

2015
PUBLISHING
PLATFORM



Quark Publishing Platform 2015 October 2017 Update Rest Interface Guide

Contents

REST framework.....	1
Session management with REST interface	2
HTTP GET and POST Requests	2
Request/Response parameter formats	3
REST Framework Services.....	29
Session management	29
logOn.....	30
getSession.....	31
getAllSessions.....	31
keepSessionAlive.....	32
terminateSession.....	33
getMaxIdleTime.....	33
validatePassword.....	34
logOff.....	34
Asset management	35
getAsset.....	36
getAssetAllVersions.....	44
importAsset.....	45
importNewAsset.....	46
updateAsset.....	46
duplicateAsset.....	48
deleteAsset.....	49
createAssetRelation.....	50
createAssetRelations.....	53
getAssetRelations.....	56
updateAssetRelation.....	57
deleteAssetRelations.....	59
deleteAssetRelationsForParent.....	59
generateTextIndexesOfAllAssets.....	60
generateTextIndexes.....	60
getAssetVersionNumbers.....	61
getAssetRenditionsInfo.....	61
getAvailableRenditions.....	62

updateChildVersionForAsset	63
updateChildAssetVersionForRelationId	64
updateAllChildAssetsVersion	65
updateChildVersionLockingFlag	65
checkIn	66
checkInNew.....	68
checkInArticle.....	70
checkInNewArticle	72
checkInRendition.....	74
checkOut.....	75
abortCheckout	76
saveDraft.....	77
promoteDraft	77
detectContentType	78
Collection management	79
createCollection	80
createCollections	82
getCollection	84
getCollectionHierarchy	100
getCollectionDefaultJobJacket	102
getCollectionDefaultTicket	102
moveCollection.....	103
duplicateCollection.....	104
updateCollection	105
updateCollections.....	107
deleteCollection	108
deleteCollectionWithAssets.....	109
createJobjackets	109
getJobjacket	110
getAllJobJackets.....	110
getJobJacketContent	111
updateJobJacket	112
updateJobJackets.....	112
parseJobJacketXml.....	113
deleteJobJacket	114
getApplicableRoutingUsers	114
getApplicablePrivileges	115
getApplicableRoutingGroups	118
getApplicableAttributeConstraints	119
getApplicableStatusTransition	119
getApplicableWorkflowInitialStatuses	120
Query management	121
search.....	121
searchAssets.....	122

searchCollections	132
createSearches	134
getSearch	135
getAllSearches	136
getSearchResultsForQuery	136
getTextSearchResults	139
setSharedSearchTrustees	140
getSharedSearchTrustees.....	141
updateSearches	142
deleteSearch	143
count	143
countAssets	144
countCollections	145
countSearchResultsForQuery	147
countTextSearchResults	149
Administration	150
createUsers	154
getUser	155
getAllUsers.....	155
getUsersWithGivenUserClass	156
getAllUserRedliningColor	157
updateUserRedliningColor	158
updateUserEnabled	158
updateUsers.....	159
deleteUser.....	160
createGroups	161
getGroup	162
getAllGroups.....	163
getGroupsWithGivenUserClass	164
getUserGroups.....	164
updateGroups.....	165
deleteGroup.....	166
createContentTypes.....	167
getContentType	169
getAllContentTypes	170
updateContentTypes	171
deleteContentType	173
createAttributes	173
setAttributes	175
getAttribute	176
getAllAttributes.....	176
updateAttributes.....	177
deleteAttribute	179
createForms	180

getAllForms	180
updateForms.....	182
deleteForm	185
createRoles	185
getRole.....	186
getAllRoles.....	187
updateRoles	187
deleteRole.....	189
createRelationType	189
getRelationType	190
getAllRelationTypes	191
updateRelationTypes	192
deleteRelationType	193
createDomains	193
getDomain	195
getAllDomains	195
updateDomains	196
deleteDomain	197
createLdapProfiles	198
getLdapProfile	201
getAllLdapProfiles.....	202
updateLdapProfiles.....	202
deleteLdapProfile.....	203
getLdapUsers.....	204
importUsersFromLdap	205
getLdapGroups.....	206
importGroupsFromLdap	207
synchronizeLdapUsersGroups.....	208
synchronizeLdapUsersGroupsBasedOnProfile.....	208
createLdapSearches.....	209
getLdapSearch	209
getAllLdapSearches	210
updateLdapSearches	210
deleteLdapSearch	211
getTrustee	212
updateTrusteeRule	212
createWorkflows.....	213
getWorkflow.....	214
getAllWorkflows	216
updateWorkflows	216
deleteWorkFlow.....	217
createPreferences	218
getPreference.....	218
getAllPreferences.....	219

getPreferenceValues	220
updatePreferences	221
updatePreferenceValues	221
deletePreference.....	222
createRepositories	222
getRepository.....	223
getAllRepositories.....	224
updateRepositories.....	225
deleteRepository.....	225
createPrivileges.....	226
getPrivilege	226
getAllPrivileges	227
getApplicableApplicationPrivileges.....	228
getApplicableContentPrivileges	229
updatePrivileges	230
deletePrivilege	230
createPrivilegeGroups.....	231
getPrivilegeGroup.....	231
getAllPrivilegeGroups	232
updatePrivilegeGroups	233
deletePrivilegeGroup.....	233
addDomainValue	234
getDomainValue	234
getDomainValueXml	235
setDomainValueXml.....	235
searchDomainValues.....	236
updateDomainValue	236
deleteDomainValue	237
getFavorites	238
setFavorites.....	238
getWorkflowInitiatingStatuses	239
getStatusTransitions.....	239
Publishing.....	240
getPublishingChannels	241
getAllPublishingChannels	241
getDeliveryChannels.....	242
publishAsset.....	242
publishBasedOnURI	244
publishBasedOnStreamContext	245
createDatadoc	246
getDatadocChannels	247
fetchDatadocAssets	248
executeDatadocChannel	249
updateDatadoc.....	250

detachDatadocChannel	251
Script execution	252
createScript	252
getScriptContent	253
getAllScripts	254
executeScript	254
deleteScript.....	255
Archive and Restore management	256
archiveCollectionAssets	256
archiveQueryAssets.....	257
archiveAssets	258
restoreAssets.....	259
Office	260
getExcelChart	261
getExcelChartBasedOnStream	262
getExcelChartBasedOnURI	263
getExcelTable.....	264
getExcelTableBasedOnStream	265
getExcelTableBasedOnURI	265
getExcelWorksheets.....	266
getExcelWorksheetsBasedOnURI	268
getExcelWorksheetsBasedOnStreamContext.....	268
getChart.....	269
getChartBasedOnURI	270
getChartBasedOnStream	271
getTable	271
getTableBasedOnURI.....	272
getTableBasedOnStreamContext	273
getNamedRange.....	274
getNamedRangeBasedOnURI	275
getNamedRangeBasedOnStream	276
getDynamicRange.....	277
getDynamicRangeBasedOnURI	278
getDynamicRangeBasedOnStream	279
getVisioPagesInfo	280
getVisioPagesInfoBasedOnURI	280
getVisioPagesInfoBasedOnStreamContext	281
getVisioPageOutput	282
getVisioPageOutputBasedOnURI	282
getVisioPageOutputBasedOnStreamContext	283
getPowerPointSlidesInfo	283
getPowerPointSlidesInfoBasedOnURI	284
getPowerPointSlidesInfoBasedOnStreamContext	285
getPowerPointSlideOutput.....	285

getPowerPointSlideOutputBasedOnURI	286
getPowerPointSlideOutputBasedOnStreamContext	287
getPowerPointSlideRangeOutput.....	287
getPowerPointSlideRangeOutputBasedOnURI	288
getPowerPointSlideRangeOutputBasedOnStreamContext.....	289
Charting	289
getChart.....	289
getChartBasedOnUri	290
getChartBasedOnStream.....	291
Configuration	291
getAllXmlMimeType	292
getXmlMimeType.....	293
createXmlMimeType	294
getMappedMimeType.....	294
mapMimeType.....	295
unmapMimeType.....	295
deleteXmlMimeType.....	296
getPublishingProcess	296
createPublishingProcess	297
deletePublishingProcess	297
getAllPublishingChannelDefinitions	298
getPublishingChannelDefinition	298
getPublishingChannelDefinitions.....	299
createPublishingChannels.....	300
createContentChannelMappings.....	301
deleteContentChannelMappings	301
createIndexingChannelMapping	302
getMappedIndexingChannel.....	303
deleteIndexingChannelMapping	303
deletePublishingChannels	304
getDeploymentProperties.....	304
getDeploymentVersion	305
reinitialize	305
Clipboard	306
addClipboardData	306
addFormattedData	307
getAllClipboardChannelsMapping	308
getAllClipboardDataInfos	309
getAllClipboardFormats	309
getClipboardDataInfo.....	310
getClipboardChannelsMapping	311
getFormattedData	312
getLatestClipboardData.....	312
getLatestClipboardDataInfo	313

getLatestFormattedData.....	314
promoteClipboardDataToAsset	314
JSON support in request parameters for POST requests.....	316
createCollection	316
checkinNew.....	317
createDomains	319
createSearches.....	319
Change Log	321
Changes in version 11.0.....	321
Changes in version 11.2.....	323
Changes in version 12.0.....	323
Changes in version 13.0.....	325
Changes in version 13.1.....	326
Contacting Quark.....	327
In the Americas	327
Outside the Americas	327
Legal notices	328

REST framework

The Quark Publishing Platform REST interface lets you access the Platform's functionality using simple HTTP requests. Rather than having to depend on complex remote access libraries such as RMI and SOAP, you can directly invoke functionality and access resources via HTTP and take advantage of standard HTTP mechanisms for transport, session management, security, and error handling. In response to REST calls, the Platform returns any requested data as XML, HTML, or a binary resource stream. If an error occurs, the Platform returns an XML response with an error message and a complete stack trace.

REST URL pattern used in Quark Publishing Platform is:

```
http://server:port/rest/service/typeofcall/resource?op=[value]&[parameter 1]=[value1]
```

For secure access to these requests, the platform server should be configured with SSL. Please refer to the "Enabling SSL for Quark Publishing Platform Web applications" chapter in the **System Admin Guide** in order to learn how to configure the Platform server with SSL.

- ➔ Request parameters, including *op* are not required for all calls.
- ➔ The REST framework is extensible, so you can expose custom services over HTTP.

REST requests return XML responses (by default) or a binary file. The schema of the XML response is specific to the request. These schemas with examples are available in the "[Request parameter formats](#)" chapter.

For all requests that return XML you can add an additional parameter (i.e. **view=jsonView**) to return **json** instead of an XML response.

You can also apply formatting to returned XML by passing a **view** parameter that points to an XSL transformation or a .jsp page that generates HTML from the XML. The following example points to a .jsp page. This example can be used with a request that returns a search result:

```
http://[server]:[port]/[type of call]?view=thumbnail
```

This example can be used with a request that returns a search result.

Sample .jsp files are provided in the Quark Publishing Platform application folder at `QPP Server/webapps/rest/jsp`

- ➔ Users can add their own .jsp file to this folder.

To use custom XSLT, you need to write a custom bean and specify the path of the XSLT file. The XSLT file can be anywhere. The classpath/file location must be mentioned in the bean definition.

Already defined beans allowing you to view responses using XSLT can be found here:

```
http://ServerName:ServerPort/rest/service/sessions?view=sessionXsltView
```

Support for Transit Session by providing QPP sessionId or username, password as request parameters. User can make a request to server without logging-in by adding request parameters loginname, loginpassword in their requests. Value for loginname and loginpassword would be the username and password of the user. Another request parameter qppsessionid can be used by providing valid qpp session id.

Session management with REST interface

The following types of sessions can be managed using the REST interface:

- Transient sessions - these sessions are not actively managed they are created just for the time period it takes to work on a request.

Users can receive support for a transient session by providing the Quark Publishing Platform's **sessionId**, **username** or **password** as request parameters. Users can make a request to the server without logging-in by adding **loginname** and **loginpassword** as request parameters.

- ➔ Values for **loginname** and **loginpassword** would be the **username** and **password** of the user.

The request parameter, **qppsessionid** can be used by providing a valid Platform session id.

- Stateful sessions

Create a stateful session by using the Session Management service. A stateful session created with this approach will last for the maximum idle time. The session can be kept alive by using the **keepsessionalive** request. When working with the session, a user does not need to pass the **qppsessionid** parameter to all requests for the session just created.

HTTP GET and POST Requests

You access the API and manipulate resources by using standard REST conventions. Each API method is invoked using one of the following HTTP methods:

- **GET** - performs a read operation
GET is for retrieving data. When you use the **GET** method, the browser encodes form data into a URL. Use the **GET** method only when the form processing is important.
- **POST** - performs a create operation.
POST can involve storing or updating data. When you use the **POST** method, form data is passed within the message body.

Request/Response parameter formats

Some response and request parameters require an XML body. The format for such parameters is described in the following table.

The same request parameters can also be sent in JSON format.

In order to pass request parameters in JSON format, either a request parameter or a request header with the name `qpp-content-type` can be passed in the request.

Following is the fallback sequence used for resolving the `qpp-content-type`:

1. Request Header `qpp-content-type`.
2. Request Parameter `qpp-content-type`.
3. Auto detection by Tika.

If none of the above resolves to the correct content type, the default value used is XML.

➔ Response in JSON format already exists by specifying extra request parameter `view=jsonView` in the REST request.

Object Name	Schema Definition	Example
assetRelationInfo	<pre><xs:element name="assetRelationInfo"> <xs:complexType base="xs:sequence"> <xs:element ref="attributeValueList" /> </xs:complexType> <xs:attribute name="id" type="xs:long" use="optional" /> <xs:attribute name="parentAssetId" type="xs:long" use="optional" default="0" /> <xs:attribute name="parentAssetName" type="xs:string" use="optional"/> <xs:attribute name="childAssetId" type="xs:long" use="optional" /> <xs:attribute name="childAssetName" type="xs:string" use="optional" /> </xs:element></pre>	<p>XML based relations: XML Component Reference</p> <pre><assetRelationInfoList> <assetRelationInfo relationType="XML Component Reference" childAssetId="227" parentAssetId="225"> <attributeValueList> <attributeValue id="417"> xpathValue </attributeValue> </attributeValueList> </assetRelationInfo> </assetRelationInfoList></pre> <p>Primary Attachment:</p>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre> /> <xs:attribute name="childAssetMajorVersion" type="xs:long" use="optional" default="0"/> <xs:attribute name="childAssetMinorVersion" type="xs:long" use="optional" default="0"/> <xs:attribute name="relationType" type="xs:string" use="optional" default="" /> <xs:attribute name="relationTypeId" type="xs:long" use="optional" default="0" /> <xs:attribute name="relationState" type="xs:string" use="optional" /> <xs:attribute name="relationStateId" type="xs:int" use="optional" /> <xs:attribute name="lockedToChildVersion" type="xs:boolean" use="optional" default="false"/> </xs:complexType> </xs:element> </pre>	<pre> <attributeValueList> <attributeValue id="401"> 1 </attributeValue> <attributeValue id="402"> 5 </attributeValue> <attributeValue id="403"> 1 </attributeValue> <attributeValue id="404"> 1 </attributeValue> <attributeValue id="412"> 1 </attributeValue> </attributeValueList> </pre>
assetRelationInfoList	<pre> <xs:element name="assetRelationInfoList"> <xs:complexType> <xs:sequence> <xs:element maxOccurs="unbounded" ref="assetRelationInfo" /> </xs:sequence> </xs:complexType> </xs:element> </pre>	<p>XML based relations: XML Component Reference</p> <pre> <assetRelationInfoList> <assetRelationInfo relationType= "XML Component Reference" childAssetId="227" parentAssetId="225"> <attributeValueList> <attributeValue id="417"> xpathValue </attributeValue> </attributeValueList> </assetRelationInfo> </assetRelationInfoList> </pre> <p>Primary Attachment:</p> <pre> <attributeValueList> <attributeValue id="401"> 1 </attributeValue> <attributeValue id="402"> 5 </attributeValue> </pre>

Object Name	Schema Definition	Example
		<pre><attributeValue id="403"> 1 </attributeValue> <attributeValue id="404"> 1 </attributeValue> <attributeValue id="412"> 1 </attributeValue> </attributeValueList></pre>
attributeValue	<pre><xs:element name="attributeValue"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long" use="optional" default="0"/> <xs:attribute name="name" type="xs:string" use="optional" default=""/> <xs:attribute name="type" type="xs:string" use="optional" default=""/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element name="attributeValueList"> <xs:complexType> <xs:sequence> <xs:element maxOccurs="unbounded" ref="attributeValue" /> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><attributeValue id="2"> Hydra.jpg </attributeValue> <attributeValue name="workflow"> publish </attributeValue> <attributeValue id="55"> 1 </attributeValue></pre>
attributeValueList	<pre><xs:element name="attributeValueList"> <xs:complexType> <xs:sequence> <xs:element ref="attributeValue" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1"> 222 </attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2"> 1511623dd8c_Duplicate. qxp </attributeValue> <attributeValue valueId="0" type="dateTime"</pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
		<pre>name="Created" id="3"> 2013-10-16T06:19:38Z </attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4"> 2013-10- 16T06:19:38Z </attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5"> Admin </attributeValue> </attributeValueList></pre>
domainValue	<pre><xs:element name="domainValue"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="valueHierarchy" type="xs:string"/> <xs:element ref="domainValueList"> <xs:element name="name" type="xs:string"/> <xs:element name="valueHierarchy" type="xs:string"/> </xs:domainValueList> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><domainValue id="1"> <name>Desktop</name> <valueHierarchy> Desktop </valueHierarchy> </domainValueList> <domainValue id="2"> <name>Research</name> <valueHierarchy> Desktop;Research </valueHierarchy> </domainValue> <domainValue id="3"> <name>Development</nam e> <valueHierarchy> Desktop;Development </valueHierarchy> </domainValue> </domainValueList> </domainValue></pre>
groupInfo	<pre><xs:element name="groupInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="emailAddress" type="xs:string"/> <xs:element name="nativeTrustee" type="xs:boolean"/> <xs:element name="defaultUserClassId" type="xs:int"/> <xs:element</pre>	<pre><groupInfo id="61"> <name>Editors</name> <emailAddress> Editors@quark.com </emailAddress> <nativeTrustee> false </nativeTrustee> <defaultUserClassId> 2 </defaultUserClassId> <defaultUserClass></pre>

Object Name	Schema Definition	Example
	<pre>name="defaultUserClass" type="xs:string"/> <xs:element name="ldapProfileId" type="xs:int"/> <xs:element name="ldapProfile" type="xs:string"/> <xs:element name="ldapDistinguishedName" type="xs:string"/> <xs:element ref="memberList"> <xs:element name="nameId" /></xs:memberList> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre>NoPrivileges </defaultUserClass> <ldapProfileId> 10 </ldapProfileId> <ldapProfile> Profile </ldapProfile> <ldapDistinguishedName > DN </ldapDistinguishedNam e> <memberList> <nameId id="62"> <name>John</name > </nameId> </memberList> </groupInfo></pre>
groupInfoList	<pre><xs:element name="groupInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="groupInfo" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType> </xs:element></pre>	
memberList	<pre><xs:element name="memberList"> <xs:complexType> <xs:sequence> <xs:element ref="nameId" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	
ldapProfileInfo	<pre><xs:element name="lapProfileInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="realm" type="xs:string"/> <xs:element name="userName" type="xs:boolean"/> <xs:element name="authenticationType" type="xs:int"/> <xs:element ref="ldapServerInfoList"> <xs:element ref="ldapServerInfo"> <xs:element name="scheme" type="xs:string"/> <xs:element name="serverName" type="xs:string"/> <xs:element name="port" type="xs:int"/> </xs:ldapServerInfo> </xs:ldapServerInfoList></pre>	<pre><ldapProfileInfo id="10"> <name>Quark</name> <realm> Editors@quark.com </realm> <userName> realm.com </userName> <authenticationType> 2 </authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName> SERVER1 </serverName> <port>389</port > </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentifica tionCriteria> <userIdentificationCri</pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre> <xs:element ref= "ldapObjectsIdentificationCriter ia"> <xs:element ref= "userIdentificationCriteria"/> <xs:element ref= "groupIdentificationCriteria"/> <xs:element name="groupMemberAttrName" type="xs:string"/> <xs:element name="membershipIdentifyingAttrN ame" type="xs:string"/> <xs:element name="baseDn" type="xs:string"/> <xs:element ref="filterToGetAllUsersOfGroup" /> <xs:element ref="filterToGetAllSubGroupsOfGr oup"/> <xs:element ref="filterToGetAllGroupsOfUser" /> </xs:ldapObjectsIdentificationCr iteria> <xs:element ref="ldapAttributeInfoList"> <xs:element ref="ldapAttributeInfo"> <xs:element name="name" type="xs:string"/> <xs:element name="value" type="xs:string"/> </xs:ldapAttributeInfo> </xs:ldapAttributeInfoList> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> teria> ... </userIdentificationCr iteria> <groupIdentificationCr iteria> ... </groupIdentificationC riteria> <groupMemberAttrName> Member </groupMemberAttrName> <membershipIdentifying AttrName> memberof </membershipIdentifyin gAttrName> <baseDn> BDN </baseDn> <filterToGetAllUsersOf Group> ... </filterToGetAllUsersO fGroup> <filterToGetAllSubGrou psOfGroup> ... </filterToGetAllSubGro upsOfGroup> <filterToGetAllGroupsO fUser> ... </filterToGetAllGroups OfUser> </ldapObjectsIdentific ationCriteria> <ldapAttributeInfoList > <ldapAttributeInfo id="5"> <name>USER_EMAIL</name > <value>mail</value > </ldapAttributeInfo> </ldapAttributeInfoLis t> </ldapProfileInfo> </pre>
ldapSearchCriteria	<pre> <xs:element name="ldapSearchCriteria"> <xs:complexType> <xs:sequence> <xs:element name="ldapProfile" <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> </pre>	<pre> <ldapSearchCriteria> <ldapProfile id="10"> QPP LDAP Profile </ldapProfile> <searchFilter> cn=qpp server </searchFilter> <baseString> dc=india, dc=quark, dc=com </baseString> <searchType id="2"> GROUPS_SEARCH </pre>

Object Name	Schema Definition	Example
	<pre><xs:element name="searchilter" type="xs:string"/> <xs:element name="baseString" type="xs:string"/> <xs:element name="searchType"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:int"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre></searchType> </ldapSearchCriteria></pre>
ldapSearchInfo	<pre><xs:element name="ldapSearchInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element ref="ldapSearchCriteria"/> </xs:sequence> <xs:attribute name="id" type="xs:long"/> </xs:complexType> </xs:element></pre>	<pre><ldapSearchInfo id="11"> <name>LDAP Group Search</name> <ldapSearchCriteria> <ldapProfile id="10"> QPP LDAP Profile </ldapProfile> <searchFilter> cn=qpp server </searchFilter> <baseString> dc=india, dc=quark, dc=com </baseString> <searchType id="2"> GROUPS_SEARCH </searchType> </ldapSearchCriteria> </ldapSearchInfo></pre>
ldapSearchInfoList	<pre><xs:element name="ldapSearchInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="ldapSearchInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	
preferenceInfo	<pre><xs:element name="preferenceInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="category"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:int"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><preferenceInfo id="1007"> <name> Display all Redline Deletions in One Color </name> <category id="0"> SYSTEM </category> <possibleValuesExist> false </possibleValuesExist></pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element name="possibleValuesExist" type="xs:boolean"/> <xs:element name="defaultValue"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element ref="preferenceValueList"/> <xs:element name="systemDefined" type="xs:boolean"/> </xs:sequence> <xs:attribute name="id" type="xs:long"/> </xs:complexType> </xs:element> </pre>	<pre> <defaultValue> No </defaultValue> <systemDefined> true </systemDefined> </preferenceInfo> </pre>
preferenceInfoList	<pre> <xs:element name="preferenceInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="preferenceInfo" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <preferenceInfoList> <preferenceInfo id="1007"> <name> Display all Redline Deletions in One Color </name> <category id="0"> SYSTEM </category> <possibleValuesExist> false </possibleValuesExist> <defaultValue> No </defaultValue> <systemDefined> true </systemDefined> </preferenceInfo> </pre>
privilegeDefinition	<pre> <xs:element name="privilegeDefinition"> <xs:complexType> <xs:sequence> <xs:element name="group"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> </pre>	<pre> <privilegeDefinition name="Check In New" id="1000001"> <group id="100000"> General </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> true </contentPrivilege> </pre>

Object Name	Schema Definition	Example
	<pre><xs:element name="statusBased" type="xs:boolean"/> <xs:element name="systemDefined" type="xs:boolean"/> <xs:element name="contentPrivilege" type="xs:boolean"/> <xs:element ref="contentTypeInfoList"/> </xs:sequence> <xs:attribute name="id" type="xs:long"/> <xs:attribute name="name" type="xs:string"/> </xs:complexType> </xs:element></pre>	<pre><contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> </contentTypeInfoList> </privilegeDefinition></pre>
privilegeDefinitionList	<pre><xs:element name="privilegeDefinitionList"> <xs:complexType> <xs:sequence> <xs:element ref="privilegeDefinition" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><privilegeDefinitionList> <privilegeDefinition name="Check In New" id="1000001"> <group id="100000"> General </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> true </contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList></pre>
privilegeGroupDefinition	<pre><xs:element name="privilegeGroupDefinition"> <xs:complexType> <xs:sequence> <xs:element ref= "privilegeDefinitionList"/> <xs:element ref= "privilegeGroupDefinitionList"/> <xs:element name= "systemDefined" type="xs:boolean"/> <xs:element name="parentGroup"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute</pre>	<pre><privilegeGroupDefinition name="Attributes" id="300000"> <privilegeDefinitionList> <privilegeDefinition name="View" id="3000007"> <group id="300000"> Attributes </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined></pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre> name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> </xs:sequence> <xs:attribute name="id" type="xs:long"> </xs:attribute> <xs:attribute name="name" type="xs:string"> </xs:attribute> </xs:complexType> </xs:element> </pre>	<pre> <contentPrivilege> false </contentPrivilege> </privilegeDefinition> <privilegeDefinition name="Create/Modify" id="3000008"> <group id="300000"> Attributes </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> false </contentPrivilege> </privilegeDefinition> <privilegeDefinition name="Delete" id="3000009"> <group id="300000"> Attributes </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> false </contentPrivilege> </privilegeDefinition> </privilegeDefinitionL ist> <privilegeGroupDefinit ionList> <systemDefined> true </systemDefined> <parentGroup id="99998"> Administration </parentGroup> </privilegeDefinitionL ist> </privilegeGroupDefini tion> </pre>
privilegeGroupDefi nitionList	<pre> <xs:element name= "privilegeGroupDefinitionList"> <xs:complexType> <xs:sequence> <xs:element ref= </pre>	<pre> <privilegeGroupDefinit ionList> <privilegeGroupDefinit ion </pre>

Object Name	Schema Definition	Example
	<pre> "privilegeGroupDefinition" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> name="Administration" id="99998"> <privilegeDefinitionLi st/> <privilegeGroupDefinit ionList> <privilegeGroupDefinit ion name="Attributes" id="300000"> <privilegeDefinitionLi st> <privilegeDefinition name="View" id="300007"> <group id="300000"> Attributes </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> false </contentPrivilege> </privilegeDefinition> <privilegeDefinition name="Create/Modify" id="300008"> <group id="300000"> Attributes </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> false </contentPrivilege> </privilegeDefinition> <privilegeDefinition name="Delete" id="300009"> <group id="300000"> Attributes </group> <statusBased> false </statusBased> <systemDefined> true </systemDefined> <contentPrivilege> false </contentPrivilege> </pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
		<pre> </privilegeDefinition> </privilegeDefinitionList> <privilegeGroupDefinition> <systemDefined> true </systemDefined> <parentGroup id="99998"> Administration </parentGroup> </privilegeGroupDefinition> </privilegeGroupDefinitionList> </pre>
revisionControlInfo	<pre> <xs:element name="revisionControlInfo"> <xs:complexType> <xs:sequence> <xs:element name="minorVersioningAllowed" type="xs:boolean" /> <xs:element name="allRevisions" type="xs:boolean" /> <xs:element name="initialRevisionsCount" type="xs:int" /> <xs:element name="recentRevisionsByPeriod" type="xs:boolean" /> <xs:element name="recentRevisionsCount" type="xs:int" /> <xs:element name="recentRevisionsDuration" type="xs:int" /> </xs:sequence> <xs:attribute name="contentType" type="xs:string" /> </xs:complexType> </xs:element> </pre>	<pre> <revisionControlInfo contentType="110211"> <minorVersioningAllowed> true </minorVersioningAllowed> <allRevisions> false </allRevisions> <initialRevisionsCount> 0 </initialRevisionsCount> <recentRevisionsByPeriod> false </recentRevisionsByPeriod> <recentRevisionsCount> 0 </recentRevisionsCount> <recentRevisionsDuration> 0 </recentRevisionsDuration> </revisionControlInfo> </pre>
revisionControlInfoList	<pre> <xs:element name="revisionControlInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="revisionControlInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <revisionControlInfoList> <revisionControlInfo contentType="110211"> <minorVersioningAllowed> true </minorVersioningAllowed> <allRevisions> </pre>

Object Name	Schema Definition	Example
		<pre> false </allRevisions> <initialRevisionsCount > 0 </initialRevisionsCoun t> <recentRevisionsByPeri od> false </recentRevisionsByPer iod> <recentRevisionsCount> 0 </recentRevisionsCoun t> <recentRevisionsDurati on> 0 </recentRevisionsDurat ion> </revisionControlInfo> </revisionControlInfoL ist> </pre>
redLineInfo	<pre> <xs:element name="redlineInfo"> <xs:complexType> <xs:sequence> <xs:element name="color"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="red" type="xs:int"/> <xs:attribute name="green" type="xs:int"/> <xs:attribute name="blue" type="xs:int"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element name="user"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <redlineInfo> <color blue="0" green="0" red="0"/> <user id="1">Admin</user> </redlineInfo> </pre>
redLineInfoList	<pre> <xs:element name="redlineInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="redlineInfo" maxOccurs="unbounded" </pre>	<pre> <redlineInfoList> <redlineInfo> <color blue="0" green="0" red="0"/> <user id="1">Admin</user> </pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre>minOccurs="0"/> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre></redlineInfo> </redlineInfoList></pre>
roleInfo	<pre><xs:element name="roleInfo"> <xs:complexType> <xs:sequence> <xs:element ref="roleInfo" macOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <roleInfo name="Administrator" id="1"> <applicationPrivList> <privilege id="190000017"> <name> Remove Link with Collection Template </name> </privilege> <privilege id="190000016"> <name> Status Based Routings </name> </privilege> </applicationPrivList> <contentTypePrivList> <contentTypePriv contentType="Asset" contentTypeid="1100"> <privilege id="10000015"> <name> Get Original Assets </name> </privilege> <privilege id="10000013"> <name> Enable Asset Archiving </name> </privilege> <privilege id="1000001"> <name> Check In New </name> </privilege> </contentTypePriv> </contentTypePrivList> </roleInfo></pre>
routingInfo	<pre><xs:element name="routingInfo"> <xs:complexType> <xs:sequence> <xs:element name="trusteeId" type="xs:long" /> <xs:element name="trustee" type="xs:string" /> <xs:element name="routingTypeId" type="xs:int" /> <xs:element name="routingType"></pre>	<pre><routingInfo status= "New Status 001" statusId="10"> <trusteeId>1</trusteeI d> <routingTypeid> 1 </routingTypeId> </routingInfo></pre>

Object Name	Schema Definition	Example
	<pre> type="xs:string" /> </xs:sequence> <xs:attribute name="statusId" type="xs:long" /> <xs:attribute name="status" type="xs:string" /> </xs:complexType> </xs:element> </pre>	
routingInfoList	<pre> <xs:element name="routingInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="routingInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <routingInfoList> <routingInfo status="New Status 001" statusId="10"> <trusteeId>1</trusteeI d> <routingTypeId> 1 </routingTypeId> </routingInfo> </routingInfoList> </pre>
savedSearchList	<pre> <xs:element name="savedSearchList"> <xs:complexType> <xs:sequence> <xs:element ref="savedSearch" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <savedSearchList> <savedSearch id="100"> <searchName> Default </searchName> <ownerId>1</ownerId> <ownerName> Admin </ownerName> </savedSearch> </savedSearchList> </pre>
scriptInfo	<pre> <xs:element name="scriptInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="language" type="xs:string"/> <xs:element name="content" type="xs:string"/> <xs:element name="createdDate" type="xs:string"/> <xs:element name="lastModified" type="xs:string"/> <xs:element name="lastExecutionTime" type="xs:string"/> <xs:element name="lastExecutionLog" type="xs:string"/> <xs:element name="creator"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <scriptInfo id="11"> <name>NewScript.js</na me> <language> javascript </language> <createdDate> 2015-02- 02T09:28:32Z </createdDate> <lastModified> 2015- 02-02T09:40:44Z </lastModified> <lastExecutionTime> 2015-02-03T05:55:09Z </lastExecutionTime> <lastExecutionLog> PASSED </lastExecutionLog> <creator id="1"> Admin </creator> <scriptExecutionInfo </pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element name="lastModifiedBy"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element ref="scriptExecutionInfo" maxOccurs="1" minOccurs="0"/> </xs:sequence> <xs:attribute name="id" type="xs:long"/> </xs:complexType> </xs:element> </pre>	<pre> triggerType="SCHEDULED "> <scheduleInfo scheduleGroup= "SCRIPT_MANAGER_GROUP" id="11"> <cronExpressionList> <cronExpression> 0/30 * * * * ? </cronExpression> </cronExpressionList> </scheduleInfo> </scriptExecutionInfo> </scriptInfo> </pre>
scriptInfoList	<pre> <xs:element name="scriptInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="scriptInfo" maxOccurs="unbounded" minOccurs="0"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <scriptInfoList> <scriptInfo id="11"> <name>NewScript.js</na me> <language> javascript </language> <createdDate> 2015-02- 02T09:28:32Z </createdDate> <lastModified> 2015- 02-02T09:40:44Z </lastModified> <lastExecutionTime> 2015-02-03T05:55:09Z </lastExecutionTime> <lastExecutionLog> PASSED </lastExecutionLog> <creator id="1"> Admin </creator> <scriptExecutionInfo triggerType="SCHEDULED "> <scheduleInfo scheduleGroup= "SCRIPT_MANAGER_GROUP" id="11"> <cronExpressionList> <cronExpression> 0/30 * * * * ? </cronExpression> </pre>

Object Name	Schema Definition	Example
		<pre> </cronExpressionList> </scheduleInfo> </scriptExecutionInfo> </scriptInfo> </scriptInfoList> </pre>
scriptExecutionInfo	<pre> <xs:element name="scriptExecutionInfo"> <xs:complexType> <xs:sequence> <xs:element ref="eventInfo"/> <xs:element ref="scheduleInfo"/> </xs:sequence> <xs:attribute name="triggerType" type="xs:string"/> </xs:complexType> </xs:element> </pre>	<pre> <scriptExecutionInfo triggerType="SCHEDULED "> <scheduleInfo scheduleGroup= "SCRIPT_MANAGER_GROUP" id="11"> <cronExpressionList> <cronExpression> 0/30 * * * * ? </cronExpression> </cronExpressionList> </scheduleInfo> </scriptExecutionInfo> </pre>
searchInfo	<pre> <xs:element name="searchInfo"> <xs:complexType> <xs:sequence> <xs:element name="searchName" type="xs:string"/> <xs:element name="owner"> <xs:complexType> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="id" type="xs:long"/> </xs:extension> </xs:simpleContent> </xs:complexType> </xs:element> <xs:element ref="queryContext"/> <xs:element ref="queryConditionList"/> <xs:element ref="queryDisplay"/> <xs:element ref="trusteeInfoList"/> </xs:sequence> <xs:attribute name="id" type="xs:long"/> <xs:attribute name="parameterized" type="xs:boolean"/> </xs:complexType> </xs:element> </pre>	<pre> <searchInfoList> <searchInfo parameterized= "false" id="100"> <searchName> Default </searchName> <owner id="1">Admin</owner> <queryContext context= "ASSET"/> <queryDisplay exploreMode= "PLAIN" displayMode="PLAIN"> <displayColumnList> <displayColumn attributeColumn="true" width="0" columnId="2"> Name </displayColumn> <displayColumn attributeColumn="true" width="0" columnId="55"> Collection </displayColumn> <displayColumn attributeColumn="true" width="0" columnId="54"> Workflow </displayColumn> </pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
		<pre><displayColumn attributeColumn="true" width="0" columnId="24"> Status </displayColumn> </displayColumnList> <sortInfoList> <sortInfo attributeColumn="true" order="descending" columnId="2"> Name </sortInfo> </sortInfoList> </queryDisplay> <trusteeInfoList> <trusteeInfo id="11"> <user>true</user> <deleted>>false</delete d> <name>DKumar</name> </trusteeInfo> </trusteeInfoList> </searchInfo> </searchInfoList></pre>
searchInfoList	<pre><xs:element name="searchInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="searchInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	
sessionInfo	<pre><xs:element name="sessionInfo"> <xs:complexType> <xs:sequence> <xs:element name="sessionId" type="xs:string"/> <xs:element name="userId" type="xs:long"/> <xs:element name="userName" type="xs:string"/> <xs:element name="timeCreated" type="xs:string"/> <xs:element name="lastAccessedTime" type="xs:string"/> <xs:element name="machineName" type="xs:string"/> <xs:element name="applicationName" type="xs:string"/> <xs:element name="timeZoneOffset" type="xs:string"/> </xs:sequence> </xs:complexType></pre>	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sessionInfo> <sessionId> 8d9676d9-1300-433c-b305- 509cb2fa2bf9 </sessionId> <userId>1</userId> <userName>Admin</userNa me> <timeCreated> 2013- 10-16T07:25:52Z </timeCreated> <lastAccessedTime> 2013- 10-16T07:25:52Z </lastAccessedTime> <machineName> localhost </machineName> <applicationName> Rest</pre>

Object Name	Schema Definition	Example
	<pre></xs:element></pre>	<pre>Client </applicationName> <timeZoneOffset> +05:30 </timeZoneOffset> </sessionInfo></pre>
sessionInfoList	<pre><xs:element name="sessionInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="sessionInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sessionInfoList> <sessionInfo> <sessionId> c191521e- 20c2-495f-91ad-bace168 1d466 </sessionId> <userId>1</userId> <userName>Admin</userN ame> <timeCreated> 2013-10-16T05:47:23Z </timeCreated> <lastAccessedTime> 2013-10- 16T12:58:07+05:30 </lastAccessedTime> <machineName> 0:0:0:0:0:0:1 </machineName> <applicationName> Quark Publishing Platform Admin </applicationName> <timeZoneOffset> +05:30 </timeZoneOffset> </sessionInfo> <sessionInfo> <sessionId> 1f8e5d68- e915-47eb-a504-ef51333 c24e4 </sessionId> </sessionInfo> <sessionInfo> <sessionId> de4323a6- d4a6-4f15-9dfd-5c8d69f 3b7d9 </sessionId> </sessionInfo> </sessionInfoList></pre>
trusteeInfo	<pre><xs:element name="trusteeInfo"> <xs:complexType> <xs:sequence> <xs:element name="user" type="xs:boolean"/> <xs:element name="deleted"></pre>	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <trusteeInfo id="122"> <user>true</user></pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
	<pre> type="xs:boolean"/> <xs:element name="name" type="xs:string"/> <xs:element name="emailAddress" type="xs:string"/> <xs:element name="nativeTrustee" type="xs:boolean"/> <xs:element name="defaultUserClassId" type="xs:int"/> <xs:element name="ldapProfileId" type="xs:int"/> <xs:element name="ldapDistinguishedName" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <deleted>>false</delete d> <name>JSmith</name> <emailAddress> John@Quark.com </emailAddress> <nativeTrustee> false </nativeTrustee> <defaultUserClassId> 2 </defaultUserClassId> <ldapProfileId> 10 </ldapProfileId> <ldapDistinguishedName > DN </ldapDistinguishedNam e> </trusteeInfo> </pre>
userInfo	<pre> <xs:element name="userInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string" /> <xs:element name="domainId" type="xs:int" /> <xs:element name="firstName" type="xs:string" /> <xs:element name="lastName" type="xs:string" /> <xs:element name="phoneNumber" type="xs:string" /> <xs:element name="emailAddress" type="xs:string" /> <xs:element name="nativeTrustee" type="xs:boolean" /> <xs:element name="enabled" type="xs:boolean" /> <xs:element name="defaultUserClassId" type="xs:long" /> <xs:element name="defaultUserClass" type="xs:string" /> <xs:element name="ldapProfileId" type="xs:long" /> <xs:element name="nameForAuthentication" type="xs:string" /> <xs:element name="user" type="xs:boolean" /> <xs:element name="deleted" type="xs:boolean" /> <xs:element name="overriddenUserClassId" type="xs:long" /> <xs:element name="ldapDistinguishedName" type="xs:long" /> </xs:sequence> <xs:attribute name="id" </pre>	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfo id="1"> <name>Admin</name> <domainId>0</domainId> <firstName> Administrator </firstName> <emailAddress> admin@quark.com </emailAddress> <nativeTrustee> false </nativeTrustee> <enabled>true</enabled> <defaultUserClassId> 1 </defaultUserClassId> <defaultUserClass> Administrator </defaultUserClass> <ldapProfileId> 0 </ldapProfileId> <user>true</user> <deleted>>false</delete d> <ldapDistinguishedName > DN </ldapDistinguishedNam e> <overriddenUserClassId > 0 </overriddenUserClassI </pre>

Object Name	Schema Definition	Example
	<pre>type="xs:long" /> </xs:complexType> </xs:element></pre>	<pre>d> </userInfo></pre>
userInfoList	<pre><xs:element name="userInfoList"> <xs:complexType> <xs:sequence> <xs:element maxOccurs="unbounded" ref="userInfo"/> </xs:sequence> </xs:complexType> </xs:element></pre>	
workflowInfo	<pre><xs:element name="workflowInfo"> <xs:complexType> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="domainId" type="xs:int"/> <xs:element name="created" type="xs:string"/> <xs:element name="createdBy" type="xs:long"/> <xs:element name="creatorName" type="xs:string"/> <xs:element name="lastModifiedBy" type="xs:long"/> <xs:element name="lastModifierName" type="xs:string"/> <xs:element name="lastModified" type="xs:string"/> <xs:element name="constrainedStatusTransitio n" type="xs:boolean" minOccurs="0"/> <xs:element name="roleBasedRouting" type="xs:boolean" minOccurs="0"/> <xs:element ref="statusInfoList"/> <xs:element ref="contentTypeInfoList"/> <xs:element ref="statusTransitionInfoList"/> <xs:element ref="roleBasedStatusTransitionIn foList"/> <xs:element ref="statusRoutingRolesInfoList" /> </xs:sequence> <xs:attribute name="id" type="xs:long"/> </xs:complexType> </xs:element></pre>	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> <domainId>18</domainId > <created> 2014-12- 24T06:12:37Z </created> <createdBy>1</createdB y> <creatorName> Admin </creatorName> <lastModifiedBy> 1 </lastModifiedBy> <lastModifierName> Admin </lastModifierName> <lastModified> 2015- 01-22T07:16:16Z </lastModified> <constrainedStatusTra nsition> true </constrainedStatusTra nsition> <roleBasedRouting> true </roleBasedRouting> <statusInfoList> <statusInfo id="10"> <name>Status1</name> <domainId>6</domainId> <colorRed>50</colorRed > <colorGreen> 150 </colorGreen> <colorBlue>10</colorBl ue> <position>1</position> <redlineTracking> false</pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
		<pre> </redlineTracking> <checkInLayoutEvaluation> false </checkInLayoutEvaluation> <outputLayoutEvaluation> false </outputLayoutEvaluation> <privilegeInfoList> <privilegeInfo> <roleName> Administrator </roleName> <roleId>1</roleId> <overridden> false </overridden> <contentTypePrivList/> </privilegeInfo> <privilegeInfo> <roleName> Editor </roleName> <roleId>20</roleId> <overridden> false </overridden> <contentTypePrivList/> </privilegeInfo> <privilegeInfo> <roleName> Publisher </roleName> <roleId>21</roleId> <overridden> false </overridden> <contentTypePrivList/> </privilegeInfo> </privilegeInfoList> </statusInfo> </statusInfoList> <contentTypeInfoList> <contentTypeInfo id="1102"> <name>Project</name> <contentTypeHierarchy> System;Asset;Project; </contentTypeHierarchy> <systemDefined> true </systemDefined> <childrenAvailable> true </pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
		<pre> </childrenAvailable> <extensible> true </extensible> </contentTypeInfo> </contentTypeInfoList> <statusTransitionInfoList> <statusTransitionInfo id="10"> <name>Status1</name> <workflowInitiatingSta tus> true </workflowInitiatingSt atus> </statusTransitionInfo > </statusTransitionInfo List> <roleBasedStatusTransi tionInfoList> <roleBasedStatusTransi tionInfo id="20"> <name>Editor</name> <statusTransitionInfoL ist> <statusTransitionInfo id="10"> <name>Status1</name> <workflowInitiatingSta tus> true </workflowInitiatingSt atus> </statusTransitionInfo > </statusTransitionInfo List> </roleBasedStatusTrans itionInfo> </roleBasedStatusTrans itionInfoList> <statusRoutingRolesInf oList> <statusRoutingRolesInf o id="10"> <name>Status1</name> <routingRolesInfoList> <routingRoleInfo id="1"> <name> Administrator </name> </pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
		<pre> </routingRoleInfo> <routingRoleInfo id="20"> <name>Editor</name> </routingRoleInfo> </routingRolesInfoList > </statusRoutingRolesIn fo> </statusRoutingRolesIn foList> </workflowInfo> </workflowInfoList> </pre>
workflowInfoList	<pre> <xs:element name="workflowInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="workflowInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	
visioPageInfo	<pre> <xs:element name="visioPageInfo"> <xs:complexType> <xs:sequence> <xs:element name="pageName" type="xs:string" /> </xs:sequence> <xs:attribute name="index" type="xs:int"/> </xs:complexType> </xs:element> </pre>	<pre> <visioPageInfo index="1"> <pageName>Overview</pa geName> </visioPageInfo> <visioPageInfo index="2"> <pageName>Final</pageN ame> </visioPageInfo> </pre>
mimeTypeInfo	<pre> <xs:element name="mimeTypeInfo"> <xs:complexType> <xs:sequence> <xs:element ref="pi" minOccurs="0" maxOccurs="1"/> <xs:element name="xpath" type="xs:string" minOccurs="0" maxOccurs="1"/> </xs:sequence> <xs:attribute name="type" type="xs:string" use="required"/> <xs:attribute name="parentMimeType" type="xs:string" use="required" /> </xs:complexType> </xs:element> </pre>	<pre> <mimeTypeInfo type= "application/xml; format=smarhtable" parentMimeType="applic ation/xml;"> <xpath>/*[local- name()='table']/@type= 'smarhtable'</xpath> </mimeTypeInfo> </pre>
visioPageInfoList	<pre> <xs:element name="visioPageInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="visioPageInfo" maxOccurs="unbounded"/> </pre>	<pre> <visioPageInfoList> <visioPageInfo index="1"> <pageName>Overview</pa geName> </pre>

Object Name	Schema Definition	Example
	<pre></xs:sequence> </xs:complexType> </xs:element></pre>	<pre></visioPageInfo> <visioPageInfo index="2"> <pageName>Final</pageName> </visioPageInfo> </visioPageInfoList></pre>
mimeTypeInfoList	<pre><xs:element name="mimeTypeInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="mimeTypeInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element></pre>	<pre><mimeTypeInfoList> <mimeTypeInfo type="application/xml; format=smarttable" parentMimeType="application/xml;"> <xpath>/*[local-name()='table']/ @type='smarttable'</xpath> </mimeTypeInfo> <mimeTypeInfo type="application/xml; format=auxiliarydictionary" parentMimeType="format=hyphenationexceptions"> <xpath>/*[local-name()='Dictionary']/ @Type='AuxiliaryDictionary'</xpath> </mimeTypeInfo> </mimeTypeInfoList></pre>
powerPointSlideInfo	<pre><xs:element name="powerPointSlideInfo"> <xs:complexType> <xs:sequence> <xs:element name="slideId" type="xs:long"/> </xs:sequence> <xs:attribute name="index" type="xs:int"/> </xs:complexType> </xs:element></pre>	<pre><powerPointSlideInfo index="1"> <slideId>256</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="2"> <slideId>257</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="3"> <slideId>316</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="4"> <slideId>330</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="5"> <slideId>325</slideId> </powerPointSlideInfo></pre>

REQUEST/RESPONSE PARAMETER FORMATS

Object Name	Schema Definition	Example
powerPointSlideInfoList	<pre> <xs:element name="powerPointSlideInfoList"> <xs:complexType> <xs:sequence> <xs:element ref="powerPointSlideInfo" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </pre>	<pre> <powerPointSlideInfoList> <powerPointSlideInfo index="1"> <slideId>256</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="2"> <slideId>257</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="3"> <slideId>316</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="4"> <slideId>330</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="5"> <slideId>325</slideId> </powerPointSlideInfo> </powerPointSlideInfoList> </pre>

REST Framework Services

The API includes support for the following functional groups:

- Session Management
- Asset Management
- Collection Management
- Query Management
- Administration
- Publishing
- Script Execution
- Archive & Restore Management
- Office Document Management
- Charts Management
- Configuration Management
- Clipboard Management

The input parameters for all functions have been identified in their description as being either mandatory or optional. An input value designated with *[value]*, indicates a value that you must provide.

Many functions require an `op` parameter as input. The `op` parameter allows you to specify which operation is to be performed. The value required for the `op` parameter for each function is given.

Session management

The session management capabilities of the API allow application developers to manage sessions and includes the following REST calls:

- *logOn*
- *getSession*
- *getAllSessions*
- *keepSessionAlive*
- *terminateSession*
- *getMaxIdleTime*

- *validatePassword*
- *logOff*

logOn

Function	Log in		
Description	This request starts a session.		
Type	POST		
URL Format	.../sessions/[user name]		
Java API	Method Name		Class Name
	logOn		com.quark.qpp.service.facadeClass SessionFacade
Parameters	Name	Type	Description
	op=logon	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	password=[password]	string	Users password (Mandatory).
	asusername=[name]	string	If password is set to the super admin password, you can perform actions as a different user by supplying that user name here. (Optional)
	machinename=[name]	string	Lets you pass the name of the computer from which you have logged in. (Optional) If this value is not provided then the machinename is set as localhost for this session.
	appname=[name]	string	Lets you pass the name of the application from which you have logged in. (Optional) If this value is not provided then the appname is set as Rest Client for this session.
Returns	Returns a <code>sessionInfo</code> object in XML format. See the table in the "Request/Response parameter formats" chapter.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/sessions/Admin?op=logon &password=Admin</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sessionInfo> <sessionId>30609864-6dd2-4eba-9031- c80ef56f3a45</sessionId> <userId>1</userId> <userName>Admin</userName> <timeCreated>2013-04-10T07:32:06Z</timeCreated> <lastAccessedTime>2013-04-10T07:32:06Z</lastAccessedTime> <machineName>localhost</machineName> <applicationName>Rest Client</applicationName></pre>		

	<pre><timeZoneOffset>+05:30</timeZoneOffset> </sessionInfo></pre>
--	---

getSession

Function	Get information about a specific session	
Description	This request returns information about a specific session.	
Type	GET	
URL Format	.../sessions/[session ID]	
Java API	Method Name	Class Name
	getSession	com.quark.qpp.service.facade Class SessionFacade
Parameters	This request requires no input parameters	
Returns	Returns a <code>sessionInfo</code> object in XML format. See the table in the “Request/Response parameter formats” chapter.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/sessions/30609864-6dd2-4eba-9031-c80ef56f3a45</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sessionInfo> <sessionId>30609864-6dd2-4eba-9031-c80ef56f3a45</sessionId> <userId>1</userId> <userName>Admin</userName> <timeCreated>2013-04-10T07:32:06Z</timeCreated> <lastAccessedTime>2013-04-10T07:32:06Z</lastAccessedTime> <machineName>localhost</machineName> <applicationName>Rest Client</applicationName> <timeZoneOffset>+05:30</timeZoneOffset> </sessionInfo></pre>	

getAllSessions

Function	Get all sessions	
Description	This request returns a list of all the active sessions.	
Type	GET	
URL Format	.../sessions	
Java API	Method Name	Class Name
	getAllSessions	com.quark.qpp.service.facade Class SessionFacade
Parameters	This request requires no input parameters	
Returns	Returns a <code>sessionInfoList</code> object containing an array of active sessions in XML format. See the table in the “Request/Response parameter formats” chapter.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/sessions</pre>	

	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sessionInfoList> <sessionInfo> <sessionId>e1a89f4c-aa30-488d-9e0e- e067e3431be4</sessionId> <userId>1</userId> <userName>Admin</userName> <timeCreated>2013-04-10T06:11:56Z</timeCreated> <lastAccessedTime>2013-04- 10T13:02:57+05:30</lastAccessedTime> <machineName>0:0:0:0:0:0:0:1</machineName> <applicationName>Quark Publishing Platform Web Client</applicationName> <timeZoneOffset>+05:30</timeZoneOffset> </sessionInfo> </sessionInfoList></pre>
--	---

keepSessionAlive

Function	Keep a specific session alive		
Description	This request keeps the current session alive. The maximum idle time for a session is 60 seconds, which can be updated in the {qpp_home}/serverApp.properties file.		
Type	GET		
URL Format	.../sessions/		
Java API	Method Name		Class Name
	keepSessionAlive		com.quark.qpp.service.facade Class SessionFacade
Parameters	Name	Type	Description
	op=keepalive	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
Returns	Returns a sessionInfo object in XML format. See the table in the "Request/Response parameter formats" chapter.		
Examples	URL:		
	http://serverName:serverPort/rest/service/sessions?op=keepalive		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sessionInfo> <sessionId>30609864-6dd2-4eba-9031- c80ef56f3a45</sessionId> <userId>1</userId> <userName>Admin</userName> <timeCreated>2013-04-10T07:32:06Z</timeCreated> <lastAccessedTime>2013-04-10T07:32:06Z</lastAccessedTime> <machineName>localhost</machineName> <applicationName>Rest Client</applicationName> <timeZoneOffset>+05:30</timeZoneOffset> </sessionInfo></pre>		

terminateSession

Function	Delete a session		
Description	This request terminates the session with the given sessionID.		
Type	POST		
URL Format	.../sessions/[session ID]		
Java API	Method Name		Class Name
	terminateSession		com.quark.qpp.service.facade Class SessionFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/sessions/fee9ea88-68cc-47b9-a576-dab6ad0181eb?op=delete		

getMaxIdleTime

Function	Get the maximum idle time for all sessions		
Description	This request returns the value for maximum idle time setting for all sessions in seconds. User can update the value for the <code>session.maxIdle</code> property by editing the <code>(qpp_home/conf) serverApp.properties</code> file.		
Type	GET		
URL Format	.../sessions		
Java API	Method Name		Class Name
	getMaxIdleTime		com.quark.qpp.service.facade Class SessionFacade
Parameters	Name	Type	Description
	op=maxidletime	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns the maximum idle time set for sessions (for example, 60 seconds (Default)).		
Examples	URL: http://serverName:serverPort/rest/service/sessions?op=maxidletime Result: 60		

validatePassword

Function	Validate a password		
Description	This request validates a password for a user.		
Type	POST		
URL Format	.../sessions/[user name]		
Java API	Method Name		Class Name
	validatePassword		com.quark.qpp.service.facade Class SessionFacade
Parameters	Name	Type	Description
	op=validate	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	password=[password]	string	Password to be validated. (Mandatory)
Returns	Returns a string value of true or false.		
Examples	URL:		
	http://serverName:serverPort/rest/service/sessions/Admin?op=validate&password=Admin		
	Result:		
	True/False		

logOff

Function	Log out		
Description	This request ends the session that you are currently logged into.		
Type	POST		
URL Format	.../sessions/[session ID]		
Java API	Method Name		Class Name
	logOff		com.quark.qpp.service.facade Class SessionFacade
Parameters	Name	Type	Description
	op=logoff	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)This operation will run only for a valid session id.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL:		
	http://serverName:serverPort/rest/service/sessions/fee9ea88-68cc-47b9-a576-dab6ad0181eb?op=logoff		

Asset management

The asset management capabilities of the API allow application developers to manage assets and relations and includes the following REST calls:

- *getAsset*
- *getAssetAllVersions*
- *importAsset*
- *importNewAsset*
- *updateAsset*
- *duplicateAsset*
- *deleteAsset*
- *createAssetRelation*
- *createAssetRelations*
- *getAssetRelations*
- *updateAssetRelation*
- *deleteAssetRelation*
- *deleteAssetRelationsForParent*
- *generateTextIndexesOfAllAssets*
- *generateTextIndexes*
- *getAssetVersionNumbers*
- *getAssetRenditionsInfo*
- *getAvailableRenditions*
- *updateChildVersionForAsset*
- *updateChildAssetVersionForRelationId*
- *updateAllChildAssets Versions*
- *updateChildVersionLockingFlag*
- *checkIn*
- *checkInNew*
- *checkInArticle*
- *checkInNewArticle*
- *checkInRendition*
- *checkOut*
- *abortCheckout*
- *saveDraft*
- *promoteDraft*

- *detectContentType*
- *generateTextIndexes*
- *generateTextIndexesOfAllAssets*

➔ If you are retrieving information about a custom relationship that could be renamed, or if you are retrieving system relationship names that have been localized, use the relationship's ID number instead of the name. To get the IDs of all relationships, use a call to `/admin/relationtypes`.

getAsset

Function	Get all metadata for an asset.		
Description	This request returns all of the metadata associated with an asset.		
Type	GET		
URL Format	<code>.../assetsbypath/[asset path]</code> or <code>/assets/[asset ID]</code>		
Java API	Method Name		Class Name
	<code>getAsset</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>attributes=all</code>	string	(Mandatory)
Returns	Returns an <code>assetInfo</code> object with the metadata of all attributes.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/assets/4?attributes=all</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">lquark2.jpg</attributeValue> </attributeValueList> </assetInfo></pre>		

Function	Get a specific piece of metadata for an asset.		
Description	This request returns a specific piece of metadata associated with an asset.		
Type	GET		
URL Format	<code>.../assetsbypath/[asset path]</code> or <code>/assets/[asset ID]</code>		
Java API	Method Name		Class Name
	<code>getAsset</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>attributes= [comma-separated list of</code>	string	For a list of attributes, use a call to <code>admin/attributes</code> . (Mandatory)

	attribute names or IDs]		
	majorversion and minorversion	string	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	includechildren	boolean	Returns the relation information of all child assets for the given asset. (Optional)
	includeparents	boolean	Returns the relation information of all parent assets for the given asset. (Optional)
Returns	Returns an <code>assetInfo</code> object with the metadata of specified attributes.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?attributes=1,name,3,4,5,6,7,8,54,collection</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> ... </attributeValueList> </assetInfo></pre> <p>...with the majorversion and minorversion parameters</p> <p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/292?attributes=1,name,3,4,5,6,7,8,54,collection&majorversion=1&minorversion=0</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="292"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">292</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">15116462.qxp</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2013-10-16T11:48:52Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-10-16T11:54:56Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue></pre>		

```

    <attributeValue valueId="0" type="numeric" name="Major
version" id="7">1</attributeValue>
    <attributeValue valueId="0" type="text" name="File path"
id="8">/01/24/292.1-0.1.1.qxp</attributeValue>
    <attributeValue valueId="10" type="popup"
name="Workflow" id="54">Workflow1</attributeValue>
    <attributeValue valueId="1" type="popup"
name="Collection" id="55">Home</attributeValue>
  </attributeValueList>
</assetInfo>

```

...with the **includechildren** parameter

URL:

http://serverName:serverPort/rest/service/assets/292?attributes=1, name, 3, 4, 5, 6, 7, 8, 54, collection&includechildren=true

Result:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <assetInfo id="292">
    <assetVersion minorVersion="0" majorVersion="1"/>
    <attributeValueList>
      <attributeValue valueId="0" type="numeric" name="Id"
id="1">292</attributeValue>
      <attributeValue valueId="0" type="text" name="Name"
id="2">15116462.qxp</attributeValue>
      <attributeValue valueId="0" type="dateTime"
name="Created" id="3">2013-10-16T11:48:52Z</attributeValue>
      <attributeValue valueId="0" type="dateTime" name="Last
modified" id="4">2013-10-16T11:54:56Z</attributeValue>
      <attributeValue valueId="1" type="popup" name="Creator"
id="5">Admin</attributeValue>
      <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
      <attributeValue valueId="0" type="numeric" name="Major
version" id="7">1</attributeValue>
      <attributeValue valueId="0" type="text" name="File path"
id="8">/01/24/292.1-0.1.1.qxp</attributeValue>
      <attributeValue valueId="10" type="popup"
name="Workflow" id="54">Workflow1</attributeValue>
      <attributeValue valueId="1" type="popup"
name="Collection" id="55">Home</attributeValue>
    </attributeValueList>
    <assetRelationInfoList>
      <assetRelationInfo relationStateId="1"
relationState="Permanent" relationTypeId="1"
relationType="Primary Attachment" childAssetMinorVersion="2"
childAssetMajorVersion="0" childAssetName="153486e3.qcd"
childAssetId="290" parentAssetName="15116462.qxp"
parentAssetId="292" id="109">
      <attributeValueList>
        <attributeValue valueId="0" type="numeric"
name="Attached Layout Uid" id="401">1</attributeValue>
        <attributeValue valueId="0" type="numeric" name="Box
Uid" id="402">5</attributeValue>
        <attributeValue valueId="0" type="text" name="Page
Name" id="403">Page 1</attributeValue>
        <attributeValue valueId="0" type="numeric" name="Page
Index" id="404">0</attributeValue>
        <attributeValue valueId="0" type="numeric"
name="Attached Component Id" id="412">136</attributeValue>
      </attributeValueList>
    </assetRelationInfo>
  </assetRelationInfoList>

```

```
</assetInfo>
```

...with the **includechildren** and **includeparents** parameters

URL:

```
http://serverName:serverPort/rest/service/assets/290?attributes=1,
name,3,4,5,6,7,8,54,collection&includechildren=true&includeparents=true
```

Result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <assetInfo id="290">
    <assetVersion minorVersion="2" majorVersion="0"/>
    <attributeValueList>
      <attributeValue valueId="0" type="numeric" name="Id"
id="1">290</attributeValue>
      <attributeValue valueId="0" type="text" name="Name"
id="2">153486e3.qcd</attributeValue>
      <attributeValue valueId="0" type="dateTime"
name="Created" id="3">2013-10-16T11:48:14Z</attributeValue>
      <attributeValue valueId="0" type="dateTime" name="Last
modified" id="4">2013-10-16T11:54:57Z</attributeValue>
      <attributeValue valueId="1" type="popup" name="Creator"
id="5">Admin</attributeValue>
      <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
      <attributeValue valueId="0" type="numeric" name="Major
version" id="7">0</attributeValue>
      <attributeValue valueId="0" type="text" name="File path"
id="8">/01/22/290.0-2.1.1.qcd</attributeValue>
      <attributeValue valueId="10" type="popup"
name="Workflow" id="54">Workflow1</attributeValue>
      <attributeValue valueId="1" type="popup"
name="Collection" id="55">Home</attributeValue>
    </attributeValueList>
    <assetRelationInfoList>
      <assetRelationInfo relationStateId="1"
relationState="Permanent" relationTypeId="4"
relationType="Article Component Reference"
childAssetMinorVersion="1" childAssetMajorVersion="0"
childAssetName="Picture2850.jpg" childAssetId="291"
parentAssetName="153486e3.qcd" parentAssetId="290" id="108">
      <attributeValueList>
        <attributeValue valueId="0" type="numeric"
name="Attached Component Id" id="412">136</attributeValue>
        <attributeValue valueId="0" type="numeric"
name="Article Component id" id="305">2</attributeValue>
        <attributeValue valueId="0" type="numeric"
name="Article Id" id="413">290</attributeValue>
      </attributeValueList>
    </assetRelationInfo>
      <assetRelationInfo relationStateId="1"
relationState="Permanent" relationTypeId="1"
relationType="Primary Attachment" childAssetMinorVersion="2"
childAssetMajorVersion="0" childAssetName="153486e3.qcd"
childAssetId="290" parentAssetName="15116462.qxp"
parentAssetId="292" id="109">
      <attributeValueList>
        <attributeValue valueId="0" type="numeric"
name="Attached Layout Uid" id="401">1</attributeValue>
        <attributeValue valueId="0" type="numeric" name="Box
Uid" id="402">5</attributeValue>
        <attributeValue valueId="0" type="text" name="Page
```



```

Name" id="403">Page 1</attributeValue>
  <attributeValue valueId="0" type="numeric"
name="Page Index" id="404">0</attributeValue>
  <attributeValue valueId="0" type="numeric"
name="Attached Component Id" id="412">136</attributeValue>
</attributeValueList>
</assetRelationInfo>
</assetRelationInfoList>
</assetInfo>
    
```

Function	Get a rendition of an asset.		
Description	This request returns a particular rendition of an asset.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	getAsset		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	rendition=highres or rendition=1		Specify the required rendition type. (Optional) If the parameter is not passed the system returns the highest rendition type of the asset. Available renditions types are: <ul style="list-style-type: none"> • HIGHRES • PREVIEW • THUMBNAIL • ARTICLE_PAGE_PICTURE • DECONSTRUCTED_XML • TEXT
	layout=1		Specify layout ID in case of QuarkXPress document or CopyDesk article. (Optional) Layout only applies to a content type of QuarkPress, QuarkXPress Template, InDesign Document and InDesign Document Template. If this parameter is not passed it is defaulted to 1.
	page=2		Specify page in case of QuarkXPress document or CopyDesk article. (Optional) Page applies to all assets that have multi-page renditions. If this parameter is not passed it is defaulted to 1.
	contentDisposition =[option]		Controls whether the response body part displays inline (automatically upon receipt of the message). Defaults to inline. (Optional)

	<code>majorversion=[x] and minorversion=[x]</code>	long	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
Returns	Returns a filestream of required assets based on given parameters.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assetsbypath/Home/15116.qxp?rendition=preview</pre> <p>Result:</p> <p>A filestream of a generated preview of the given asset.</p>		

Function	Get relationship with parent asset.		
Description	This request allows you to fetch the information of an asset and its relationship to parent assets.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	<code>getAsset</code>		<code>com.quark.qpp.service.facade Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>includeparents=true</code>	boolean	Parameter to fetch metadata of its parent assets. (Mandatory)
	<code>relationtypes=[comma-separated list of relationship names or IDs]</code>	long (relation type id) or string (relation type name)	<p>You can retrieve the list of relationship types with <code>/admin/relationtypes</code>. (Optional)</p> <p>If not specified, the metadata related to all types of existing relations will be returned.</p> <p>The available relation types are:</p> <ul style="list-style-type: none"> • Primary Attachment • Secondary Attachment • Article Component Ref Attachment • Overlay Attachment • XML Component Reference • InDesign Behavior Reference • InDesign Hyperlink Reference

Returns	Returns an <code>assetInfo</code> object containing metadata of the relationship to the parent assets.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/[asset id]?includeparents=true</pre> <p>Result:</p> <p>The <code>assetInfoobject</code> containing metadata of the relationship to the parent assets.</p>

Function	Get relationship with child assets.		
Description	This request allows you to fetch the information of an asset and its relationship to its child assets.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	<code>getAsset</code>		<code>com.quark.qpp.service.facade</code> Class <code>AssetFacade</code>
Parameters	Name	Type	Description
	<code>includechildren=true</code>	boolean	Parameter to fetch metadata of its child assets. (Mandatory)
	<code>relationtypes</code> = <code>[comma-separated list of relationship names or IDs]</code>	long (relation type id) or string (relation type name)	<p>You can retrieve the list of relationship types with <code>/admin/relationtypes</code>. (Optional)</p> <p>If not specified, the metadata related to all types of existing relations will be returned.</p> <p>The available relation types are:</p> <ul style="list-style-type: none"> • Primary Attachment • Secondary Attachment • Article Component Ref Attachment • Overlay Attachment • XML Component Reference • InDesign Behavior Reference • InDesign Hyperlink Reference
Returns	Returns an <code>assetRelationInfoList</code> . object containing metadata of relations to parent and child assets. See the table in the “Request/Response parameter formats” chapter.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/[asset id]?includechildren=true</pre>		

	<p>Result:</p> <p>The <code>assetInfo</code> object containing metadata of the relationship to child assets.</p>
--	---

Function	Get relationship with parent and child assets.		
Description	This request allows you to retrieve an assets information and information about its relationship to parent and child assets.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	<code>getAsset</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>includeparents=true&includechildren=true</code>	boolean	Specify if the relation metadata to parent assets or child assets is to be included or not. (Mandatory)
Returns	Returns an <code>assetRelationInfoList</code> . object containing metadata of relations to parent and child assets. See the table in the “Request/Response parameter formats” chapter.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/assets/[assetid]?includechildren=true&includeparents=true</code>		
	Result:		
	The <code>assetInfo</code> object containing metadata of relations to parent and child assets.		

Function	Get a Draft copy of an asset.		
Description	This request returns a draft version of the required asset. If the draft doesn't exists, the last saved version of the asset will be returned.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	<code>getAsset</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>attributes= [comma-separated list of attribute names or IDs]</code>	string	For a list of attributes, use a call to <code>admin/attributes</code> . (Mandatory)
	<code>draft=[true]</code>	boolean	(Mandatory)
Returns	Returns an <code>assetInfo</code> object with the metadata of all attributes.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/assets/4?attributes=all&draft=true</code>		

	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">1quark2.jpg</attributeValue> </attributeValueList> </assetInfo></pre>
--	---

getAssetAllVersions

Function	Get all attributes for all versions of an asset.		
Description	This request returns all attributes for all versions of an asset.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	getAssetAllVersions		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=allversions	string	This operation will list all the metadata of all the versions. (Mandatory)
Returns	Returns an <code>assetInfoList</code> object containing an array of all versions of the same asset with all of its' metadata. See the table in the "Request/Response parameter formats" chapter.		
Examples	URL:		
	http://serverName:serverPort/rest/service/assets/4?op=allversions		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="4"assetVersion minorVersion="1"majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="4" assetVersion minorVersion="2" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

importAsset

Function	Import a new version of an existing asset.		
Description	This request imports the new version of an asset.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	importAsset		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=import	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request will insert an asset into the system without giving its stream.
	attributevalues=<xml/>	string	The set of attribute values. See "Request/Response parameter formats."(Mandatory)
	createminorversion=[true false]	boolean	Defaults to false.(Optional)
	index=[true false]	boolean	Defaults to true.(Optional)
Returns	Returns an <code>assetInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/[asset name]?op=import <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">1quark2.jpg</attributeValue> </attributeValueList> </assetInfo></pre> <p>Result:</p> <p>The <code>assetInfo</code> object in XML format.</p>		

importNewAsset

Function	Import a new asset		
Description	This request imports a new asset.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset name]		
Java API	Method Name		Class Name
	importNewAsset		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=importnew	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request will insert an asset into the system without giving its stream.
	attributevalues=<xml/>	string	The set of attribute values. See "Request/Response parameter formats." (Mandatory)
	index=[true false]	boolean	Defaults to true. (Optional)
Returns	Returns an <code>assetInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/[asset name]?op=importnew</pre> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?></pre> <pre> <assetInfo id="4"></pre> <pre> <assetVersion minorVersion="1" majorVersion="0"/></pre> <pre> <attributeValueList></pre> <pre> <attributeValue type="numeric" name="Id" id="1">4</attributeValue></pre> <pre> <attributeValue type="text" name="Name" id="2">1quark2.jpg</attributeValue></pre> <pre> </pre> <pre> </attributeValueList></pre> <pre> </assetInfo></pre> <p>Result:</p> <p>The <code>assetInfo</code> object in XML format.</p>		

updateAsset

Function	Update repository attributes.
-----------------	-------------------------------

Description	This requests updates the attribute values that are owned by a given repository adaptor.		
Type	POST		
URL Format	.../assets/[asset id]?op=update&repotoken=[repo token]&attributes=....		
Java API	Method Name		Class Name
	updateAsset		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributes=<xml/>	string	The set of attribute values. See "Request/Response parameter formats."(Mandatory)
	repotoken=[token]	GUID	The repository token provided to the repository adaptor during repository adaptor registration. (Mandatory)
	majorversion=[x] and minorversion=[x]	long	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
Returns	Returns an <code>assetInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/24?op=update&attributevalues= <attributeValueList> <attributeValue name="name">blag_new</attributeValue> <attributeValue name="status">Status2</attributeValue> <attributeValue name="workflow">Workflow2</attributeValue> </attributeValueList> &repotoken=[repository token]</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">lquark2.jpg</attributeValue> </attributeValueList></pre>		

	<code></assetInfo></code>
--	---------------------------------

Function	Update an asset's attributes.		
Description	This request allows you to update attributes of specific version of an asset.		
Type	POST		
URL Format	<code>/assetsbypath/[asset path]</code> or <code>/assets/[asset ID]</code>		
Java API	Method Name		Class Name
	<code>updateAsset</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>op=update</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributevalues=<xml/></code>	string	The set of attribute values. See "Request/Response parameter formats."(Mandatory)
Returns	Return an <code>assetInfo</code> object of the specified asset with updated metadata.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/[parent asset id]?op=update&attributevalues= <attributeValueList> <attributeValue name="name">blog_new</attributeValue> <attributeValue name="status">Status2</attributeValue> <attributeValue name="workflow">Workflow2</attributeValue> </attributeValueList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">1quark2.jpg</attributeValue> </attributeValueList> </assetInfo></pre>		

duplicateAsset

Function	Duplicate an asset		
Description	This request creates a duplicate of an asset.		
Type	POST		
URL Format	<code>.../assetsbypath/[asset path]</code> or <code>/assets/[asset ID]</code>		
Java API	Method Name		Class Name
	<code>duplicateAsset</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>

	Name	Type	Description
Parameters	<code>op=duplicate</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributevalues=<xml/></code>	string	The set of attribute values. See "Request/Response parameter formats." The following attributes are Mandatory: <ul style="list-style-type: none"> NAME COLLECTION (There is a choice between collection ID, Name and Path when providing a value for the Collection attribute).
	<code>createminorversion=[true false]</code>	boolean	specify if the duplicated asset is to be created as a Major or Minor Version. Defaults to <code>false</code> . (Optional)
	<code>relationtypes=[comma-separated list of relationship names or IDs]</code>	string or long	You can retrieve the list of relationship types with <code>/admin/relationtypes</code> . (Optional)
Returns	Returns an <code>assetInfo</code> object of the newly created asset in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?op=duplicate&attributevalues= <attributeValueList> <attributeValue name="name">Duplicate_asset</attributeValue> <attributeValue name="collection">1</attributeValue> </attributeValueList> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="11"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">11</attributeValue> <attributeValue type="text" name="Name" id="2">Duplicate_asset</attributeValue> . . . </attributeValueList> </assetInfo></pre> <p>Result:</p> <p>The <code>assetInfo</code> object of the newly created asset in XML format.</p>		

deleteAsset

Function	Delete an asset.
Description	This request deletes a specified asset from the QPP server.

Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	deleteAsset		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request will lock the asset internally and delete it from system.
	majorversion=[x] and minorversion=[x]	long	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	deleteminorver=True/F alse	boolean	Indicates if all minor versions of the given asset version are to be deleted or not. If True - all minor versions (associated with the given major version) of the asset would also be deleted. If False , only the specified version will be deleted. This parameter would be considered only when the given asset version is major. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/25?op=delete</code>		

createAssetRelation

Function	Create a relationship between two assets.		
Description	This request creates a relationship between two assets.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	createAssetRelation		com.quark.qpp.service.facade Class AssetFacade

	Name	Type	Description
Parameters	<code>op=createrelation</code>	string	<p>The op parameter allows you to specify which operation is to be performed. (Mandatory)</p> <p>This operation requires parent asset to be in a Checked-Out state.</p>
	<code>childassetid=[asset id]</code>	long	The ID of the other asset in the relationship. (Mandatory)
	<code>relationtype=[x]</code>	long	<p>1 = primary attachment, 2 = secondary attachment. (Mandatory)</p> <p>The availabel relation types are:</p> <ul style="list-style-type: none"> • Primary Attachment • Secondary Attachment • Article Component Ref Attachment • Overlay Attachment • XML Component Reference • InDesign Behavior Reference • InDesinn Hyperlink Reference
	<code>attributevalues=<xml/></code>	string	<p>The set of attribute values.</p> <p>See “Request/Response parameter formats.”</p> <p>For Secondary Attachment and Overlay Attachment the following attributes are Mandatory:</p> <ul style="list-style-type: none"> • ATTACHED_LAYOUT_ID • BOX_ID • PAGE_NAME • PAGE_INDEX <p>For Primary Attachment the following attribute is also Mandatory ins addition to the above mentioned four attributes:</p> <ul style="list-style-type: none"> • ATTACHED_COMPONENT_ID <p>For ArticleComponentRef Attachment the following two attributes are also Mandatory ins addition to the above mentioned four attributes:</p> <ul style="list-style-type: none"> • ARTICLE_ID • ARTICLE_COMPONENT_ID
	<code>childmajorversion=[x]</code>	long	Specify the required childmajor and

	<p>and childminorversion=[x]</p>		<p>childminor version for which the given rendition of an asset is required. (Optional)</p> <p>These parameters will be used collectively.</p> <p>If neither parameter is passed, the latest version of the asset will be considered.</p> <p>If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.</p>
	<p>lockedtochildversion =[true false]</p>	<p>boolean</p>	<p>Defaults to false. Allows you to pin relationships to a particular version if true. (Optional)</p>
<p>Returns</p>	<p>Returns an <code>assetRelationInfo</code> object containing metadata of the relation, parent and child asset. See the table in the “Request/Response parameter formats” chapter.</p>		
<p>Examples</p>	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/[parent asset Id]?op=createrelation &childassetid=[child asset Id]&relationtype=Secondary Attachment&attributevalues=<attributeValueList> <attributeValue id="401">1</attributeValue> <attributeValue id="402">5</attributeValue> <attributeValue id="403">1</attributeValue> <attributeValue id="404">1</attributeValue> <attributeValue id="412">1</attributeValue> </attributeValueList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetRelationInfo relationStateId="3" relationState="Related Temporarily" relationTypeId="2" relationType="Secondary Attachment" childAssetMinorVersion="2" childAssetMajorVersion="0" childAssetName="1quark2.jpg" childAssetId="4" parentAssetName="15116.qxp" parentAssetId="5" id="13"> <attributeValueList> <attributeValue type="numeric" name="Attached Layout Uid" id="401">1</attributeValue> <attributeValue type="numeric" name="Box Uid" id="402">5</attributeValue> <attributeValue type="text" name="Page Name" id="403">1</attributeValue> <attributeValue type="numeric" name="Page Index" id="404">1</attributeValue> <attributeValue type="numeric" name="Attached Component Id" id="412">1</attributeValue> </attributeValueList> </assetRelationInfo></pre>		

createAssetRelations

Function	Create several relationships between an asset and other assets at one time.		
Description	This request creates several relationships between an asset and other assets in a single request.		
Type	POST		
URL Format	.../assetsbypath or /assets		
Java API	Method Name		Class Name
	createAssetRelations		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=createrelations	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request requires that the parent asset to be in checked-out state.
	relations=<xml/>		See "Request/Response parameter formats." (Mandatory) The relationInfoList returned will contain the metadata of all the relations that have to be created. The following is an example of creating several relationships at one time: Overlay Attachment: <pre><?xml version="1.0" encoding="UTF-8"?> <assetRelationInfoList> <assetRelationInfo id="112" parentAssetId="306" parentAssetName="15116d33.qxp" childAssetId="307" childAssetName="Picture2503.jpg" childAssetMinorVersion="1" relationType="Overlay Attachment" relationTypeId="3" relationState="Related Temporarily" relationStateId="3"> <attributeValueList> <attributeValue id="401" name="Attached Layout Uid" type="numeric">1</attributeValue> <attributeValue id="402" name="Box Uid" type="numeric">17</attributeValue> <attributeValue id="403" name="Page Name" type="text">Page 1</attributeValue> <attributeValue id="404" name="Page Index" type="numeric">0</attributeValue></pre>

		<pre> ue> </attributeValueList> </assetRelationInfo> <assetRelationInfo id="113" parentAssetId="306" parentAssetName="15116d33.qxp" childAssetId="308" childAssetName="Picture2aba.jp g" childAssetMinorVersion="1" relationType="Overlay Attachment" relationTypeId="3" relationState="Related Temporarily" relationStateId="3"> <attributeValueList> <attributeValue id="401" name="Attached Layout Uid" type="numeric">1</attributeVal ue> <attributeValue id="402" name="Box Uid" type="numeric">17</attributeVa lue> <attributeValue id="403" name="Page Name" type="text">Page 1</attributeValue> <attributeValue id="404" name="Page Index" type="numeric">0</attributeVal ue> </attributeValueList> </assetRelationInfo> <assetRelationInfo id="114" parentAssetId="306" parentAssetName="15116d33.qxp" childAssetId="309" childAssetName="Picture2810.jp g" childAssetMinorVersion="1" relationType="Overlay Attachment" relationTypeId="3" relationState="Related Temporarily" relationStateId="3"> <attributeValueList> <attributeValue id="401" name="Attached Layout Uid" type="numeric">1</attributeVal ue> <attributeValue id="402" name="Box Uid" type="numeric">17</attributeVa lue> <attributeValue id="403" name="Page Name" type="text">Page 1</attributeValue> <attributeValue id="404" name="Page Index" type="numeric">0</attributeVal ue> </attributeValueList> </assetRelationInfo> <assetRelationInfo id="115" parentAssetId="306" parentAssetName="15116d33.qxp" childAssetId="310" childAssetName="Picture225d.jp g" childAssetMinorVersion="1" relationType="Overlay Attachment" relationTypeId="3" relationState="Related </pre>
--	--	--

		<pre> Temporarily" relationStateId="3"> <attributeValueList> <attributeValue id="401" name="Attached Layout Uid" type="numeric">1</attributeVal ue> <attributeValue id="402" name="Box Uid" type="numeric">17</attributeVa lue> <attributeValue id="403" name="Page Name" type="text">Page 1</attributeValue> <attributeValue id="404" name="Page Index" type="numeric">0</attributeVal ue> </attributeValueList> </assetRelationInfo> <assetRelationInfo id="116" parentAssetId="306" parentAssetName="15116d33.qxp" childAssetId="311" childAssetName="Picture24da.jp g" childAssetMinorVersion="1" relationType="Overlay Attachment" relationTypeId="3" relationState="Related Temporarily" relationStateId="3"> <attributeValueList> <attributeValue id="401" name="Attached Layout Uid" type="numeric">1</attributeVal ue> <attributeValue id="402" name="Box Uid" type="numeric">17</attributeVa lue> <attributeValue id="403" name="Page Name" type="text">Page 1</attributeValue> <attributeValue id="404" name="Page Index" type="numeric">0</attributeVal ue> </attributeValueList> </assetRelationInfo> </assetRelationInfoList> Primary Attachment: <?xml version="1.0" encoding="UTF-8"?> <assetRelationInfoList> <assetRelationInfo parentAssetId="304" childAssetId="305" relationType="Primary Attachment"> <attributeValueList> <attributeValue id="401">1</attributeValue> <attributeValue id="402">5</attributeValue> <attributeValue id="403">Page 1</attributeValue> <attributeValue id="404">0</attributeValue> </pre>
--	--	--

			<pre><attributeValue id="412">145</attributeValue> </attributeValueList> </assetRelationInfo> </assetRelationInfoList></pre>
Returns	Returns an <code>assetRelationInfo</code> object containing metadata of the relation, parent and child asset. See the table in the "RequestResponse parameter formats" chapter.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/[parent asset id]?op=createrelations</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetRelationInfoList> <assetRelationInfo relationStateId="3" relationState="Related Temporarily" relationTypeId="2" relationType="Secondary Attachment" childAssetMinorVersion="2" childAssetMajorVersion="0" childAssetName="1quark2.jpg" childAssetId="4" parentAssetName="15116.qxp" parentAssetId="24" id="14"> <attributeValueList> <attributeValue type="numeric" name="Attached Layout Uid" id="401">1</attributeValue> <attributeValue type="numeric" name="Box Uid" id="402">5</attributeValue> <attributeValue type="text" name="Page Name" id="403">1</attributeValue> <attributeValue type="numeric" name="Page Index" id="404">1</attributeValue> <attributeValue type="numeric" name="Attached Component Id" id="412">1</attributeValue> </attributeValueList> </assetRelationInfo> </assetRelationInfoList></pre>		

getAssetRelations

Function	Get metadata of the relations between two assets	
Description	This request returns information about the relations between two assets.	
Type	GET	
URL Format	.../assetsbypath/relations/[RelationId] or /assets/relations/[RelationId]	
Java API	Method Name	Class Name
	getAssetRelations	com.quark.qpp.service.facade Class AssetFacade
Parameters	This request requires no input parameters.	
Returns		

Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/relations/1</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetRelationInfo relationStateId="1" relationState="Permanent" relationTypeId="1" relationType="Primary Attachment" childAssetMinorVersion="1" childAssetMajorVersion="0" childAssetName="15348eb3.qcd" childAssetId="51" parentAssetName="15116bb0.qxp" parentAssetId="50" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Attached Layout Uid" id="401">1</attributeValue> <attributeValue valueId="0" type="numeric" name="Box Uid" id="402">5</attributeValue> <attributeValue valueId="0" type="text" name="Page Name" id="403">Page 1</attributeValue> <attributeValue valueId="0" type="numeric" name="Page Index" id="404">0</attributeValue> <attributeValue valueId="0" type="numeric" name="Attached Component Id" id="412">10</attributeValue> </attributeValueList> </assetRelationInfo></pre>
-----------------	---

updateAssetRelation

Function	Update the relation type and attributes of a relationship.		
Description	This request updates an asset relation with the given Id. In order to update the relation type, specify <code>newRelationType</code> and the corresponding attributes to be set. If the relation type is not supplied then it will update only the relation attributes for the current relation type of the given relation.		
Type	POST		
URL Format	.../assets/relations/[relation ID]		
Java API	Method Name		Class Name
	updateAssetRelation		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request requires that the parent asset is in a checked-out state.
	relationtype=[relation type name or ID]	long (relation type id) or string	The new relation type. (Mandatory)

	<pre>attributevalues=<xml/></pre>	(relation type name) string	<p>The set of relation attribute values.</p> <p>See “Request/Response parameter formats.”</p> <p>For Secondary Attachment and Overlay Attachment the following attributes are Mandatory:</p> <ul style="list-style-type: none"> • ATTACHED_LAYOUT_ID • BOX_ID • PAGE_NAME • PAGE_INDEX <p>For Primary Attachment the following attribute is also Mandatory ins addition to the above mentioned four attributes:</p> <p>ATTACHED_COMPONENT_ID</p> <p>For ArticleComponent Ref Attachment the following two attributes are also Mandatory ins addition to the above mentioned four attributes:</p> <ul style="list-style-type: none"> • ARTICLE_ID • ARTICLE_COMPONENT_ID
Returns	Returns an <code>assetRelationInfo</code> object in XML format with an updated set of attributes. See the table in the “Request/Response parameter formats” chapter.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/relations/[relation id]?op=update</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetRelationInfo relationStateId="3" relationState="Related Temporarily" relationTypeId="2" relationType="Secondary Attachment" childAssetMinorVersion="2" childAssetMajorVersion="0" childAssetName="1quark2.jpg" childAssetId="4" parentAssetName="15116.qxp" parentAssetId="5" id="13"> <attributeValueList> <attributeValue type="numeric" name="Attached Layout Uid" id="401">1</attributeValue> <attributeValue type="numeric" name="Box Uid" id="402">5</attributeValue> <attributeValue type="text" name="Page Name"</pre>		

	<pre> id="403">1</attributeValue> <attributeValue type="numeric" name="Page Index" id="404">1</attributeValue> <attributeValue type="numeric" name="Attached Component Id" id="412">1</attributeValue> </attributeValueList> </assetRelationInfo> </pre>
--	---

deleteAssetRelations

Function	Delete a relationship by its id.		
Description	This request deletes a specific relationship between two assets.		
Type	POST		
URL Format	.../assets/relations/[relation ID]		
Java API	Method Name		Class Name
	deleteAssetRelations		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request requires that the parent asset is in a checked-out state.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/assets/relations/[relation id]?op=delete		

deleteAssetRelationsForParent

Function	Delete an asset's relationship.		
Description	This request removes all child relations of a specified asset.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	deleteAssetRelationsForParent		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=deleterelations	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request requires that the parent asset is in a checked-out state.
	childassetid=[x]	long	The ID of the child asset in the

			relationship to be deleted. (Optional) If this parameter is not supplied, all the asset's relation to its child assets will be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/27?op=deleterelations</code>		

generateTextIndexesOfAllAssets

Function	Generate text indexes for all assets.		
Description	This request asynchronously initiates text indexes for all Platform Server assets, including asset attributes and content.		
Type	POST		
URL Format	<code>.../assets/</code>		
Java API	Method Name		Class Name
	<code>generateTextIndexesOfAllAssets</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>op=generatetextindexes</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution, along with the following message: The text indexes generation process has been initiated successfully.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets?op=generatetextindexes</code>		

generateTextIndexes

Function	Generate text indexes for a specified asset.		
Description	This requests initiates text indexes for a specified asset in Platform Server, including asset attributes and content.initiates.		
Type	POST		
URL Format	<code>.../assets/[asset id]?op=generatetextindexes...</code>		
Java API	Method Name		Class Name
	<code>generateTextIndexes</code>		<code>com.quark.qpp.service.facade</code> <code>Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>op=generatetextindexes</code>	string	The op parameter allows you to specify

	s		which operation is to be performed. (Mandatory)
	assetid=[asset Id]	long	The ID of the asset. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/<assetId>?op=generatetextindexes</code>		

getAssetVersionNumbers

Function	Get a list of all versions of an asset.		
Description	This request returns a list of versions for an asset.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	getAssetVersionNumbers		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=listversions	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns an <code>assetVersionList</code> object containing an array of all versions of the same asset.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/4?op=listversions</code> Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetVersionList> <assetVersion minorVersion="1" majorVersion="0"/> <assetVersion minorVersion="2" majorVersion="0"/> </assetVersionList></pre>		

getAssetRenditionsInfo

Function	Get information about an asset's renditions.		
Description	This request returns all information about an asset's renditions.		
Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	getAssetRenditionsInfo		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description

	<code>op=renditioninfo</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>rendition=preview</code> or <code>rendition=2</code>	string	By specifying rendition type, metadata of that rendition will be returned. (Optional) Available renditions are: <ul style="list-style-type: none"> • PREVIEW • THUMBNAIL • ARTICLE_PAGE_PICTURE • DECONSTRUCTED_XML • TEXT
	<code>majorversion=[x]</code> and <code>minorversion=[x]</code>	string	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
Returns	Returns an <code>assetRenditionInfoList</code> object containing an array of all available renditions of the same asset.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?op=renditioninfo</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetRenditionInfoList> <assetRenditionInfo fileSize="80357" fileExtension="jpg" filePath="/00/04/4.0-2.2.1.jpg" repositoryId="1" pageNumber="1" layoutNumber="1" renditionTypeId="2" renditionType="PREVIEW" minorVersion="2" majorVersion="0" assetId="4"/> <assetRenditionInfo fileSize="47142" fileExtension="jpg" filePath="/00/04/4.0-2.3.1.jpg" repositoryId="1" pageNumber="1" layoutNumber="1" renditionTypeId="3" renditionType="THUMBNAIL" minorVersion="2" majorVersion="0" assetId="4"/> </assetRenditionInfoList></pre>		

getAvailableRenditions

Function	Get a list of an asset's available rendition types
Description	This request returns a list of all the rendition types that are applicatble to a specified asset.

Type	GET		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	getAvailableRenditions		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=listrenditiontypes	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	majorversion=[x] and minorversion=[x]	string	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
Returns	Returns a <code>namedList</code> object containing an array of all renditions for the specified asset.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?op=listrenditiontypes</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <nameIdList> <nameId id="2" name="PREVIEW"/> <nameId id="3" name="THUMBNAIL"/> </nameIdList></pre>		

updateChildVersionForAsset

Function	Update the version number of an asset's child asset.		
Description	This request allows you to update the version number of an asset's child asset. The parent asset must be in a checked out state.		
Type	POST		
URL Format	.../assetsbypath/[parent asset path] or /assets/[parent asset ID]		
Java API	Method Name		Class Name
	updateChildVersionForAsset		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=updatechildversion	string	The op parameter allows you to specify which operation is to be performed.

			(Mandatory)This operation requires parent asset to be in a Checked-Out state.
	<code>childassetid=[asset ID]</code>	long	The ID of the child asset. (Mandatory)
	<code>majorversion=[x] and minorversion=[x]</code>	long	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/[parent asset id]?op=updatechildversion&childassetid=4</code>		

updateChildAssetVersionForRelationId

Function	Update the version number of a relationship's child asset.		
Description	This request updates a child asset to a given asset version for the given asset relation.		
Type	POST		
URL Format	<code>.../assets/relations/[relation ID]</code>		
Java API	Method Name		Class Name
	updateChildAssetVersionForRelationId		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	<code>op=updatechildversion</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation requires the parent asset to be in a checked-out state.
	<code>majorversion=[x] and minorversion=[x]</code>	string	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter

		is assumed to be 0.
Returns	A status of 200 is returned in response to a successful request execution.	
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/relations/[relation id]?op=updatechildversion</code>	

updateAllChildAssetsVersion

Function	Update the version number of all of an asset's child assets to the parent asset's current version.		
Description	This request allows you to update the asset versions of all the children of an asset at one time. the parent asset must be in a checked out state.		
Type	POST		
URL Format	<code>.../assetsbypath/[parent asset path] or /assets/[parent asset ID]</code>		
Java API	Method Name		Class Name
	updateAllChildAssetsVersion		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	<code>op=updateallchildversion</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>relationtype=[x]</code>	long (relation type id) or string (relation type name)	You can retrieve the list of relationship types with /admin/relationtypes. (Optional) If the relation types are not specified, then the child asset version will be updated in all the relations with the given parent asset id.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/[parent asset id]?op=updateallchildversion</code>		

updateChildVersionLockingFlag

Function	Update the "locked" flag for a child asset in a relationship.	
Description	This request updates the lockedToChildVersion flag of the given Asset relation.	
Type	POST	
URL Format	<code>.../assets/relations/[relation ID]</code>	
Java API	Method Name	
	updateChildVersionLockingFlag	
	Class Name	
	com.quark.qpp.service.facade Class AssetFacade	

	Name	Type	Description
Parameters	<code>op=updatelockflag</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>flag=[true or false]</code>	boolean	True for locked, false for unlocked. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/assets/relations/19?op=updatelockflag&flag=true</code>		

checkIn

Function	Check in a checked out asset.		
Description	This request checks in a checked out asset. Include the file in the request body or as file part.		
Type	POST		
URL Format	<code>.../assetsbypath/[asset path] or /assets/[asset ID]</code>		
Java API	Method Name		Class Name
	checkIn		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	<code>op=checkin</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request will not work if the asset is not checked out.
	<code>attributevalues=<xml/></code>		See "Request/Response parameter formats." The following attributes are Mandatory: <ul style="list-style-type: none"> Collection (There is a choice between collection ID, Name and Path when providing a value for the Collectionattribute.) Workflow Status Name
	<code>relations=<xml/></code>		See "Request/Response parameter formats." If given a value in this request parameter, the Quark Publishing Platform Server will check-in a checked-out asset and create a relation with the secified parent asset at

			<p>the same time. (Optional)</p> <p>For Secondary Attachment and Overlay Attachment the following attributes are Mandatory:</p> <ul style="list-style-type: none"> • ATTACHED_LAYOUT_ID • BOX_ID • PAGE_NAME • PAGE_INDEX <p>For Primary Attachment the following attribute is also Mandatory ins addition to the above mentioned four attributes:</p> <ul style="list-style-type: none"> • ATTACHED_COMPONENT_ID <p>For ArticleComponent Ref Attachment the following two attributes are also Mandatory ins addition to the above mentioned four attributes:</p> <ul style="list-style-type: none"> • ARTICLE_ID • ARTICLE_COMPONENT_ID
	<pre>createMinorVersion =[true false]</pre>	<p>boolean</p>	<p>Specify whether to create a minor version or not.</p> <p>Defaults to false. (Optional)</p>
<p>Returns</p>	<p>Returns the <code>assetInfo</code> object of the latest version of the asset to be checked into the QPP server.</p>		
<p>Examples</p>	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?op=checkin&attributevalues= <attributeValueList > <attributeValue id="2">1quark2.jpg</attributeValue> <attributeValue id="54">1</attributeValue> <attributeValue id="55">1</attributeValue> <attributeValue id="24">1</attributeValue> <attributeValue id="11">jpg</attributeValue> <attributeValue id="33">1quark2.jpg</attributeValue> <attributeValue id="25">admin</attributeValue> </attributeValueList> <assetInfo id="4"> <assetVersion minorVersion="1" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">1quark2.jpg</attributeValue> </attributeValueList> </assetInfo></pre> <p>Result:</p> <p>The <code>assetInfo</code> object of the latest version of the asset to be checked into the QPP</p>		

server.

checkInNew

Function	Check in a new asset.		
Description	Include the file in the request body or as file part.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[New Asset Name]		
Java API	Method Name		Class Name
	checkInNew		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=checkinnew	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributevalues=<xml/>		See "Request/Response parameter formats." <pre><attributeValueList> <attributeValue id="62"> 1103 </attributeValue> <attributeValue id="54"> 10 </attributeValue> <attributeValue id="24"> 10 </attributeValue> <attributeValue id="11"> jpg </attributeValue> <attributeValue id="33"> Original file </attributeValue> <attributeValue id="25"> 1 </attributeValue> </attributeValueList></pre> <p>The following attributes are Mandatory:</p> <ul style="list-style-type: none"> • Content type (There is a choice between content type ID, Name and Hierarchy when providing a value for the Content type attribute.) • Collection (There is a choice between collection ID, Name and Path when providing a value for the Collection attribute). • Workflow • Status • File extension • Original filename
relations=<xml/>			See "Request/Response parameter formats."

		<p>If this parameter is passed, the QPP Server will check-in the asset and create child relations with the new asset being checked in at the same time. (Optional).</p>
	<p>createmajorversion =[true false]</p>	<p>boolean Specify if this version of the asset is to be checked in to the QPP server as a major or a minor version. (Optional)</p>
<p>Returns</p>	<p>Returns an <code>assetInfo</code> object of the newly checked in asset with its metadata.</p>	
<p>Examples</p>	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assetsbypath/Home/Tulips?attributevalues= <attributeValueList> <attributeValue id="62">System;Asset;Picture;</attributeValue> <attributeValue id="54">10</attributeValue> <attributeValue id="24">10</attributeValue> <attributeValue id="11">jpg</attributeValue> <attributeValue id="33">Original file</attributeValue> <attributeValue id="25">1</attributeValue> </attributeValueList> &op=checkinnew</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="10"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">10</attributeValue> <attributeValue type="text" name="Name" id="2">Tulips</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2013-04-11T05:12:51Z</attributeValue> <attributeValue type="dateTime" name="Last modified" id="4">2013-04-11T05:12:51Z</attributeValue> <attributeValue type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue type="numeric" name="Major version" id="7">1</attributeValue> <attributeValue type="text" name="File path" id="8">/00/0a/10.1-0.1.1.jpg</attributeValue> <attributeValue type="text" name="File extension" id="11">jpg</attributeValue> <attributeValue type="numeric" name="File size" id="12">115191</attributeValue> <attributeValue type="boolean" name="Is checked out" id="13">>false</attributeValue> <attributeValue type="dateTime" name="Last content update" id="19">2013-04-11T05:12:51Z</attributeValue> <attributeValue type="popup" name="Last content updated by" id="20">Admin</attributeValue> <attributeValue type="popup" name="Routed to" id="25">Admin</attributeValue> <attributeValue type="popup" name="Status" id="24">Status1</attributeValue> <attributeValue type="popup" name="Repository type" id="27">fileRepositoryAdapter</attributeValue> <attributeValue type="popup" name="Indexing status" id="29">Pending</attributeValue></pre>	

	<pre> <attributeValue type="popup" name="Repository" id="28">QPS_Repo</attributeValue> <attributeValue type="boolean" name="Is placeholder" id="30">>false</attributeValue> <attributeValue type="popup" name="Content creator" id="32">Admin</attributeValue> <attributeValue type="text" name="Original filename" id="33">Original file</attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue type="popup" name="Content Type" id="62">Picture</attributeValue> <attributeValue type="numeric" name="Minor version" id="61">0</attributeValue> <attributeValue type="boolean" name="Has Children" id="65">>false</attributeValue> <attributeValue type="text" name="Collection Path" id="57">Home</attributeValue> <attributeValue type="text" name="Content Type Hierarchy" id="63">System;Asset;Picture</attributeValue> </attributeValueList> </assetInfo> </pre>
--	--

checkInArticle

Function	Check in a checked-out QuarkCopyDesk article with components.		
Description	This request checks in a checked out article. Include the file in the request body or as file part.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset name]		
Java API	Method Name		Class Name
	checkInArticle		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=checkinarticle	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request will not work if the asset is not checked out.
	attributevalues=<xml/>		See "Request/Response parameter formats." The following attributes are Mandatory: <ul style="list-style-type: none"> Collection (There is a choice between collection ID, Name and Path when providing a value for the Collection attribute.) Workflow Status

	<pre>relations=<xml/></pre>		<ul style="list-style-type: none"> Name <p>See "Request/Response parameter formats."</p> <p>If given a value in this request parameter, the Quark Publishing Platform Server will check-in a checked-out asset and create a relation with the specified parent asset at the same time. (Optional)</p>
	<pre>createminorversion =[true false]</pre>	<p>boolean</p>	<p>Specify whether to create a minor version or not.</p> <p>Defaults to false. (Optional)</p>
Returns	<p>Returns the assetInfo object of the latest version of the asset to be checked into the QPP server.</p>		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assetsbypath/Home/15116d12.qcd?op=checkin&attributevalues= <attributeValueList> <attributeValue id="2">15348a77.qcd</attributeValue> <attributeValue id="54">Workflow1</attributeValue> <attributeValue id="55">Home</attributeValue> <attributeValue id="24">Status1</attributeValue> <attributeValue id="11">qcd</attributeValue> <attributeValue id="33">15348a77.qcd</attributeValue> <attributeValue id="25">1</attributeValue> </attributeValueList> & components= <articleComponentInfoList> <articleComponentInfo componentId="15"> <attributeValueList> <attributeValue name="Component Name">Component 1</attributeValue> <attributeValue name="Content Type">1250</attributeValue> <attributeValue name="Article Component Id">1</attributeValue> </attributeValueList> </articleComponentInfo> <articleComponentInfo componentId="14"> <attributeValueList> <attributeValue name="Component Name">Component2</attributeValue> <attributeValue name="Content Type">1300</attributeValue> <attributeValue name="Article Component Id">2</attributeValue> </attributeValueList> </articleComponentInfo> </articleComponentInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="53"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">53</attributeValue> <attributeValue valueId="0" type="text" name="Name"</pre>		


```

id="2">15116d12.qcd</attributeValue>
  <attributeValue valueId="0" type="dateTime"
name="Created" id="3">2014-04-09T09:51:45Z</attributeValue>
  <attributeValue valueId="0" type="dateTime" name="Last
modified" id="4">2014-04-09T09:51:45Z</attributeValue>
  <attributeValue valueId="1" type="popup" name="Creator"
id="5">Admin</attributeValue>
  <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
  <attributeValue valueId="0" type="numeric" name="Major
version" id="7">1</attributeValue>
  <attributeValue valueId="0" type="text" name="File path"
id="8">/00/35/53.1-0.1.1.qcd</attributeValue>
  <attributeValue valueId="0" type="text" name="File
extension" id="11">qcd</attributeValue>
  <attributeValue valueId="0" type="numeric" name="File
size" id="12">284672</attributeValue>
  <attributeValue valueId="0" type="boolean" name="Is
checked out" id="13">>false</attributeValue>
  <attributeValue valueId="0" type="dateTime" name="Last
content update" id="19">2014-04-09T09:51:45Z</attributeValue>
  <attributeValue valueId="1" type="popup" name="Last
content updated by" id="20">Admin</attributeValue>
  <attributeValue valueId="1" type="popup" name="Routed
to" id="25">Admin</attributeValue>
  <attributeValue valueId="10" type="popup" name="Status"
id="24">Status1</attributeValue>
  <attributeValue valueId="1" type="popup"
name="Repository type"
id="27">fileRepositoryAdapter</attributeValue>
</attributeValueList>
</assetInfo>
    
```

checkInNewArticle

Function	Check in a new QuarkCopyDesk Article with components.		
Description	Include the file in the request body or as file part.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset name]		
Java API	Method Name		Class Name
	checkInNewArticle		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=checkinnewarticle	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributevalues=<xml/>		See "Request/Response parameter formats." <attributeValueList> <attributeValue id="62"> 1103 </attributeValue> <attributeValue id="54"> 10 </attributeValue> <attributeValue id="24"> 10 </attributeValue> <attributeValue id="11"> jpg

			<pre> </attributeValue> <attributeValue id="33"> Original file </attributeValue> <attributeValue id="25"> 1 </attributeValue> </attributeValueList> </pre> <p>The following attributes are Mandatory:</p> <ul style="list-style-type: none"> • Content type (There is a choice between content type ID, Name and Hierarchy when providing a value for the Content type attribute.) • Collection (There is a choice between collection ID, Name and Path when providing a value for the Collection attribute) • Workflow • Status • File extension • Original filename
	<pre>relations=<xml/></pre>		<p>See "Request/Response parameter formats."</p> <p>If this parameter is passed, the QPP Server will check-in the asset and create child relations with the new asset being checked in at the same time. (Optional).</p>
	<pre>createminorversion =[true false]</pre>	boolean	Specify if this version of the asset is to be checked in to the QPP server as a major or a minor version. (Optional)
Returns	Returns an <code>assetInfo</code> object of the newly checked in asset with its metadata.		
Examples	<p>URL:</p> <pre> http://serverName:serverPort/rest/service/assets/15116d12.qcd? op=checkinnew&attributevalues= <?xml version="1.0" encoding="UTF-8"?> <attributeValueList> <attributeValue id="62">11011</attributeValue> <attributeValue name="workflow">Workflow1</attributeValue> <attributeValue name="collection">1</attributeValue> <attributeValue id="24">10</attributeValue> <attributeValue id="11">qcd</attributeValue> <attributeValue id="33">15348.qcd</attributeValue> <attributeValue name="routed to">1</attributeValue> </attributeValueList> &createminorversion=false&components= <articleComponentInfoList> <articleComponentInfo> <attributeValueList> <attributeValue name="Component Name">Component1</attributeValue> <attributeValue name="Content Type">1250</attributeValue> </pre>		

	<pre> <attributeValue name="Article Component Id">1</attributeValue> </attributeValueList> </articleComponentInfo> <articleComponentInfo> <attributeValueList> <attributeValue name="Component Name">Component 2</attributeValue> <attributeValue name="Content Type">1300</attributeValue> <attributeValue name="Article Component Id">2</attributeValue> </attributeValueList> </articleComponentInfo> </articleComponentInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="53"> <assetVersion minorVersion="0"majorVersion="1"/> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">53</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">15116d12.qcd</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2014-04-09T09:51:45Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2014-04-09T09:51:45Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="numeric" name="Major version" id="7">1</attributeValue> <attributeValue valueId="0" type="text" name="File path" id="8">/00/35/53.1-0.1.1.qcd</attributeValue> <attributeValue valueId="0" type="text" name="File extension" id="11">qcd</attributeValue> <attributeValue valueId="0" type="numeric" name="File size" id="12">284672</attributeValue> <attributeValue valueId="0" type="boolean" name="Is checked out" id="13">>false</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last content update" id="19">2014-04-09T09:51:45Z</attributeValue> <attributeValue valueId="1" type="popup" name="Last content updated by" id="20">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Routed to" id="25">Admin</attributeValue> <attributeValue valueId="10" type="popup" name="Status" id="24">Status1</attributeValue> <attributeValue valueId="1" type="popup" name="Repository type" id="27">fileRepositoryAdapter</attributeValue> </attributeValueList> </assetInfo> </pre>
--	---

checkInRendition

Function	Check in a rendition of an asset.
Description	Include the file in the request body or as file part.

Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[assetId]		
Java API	Method Name		Class Name
	checkInRendition		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=checkinrendition	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	rendition=highres or rendition=1	string	Specify the required rendition type. (Optional) If the parameter is not passed the system returns the highres rendition type of the asset. Available renditions types are: PREVIEWTHUMBNAIARTICLE_PAGE_PICTUREDECONSTRUCTED_XMLTEXT
	fileext=[file extension]	string	The file extension of the rendition files being checked in. (Mandatory)
	pages=[pages]	long	The number of pages contained in the rendition being created. (Optional)
	majorversion=[x] and minorversion=[x]	long	Specify the required major and minor version for which the given rendition of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/assets/4?rendition=preview&fileext=jpg&majorversion=0&minorversion=2&op=checkinrendition		

checkOut

Function	Check out an asset	
Description	This request checks out the specified asset and returns its stream.	
Type	POST	
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]	
Java API	Method Name	Class Name
	checkOut	com.quark.qpp.service.facade

			Class <code>AssetFacade</code>
Parameters	Name	Type	Description
	<code>op=checkout</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributevalues=<xml/></code>	string in XML format	See “Request/Response parameter formats.” <pre><attributeValueList> <attributeValue id="16"> E:\REST\QPP Rest\TestData\CheckoutFilePath </attributeValue> <attributeValue id="34"> REST _CLIENT </attributeValue> </attributeValueList></pre> <p>The following attributes are Mandatory:</p> <ul style="list-style-type: none"> CHECKOUT_FILE_PATH CHECKOUT_APPLICATION
Returns	Returns an asset stream after marking it checked out.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?attributevalues= <attributeValueList> <attributeValue id="16">E:\REST\QPP Rest\TestData\CheckoutFilePath</attributeValue> <attributeValue id="34">REST_CLIENT</attributeValue> </attributeValueList> &op=checkout</pre> <p>Result:</p> <p>An asset stream after marking it checked out.</p>		

abortCheckout

Function	Abort checkout of an asset		
Description	This request aborts the check-out of an asset that is marked as checked-out.		
Type	POST		
URL Format	.../assetsbypath/[asset path] or /assets/[asset ID]		
Java API	Method Name		Class Name
	<code>abortCheckout</code>		<code>com.quark.qpp.service.facade</code> Class <code>AssetFacade</code>
Parameters	Name	Type	Description
	<code>op=abortcheckout</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory) This request will not work if the asset is not checked out.

Returns	Returns an <code>assetInfo</code> object in XML format.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/4?op=abortcheckout <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="4"> <assetVersion minorVersion="2" majorVersion="0"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">4</attributeValue> <attributeValue type="text" name="Name" id="2">lquark2.jpg</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2013-04-10T09:37:56Z</attributeValue> </attributeValueList> </assetInfo></pre> <p>Result:</p> <p>The <code>assetInfo</code> object in XML format.</p>

saveDraft

Function	Saves the draft copy of an asset.		
Description	This request saves the asset without creating a new version of the asset.		
Type	POST		
URL Format	/assets/[asset ID]		
Java API	Method Name		Class Name
	saveDraft		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=savedraft	string	Include the file in the request body or as file part. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/{asset Id}?op=savedraft</pre>		

promoteDraft

Function	Promotes the draft version to the actual version of an asset.		
Description	This request saves the asset without creating a new version of the asset.		
Type	POST		
URL Format	/assets/[asset ID]		
Java API	Method Name		Class Name
	promoteDraft		com.quark.qpp.service.facade Class AssetFacade

	Name	Type	Description
Parameters	<code>op=promotedraft</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributes= [comma-separated list of attribute names or IDs]</code>	string	List of attribute values to be set for this new asset version. For a list of attributes, use a call to admin/attributes. (Mandatory)
	<code>Createminorversion={true false}</code>	string	By default value is false.
Returns	Returns an <code>assetInfo</code> object along with a list of attribute values for this new asset version.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets/{assetId}?op=promotedraft</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="40"> <assetVersion minorVersion="0" majorVersion="3"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">40</attributeValue> <attributeValue type="text" name="Name" id="2">abc.xlsx</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2014-08-18T07:23:30Z</attributeValue> </attributeValueList/> </assetInfo></pre>		

detectContentType

Function	Detects content type of provided asset. Pass asset as stream in the request body.		
Description	This request detects the content type of a provided asset.		
Type	POST		
URL Format	.../assets		
Java API	Method Name		Class Name
	<code>detectContentType</code>		<code>com.quark.qpp.service.facade Class AssetFacade</code>
Parameters	Name	Type	Description
	<code>fileextension</code>	string	The file extension of the asset.
	<code>mimetype</code>	string	The mime type of the asset.
Returns			
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/assets?op=detectcontenttype</pre>		

	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <contentTypeInfo id="11011"> <name>QuarkCopyDesk Article</name> <contentTypeHierarchy>System;Asset;Document;QuarkCopyDesk Article;</contentTypeHierarchy> <systemDefined>true</systemDefined> <childrenAvailable>false</childrenAvailable> <extensible>true</extensible> </contentTypeInfo></pre>
--	--

Collection management

The collection management capabilities of the API allow application developers to manage collections and includes the following REST calls:

- *createCollection*
- *createCollections*
- *getCollection*
- *getCollectionHierarchy*
- *getCollectionDefaultJobJacket*
- *getCollectionDefaultTicket*
- *moveCollection*
- *duplicateCollection*
- *updateCollection*
- *updateCollections*
- *deleteCollection*
- *deleteCollectionWithAssets*
- *createJobJackets*
- *getJobJacket*
- *getAllJobJackets*
- *getJobJacketContent*
- *updateJobJacket*
- *updateJobJackets*
- *parseJobJacketXml*
- *deleteJobJacket*
- *getApplicableRoutingUsers*
- *getApplicablePrivileges*
- *getApplicableRoutingGroups*
- *getApplicableAttributeConstraints*

- [getApplicableStatusTransition](#)
- [getApplicableWorkflowInitialStatuses](#)

createCollection

Function	Create a collection.		
Description	This request creates a new collection according to the given attributes.		
Type	POST		
URL Format	.../collections/[collection name] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	createCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	targetparent=[collection path or ID]	long or string	The parent collection for the new collection. (Mandatory) There is a choice between collection ID , Name and Path when providing a value.
	attributevalues=<xml/>	string	The set of attribute values. See "Request/Response parameter formats."(All attributes are Optional)
	contenttype=[content type name]	string	Defaults to GENERAL . The default content type IDs are documented in the JavaDoc SDK at <code>com.quark.qpp.core.content.service.constants.DefaultContentTypes</code> . To get a list of content types, use a call to <code>../admin/contenttypes</code> .(Optional) There is a choice between content type ID , Name and Hierarchy when providing a value.
Returns	Returns the <code>collectionInfo</code> object of the newly created collection.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/newChildCollection? op=create&targetparent=Home/childCollection</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfo templateId="0" collectionClass="General"</pre>		

```

collectionClassId="1416"
collectionTemplate="false"
accessibleChildrenAvailable="true"
domainId="19"
name="newChildCollection"
id="15">
  <attributeValueList>
    <attributeValue type="numeric"
      name="Id"
      id="1">1</attributeValue>
    ...
  </attributeValueList>
  <workflowInfoList>
    <workflowInfo id="10">
      <name>Workflow1</name>
    ...
    <statusInfoList>
      <statusInfo id="10">
        <name>Status1</name>
      ...
    </statusInfo>
  </statusInfoList>
</workflowInfo>
....
....
</collectionInfo>

```

Function	Create a collection template.		
Description	This request creates a new collection template under the given parent collection.		
Type	POST		
URL Format	.../collections/[collection name] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	createCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	istemplate=[true false]	boolean	Defaults to false . (Mandatory)
	targetparent=[collection path or ID]	long or string	The parent collection for the new collection. (Mandatory)
	attributevalues=<xml/>	string	The set of attribute values. See "RequestResponse parameter formats." All attributes are Optional.
	contenttype=[content type name]	string	Defaults to GENERAL . The default content type IDs are documented in the JavaDoc SDK at com.quark.qpp.core.content.service.constants.DefaultContentTypes.

			To get a list of content types, use a call to <code>../admin/contenttypes</code> . (Optional)
Returns	Returns the <code>collectionInfo</code> object of the newly created collection template.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/newChildCollection? op=create&targetparent=Home/childCollection&istemplate=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfo templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" name="newChildCollection" id="15"> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">1</attributeValue> ... </attributeValueList> <workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> ... </workflowInfoList> <statusInfoList> <statusInfo id="10"> <name>Status1</name> ... </statusInfoList> </workflowInfo> </collectionInfo></pre>		

createCollections

Function	Create multiple collections.		
Description	This request creates multiple collections in a single request.		
Type	POST		
URL Format	<code>.../collections</code>		
Java API	Method Name	Class Name	
	<code>createCollections</code>	<code>com.quark.qpp.service.facade</code> Class <code>CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>op=create</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.

	collectioninfoList=<xml/>		See "Request/Response parameter formats."
Returns	Returns a <code>collectioninfoList</code> object in XML format that contains the metadata of all successfully created collections.		
Examples	<p>URL:</p> <pre>https://serverName:serverPort/rest/service/collections?op=create&collectioninfoList=<?xml version="1.0" encoding="UTF-8"?> <collectionInfoList> <collectionInfo collectionClassId="1416" collectionTemplate="false" collectionPath="Home/Home3" > <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">Home3</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">>false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home/Home1</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> </collectionInfo> </collectionInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId ="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="false" domainId="19" collectionPath="Home/Home3" name="Home3" id="23"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">23</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home3</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2014-04-10T06:09:46Z</ attributeValue > </attributeValueList> </collectionInfo> </collectionInfoList></pre>		

	<pre> <routingInfoList/> <userInfoList> <userInfo id="1"> <name>Admin</name > </userInfo> </userInfoList> <groupInfoList/> <jobJacketInfoList> <jobJacketInfo id="1"> <name>Default Job Jacket</name> </jobJacketInfo> </jobJacketInfoList> </collectionInfoList> </collectionInfo> </collectionInfoList> </pre>
--	--

getCollection

Function	Get all of a collection's metadata.	
Description	This request returns all of the metadata associated with an entire collection.	
Type	GET	
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]	
Java API	Method Name	Class Name
	getCollection	com.quark.qpp.service.facade Class CollectionFacade
Parameters	This request requires no input parameters.	
Returns	Returns a collectionInfo object with all its metadata and a list of its attributes.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1</pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" </pre>	

```

name="Created" id="3">2010-04-01T10:00:00Z</attributeValue>
  <attributeValue valueId="0" type="dateTime" name="Last
modified" id="4">2013-12-23T12:07:27Z</attributeValue>
  <attributeValue valueId="1" type="popup"
name="Creator" id="5">Admin</attributeValue>
  <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
  <attributeValue valueId="0" type="boolean" name="Is
Template" id="59">>false</attributeValue>
  <attributeValue valueId="0" type="text"
name="Collection Path" id="57">Home</attributeValue>
  <attributeValue valueId="1416" type="popup"
name="Content Type" id="62">General</attributeValue>
  <attributeValue valueId="0" type="text" name="Content
Type Hierarchy"
id="63">System;Collection;General</attributeValue>
</attributeValueList>
</collectionInfo>
</collectionInfoList>

```

Function	Get specific collection attributes.		
Description	This request returns all of the attributes associated with a specific collection.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	attributes=[attribute name or ID], [attribute name or ID],...	string	To get a list of all attribute IDs, use a call to ../admin/attributes. (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the specified attributes.		
Examples	URL:		
	http://serverName:serverPort/rest/service/collections/1?attributes=1,name,creator		
Examples	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfo id="1"> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">1</attributeValue> <attributeValue type="text" name="Name" id="2">Home</attributeValue> <attributeValue type="popup" name="Creator" id="5">Admin</attributeValue> </attributeValueList> </collectionInfo> </pre>		

Function	Get a collection's workflows.
Description	This request returns all of the workflows associated with a collection.

Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	workflows=[true false]	boolean	Defaults to false. (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the metadata of the workflows only.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?workflows=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-12-23T12:07:27Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">>false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> <workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> <domainId>18</domainId> </workflowInfo> </workflowInfoList> </collectionInfo> </collectionInfoList></pre>		

```

<statusInfoList>
  <statusInfo id="10">
    <name>Status1</name>
    <domainId>6</domainId>
    .....
  </statusInfo>
</statusInfoList>
<contentTypeInfoList>
  <contentTypeInfo id="1100">
    .....
  </contentTypeInfo>
  <contentTypeInfo id="1101">
    .....
  </contentTypeInfo>
</contentTypeInfoList>
</workflowInfo>
</workflowInfoList>
</collectionInfo>
</collectionInfoList>

```

Function	Get a collection’s workflow for a specific content type.		
Description	This request returns all of the workflows associated with a specific content type.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	getCollection	com.quark.qpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	workflows=[true false]	boolean	Defaults to false. (Mandatory)
	contenttypes=[content type ID] or [content type name]	long	The default content type IDs are documented in the JavaDoc SDK at com.quark.qpp.core.content.service.constants.DefaultContentTypes. To get a list of content types, use a call to ../admin/contenttypes.(Optional) If not provided, the metadata of all workflows will be returned.
Returns	Returns a collectionInfo object with the metadata of the workflows associated with the specified content type.		
Examples	URL: http://serverName:serverPort/rest/service/collections/1?workflows=true&contenttype=Picture		
	Result: <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" > </pre>		


```

defaultJobJacketId="1"
templateId="0"
collectionClass="General"
collectionClassId="1416"
collectionTemplate="false"
accessibleChildrenAvailable="true"
domainId="19"
collectionPath="Home"
name="Home"
id="1">
  <attributeValueList>
    <attributeValue valueId="0" type="numeric" name="Id"
id="1">1</attributeValue>
    <attributeValue valueId="0" type="text" name="Name"
id="2">Home</attributeValue>
    <attributeValue valueId="0" type="dateTime"
name="Created" id="3">2010-04-01T10:00:00Z</attributeValue>
    <attributeValue valueId="0" type="dateTime"
name="Last modified" id="4">2013-12-
23T12:07:27Z</attributeValue>
    <attributeValue valueId="1" type="popup"
name="Creator" id="5">Admin</attributeValue>
    <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
    <attributeValue valueId="0" type="boolean" name="Is
Template" id="59">false</attributeValue>
    <attributeValue valueId="0" type="text"
name="Collection Path" id="57">Home</attributeValue>
    <attributeValue valueId="1416" type="popup"
name="Content Type" id="62">General</attributeValue>
    <attributeValue valueId="0" type="text" name="Content
Type hierarchy"
id="63">System;Collection;General</attributeValue>
  </attributeValueList>
  <workflowInfoList>
    <workflowInfo id="10">
      <name>Workflow1</name>
      <domainId>18</domainId>
      .....
    <statusInfoList>
      <statusInfo id="10">
        <name>Status1</name>
        <domainId>6</domainId>
        .....
      </statusInfo>
    </statusInfoList>
    <contentTypeInfoList>
      <contentTypeInfo id="1103">
        .....
      </contentTypeInfo>
    </contentTypeInfoList>
  </workflowInfo>
</workflowInfoList>
</collectionInfo>
</collectionInfoList>

```

Function	Get a collection's users.
Description	This request returns a list of the users of a collection.
Type	GET
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]

Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	users=[true false]	boolean	Defaults to false . (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the metadata of users only.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?users=true</pre>		
	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-12- 23T12:07:27Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">>false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> <userInfoList> <userInfo id="1"> <name>Admin</name> <firstName>Administrator</firstName> <nativeTrustee>trie</nativeTrustee> <enabled>>true</enabled> <defaultUserClassid>1</defaultUserClassid> <IdapProfileId>0</firstIdapProfileIdtName> <deleted>>false</deleted> </pre>		

```

        <ldapDistinguishedName>Dist
Name</ldapDistinguishedName>
    </userInfo>
    .....
</userInfoList>
</collectionInfo>
</collectionInfoList>
    
```

Function	Get a collection's groups.		
Description	This request returns a list of the groups associated with a collection.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	getCollection	com.quark.qpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	groups=[true false]	boolean	Defaults to false . (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the metadata of groups only.		
Examples	URL:		
	http://serverName:serverPort/rest/service/collections/1?groups=true		
Examples	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-12- 23T12:07:27Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">>false</attributeValue> <attributeValue valueId="0" type="text" </pre>		

```

name="Collection Path" id="57">Home</attributeValue>
  <attributeValue valueId="1416" type="popup"
name="Content Type" id="62">General</attributeValue>
  <attributeValue valueId="0" type="text" name="Content
Type Hierarchy"
id="63">System;Collection;General</attributeValue>
</attributeValueList>
<groupInfoList>
  <groupInfo id="12">
    <name>Group1</name>
    <emailAddress>Editors@company.com</emailAddress>
    <nativeTrustee>>false</nativeTrustee>
    <defaultUserClassId>2</defaultUserClassId>
    <defaultUserClass>NoPrivileges</defaultUserClass>
    <ldapProfileId>10</ldapProfileId>
    <ldapProfile>Profile Name</ldapProfile>
    <ldapDistinguishedName>Dist
Name</ldapDistinguishedName>
    <memberList>
      <nameId id="1">
        <name>Admin</name>
      </nameId>
      .....
    </memberList>
  </groupInfo>
</collectionInfo>
</collectionInfoList>

```

Function	Get the routings for a collection's workflow.		
Description	This request returns all of the routings for a collection's workflow.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	routings=[true false]	boolean	Defaults to false . (Mandatory)
	workflow=[workflow name]	long or string	Specify the workflow name or ID . (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the metadata of the routings of the specified workflow.		
Examples	URL:		
	http://serverName:serverPort/rest/service/collections/1?routings=true&workflow=Workflow1		
	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" </pre>		

```

collectionClass="General"
collectionClassId="1416"
collectionTemplate="false"
accessibleChildrenAvailable="true"
domainId="19"
collectionPath="Home"
name="Home"
id="1">
  <attributeValueList>
    <attributeValue valueId="0" type="numeric" name="Id"
id="1">1</attributeValue>
    <attributeValue valueId="0" type="text" name="Name"
id="2">Home</attributeValue>
    <attributeValue valueId="0" type="dateTime"
name="Created" id="3">2010-04-01T10:00:00Z</attributeValue>
    <attributeValue valueId="0" type="dateTime"
name="Last modified" id="4">2013-12-
23T12:07:27Z</attributeValue>
    <attributeValue valueId="1" type="popup"
name="Creator" id="5">Admin</attributeValue>
    <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
    <attributeValue valueId="0" type="boolean" name="Is
Template" id="59">false</attributeValue>
    <attributeValue valueId="0" type="text"
name="Collection Path" id="57">Home</attributeValue>
    <attributeValue valueId="1416" type="popup"
name="Content Type" id="62">General</attributeValue>
    <attributeValue valueId="0" type="text" name="Content
Type Hierarchy"
id="63">System;Collection;General</attributeValue>
  </attributeValueList>
  <routingInfoList>
    <routingInfo>
      <trusteeId>1</trusteeId>
      <trustee>Admin</trustee>
      <routingTypeId>1</routingTypeId>
      <routingType>User</routingType>
      <workflowId>10</workflowId>
      <workflow>Workflow1</workflow>
      <statusId>10</statusId>
      <status>Status1</status>
    </routingInfo>
    .....
  </routingInfoList>
</collectionInfo>
</collectionInfoList>

```

Function	Get a collection's revision controls.		
Description	This request returns all of the revision controls for a collection.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	getCollection	com.quark.qpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	revisioncontrols=[true false]	boolean	Defaults to false . (Mandatory)

Returns	Returns a <code>collectionInfo</code> object with metadata of the revision controls settings.
	Examples
<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?revisioncontrols=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-12- 23T12:07:27Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> <revisionControlInfoList> <revisionControlInfo contentType="Asset" contentTypeId="1100"> <minorVersioningAllowed>true</minorVersioningAllowed> <allRevisions>true</allRevisions> <initialRevisionsCount>0</initialRevisionsCount> <recentRevisionsByPeriod>false</recentRevisionsByPeriod> <recentRevisionsCount>0</recentRevisionsCount> <recentRevisionsDuration>0</recentRevisionsDuration> </revisionControlInfo> </revisionControlInfoList> </collectionInfo> </collectionInfoList></pre>	

Function	Get a collection's Job Jackets files.		
Description	This request returns all of the job jackets for a collection.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	jobjackets=[true false]	boolean	Defaults to false . (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the metadata of associated job jackets.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?jobjackets=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-12-23T12:07:27Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> <jobJacketInfoList> <jobJacketInfo id="1"> <name>Default Job Jacket</name></pre>		

```

        <domainId>0</domainId>
        <defaultTicketTemplate>Default Job
Ticket</defaultTicketTemplate>
        .....
        <ticketTemplateList>
        <ticketTemplate>Default Job
Ticket</ticketTemplate>
        </ticketTemplateList>
        <content>
        .....
        </content>
        </jobJacketInfo>
        </jobJacketInfoList>
        </collectionInfo>
        </collectionInfoList>
    
```

Function	Get a collection's child collections.		
Description	This request returns immediate child collections accessible to the logged on user.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	getCollection	com.quark.qpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	childcollections =[true false]	boolean	Defaults to false . (Mandatory)
Returns	Returns a <code>collectionInfo</code> object with the list of child collections that is accessible to the logged-in user.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?childcollections=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1"></attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime"</pre>		


```

name="Created" id="3">2010-04-01T10:00:00Z</attributeValue>
  <attributeValue valueId="0" type="dateTime"
name="Last modified" id="4">2013-12-
23T12:07:27Z</attributeValue>
  <attributeValue valueId="1" type="popup"
name="Creator" id="5">Admin</attributeValue>
  <attributeValue valueId="1" type="popup" name="Last
modifier" id="6">Admin</attributeValue>
  <attributeValue valueId="0" type="boolean" name="Is
Template" id="59">>false</attributeValue>
  <attributeValue valueId="0" type="text"
name="Collection Path" id="57">Home</attributeValue>
  <attributeValue valueId="1416" type="popup"
name="Content Type" id="62">General</attributeValue>
  <attributeValue valueId="0" type="text" name="Content
Type Hierarchy"
id="63">System;Collection;General</attributeValue>
</attributeValueList>
<collectionInfoList>
  <collectionInfo
  defaultJobJacketTicket="Default Job Ticket"
  defaultJobJacketName="Default Job Jacket"
  defaultJobJacketId="1"
  templateId="0"
  collectionClass="General"
  collectionClassId="1416"
  collectionTemplate="false"
  accessibleChildrenAvailable="true"
  domainId="19"
  collectionPath="Home/Collection"
  name="Collection"
  id="10">
  <attributeValueList>
    <attributeValue valueId="0" type="numeric"
name="Id" id="1">10</attributeValue>
    .....
  </attributeValueList>
</collectionInfo>
<collectionInfo
  templateId="0"
  collectionClass="General"
  collectionClassId="1416"
  collectionTemplate="false"
  accessibleChildrenAvailable="false"
  domainId="19"
  collectionPath="Home/Collection_84546"
  name="Collection_84546"
  id="13">
  <attributeValueList>
    <attributeValue valueId="0" type="numeric"
name="Id" id="1">13</attributeValue>
    .....
  </attributeValueList>
</collectionInfo>
</collectionInfoList>
</collectionInfo>
</collectionInfoList>

```

Function	Get a collection's parent collection.
Description	This request returns all of the accessible parent collections for a collection.
Type	GET

URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	parentcollections =[true false]	boolean	Defaults to false . (Mandatory)
Returns	Returns a <code>collectionInfo</code> object of ancestor collections up the hierarchy based on access control. While going up the hierarchy, if there exists any inaccessible collection (for which the logged on user does not have access to), the collection path array will contain <code>collectionInfo</code> objects of only those collections which are lying below that collection.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/10?parentcollections=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2013-12- 23T12:07:27Z</attributeValue> <attributeValue valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">>false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> </collectionInfo> </collectionInfoList></pre>		

```

<collectionInfo
  defaultJobJacketTicket="Default Job Ticket"
  defaultJobJacketName="Default Job Jacket"
  defaultJobJacketId="1"
  templateId="0"
  collectionClass="General"
  collectionClassId="1416"
  collectionTemplate="false"
  accessibleChildrenAvailable="true"
  domainId="0"
  collectionPath="Home"
  name="Home"
  id="1">
  <attributeValueList>
    <attributeValue valueId="0" type="numeric"
name="Id" id="1">1</attributeValue>
    <attributeValue valueId="0" type="text"
name="Name" id="2">Home</attributeValue>
    . . . . .
  </attributeValueList>
</collectionInfo>
</parentColInfoList>
</collectionInfo>
</collectionInfoList>

```

Function	Get collection assets and specify asset display attributes.		
Description	This request returns collection assets by considering asset display attributes.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	getCollection	com.quark.gpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	assets=[true]	boolean	(Mandatory)
	assetattributes	long or string	An array of display attributes IDs or names to be used while querying for assets within a collection.
Returns	Returns a collectionInfoList object containing a list of collections assets.		
Examples	URL:		
	http://serverName:serverPort/rest/service/collections/1?assets=true&assetattributes=Name,collection		
	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo id="1" name="Home" collectionPath="Home" domainId="19" accessibleChildrenAvailable="true" collectionTemplate="false" collectionClassId="1416" </pre>		

```

collectionClass="General"
templateId="0"
defaultJobJacketId="1"
defaultJobJacketName="Default Job Jacket"
defaultJobJacketTicket="Default Job Ticket"
<attributeValueList>
  <attributeValue id="1" name="Id"
type="numeric">1</attributeValue>
  .....
</attributeValueList>
<assetInfoList>
  <assetInfo id="10">
  <attributeValueList>
    <attributeValue id="2" name="Name" type="text">
Workflows.json.txt </attributeValue>
    <attributeValue id="55" name="Collection"
type="popup" valueId="1" multiValued="false"> Home
  </attributeValue>
  </attributeValueList>
  </assetInfo>
  <assetInfo id="11">
  <attributeValueList>
    <attributeValue id="2" name="Name" type="text">
Workflows.json.txt </attributeValue>
    <attributeValue id="55" name="Collection"
type="popup" valueId="1" multiValued="false"> Home
  </attributeValue>
  </attributeValueList>
  </assetInfo>
</assetInfoList>
</collectionInfo>
</collectionInfoList

```

Function	Get all collections in the hierarchy of the given collection along with assets and specific asset display attributes.		
Description	This request returns a complete collection hierarchy of the given collection along with collection assets with specific asset display attributes.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	hierarchy=[true]	boolean	(Mandatory)
	assets=[true]	boolean	(Mandatory)
	assetattributes	long or string	An array of display attributes IDs or names to be used while querying for assets within a collection.
Returns	Returns a collectionInfoList object containing all collections in the hierarchy of a given collection along with assets specific metadata as specified in the assetattributes parameter.		
Examples	URL:		

	<p><code>http://serverName:serverPort/rest/service/collections/1?hierarchy=true& assets=true&assetattributes=Name,collection</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo id="1" name="Home" collectionPath="Home"> <assetInfoList> <assetInfo id="10"> </assetInfoList> </collectionInfoList> <collectionInfo id="10" name="Demo" collectionPath="Home/Demo"> <assetInfoList> <assetInfo id="1"> <attributeValueList> <attributeValue id="2" name="Name" type="text">Hydrangeas.PNG </attributeValue> <attributeValue id="55" name="Collection" type="popup" valueId+'10" multivalued="false">Demo</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="3"> </assetInfoList> </collectionInfo> </collectionInfoList> </collectionInfo></pre>
--	---

getCollectionHierarchy

Function	Get the metadata of each collection in the hierarchy.		
Description	This request returns all of the metadata associated with each collection in the hierarchy		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	getCollectionHierarchy	com.quark.qpp.service.facade Class CollectionFacade	
Parameters Returns	Name	Type	Description
	hierarchy=[true false]	boolean	Defaults to true. (Mandatory)
	attributes=[attribute name or ID], [attribute name or ID],...	string	To get a list of all attribute IDs, use a call to ../admin/attributes. (Optional)
	workflows=[true false]	boolean	Defaults to false. (Optional)
	users=[true false]	boolean	Defaults to false. (Optional)
	groups=[true false]	boolean	Defaults to false. (Optional)

	<code>routings=[true false]</code>	boolean	Defaults to false . (Optional)
	<code>revisioncontrols=[true false]</code>	boolean	Defaults to false . (Optional)
	<code>jobjackets=[true false]</code>	boolean	Defaults to false . (Optional)
	<code>contenttypes=[content type ID] or [content type name]</code>	string	<p>The default content type IDs are documented in the JavaDoc SDK at <code>com.quark.qpp.core.content.service.constants.DefaultContentTypes</code>. To get a list of content types, use a call to <code>../admin/contenttypes</code>.(Optional)</p> <p>This parameter can be concatenated with above mentioned parameters workflows and revision controls.</p>
	Returns a <code>collectionInfoList</code> object with the metadata of each collection in the hierarchy.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?hierarchy=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <collectionInfoList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">1</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home</attributeValue> </attributeValueList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" collectionPath="Home/Collection" name="Collection" id="10"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">10</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Collection</attributeValue> </attributeValueList> <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="Book" collectionClassId="1403" collectionTemplate="false" accessibleChildrenAvailable="false" domainId="19" collectionPath="Home/Collection/123" name="123" id="14"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">14</attributeValue></pre>		

	<pre><attributeValue valueId="0" type="text" name="Name" id="2">123</attributeValue> </attributeValueList> </collectionInfo> </collectionInfo> </collectionInfo> </collectionInfoList></pre>
--	--

getCollectionDefaultJobJacket

Function	Get the default Job Jackets file for a collection.		
Description	This request returns the default job jacket set for a collection. Returns null if no job jacket is specified as default for this collection or the content of the job jacket is not found.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getCollectionDefaultJobJacket		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=defaultjobjacket	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns a <code>jobJacketInfo</code> object with its metadata.		
Examples	URL:		
	http://serverName:serverPort/rest/service/collections/1?op=defaultjobjacket		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <jobJacketInfo id="1"> <name>Default Job Jacket</name> <domainId>0</domainId> <defaultTicketTemplate>Default Job Ticket</defaultTicketTemplate> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <created>2008-06- 01T10:00:00Z</created> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2008- 06-01T10:00:00Z</lastModified> <ticketTemplateList> <ticketTemplate>Default Job Ticket</ticketTemplate> </ticketTemplateList> <content> ... </content> </jobJacketInfo></pre>		

getCollectionDefaultTicket

Function	Get the default Job Ticket for a collection.	
Description	This request returns the default Job Ticket for the collection.	
Type	GET	
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]	
Java API	Method Name	Class Name
	getCollectionDefaultTicket	com.quark.qpp.service.facade

			Class <code>CollectionFacade</code>
Parameters	Name	Type	Description
	<code>op=defaultticket</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns the name of the default job ticket for a collection.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/collections/1?op=defaultticket</code>		
Examples	Result:		
	Default Job Ticket. The name of the default job ticket for a collection.		

moveCollection

Function	Move a collection.		
Description	This request moves a collection or collection template from one level to another.		
Type	POST		
URL Format	<code>.../collections/[collection ID] or collectionsbypath/[collection path]</code>		
Java API	Method Name		Class Name
	<code>moveCollection</code>		<code>com.quark.qpp.service.facade</code> Class <code>CollectionFacade</code>
Parameters	Name	Type	Description
	<code>op=move</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request moves a collection along with its child collections and assets under a new parent collection.
	<code>targetparent=[collection path or ID]</code>	long or string	The collection to which this collection should be moved. (Mandatory)
Returns	Returns a <code>collectionInfo</code> object of the collection or the collection template with updated metadata.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/collections/colTemplate? op=move&targetparent=Home/Collection/XDK</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfo templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="true" accessibleChildrenAvailable="true" domainId="19" name="colTemplate" id="16"> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">1</attributeValue> ... </attributeValueList> <workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> ... <statusInfoList> <statusInfo</pre>		


```
id="10"> <name>Status1</name> ... </statusInfo>
</statusInfoList> </workflowInfo> .... </collectionInfo>
```

duplicateCollection

Function	Duplicate a collection.		
Description	This request duplicates a given collection or collection template.		
Type	POST		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	duplicateCollection	com.quark.gpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	op=duplicate	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	name=[new collection name]	string	The name of the new collection. (Mandatory)
	targetparent=[collection path or ID]	long or string	The collection where the duplicate collection should be put. (Mandatory)There is a choice between collection ID, Name and Path when providing a value.
	assets=[true or false]	boolean	True to also duplicate assets. Defaults to false.(Optional) If assets = true , then the metadata flags like duplicateHierarchy, duplicateWorkflows, duplicateMembers, etc will be overwritten as true .
	createminorversion=[true false]	boolean	This parameter will only work if assets parameter is given. Defaults to false.(Optional)
	duphierarachy=[true false]	boolean	Parameter to duplicate collection with or without its child collections. Defaults to true .(Optional)
	dupworkflows=[true false]	boolean	Parameter to duplicate collection with or without workflows. Defaults to true .(Optional)
	dupmembers=[true false]	boolean	Parameter to duplicate collection with or without users. Defaults to true .(Optional)
	duproutings=[true false]	boolean	Parameter to duplicate collection with or without Status Routing Settings.

			Defaults to true .(Optional)
	<code>dupjobjackets</code> <code>=[true false]</code>	boolean	Parameter to duplicate collection with or without JobJackets. Defaults to true .(Optional)
	<code>duprevisioncontrols</code> <code>=[true false]</code>	boolean	Parameter to duplicate collection with or without RevisionControl Settings. Defaults to true .(Optional)
Returns	Returns the <code>collectionInfo</code> object of the duplicated collection.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/24?op=duplicate&name=dup12_Children&targetparent=Home</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfo templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="true" accessibleChildrenAvailable="true" domainId="19" name="dup12_Children" id="17"> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">1</attributeValue> ... </attributeValueList> <workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> ... <statusInfoList> <statusInfo id="10"> <name>Status1</name> ... </statusInfo> </statusInfoList> </workflowInfo> </collectionInfo></pre>		

updateCollection

Function	Update a collection.		
Description	This request updates the metadata of a given collection or collection template.		
Type	POST		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name	Class Name	
	<code>updateCollection</code>	<code>com.quark.qpp.service.facade</code> <code>Class CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>op=udpate</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributevalues=<xml/></code>	string	The set of attribute values. See "Request/Response parameter formats."(Mandatory)
	<code>workflows =[comma-separated list of workflow names or IDs]</code>	long or string	To update given collection's workflows. To get the IDs of all workflows, use a call to <code>../admin/workflows</code> . (Optional)
	<code>groups=[group name or</code>	GroupInf	To update given collection's workflows.To

	ID], [group name or ID],...	oList	get the IDs of all groups, use a call to <code>../admin/groups</code> . (Optional)
	<code>jobjackets =[comma-separated list of Job Jackets IDs]</code>	long or string	To update given collection's JobJackets. To get a list of Job Jackets and their IDs for a collection, use a call to <code>../collections/colID?jobjackets=true</code> . (Optional)
	<code>routings=<xml/></code>	xml	To update given collection's Status Routing Settings. See "Request/Response parameter formats." (Optional)
	<code>ticketname=[name of default Job Ticket]</code>	string	To update given collection's Default JobJacket Ticket Name. To get the name of Job Ticket for a collection, use a call to <code>../collections/1?op=defaultticket</code> . (Optional)
	<code>applytochildren =[true false]</code>	boolean	Parameter to override settings on its child collection. Defaults to false . (Optional) If this parameter is supplied, it will override the previous settings on its child collections. Used parameters (e.g. Workflows) will not be removed in case the used workflow doesn't exist in the newly supplied parameters.
	<code>appendtochildren =[true false]</code>	boolean	Parameter to merge settings on its child collection. (Optional) Defaults to false . true is equivalent to Apply settings to child Collections: Merge in Platform. Supplied parameter settings will be merged to previous settings of its child collections.
	<code>users=<xml/></code>	xml	To update given collection's Users. See "Request/Response parameter formats." (Optional)
	<code>revisioncontrols=<xml /></code>	xml	To update given collection's Revision Control Settings. See "Request/Response parameter formats." (Optional)
Returns	Returns the <code>collectionInfo</code> object of the updated collection.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/13?op=update&attributevalues=<attributeValueList><attributeValue name="name">dup_col</attributeValue></attributeValueList>&workflows=10&users= <userInfoList><userInfo id="1"></userInfo></userInfoList> &groups=Group1</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfo templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="true" accessibleChildrenAvailable="true" domainId="19"</pre>		

```

name="dup12_Children" id="17"> <attributeValueList>
<attributeValue type="numeric" name="Id"
id="1">1</attributeValue> ... </attributeValueList>
<workflowInfoList> <workflowInfo id="10">
<name>Workflow1</name> ... <statusInfoList> <statusInfo
id="10"> <name>Status1</name> ... </statusInfo>
</statusInfoList> </workflowInfo> ....
</collectionInfo><?xml version="1.0" encoding="UTF-8"
standalone="yes"?> <collectionInfo templateId="0"
collectionClass="General" collectionClassId="1416"
collectionTemplate="false" accessibleChildrenAvailable="false"
domainId="19" collectionPath="Home/Collection/XDK/dup_col"
name="dup_col" id="13"> <attributeValueList> <attributeValue
valueId="0" type="numeric" name="Id"
id="1">13</attributeValue> <attributeValue valueId="0"
type="text" name="Name" id="2">dup_col</attributeValue> .
..... </attributeValueList> <workflowInfoList> <workflowInfo
id="10"> <name>Workflow1</name> <domainId>18</domainId>
<created>2013-12-23T12:06:33Z</created>
<createdBy>1</createdBy> <lastModifiedBy>1</lastModifiedBy>
<lastModified>2013-12-23T12:06:33Z</lastModified>
</workflowInfo> </workflowInfoList> ..... </collectionInfo>

```

updateCollections

Function	Update multiple collections.		
Description	This request updates multiple collections in a single request.		
Type	POST		
URL Format	.../collections		
Java API	Method Name	Class Name	
	updateCollections	com.quark.qpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
	collectioninfolist=<xml/>		See "Request/Response parameter formats."
Returns	Returns a collectioninfolist object in XML format that contains the metadata of all successfully updated collections.		
Examples	URL: https://serverName:serverPort/rest/service/collections?op=update&collectioninfolist=<?xml version="1.0" encoding="UTF-8"?> <collectionInfoList> <collectionInfo collectionClassId="1416" collectionTemplate="false" collectionPath="Home/Home3" > <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">Home3</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="0" type="dateTime" name="Last modified" id="4">2010-04-01T10:00:00Z</attributeValue> <attributeValue valueId="1"		

	<pre> type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue valueId="0" type="boolean" name="Is Template" id="59">>false</attributeValue> <attributeValue valueId="0" type="text" name="Collection Path" id="57">Home/Home1</attributeValue> <attributeValue valueId="1416" type="popup" name="Content Type" id="62">General</attributeValue> <attributeValue valueId="0" type="text" name="Content Type Hierarchy" id="63">System;Collection;General</attributeValue> </attributeValueList> </collectionInfo> </collectionInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList > <collectionInfo defaultJobJacketTicket="Default Job Ticket" defaultJobJacketName="Default Job Jacket" defaultJobJacketId="1" templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="false" domainId="19" collectionPath="Home/Home3" name="Home3" id="23"> <attributeValueList> <attributeValue valueId="0" type="numeric" name="Id" id="1">23</attributeValue> <attributeValue valueId="0" type="text" name="Name" id="2">Home3</attributeValue> <attributeValue valueId="0" type="dateTime" name="Created" id="3">2014-04- 10T06:09:46Z</attributeValue > ... </attributeValueList> <workflowInfoList/> <routingInfoList/> <userInfoList> <userInfo id="1"> <name>Admin</name> </userInfo> </userInfoList> <groupInfoList/> <jobJacketInfoList > <jobJacketInfo id="1"> <name>Default Job Jacket</name> </jobJacketInfo> </jobJacketInfoList> </collectionInfo> <collectionInfo> </collectionInfo> </collectionInfoList> </pre>
--	--

deleteCollection

Function	Delete a collection.		
Description	This request deletes the specified collection or collection template.		
Type	POST		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	deleteCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request works only for empty collections. This means a collection can only be deleted if the collection and all of its child collection do not have any assets in them.

Returns	A status of 200 is returned in response to a successful request execution.
Examples	URL: <code>http://serverName:serverPort/rest/service/collections/142?op=delete</code>

deleteCollectionWithAssets

Function	Delete a collection with assets.		
Description	This request deletes the given collection from the system along with its sub-collections and assets. Only users who have access to the given collection are allowed to delete the collection.		
Type	POST		
URL Format	<code>.../collections/[collection ID] or collectionsbypath/[collection path]</code>		
Java API	Method Name	Class Name	
	<code>deleteCollectionWithAssets</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>op=deletewithassets</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/collections/142?op=deletewithassets</code>		

createJobjackets

Function	Create a Job Jackets file.		
Description	This request creates new job jacket after parsing the content supplied and returns and object of the newly created Job Jacket. Stream the Job Jackets XML into the request body or send it as a file part.		
Type	POST		
URL Format	<code>.../collections/jobjackets</code>		
Java API	Method Name	Class Name	
	<code>createJobjackets</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>op=create:</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns a <code>jobJacketInfoList</code> object in XML format.		

Examples	URL: <code>http://serverName:serverPort/rest/service/collections/jobjackets?op=create</code>
	Result: A <code>jobJacketInfoList</code> object in XML format.

getJobjacket

Function	Get a Job Jackets file with a specific ID.	
Description	This request returns the specified job jacket.	
Type	GET	
URL Format	<code>.../collections/jobjackets/[Job Jackets ID]</code>	
Java API	Method Name	Class Name
	<code>getJobjacket</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>
Parameters	This request requires no input parameters.	
Returns	Returns a <code>jobJacketInfoList</code> object with its metadata.	
Examples	URL: <code>http://serverName:serverPort/rest/service/collections/jobjackets/1</code>	
	Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <jobJacketInfoList> <jobJacketInfo id="1"> <name>Default Job Jacket</name> <domainId>0</domainId> <defaultTicketTemplate>Default Job Ticket</defaultTicketTemplate> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <created>2008-06-01T10:00:00Z</created> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2008-06-01T10:00:00Z</lastModified> <ticketTemplateList> <ticketTemplate>Default Job Ticket</ticketTemplate> </ticketTemplateList> <content>... </content> </jobJacketInfo> </jobJacketInfoList></pre>	

getAllJobJackets

Function	Get all Job Jackets.	
Description	This request returns all the metadata of Job Jackets available in the system.	
Type	GET	
URL Format	<code>.../collections/jobjackets</code>	
Java API	Method Name	Class Name
	<code>getAllJobJackets</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>
Parameters	This request requires no input parameters.	

Returns	Returns a <code>jobJacketInfoList</code> object that contains a list of Job Jackets.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/jobjackets</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <jobJacketInfoList> <jobJacketInfo id="1"> <name>Default Job Jacket</name> <domainId>0</domainId> <defaultTicketTemplate>Default Job Ticket</defaultTicketTemplate> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <created>2008-06- 01T10:00:00Z</created> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2008- 06-01T10:00:00Z</lastModified> <ticketTemplateList> <ticketTemplate>Default Job Ticket</ticketTemplate> </ticketTemplateList> <content>.....</content> </jobJacketInfo> </jobJacketInfoList></pre>

getJobJacketContent

Function	Get the contents of a Job Jackets file.		
Description	This request returns the actual content of a specified job jacket in xml format.		
Type	GET		
URL Format	.../collections/[collection ID] or collectionsbypath/[collection path]		
Java API	Method Name		Class Name
	getJobJacketContent		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=jobjacketcontent	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	jobjacket=[Job Jackets file ID]		To get a list of Job Jackets and their IDs for a collection, use a call to .../collections/colID?jobjackets=true. (Mandatory)
Returns	Returns the actual content of a specified job jacket in XML format		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?op=jobjacketcontent&jobjacket=1</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <JDF xmlns="http://www.CIP4.org/JDFSchema_1_1" xmlns:qxp="http://www.quark.com/QuarkXPress/JDF_1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Version="1.2" JacketVersion="1.0" ID="QuarkXPressDefault_Job_Jacket" Type="QuarkXPressJacket" xsi:type="qxp:QuarkXPressJacket" Status="Waiting" Name="Default Job Jacket" Template="true"> <ResourcePool></pre>		

	<pre> </ResourcePool> <JDF Name="Default Job Ticket" ID="JobTicket_0" IsDefault="1" Status="Waiting" Template="true" Type="QuarkXPressTemplateTicket" xsi:type="qxp:QuarkXPressTemplateTicket"> <ResourceLinkPool> </ResourceLinkPool> <ResourcePool> </ResourcePool> </JDF> </JDF> </pre>
--	--

updateJobJacket

Function	Update a job jacket.		
Description	This request updates the contents of a specified job jacket.		
Type	POST		
URL Format	.../collections/jobjackets/[jobJacketId]		
Java API	Method Name		Class Name
	updateJobJacket		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id. Pass the xml of the job jacket file as file stream in the request body.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/collections/jobjackets/{jobJacketId} ?op=update		

updateJobJackets

Function	Update multiple job jackets		
Description	This request updates the contents of multiple job jackets in a single request.		
Type	POST		
URL Format	.../collections/jobjackets		
Java API	Method Name		Class Name
	updateJobJackets		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
	jobjacketinfolist=<xml		See "Request/Response parameter

	l/>		formats."
Returns	Returns a <code>jobjacketinfo</code> object in XML format that contains the metadata of all successfully updated job jackets.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/collections/jobjackets?op=update&jobjacketinfo=<xml /></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <jobJacketInfoList> <jobJacketInfo id="1"> <name>Default Job Jacket</name> <domainId>0</domainId> <defaultTicketTemplate>Default Job Ticket</defaultTicketTemplate> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <created>2008-06-01T10:00:00Z</created> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2008-06-01T10:00:00Z</lastModified> <ticketTemplateList> <ticketTemplate>Default Job Ticket</ticketTemplate> </ticketTemplateList> <content><?xml version="1.0" encoding="UTF-8"?><JDF xmlns="http://www.CIP4.org/JDFSchema_1_1" xmlns:qxp="http://www.quark.com/QuarkXPress/JDF_1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Version="1.2" JacketVersion="1.0" ID="QuarkXPressDefault_Job_Jacket" Type="QuarkXPressJacket" xsi:type="qxp:QuarkXPressJacket" Status="Waiting" Name="Default Job Jacket" Template="true"> <ResourcePool></ResourcePool> <JDF Name="Default Job Ticket" ID="JobTicket_0" IsDefault="1" Status="Waiting" Template="true" Type="QuarkXPressTemplateTicket" xsi:type="qxp:QuarkXPressTemplateTicket"> <ResourceLinkPool></ResourceLinkPool> <ResourcePool></ResourcePool> </JDF></JDF> </content> </jobJacketInfo> </jobJacketInfoList></pre>		

parseJobJacketXml

Function	Parse a Job Jackets file. (Include the file in the request body or as file part.)		
Description	This request parses the jobjacket xml and returns a JobJacket object containing the jobjacket name and ticket names.		
Type	POST		
URL Format	.../collections/jobjackets		
Java API	Method Name	Class Name	
	<code>parseJobJacketXml</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>op=parsexml</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns a <code>jobjacketinfo</code> object in XML format.		
Examples	URL:		

	<p><code>http://serverName:serverPort/rest/service/collections/jobjackets?op=parsexml</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <jobJacketInfo id="0"> <name>Project1 Job Jacket</name> <domainId>0</domainId> <defaultTicketTemplate>Default Job Ticket</defaultTicketTemplate> <createdBy>0</createdBy> <lastModifiedBy>0</lastModifiedBy> <ticketTemplateList> <ticketTemplate>Default Job Ticket</ticketTemplate> </ticketTemplateList> </jobJacketInfo></pre>
--	--

deleteJobJacket

Function	Delete a Job Jackets file.		
Description	This request deletes the jobjacket with the given id.		
Type	POST		
URL Format	<code>.../collections/jobjackets/[Job Jacket ID]</code>		
Java API	Method Name		Class Name
	<code>deleteJobJacket</code>		<code>com.quark.qpp.service.facade Class CollectionFacade</code>
Parameters	Name	Type	Description
	<code>op=delete</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/collections/jobjackets/10?op=delete</code></p>		

getApplicableRoutingUsers

Function	Get Applicable Routing Users		
Description	This request returns a list of applicable routing users for the given collection for specified workflow and status. This will return all collection mapped users having the same userclass as any of the status mapped userclasses.		
Type	GET		
URL Format	<code>.../collections/{collectionId}</code>		
Java API	Method Name		Class Name
	<code>getApplicableRoutingUsers</code>		<code>com.quark.qpp.service.facade Class CollectionFacade</code>
Parameters	Name	Type	Description
	<code>routingusers =true</code>	boolean	
	<code>workflow={workflowIdOrName}</code>	string	The ID or Name of the workflow whose Routing Users you wish to be retrieved.

	<code>status={statusIdOrName} or workflowStatusCombination</code>	string	The ID or Name of the status whose Routing Users you wish to be retrieved.
Returns	Returns an <code>userInfoList</code> object containing a list of all collection users applicable for given workflow or status.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?routingusers=true&status=Status1</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="1"> <name>Admin</name> <firstName>Administrator</firstName> <nativeTrustee>true</nativeTrustee> <enabled>true</enabled> <defaultUserId>1</defaultUserId> <ldapProfileId>0</ldapProfileId> <deleted>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList></pre>		

getApplicablePrivileges

Function	Get applicable privileges for the logged in user.		
Description	This request returns the privileges applicable for the logged in user for the given collection. These privileges are a union of the privileges of the following user classes:Userclass of the logged in user in the given collection, only if the user is mapped to the given collection.Userclasses of collection mapped groups of which the logged in user is a member.		
Type	GET		
URL Format	<code>../collections/[collection ID]</code>		
Java API	Method Name	Class Name	
	<code>getApplicablePrivileges</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>applicableprivileges=[true]</code>	boolean	
	<code>status=[statusId] or [Name] or [StatusWorkFlowCombination]</code>	long	The ID or Name of the status whose privileges you wish to be retrieved.
	<code>contenttype=[contentTypeId] or [Name] or [Path]</code>	long	The default content type IDs are documented in the JavaDoc SDK at <code>com.quark.qpp.core.content.service.constants.DefaultContentTypes</code> . To get a list of content types, use a call to <code>../admin/contenttypes.(Optional)</code>

			If not provided, the metadata of all content types will be returned.
	privileges=[comma-separated list of privilege ids or names]	string	
Returns	Returns a <code>ApplicablePrivilegesInfo</code> object containing either application and content level privileges or both, depending upon the parameters.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/192?applicableprivileges=true</pre> <p>Result:</p> <pre><applicablePrivilegesInfo> <applicationPrivList> <privilege id="140000023"> <name>Delete</name> <privilegeHierarchy> Administration;Storage;StorageRules;Delete </privilegeHierarchy> </privilege> <privilege id="140000022"> <name>Create/Modify</name> <privilegeHierarchy> Administration;Storage;StorageRules;Create/Modify </privilegeHierarchy> </privilege> </applicationPrivList> <contentTypePrivList> <contentTypePriv contentTypeId="1100" contentType="Asset" contentTypeHierarchy="System;Asset;"> <privilege id="3000006"> <name>Change Workflow of Checked In Asset</name> <privilegeHierarchy> Edit Attributes;Change Workflow of Checked In Asset </privilegeHierarchy> </privilege> </contentTypePriv> </contentTypePrivList> </applicablePrivilegesInfo></pre>		

Function	Get enabled content type privileges.		
Description	This request returns only contenttype based privileges applicable for the logged in user for the given collection. These privileges are a union of the privileges of the following user classes:Userclass of the logged in user in the given collection, only if the user is mapped to the given collection.Userclasses of collection mapped groups of which the logged in user is a member.		
Type	GET		
URL Format	.../collections/[collection ID]		
Java API	Method Name	Class Name	
	<code>getApplicablePrivileges</code>	<code>com.quark.qpp.service.facade Class CollectionFacade</code>	
Parameters	Name	Type	Description
	<code>applicableprivileges=[true]</code>	boolean	(Mandatory)
	<code>contenttype=[contentTypeid] or [Name] or [Path]</code>	long	The default content type IDs are documented in the JavaDoc SDK at <code>com.quark.qpp.core.content.service.constants.DefaultContentTypes</code> . To get a list of content types, use a call to <code>.../admin/contenttypes.(Optional)</code>

			If not provided, the metadata of all content types will be returned.
Returns	Returns a <code>applicablePrivilegesInfo</code> object containing either application and content level privileges or both, depending upon the parameters.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/192?applicableprivileges=true&contenttype=Picture</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <applicablePrivilegesInfo> <contentTypePrivList> <contentTypePriv contentTypeId="1103" contentType="Picture" contentTypeHierarchy="System;Asset;Picture;"> <privilege id="10000015"> <name>Get Original Assets</name> <privilegeHierarchy>General;Get Original Assets</privilegeHierarchy> </privilege> <privilege id="10000012"> <name>Check In Asset without Displaying the Check In Dialog Box</name> <privilegeHierarchy> General;Check In Asset without Displaying the Check In Dialog Box </privilegeHierarchy> </privilege></pre>		

Function	Get all enabled status privileges.		
Description	This request returns the privileges applicable for the logged in user for the given collection and status (Status based overridden privileges) . These privileges are a union of the privileges of the following user classes: Userclass of the logged in user in the given collection, only if the user is mapped to the given collection. Userclasses of collection mapped groups of which the logged in user is a member.		
Type	GET		
URL Format	.../collections/[collection ID]		
Java API	Method Name		Class Name
	<code>getApplicablePrivileges</code>		<code>com.quark.qpp.service.facade Class CollectionFacade</code>
Parameters	Name	Type	Description
	<code>applicableprivileges=[true]</code>	boolean	
	<code>status=[statusId] or [Name] or [StatusWorkflowCombination]</code>	boolean	The ID or Name of the status whose privileges you wish to be retrieved.
	<code>privileges=[comma-separated list of privilege ids or names]</code>	long or string	
Returns	Returns a <code>applicablePrivilegesInfo</code> object containing either application and content level privileges or both, depending upon the parameters.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/192</pre>		

	<pre>?applicableprivileges=true&status=10&privileges=...,...,...</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <applicablePrivilegesInfo> <contentTypePrivList> <contentTypePriv contentTypeId="1103" contentType="Picture" contentTypeHierarchy="System;Asset;Picture;"> <privilege id="10000015"> <name>Get Original Assets</name> <privilegeHierarchy>General;Get Original Assets</privilegeHierarchy> </privilege> <privilege id="10000012"> <name>Check In Asset without Displaying the Check In Dialog Box</name> <privilegeHierarchy> General;Check In Asset without Displaying the Check In Dialog Box </privilegeHierarchy> </privilege></pre>
--	--

getApplicableRoutingGroups

Function	Get applicable routing groups for the logged in user.		
Description	This request returns an array of applicable routings groups for the given collection for the specified workflow and status. This will return all collection mapped groups whose userclass is the same as one of the status mapped userclasses.		
Type	GET		
URL Format	.../collections/[collection ID]		
Java API	Method Name		Class Name
	getApplicableRoutingGroups		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	routinggroups=[true]	boolean	
	workflow=[Id] or [Name]	long or string	The ID or Name of the workflow whose routing groups you wish to be retrieved.
	status=[statusId] or [Name] or [StatusWorkflowCombination]	long or string	The ID or Name of the status whose routing groups you wish to be retrieved.
Returns	Returns a list of applicable routing <code>GroupInfo</code> objects for the given collection and status.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/collections/1 ?routinggroups=true&workflow=Default Workflow&status=Edit</pre> <p>Result:</p> <pre><groupInfoList> <groupInfo id="60"> <name>Editors</name> <emailAddress/> <nativeTrustee>true</nativeTrustee> <defaultUserId>1</defaultUserId> <defaultUserClass>Administrator</defaultUserClass> </groupInfo> </groupInfoList></pre>		

getApplicableAttributeConstraints

Function	Get applicable attribute constraints for the logged in user.		
Description	This request returns attribute constraints applicable to the given status and collection for the logged in user. The user may have multiple userclasses, due to its membership in multiple groups which are mapped to the collection with different userclasses. Since attribute constraints are defined per userclass, multiple constraints in such cases are evaluated and the most permissive set of attribute constraints is returned.		
Type	GET		
URL Format	.../collections/[collection ID]		
Java API	Method Name	Class Name	
	getApplicableAttributeConstraints	com.quark.qpp.service.facade Class CollectionFacade	
Parameters	Name	Type	Description
	applicableconstraints=[true]	boolean	
	status=[statusId] or [Name] or [StatusWorkflowCombination]	long or string	The ID or Name of the status whose attributes constraints you wish to be retrieved.
	attributes=[comma-separated list of attributes ids or names]	long or string	
Returns	Returns a list of applicable <code>AttributeConstraintsInfo</code> objects.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/191?applicableconstraints=true&status=DefaultWorkflow&attributes=2,.....</pre> <p>Result:</p> <pre><attributeConstraintsInfoList> <attributeConstraintsInfo> <attributeId>2</attributeId> <attributeName>Name</attributeName> <constraintId>1</constraintId> <constraintName>PREVENT CHANGE</constraintName> </attributeConstraintsInfo> </attributeConstraintsInfoList></pre>		

getApplicableStatusTransition

Function	Get applicable status transitions for the logged in user.
Description	This request returns the applicable status transition for the given workflow, status and collection for the logged in user. If the status transition is overridden for the given userclass, then overridden will be considered.
Type	GET
URL Format	.../collections/[collection ID]

Java API	Method Name		Class Name
	getApplicableStatusTransition		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	statustransitions=[true]	boolean	
	workflow=[workflowId] or [Name]	long or string	The ID or Name of the workflow whose status transitions you wish to be retrieved.
	status=[statusId] or [Name]	long or string	The ID or Name of the status whose status transitions you wish to be retrieved.
Returns	Returns a <code>statusTransitionInfoList</code> object containing a list of applicable status transitions.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1?statustransitions=true&workflow=Default Workflow</pre> <p>Result:</p> <pre><statusTransitionInfoList> <statusTransitionInfo id="10"> <name>Edit</name> <workflowInitiatingStatus>true</workflowInitiatingStatus> <nextPossibleStatusInfoList> <nextPossibleStatusInfo id="11"> <name>Publish</name> </nextPossibleStatusInfo> </nextPossibleStatusInfoList> </statusTransitionInfo> <statusTransitionInfo id="11"> <name>Publish</name> <workflowInitiatingStatus>true</workflowInitiatingStatus> <nextPossibleStatusInfoList> <nextPossibleStatusInfo id="10"> <name>Edit</name> </nextPossibleStatusInfo> </nextPossibleStatusInfoList> </statusTransitionInfo> </statusTransitionInfoList></pre>		

getApplicableWorkflowInitialStatuses

Function	Get applicable workflow initial statuses.		
Description	This request returns the initial applicable statuses for the given workflow and collection.		
Type	GET		
URL Format	.../collections/[collection ID]		
Java API	Method Name		Class Name
	getApplicableWorkflowInitialSt atuses		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	workflowinitiatingsta tuses=[true]	boolean	
	workflow=[workflowId] or [Name]	long or string	The ID or Name of the workflow whose workflow Initiating Statuses you wish to be retrieved.
Returns	Returns a list of <code>StatusInfo</code> objects referring to the initial applicable statuses.		

Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/collections/1 ?workflowinitiatingstatuses=true&workflow=Default Workflow</pre> <p>Result:</p> <pre><statusInfoList> <statusInfo id="10"> <name>Edit</name> </statusInfo> <statusInfo id="11"> <name>Publish</name> </statusInfo> </statusInfoList></pre>
-----------------	---

Query management

The query management capabilities of the API allow application developers to manage searches and includes the following REST calls:

- *search*
- *searchAssets*
- *searchCollections*
- *createSearches*
- *getSearch*
- *getAllSearches*
- *getSearchResultsForQuery*
- *getTextSearchResults*
- *setSharedSearchTrustees*
- *getSharedSearchTrustees*
- *updateSearches*
- *deleteSearch*
- *count*
- *countAssets*
- *countCollections*
- *countSearchResultsForQuery*
- *countTextResults*

search

Function	Search assets/collection using a searchDefinition.	
Description	This request searches assets/collections based on a given searchDefinition.	
Type	GET	
URL Format	.../search	
Java API	Method Name	Class Name

	search	com.quark.qpp.service.facade Class QueryFacade	
Parameters	Name	Type	Description
	searchinfo=[xml]	string	An XML object containing metadata of the asset/collections to be found.
Returns	Returns a <code>assetInfoList</code> object containing the metadata of the asset to be found , if the context of the given searchInfo is ASSET . Returns a <code>collectionInfoList</code> object containing the metadata of the asset to be found , if the context of the given searchInfo is COLLECTION .		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search?search=<xml/></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="66"> <attributeValueList> <attributeValue valueId="0" type="text"name="Name" id="2">Alberi.jpg</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="72"> <attributeValueList> <attributeValue valueId="0" type="text"name="Name" id="2">QuarkImage.jpg</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

searchAssets

Function	Search assets for a particular attribute value.		
Description	This request searches for assets matching specific search criteria.		
Type	GET		
URL Format	.../search/assets		
	Method Name	Class Name	
	searchAssets	com.quark.qpp.service.facade Class QueryFacade	
Parameters	Name	Type	Description
	[attribute name or id]=[attribute value]	long or string	<p>The attribute name and value that you are searching for. (Mandatory)</p> <p>The attributes must be queryable.</p> <p>The following is a lit of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor version

		<ul style="list-style-type: none"> • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment Checksum • Auxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page NameRelationship Status • Status • XPath <p>Functions to get assets based on two non-queryable attributes (Collection, Status) are given below.</p> <p>You can pass more than one attribute as a parameter with the value to search for.</p>	
	<code>includechildcontenttypes = [true false]</code>	boolean	Parameter to include child content types in search criteria.
	<code>attributes = [comma-separated list of attribute names or IDs]</code>		You can pass the appropriate value for any queryable attribute. For a list of attributes, use a call to <code>..admin/attributes</code> .
	<code>displaymode=[x]</code>	long	The default display modes are documented in the JavaDoc SDK at <code>com.quark.qpp.core.query.service.constants.DefaultDisplayModes</code> . (Optional) Defaults to 1.
	<code>exploremode=[x]</code>	long	For a list of explore modes, see <code>ExploreModeTypes.java</code> . (Optional) Defaults to 0.

	<code>collections</code>	string	Comma separated ids of collections in which assets are to be searched.
	<code>recursive=[true false]</code>	boolean	Indicates whether all of the hierarchy below and including collections are to be searched. Default value= true
Returns	Returns search results based upon specified attribute values.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/assets?Name=1quark2.jpg&Collection=Home</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="4"> <attributeValueList> <attributeValue type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue type="popup" name="Routed to" id="25">Admin</attributeValue> <attributeValue type="popup" name="Status" id="24">Status1</attributeValue> <attributeValue type="popup" name="Content Type" id="62">Picture</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

Function	Search for assets that contain a particular text string.		
Description	This request will search for assets that contain a given text string. This is specific to content based searches in workspace and ConnectClient.		
Type	GET		
URL Format	.../search/assets		
Java API	Method Name		Class Name
	<code>searchAssets</code>		<code>com.quark.qpp.service.facade</code> <code>Class QueryFacade</code>
Parameters	Name	Type	Description
	<code>text=[text to search for]</code>	string	Specify the string to search for. (Mandatory)
	<code>attributes=[comma-separated list of attribute names or IDs]</code>	string	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to <code>..admin/attributes</code>.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id •

		<ul style="list-style-type: none"> • Article Component Reference Asset Major Version • Article Component Reference Asset Minor version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment Checksum • Auxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns an <code>assetInfoList</code> object containing the assets that contain the specified text string.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/assets?text=box&attributes=Name</pre> <p>Result:</p> <pre>Example of a text search with specific attribute:<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="292"> <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">15116462.qxp</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="304"> <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">15116c9a.qxp</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="306"> <attributeValueList> <attributeValue valueId="0" type="text" name="Name"</pre>	

	<code>id="2">15116d33.qxp</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></code>
--	--

Function	Search for a particular attachment.		
Description	This request searches for all attachments for a given document.		
Type	GET		
URL Format	<code>.../search/assets</code>		
Java API	Method Name		Class Name
	<code>searchAssets</code>		<code>com.quark.qpp.service.facade Class QueryFacade</code>
Parameters	Name	Type	Description
	<code>document=[ID of the project]</code>	long	ID of the document whose attachments you want found. (Mandatory)
	<code>attributes=[comma-separated list of attribute names or IDs]</code>	long or string	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to <code>..admin/attributes</code>.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment Checksum • Auxillary Data • Box Height • Box Uid • Box Width • Collection • Component position

			<ul style="list-style-type: none"> • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns an <code>assetInfoList</code> object containing the assets that contain child assets.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/assets?document=22</pre> <p>Result:</p> <p>A <code>QueryResultInfo</code> object in XML format.</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="290"> <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">153486e3.qcd</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

Function	Search within an existing query.		
Description	This request searches an existing query in the platform.		
Type	GET		
URL Format	.../search/assets		
Java API	Method Name		Class Name
	<code>searchAssets</code>		<code>com.quark.qpp.service.facade</code> <code>Class QueryFacade</code>
Parameters	Name	Type	Description
	<code>query=[ID of the query]</code>	long or string	Use /search to get IDs. (Mandatory)
	<code>attributes=[comma-separated list of attribute names or IDs]</code>	long or string	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to <code>..admin/attributes</code>.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id

REST FRAMEWORK SERVICES

			<ul style="list-style-type: none"> • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment Checksum • Auxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns a <code>QueryResultInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/assets?query=12</pre> <p>Result:</p> <p>A <code>QueryResultInfo</code> object in XML format.</p>		

Function	Search for assets with a particular status.
Description	This request returns all assets associated with given status.
Type	GET
URL Format	<code>.../search/assets</code>

Java API	Method Name		Class Name
	searchAssets		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	status=[status name] and workflow=[workflow name]	string	Specify Status of a workflow. Specify Name or ID of a workflow. (Mandatory) Status and workflow will go hand in hand.
	attributes=[comma-seperated list of attribute names or IDs]	long or string	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to <code>..admin/attributes</code>.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type

			<ul style="list-style-type: none"> • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns an <code>assetInfoList</code> object containing an array of assets related to a given status.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/assets?workflow=Workflow1 &status=Status1&attributes=name,5,7,62,54,collection</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="4"> <attributeValueList> <attributeValue type="text" name="Name" id="2">lquark2.jpg</attributeValue> ... </attributeValueList> </assetInfo> <assetInfo id="10"> <attributeValueList> <attributeValue type="text" name="Name" id="2">Tulips</attributeValue> ... </attributeValueList> </assetInfo> </assetInfoList></pre>		

Function	Search for assets in a particular collection.		
Description	This request searches for the assets checked-in at given collection.		
Type	GET		
URL Format	.../search/assets		
Java API	Method Name		Class Name
	searchAssets		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	collections =[comma-separated list of collection names or IDs]	long or string	To get a list of collection IDs for a particular collection, use a call to <code>..collectionsbypath/[path]</code> . (Mandatory)
	recursive=[true false]	boolean	Parameter to include child collections. Defaults to true . (Optional)
	attributes=[comma-separated list of attribute names or IDs]	long or string	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to <code>..admin/attributes</code>.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id

		<ul style="list-style-type: none"> • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns an <code>assetInfoList</code> object containing an array of assets checked-in at a specified collection.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/assets?collections=Home <?xml version="1.0" encoding="UTF-8" standalone="yes"?)</pre> <p>Result:</p> <pre><assetInfoList> <assetInfo id="4"> <attributeValueList> <attributeValue type="text" name="Name" id="2">1quark2.jpg</attributeValue> ... </attributeValueList> </assetInfo> <assetInfo id="10"> <attributeValueList> <attributeValue type="text" name="Name" id="2">Tulips</attributeValue> ... </attributeValueList> </assetInfo> </assetInfoList></pre>	

searchCollections

Function	Search for collections in a particular collection.		
Description	This request searches all nested collections accessible to the user under a given collection ID or Name.		
Type	GET		
URL Format	.../search/collections		
Java API	Method Name		Class Name
	searchCollections		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	collections =[comma-separated list of collection names or IDs]	long or string	Parameter to include child collections. To get a list of collection IDs for a particular collection, use a call to ..collectionsbypath/[path].(Mandatory)
	recursive=[true false]	boolean	Defaults to true .(Optional)
	includechildcontenttypes=[true false]	boolean	Parameter to include child content types in search criteria.
	attributes=[comma-separated list of attribute names or IDs]	long or string	Specify the string to search for. (Mandatory) You can pass the appropriate value for any queryable attribute to fetch specific attributes. For a list of attributes, use a call to ..admin/attributes. The attributes must be queryable. The following is a list of non-queryable attributes: <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid

			<ul style="list-style-type: none"> • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns a <code>collectionInfoList</code> object containing an array of existing and accessible collections.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/collections?collection_path=Home/Collection&Creator=Admin</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" name="Collection" id="10"> <attributeValueList> <attributeValue type="text" name="Name" id="2">Collection</attributeValue> <attributeValue type="popup" name="Workflow" id="54"></attributeValue> <attributeValue type="popup" name="Collection" id="55"></attributeValue> <attributeValue type="popup" name="Routed to" id="25"></attributeValue> <attributeValue type="popup" name="Status" id="24"></attributeValue> <attributeValue type="popup" name="Content Type" id="62">General</attributeValue> </attributeValueList> </collectionInfo> </collectionInfoList></pre>		

Function	Search for collections in a collection with particular attributes.
Description	This request searches for assets in a specified collection and displays results with the specified attributes.
Type	GET
URL Format	<code>.../search/collections</code>

Java API	Method Name		Class Name
	searchCollections		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	attributes =[comma-separated list of attribute names or IDs]	long or string	For a list of attributes, use a call to ..admin/attributes.(Mandatory)
	collecons =[comma-separated list of collections names or IDs]	long or string	Parameter to include child collections. To get a list of collection IDs for a particular collection, use a call to ..collectionsbypath/[path]. (Mandatory)
	recursive=[true false]	boolean	Defaults to true . (Optional)
Returns	Returns a <code>collectionInfoList</code> object containing an array of existing and accessible collections.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/collections?collection path=Home&Creator=Admin&attributes=Name</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <collectionInfoList> <collectionInfo templateId="0" collectionClass="General" collectionClassId="1416" collectionTemplate="false" accessibleChildrenAvailable="true" domainId="19" name="Home" id="1"> <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">Home </attributeValue> </attributeValueList> <groupInfoList/> </collectionInfo> </collectionInfoList></pre>		

createSearches

Function	Create searches.		
Description	This request creates a single or multiple searches at one time.		
Type	POST		
URL Format	.../searchdefinitions		
Java API	Method Name		Class Name
	createSearches		com.quark.qpp.service.facade Class QueryDefinitionFacade
Parameters	Name	Type	Description
	op=create	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	searchinfolist=[xml]	string	An XML object containing metadata of searches to be created.

Returns	Returns a <code>searchInfoList</code> object containing metadata of newly created searches.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/searchdefinitions?op=create&searchinfilist=<xml/></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <searchInfoList> <searchInfo parameterized="false" id="110"> <searchName>AllPictures</searchName> <owner id="1">Admin</owner> <queryContext context="ASSET"/> <queryConditionList> <queryCondition recursive="false" parameterized="false" negated="false" type="CONTENT_TYPE_CONDITION" comparisonOperator="IS" logicalOperator="OR" nestingLevel="1"> <searchValue id="1103">System;Asset;Picture;</searchValue> </queryCondition> </queryConditionList> <queryDisplay exploreMode="PLAIN" displayMode="SNIPPET"> <displayColumnList> <displayColumn attributeColumn="true" width="150" columnId="2">Name</displayColumn> <displayColumn attributeColumn="false" width="150" columnId="29"/> <displayColumn attributeColumn="false" width="150" columnId="62"/> <displayColumn attributeColumn="false" width="150" columnId="54"/> <displayColumn attributeColumn="false" width="150" columnId="25"/> <displayColumn attributeColumn="false" width="150" columnId="55"/> </displayColumnList> <sortInfoList> <sortInfo attributeColumn="true" order="ascending" columnId="2">Name</sortInfo> </sortInfoList> </queryDisplay> </searchInfo> </searchInfoList></pre>

getSearch

Function	Search for a saved search.		
Description	This request returns the metadata of a specified search.		
Type	GET		
URL Format	.../searchdefinitions/{searchIdOrName}?		
Java API	Method Name		Class Name
	getSearch		com.quark.qpp.service.facade Class QueryDefinitionFacade
Parameters	Name	Type	Description
	context=[true false]	boolean	Defaults to false .
	display=[true false]	boolean	Defaults to false .
	conditions=[true false]	boolean	Defaults to false .
	getSharedSearchTrustees=[true false]	boolean	Defaults to false .
Returns	Returns a <code>searchInfoList</code> object containing metadata of all saved and shared searches. Metadata of QueryContext, QueryDisplay, QueryConditions will be added per the parameters.		
Examples	URL:		

<pre>http://serverName:serverPort/rest/service/searchdefinitions/Default</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <searchInfoList> <searchInfo parameterized="false" id="100"> <searchName>Default</searchName> <owner id="1">Admin</owner> </searchInfo> </searchInfoList></pre>

getAllSearches

Function	Get all saved searches.		
Description	This request returns all saved, shared searches.		
Type	GET		
URL Format	.../searchdefinitions?		
Java API	Method Name		Class Name
	getAllSearches		com.quark.qpp.service.facade QueryDefinitionFacade
Parameters	Name	Type	Description
	context=[true false]	boolean	Defaults to false .
	display=[true false]	boolean	Defaults to false .
	conditions=[true false]	boolean	Defaults to false .
	getSharedSearchTrustees=[true false]	boolean	Defaults to false .
Returns	Returns a <code>searchInfoList</code> object containing metadata of all saved and shared searches. Metadata of QueryContext, QueryDisplay, QueryConditions will be added per the parameters.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/searchdefinitions</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <searchInfoList> <searchInfo parameterized="false" id="100"> <searchName>Default</searchName> <owner id="1">Admin</owner> </searchInfo> <searchInfo parameterized="false" id="101"> <searchName>Expired Content</searchName> <owner id="1">Admin</owner> </searchInfo> </searchInfoList></pre>		

getSearchResultsForQuery

Function	Get the results of an existing search.	
Description	This request returns the results of the execution of an already saved search.	
Type	GET	
URL Format	.../search/[search name or ID]	
Java API	Method Name	
	Class Name	

	getSearchResultsForQuery		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	displaymode=[x].	long	Defaults to 1. The default display modes are documented in the JavaDoc SDK at com.quark.qpp.core.query.service.constants.DefaultDisplayModes.(Optional)
	exploremode=[x]	long	Defaults to 0. For a list of explore modes, see ExploreModeTypes.java.
Returns	Returns an <code>assetInfoList</code> object containing the results of a specified search.		
Examples	URL:		
	http://serverName:serverPort/rest/service/search/Default		
	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="10"> <attributeValueList> <attributeValue type="text" name="Name" id="2">Tulips</attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue type="popup" name="Status" id="24">Status1</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="11"> <attributeValueList> <attributeValue type="text" name="Name" id="2">Duplicate_asset</attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue type="popup" name="Status" id="24">Status1</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="4"> <attributeValueList> <attributeValue type="text" name="Name" id="2">lquark2.jpg</attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue type="popup" name="Status" id="24">Status1</attributeValue> </attributeValueList> </assetInfo> <assetInfo id="5"> <attributeValueList> <attributeValue type="text" name="Name" id="2">15116.qxp</attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue type="popup" name="Status" id="24">Status1</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

Function	Get the results of an existing search with specific display attributes.	
Description	This request returns the results of the execution of an already saved search and displays only the specified attributes.	
Type	GET	
URL Format	.../search/[search name or ID]	
Java API	Method Name	Class Name
	getSearchResultsForQuery	com.quark.qpp.service.facade Class

		QueryFacade	
Parameters	Name	Type	Description
		<code>attributes = [comma-separated list of attribute names or IDs]</code>	long or string
Returns	Returns an <code>assetInfoList</code> object containing the results of a specified search with only the specified attributes.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/Default?attributes=Name,2,10,Collection</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfoList> <assetInfo id="10"> <attributeValueList> <attributeValue type="text" name="Name" id="2">Tulips</attributeValue> <attributeValue type="text" name="Name" id="2">Tulips</attributeValue> <attributeValue type="dateTime" name="Job jacket last modified" id="10"></attributeValue> <attributeValue type="popup" name="Collection" id="55">Home</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

Function	Get the results of an already saved "Ask" queries.		
Description	This request returns the result of an already saved search (Ask Queries) by accepting required values of search criteria as its parameters.		
Type	GET		
URL Format	<code>.../search/[search name OR id]</code>		
Java API	Method Name		Class Name
	<code>getSearchResultsForQuery</code>		<code>com.quark.qpp.service.facade</code> Class <code>QueryFacade</code>
Parameters	Name	Type	Description
	<code>text=[text to search for]</code>	string	Provide value for search criteria as its parameter. Parameter will be based upon the saved query. (Mandatory)
	<code>attributes=[comma-separated list of attribute names or IDs]</code>	string or long	You can pass the appropriate value for any queryable attribute to fetch specific attributes. For a list of attributes, use a call to <code>..admin/attributes.</code>
Returns	Returns an <code>assetInfoList</code> containing the results of a specified search based on the given parameter and its value.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/myAskQuery?File%20extension=qxp</pre> <p>Result:</p>		

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assetInfoList> <assetInfo id="1"> <attributeValueList>
<attributeValue valueId="0" type="text" name="Name"
id="2">Sample.qxp</attributeValue> </attributeValueList>
</assetInfo> </assetInfoList>
```

getTextSearchResults

Function	Search for assets using wild cards.		
Description	This request executes a quick search for assets based on their name. Both simple and wildcard based comparison is performed for name.		
Type	GET		
URL Format	.../search/quicksearch/		
Java API	Method Name		Class Name
	getTextSearchResults		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	collections =[comma-separated list of collection names or IDs]	long or string	To get a list of collection IDs for a particular collection, use a call to ..collectionsbypath/[path].(Mandatory)
	recursive=[true false]	boolean	Parameter to include child collections. Defaults to true . (Optional)
	attributes=[comma-separated list of attribute names or IDs]	long or string	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to ..admin/attributes.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid

			<ul style="list-style-type: none"> • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
	<code>includechildcontenttypes=[true false]</code>	boolean	Parameter to include child content types in search criteria.
	<code>orderby</code>	long or string	attributeIds used for determining the sequence of asset elements in query results.
Returns	Returns an <code>assetInfoList</code> object .		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/quicksearch?text=*5*></pre> <p>Result:</p> <pre><assetInfoList> <assetInfo id="1"> <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">151161dc.qxp</attributeValue> <attributeValue valueId="10" type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue valueId="1" type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue valueId="1" type="popup" name="Routed to" id="25">Admin</attributeValue> <attributeValue valueId="10" type="popup" name="Status" id="24">Status1</attributeValue> <attributeValue valueId="11021" type="popup" name="Content Type" id="62">QuarkXPress Project</attributeValue> </attributeValueList> </assetInfo> </assetInfoList></pre>		

setSharedSearchTrustees

Function	Share a search.
-----------------	-----------------

Description	This request shares any specified search to any user or group.		
Type	POST		
URL Format	.../searchdefinitions/{searchIdOrName}?		
Java API	Method Name		Class Name
	getSharedSearchTrustees		com.quark.qpp.service.facade Class QueryDefinitionFacade
Parameters	Name	Type	Description
	op=setsharedsearchtrustees	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	trusteeinfo=[xml]	string	An XML object containing metadata of all trustees (users, groups) that the search will be shared with.
Returns	Returns a <code>trusteeinfo</code> object containing metadata of all trustees (users, groups) that the search will be shared with.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/searchdefinitions?op=setsharedsearchtrustees&trusteeinfo=<xml/></code>		
	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <trusteeInfoList> <trusteeInfo id="10"> <user>true</user> <deleted>>false</deleted> <name>Admin2</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <ldapProfileId>10</ldapProfileId> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </trusteeInfo> <trusteeInfo id="12"> <user>>false</user> <deleted>>false</deleted> <name>Group1</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <ldapProfileId>10</ldapProfileId> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </trusteeInfo> </trusteeInfoList></pre>		

getSharedSearchTrustees

Function	Get the trustees of a shared search.		
Description	This request returns the trustees that a specified search has been shared with.		
Type	GET		
URL Format	.../searchdefinitions/{searchIdOrName}?		
Java API	Method Name		Class Name
	getSharedSearchTrustees		com.quark.qpp.service.facade Class QueryDefinitionFacade
Parameters	Name	Type	Description
	op=getsharedsearchtrustees	long	The op parameter allows you to specify which operation is to be performed.

			(Mandatory)
Returns	Returns a <code>trusteeInfoList</code> object containing metadata of all trustees (users, groups) that a specified search has been shared with.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/searchdefinitions?op=getsharedsearchtrustees</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <trusteeInfoList> <trusteeInfo id="10"> <user>true</user> <deleted>>false</deleted> <name>Admin2</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <ldapProfileId>10</ldapProfileId> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </trusteeInfo> <trusteeInfo id="12"> <user>>false</user> <deleted>>false</deleted> <name>Group1</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <ldapProfileId>10</ldapProfileId> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </trusteeInfo> </trusteeInfoList></pre>		

updateSearches

Function	Update searches.		
Description	This request updates a single or multiple searches at one time.		
Type	POST		
URL Format	.../searchdefinitions		
Java API	Method Name	Class Name	
	updateSearch	com.quark.qpp.service.facade Class QueryDefinitionFacade	
Parameters	Name	Type	Description
	op=update	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	searchinfolist=[xml]	string	An XML object containing metadata of searches to be updated.
Returns	Returns a <code>searchinfolist</code> object containing metadata of successfully updated searches.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/searchdefinitions?op=update&searchinfolist=<xml/></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <searchInfoList> <searchInfo parameterized="false" id="110"> <searchName>AllPictures</searchName> <ownerid="1">Admin</owner> <queryContextcontext="ASSET"/></pre>		

	<pre> <queryConditionList> <queryCondition recursive="false" parameterized="false" negated="false" type="CONTENT_TYPE_CONDITION"comparisonOperator="IS" logicalOperator="OR" nestingLevel="1"> <searchValueid="1103">System;Asset;Picture;</searchValue> </queryCondition> </queryConditionList> <queryDisplay exploreMode="PLAIN"displayMode="SNIPPET"> <displayColumnList> <displayColumn attributeColumn="true"width="150" columnId="2">Name</displayColumn> <displayColumn attributeColumn="false"width="150" columnId="29"/> <displayColumn attributeColumn="false"width="150" columnId="62"/> <displayColumn attributeColumn="false"width="150" columnId="54"/> <displayColumn attributeColumn="false"width="150" columnId="25"/> <displayColumn attributeColumn="false"width="150" columnId="55"/> </displayColumnList> <sortInfoList> <sortInfo attributeColumn="true"order="ascending" columnId="2">Name</sortInfo> </sortInfoList> </queryDisplay> </searchInfo> </searchInfoList> </pre>
--	---

deleteSearch

Function	Delete searches.		
Description	This request deletes a specified search.		
Type	POST		
URL Format	.../searchdefinitions/{searchIdOrName}?		
Java API	Method Name		Class Name
	updateSearch		com.quark.qpp.service.facade Class QueryDefinitionFacade
Parameters	Name	Type	Description
	op=delete	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/searchdefinitions/{Save dSearchIdORName}?op=delete		

count

Function	Count assets/collection using a searchDefinition.		
Description	This request counts assets/collections based on a given searchDefinition.		
Type	GET		
URL Format	.../search		
Java API	Method Name		Class Name
	count		com.quark.qpp.service.facade Class QueryFacade

Parameters	Name	Type	Description
	<code>searchinfo=[xml]</code>	string	An XML object containing metadata of the asset/collections to be counted.
Returns	Returns a count of the assets, if the context of the given searchInfo is ASSET . Returns a count of the collections, if the context of the given searchInfo is COLLECTION		
Examples	URL: <code>http://serverName:serverPort/rest/service/search?search=<xml/>&op=count</code> Result: 30		

countAssets

Function	Count assets.		
Description	This request counts assets based on given conditions.		
Type	GET		
URL Format	<code>.../search</code>		
Java API	Method Name	Class Name	
	<code>countAssets</code>	<code>com.quark.qpp.service.facade Class QueryFacade</code>	
Parameters	Name	Type	Description
	<code>op=count</code>	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>assets =[comma-separated list of assets names or IDs]</code>	long or string	(Mandatory)
	<code>recursive=[true false]</code>	boolean	Defaults to true . (Optional)
	<code>includechildcontenttypes=[true false]</code>	boolean	Parameter to include child content types in search criteria.
	<code>attributes=[comma-separated list of attribute names or IDs]</code>	long or string	Specify the string to search for. (Mandatory) You can pass the appropriate value for any queryable attribute to fetch specific attributes. For a list of attributes, use a call to <code>..admin/attributes</code> . The attributes must be queryable. The following is a list of non-queryable attributes:

			<ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns a count of the assets satisfying the condition parameters specified.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/search/assets?op=count</code></p> <p>Result:</p> <p>25</p>		

countCollections

Function	Count collections.
-----------------	--------------------

Description	This request counts collections based on given conditions.		
Type	GET		
URL Format	.../search		
Java API	Method Name		Class Name
	countCollections		com.quark.qpp.service.facade Class QueryFacade
Parameters	Name	Type	Description
	op=count	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	collections =[comma-separated list of collection names or IDs]	long or string	Parameter to include child collections. To get a list of collection IDs for a particular collection, use a call to ..collectionsbypath/[path].(Mandatory)
	recursive=[true false]	boolean	Defaults to true .(Optional)
	includechildcontenttypes=[true false]	boolean	Parameter to include child content types in search criteria.
	attributes=[comma-separated list of attribute names or IDs]	long or string	Specify the string to search for. (Mandatory) You can pass the appropriate value for any queryable attribute to fetch specific attributes. For a list of attributes, use a call to ..admin/attributes. The attributes must be queryable. The following is a list of non-queryable attributes: <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid •

			<ul style="list-style-type: none"> • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns a count of the collections satisfying the condition parameters specified.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/collections?op=count</pre> <p>Result:</p> <pre>25</pre>		

countSearchResultsForQuery

Function	Count assets based on saved query.		
Description	This request counts resultant assets of saved query.		
Type	GET		
URL Format	.../search/[Query ID or Name]		
Java API	Method Name		Class Name
	<code>countSearchResultsForQuery</code>		<code>com.quark.qpp.service.facade Class QueryFacade</code>
Parameters	Name	Type	Description
	<code>op=count</code>	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributes=[comma-separated list of</code>	long or string	Specify the string to search for. (Mandatory)

	<p>attribute names or IDs]</p>	<p>You can pass the appropriate value for any queryable attribute to fetch specific attributes.</p> <p>For a list of attributes, use a call to <code>..admin/attributes</code>.</p> <p>The attributes must be queryable.</p> <p>The following is a list of non-queryable attributes:</p> <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns a count of the assets satisfying the specified query.	

Examples	URL: <code>http://serverName:serverPort/rest/service/search/AllAssets?op=count</code>
	Result: 2

countTextSearchResults

Function	Count text search results.		
Description	This request counts assets based on given conditions.		
Type	GET		
URL Format	<code>.../search/quicksearch</code>		
Java API	Method Name		Class Name
	<code>countTextSearchResults</code>		<code>com.quark.qpp.service.facade Class QueryFacade</code>
Parameters	Name	Type	Description
	<code>op=count</code>	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>collections =[comma-separated list of collections names or IDs]</code>	long or string	Parameter to include child collections. To get a list of collection IDs for a particular collection, use a call to <code>..collectionsbypath/[path]</code> . (Mandatory)
	<code>recursive=[true false]</code>	boolean	Defaults to true . (Optional)
	<code>includechildcontenttypes=[true false]</code>	boolean	Parameter to include child content types in search criteria.
	<code>attributes=[comma-separated list of attribute names or IDs]</code>	long or string	Specify the string to search for. (Mandatory) You can pass the appropriate value for any queryable attribute to fetch specific attributes. For a list of attributes, use a call to <code>..admin/attributes</code> . The attributes must be queryable. The following is a list of non-queryable attributes: <ul style="list-style-type: none"> • Article Component id • Article Component Reference Asset Id

			<ul style="list-style-type: none"> • Article Component Reference Asset Major Version • Article Component Reference Asset Minor Version • Article Id • Asset Checksum • Attached Component Id • Attached Layout Uid • Attachment ChecksumAuxillary Data • Box Height • Box Uid • Box Width • Collection • Component position • Dependent On Collection Resources • InDesign Object UID • Item Name • Item Type • Page Index • Page Name • Relationship Status • Status • XPath
Returns	Returns a count of the assets satisfying the condition parameters specified.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/search/quicksearch?text=*quark*&op=count</pre> <p>Result:</p> <p>2</p>		

Administration

The administration capabilities of the API allow application developers to perform administrative tasks and includes the following REST calls:

- *createUsers*
- *getUser*

- *getAllUsers*
- *getUsersWithGivenUserClass*
- *getAllUserRedliningColor*
- *updateUserRedliningColor*
- *updateUserEnabled*
- *updateUsers*
- *deleteUser*
- *createGroups*
- *getGroup*
- *getAllGroups*
- *getGroupsWithGivenUserClass*
- *getuserGroups*
- *updateGroups*
- *deleteGroup*
- *createContentTypes*
- *getContentTypes*
- *getAllContentTypes*
- *updateContentTypes*
- *deleteContentType*
- *createAttributes*
- *setAttributes*
- *getAttribute*
- *getAllAttributes*
- *updateAttributes*
- *deleteAttribute*
- *createForms*
- *getAllForms*
- *updateForms*
- *deleteForm*
- *createRoles*
- *getRole*
- *getAllRoles*
- *updateRoles*
- *deleteRole*

REST FRAMEWORK SERVICES

- *createRelationType*
- *getRelationType*
- *getAllRelationTypes*
- *updateRelationTypes*
- *deleteRelationType*
- *createDomains*
- *getDomain*
- *getAllDomains*
- *updateDomains*
- *deleteDomain*
- *createLdapProfiles*
- *getLdapProfile*
- *getAllLdapProfiles*
- *updateLdapProfiles*
- *deleteLdapProfile*
- *getLdapUsers*
- *importUsersFromLdap*
- *getLdapGroups*
- *importGroupsFromLdap*
- *synchronizeLdapUsersGroups*
- *synchronizeLdapUsersGroupsBasedOnProfile*
- *createLdapSearches*
- *getLdapSearch*
- *getAllLdapSearches*
- *updateLdapSearches*
- *deleteLdapSearch*
- *getTrustee*
- *updateTrusteeRule*
- *createWorkflows*
- *getWorkflow*
- *getAllWorkflows*
- *updateWorkflows*
- *deleteWorkFlow*
- *createPreferences*

- *getPreference*
- *getAllPreferences*
- *updatePreferences*
- *getPreferenceValues*
- *updatePreferenceValues*
- *deletePreference*
- *createRepositories*
- *getRepository*
- *getAllRepositories*
- *updateRepositories*
- *deleteRepository*
- *createPrivileges*
- *getPrivilege*
- *getAllPrivileges*
- *getApplicableApplicationPrivileges*
- *getApplicableContentPrivileges*
- *updatePrivileges*
- *deletePrivilege*
- *createPrivilegeGroups*
- *getPrivilegeGroup*
- *getAllPrivilegeGroups*
- *updatePrivilegeGroups*
- *deletePrivilegeGroup*
- *addDomainValue*
- *getDomainValue*
- *getDomainValueXML*
- *setDomainValueXML*
- *searchDomainValues*
- *updateDomainValue*
- *deleteDomainValue*
- *getFavorites*
- *setFavorites*
- *getWorkflowInitiatingStatuses*
- *getStatusTransitions*

createUsers

Function	Create users.		
Description	This request is used to create users. The new user will be created without a password, client needs to set the password for the user separately.		
Type	POST		
URL Format	.../admin/users		
Java API	Method Name		Class Name
	createUsers		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	userinfolist		Object containing a list of the users (userInfo) to be created. See the userInfoList parameter in "Request/Response parameter formats". The following are mandatory attributes required to create a user: <ul style="list-style-type: none"> • Name • DefaultUserClass • Enabled,NativeTrustee • DefaultUserClassId
Returns	Returns a <code>userInfoList</code> object containing metadata of a newly created user.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/users?op=create&userinfolist= <?xml version="1.0" encoding="UTF-8"?> <userInfoList> <userInfo> <name>SampleQPPGroup</name> <nativeTrustee>true</nativeTrustee> <enabled>true</enabled> <defaultUserClassId>1</defaultUserClassId> <defaultUserClass>Administrator</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="14"> <name>SampleQPPGroup</name> <nativeTrustee>true</nativeTrustee> <enabled>true</enabled> <defaultUserClassId>1</defaultUserClassId> <defaultUserClass>Administrator</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList>Example with multiple usersURL:http://serverName:serverPort/rest/service/admin/users ?op=create&userinfolist= <?xml version="1.0" encoding="UTF-8"?> <userInfoList> <userInfo> <name>SampleQPPGroup1</name> <nativeTrustee>true</nativeTrustee> <enabled>true</enabled> <defaultUserClassId>1</defaultUserClassId></pre>		

```
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <deleted>>false</deleted>
<ldapDistinguishedName>Dist Name</ldapDistinguishedName>
</userInfo> <userInfo> <name>SampleQPPGroup2</name>
<nativeTrustee>>true</nativeUser> <enabled>>true</enabled>
<defaultUserClassId>1</defaultUserClassId>
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <deleted>>false</deleted>
<ldapDistinguishedName>Dist Name</ldapDistinguishedName>
</userInfo> </userInfoList>Result: <?xml version="1.0"
encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo
id="14"> <name>SampleQPPGroup1</name>
<nativeTrustee>>true</nativeTrustee> <enabled>>true</enabled>
<defaultUserClassId>1</defaultUserClassId>
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <deleted>>false</deleted>
<ldapDistinguishedName>Dist Name</ldapDistinguishedName>
</userInfo> <userInfo id="15"> <name>SampleQPPGroup2</name>
<nativeTrustee>>true</nativeTrustee> <enabled>>true</enabled>
<defaultUserClassId>1</defaultUserClassId>
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <deleted>>false</deleted>
<ldapDistinguishedName>Dist Name</ldapDistinguishedName>
</userInfo> </userInfoList>
```

getUser

Function	Get specific user.	
Description	This request is used to fet the metadata of a specified user.	
Type	GET	
URL Format	.../admin/users/[userIdORuserName]	
Java API	Method Name	Class Name
	getUser	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters	
Returns	Returns an <code>userInfoList</code> object with its metadata.	
Examples	URL: http://serverName:serverPort/rest/service/users/Admin	
	Result: <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="1"> <name>Admin</name> <firstName>Administrator</firstName> <nativeTrustee>>true</nativeTrustee> <enabled>>true</enabled> <defaultUserClassId>1</defaultUserClassId> <defaultUserClass>Administrator</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList>	

getAllUsers

Function	Get all users.
-----------------	----------------

Description	This request is used to fetch all users.		
Type	GET		
URL Format	.../admin/users		
Java API	Method Name	Class Name	
	getAllUsers	com.quark.qpp.service.facade Class AdminFacade	
Parameters	This request requires no input parameters		
Returns	Returns an <code>userInfoList</code> object containing metadata of users in the form of an array.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/users</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="1"> <name>Admin</name> <domainId>0</domainId> <firstName>Administrator</firstName> <phoneNumber>+911722299000</phoneNumber> <emailAddress>platformadmin@quark.com</emailAddress> <nativeTrustee>true</nativeTrustee> <enabled>true</enabled> <defaultUserId>1</defaultUserId> <defaultUserClass>Administrator</defaultUserClass> <ldapProfileId>0</ldapProfileId> <user>true</user> <deleted>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <overriddenUserId>0</overriddenUserId> </userInfo> </userInfoList></pre>		

getUsersWithGivenUserClass

Function	Get users with the default given userclass.		
Description	This request returns the platform users with the specified user class.		
Type	GET		
URL Format	.../admin/roles/{roleIdORroleName}		
Java API	Method Name	Class Name	
	getUsersWithGivenUserClass	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	users	string	true
Returns	Returns a <code>userInfoList</code> containing metadata of all the users belonging to the specified userClass in array format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/roles/Administrator?users=true</pre> <p>Result:</p>		

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<userInfoList> <userInfo id="1"> <name>Admin</name>
<domainId>0</domainId> <firstName>Administrator</firstName>
<emailAddress>qppadministrator@quark.com</emailAddress>
<nativeTrustee>true</nativeTrustee> <enabled>true</enabled>
<defaultUserClassId>1</defaultUserClassId>
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <user>true</user>
<deleted>false</deleted>
<overriddenUserClassId>0</overriddenUserClassId>
<ldapDistinguishedName>Dist Name</ldapDistinguishedName>
</userInfo> <userInfo id="10"> <name>Admin2</name> .....
</userInfo> </userInfoList>
```

getAllUserRedliningColor

Function	Get a list of all users along with redline colors.		
Description	This request returns a list of all the users along with their redlining colors.		
Type	GET		
URL Format	.../redline		
Java API	Method Name	Class Name	
	getAllUserRedliningColor	com.quark.qpp.service.facade Class AdminFacade	
Parameters	This request requires no input parameters.		
Returns	Returns list of all the users along with their redlining colors.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/redline		
Examples	Result:		
	<pre><redlineInfoList> <redlineInfo> <color blue="0" green="0" red="0"/> <user id="11">BOT</user> </redlineInfo> <redlineInfo> <color blue="0" green="0" red="0"/> <user id="1">Admin</user> </redlineInfo> <redlineInfo> <color blue="0" green="0" red="0"/> <user id="10">Admin2</user> </redlineInfo> </redlineInfoList></pre>		

Function	Get Redline color for a given user.		
Description	This request returns the defined redlining color for the given user.		
Type	GET		
URL Format	.../redline		
Java API	Method Name	Class Name	
	getAllUserRedliningColor	com.quark.qpp.service.facadeClass AdminFacade	
Parameters	Name	Type	Description
	user	string	Name of the user for which redline color information is to be fetched.
Returns	Returns the defined redlining color for the given user. Returns specific privilege(s) in system with given id or name. If privilege name is		

	specified and there exists multiple privileges in different privilege groups, with the given name, list of all such privileges will be returned.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/redline?user=Admin</pre> <p>Result:</p> <pre><redlineInfoList> <redlineInfo> <color blue="252" green="8" red="126"/> <user id="1">Admin</user> </redlineInfo> </redlineInfoList></pre>

updateUserRedliningColor

Function	Updates the redlining color for the given user(s).		
Description	This request updates the redlining color for the given user(s).		
Type	POST		
URL Format	.../redline? op=update		
Java API	Method Name		Class Name
	updateUserRedliningColor		com.quark.qpp.service.facadeClass AdminFacade
Parameters	Name	Type	Description
	redlineinfolist=<XML/>	string	A redLineInfoList object in XML format. See "Request/Response parameter formats."
Returns	Returns the defined redlining color for the given user(s).		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/redline?user=BOT</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <redlineInfoList> <redlineInfo> <color blue="10" green="56" red="115"/> <user id="11">BOT</user> </redlineInfo> </redlineInfoList></pre>		

updateUserEnabled

Function	Enables/disables the Platform Server Logon for the given user.		
Description	This request enables/disables the Platform Server logon for the given user.		
Type	POST		
URL Format	.../admin/users/{userIdOrName}		
Java API	Method Name		Class Name
	updateUserEnabled		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=setenabled	string	The op parameter allows you to specify

			<p>which operation is to be performed. (Mandatory)</p> <p>This operation will run only for a valid session id.</p>
	enabled=[True] or [False]	boolean	
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/admin/users/John?op=setenabled&enabled=true:</code></p>		

updateUsers

Function	Update users.		
Description	This request is used to update a single user or multiple users at one time.		
Type	POST		
URL Format	.../admin/users		
Java API	Method Name		Class Name
	updateUsers		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	userinfoList=<xml/>		Object containing a list of the successfully updated users .
Returns	Returns a <code>UserInfoList</code> object containing a list of all the successfully updated users.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/admin/users?userinfoList= <userInfoList> <userInfo id="33"> <name>Tom</name> <firstName>Tom</firstName> <lastName>Rodrigues</lastName> <nativeTrustee>true</nativeTrustee> <enabled>>false</enabled> <defaultUserId>12</defaultUserId> <defaultUserClass>Editor</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList></code></p> <p>Result:</p> <p><code><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="33"> <name>Tom</name> <firstName>Tom</firstName> <lastName>Rodrigues</lastName> <nativeTrustee>true</nativeTrustee> <enabled>>false</enabled> <defaultUserId>12</defaultUserId> <defaultUserClass>Editor</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList></code></p>		

	<p>Example with multiple content types</p> <p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/users?userinfo list= <userInfoList> <userInfo id="33"> <name>Tom</name> <firstName>Tom</firstName> <lastName>Rodrigues</lastName> <nativeTrustee>true</nativeTrustee> <enabled>>false</enabled> <defaultUserId>12</defaultUserId> <defaultUserClass>Editor</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> <userInfo id="34"> <name>Sam</name> <firstName>Sam</firstName> <lastName>Rodrigues</lastName> <nativeTrustee>true</nativeTrustee> <enabled>>false</enabled> <defaultUserId>12</defaultUserId> <defaultUserClass>Editor</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="33"> <name>Tom</name> <firstName>Tom</firstName> <lastName>Rodrigues</lastName> <nativeTrustee>true</nativeTrustee> <enabled>>false</enabled> <defaultUserId>12</defaultUserId> <defaultUserClass>Editor</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> <userInfo id="34"> <name>Sam</name> <firstName>Sam</firstName> <lastName>Rodrigues</lastName> <nativeTrustee>true</nativeTrustee> <enabled>>false</enabled> <defaultUserId>12</defaultUserId> <defaultUserClass>Editor</defaultUserClass> <ldapProfileId>0</ldapProfileId> <deleted>>false</deleted> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> </userInfo> </userInfoList></pre>
--	---

deleteUser

Function	Delete a user.		
Description	This request is used to delete the specified user.		
Type	POST		
URL Format	.../admin/users/[userIdORName]		
Java API	Method Name	Class Name	
	deleteUser	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	userIdORName		Id or Name of the user to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL:		

	<code>http://serverName:serverPort/rest/service/admin/users/32?op=delete</code>
--	---

createGroups

Function	Create groups.		
Description	This request is used to create a single or multiple groups at one time.		
Type	POST		
URL Format	.../admin/groups		
Java API	Method Name		Class Name
	<code>createGroups</code>		<code>com.quark.qpp.service.facade</code> Class <code>AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>groupinfolist</code>		Object containing list of groups to be created. See "Request/Response parameter formats."
Returns	Returns a <code>groupInfoList</code> object containing its metadata and memberlist.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/admin/groups?op=create &groupInfo=<groupInfoList> <name>Editorial</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre>		
	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <groupInfo id="48"> <name>SampleQPdzxcdPGroup</name> <domainId>0</domainId> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1" name="Admin"/> <nameId id="10" name="Admin2"/> </memberList> </groupInfo><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <groupInfoList> <groupInfo id="20"> <name>Editoria</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist</pre>		

	<pre>Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre> <p>Multiple Groups Example:</p> <p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/groups?op=create&groupinfolist= <groupInfoList> <groupInfo> <name>Department</name> <deleted>>false</deleted> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> <groupInfo> <name>Production</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <groupInfoList> <groupInfo id="21"> <name>Department</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> <groupInfo id="22"> <name>Production</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre>
--	--

getGroup

Function	Get a group with a particular ID or Name.	
Description	This request returns all the metadata of specified group.	
Type	GET	
URL Format	.../admin/groups/[groupIDORgroupName]	
Java API	Method Name	Class Name
	getGroup	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns a <code>groupInfoList</code> object in XML format of specified group. See the table in the "Request/Response parameter formats" chapter.	
Examples	URL:	

	<p><code>http://serverName:serverPort/rest/service/admin/groups/21</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <groupInfoList> <groupInfo id="21"> <name>Department</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>false</nativeTrustee> <defaultUserId>2</defaultUserId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre>
--	--

getAllGroups

Function	Get all groups.	
Description	This request returns all groups including its members.	
Type	GET	
URL Format	.../admin/groups	
Java API	Method Name	Class Name
	getAllGroups	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns a <code>groupInfoList</code> object containing an array of groups that exist on the QPP server. See the table in the "Request/Response parameter formats" chapter.	
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/admin/groups</code></p> <p>Result:</p> <pre><groupInfoList> <groupInfo> <name>gp34</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>false</nativeTrustee> <defaultUserId>2</defaultUserId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId name="Sam" /> <nameId id="1"/> </memberList> </groupInfo> <groupInfo > <name>gp35</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>false</nativeTrustee> <defaultUserId>2</defaultUserId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId name="Sam" /> <nameId id="1"/> </memberList> </groupInfo> </groupInfoList></pre>	

getGroupsWithGivenUserClass

Function	Get groups with the given role.		
Description	This request returns the Platform Server groups with the given role.		
Type	GET		
URL Format	.../admin/roles/{roleIdOrName}		
Java API	Method Name		Class Name
	getGroupsWithGivenUserClass		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	groups=[true]	boolean	
Returns	Returns a <code>groupInfoList</code> object containing a list of groups that have the default role.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/roles/Administrator? groups=true</pre> <p>Result:</p> <pre><groupInfoList> <groupInfo id="12"> <name>Editors</name> <nativeTrustee>true</nativeTrustee> <defaultUserClassId>1</defaultUserClassId> <defaultUserClass>Administrator</defaultUserClass> <memberList> <nameId id="3680"> <name>John</name> </nameId> </memberList> </groupInfo> <groupInfo id="13"> <name>Publishers</name> <nativeTrustee>true</nativeTrustee> <defaultUserClass>Administrator</defaultUserClass> <memberList> <nameId id="10"> <name>Mathias</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre>		

getUserGroups

Function	Get groups of a user.		
Description	This request returns all groups to which the given user belongs.		
Type	GET		
URL Format	.../admin/users/{userIdOrName}		
Java API	Method Name		Class Name
	getUserGroups		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	userIdOrName	string	
Returns	Returns a <code>groupInfoList</code> object containing a list of groups of which the given user is a member.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/users/John?groups=true</pre>		

	<p>Result:</p> <pre><groupInfoList> <groupInfo id="12"> <name>Editors</name> <nativeTrustee>true</nativeTrustee> <defaultUserClassId>123/defaultUserClassId> <defaultUserClass>Editors</defaultUserClass> </groupInfo> <groupInfo id="3681"> <name>QPP Server</name> <nativeTrustee>false</nativeTrustee> <defaultUserClassId>10/defaultUserClassId> <defaultUserClass>UserClass2</defaultUserClass> <ldapProfileId>764/ldapProfileId> <ldapProfile>QPP LDAP Profile</ldapProfile> <ldapDistinguishedName> CN=QPP Server,OU=DistributionLists,OU=QMHI,DC=india,DC=quark,DC=com </ldapDistinguishedName> </groupInfo> </groupInfoList></pre>
--	---

updateGroups

Function	Update groups.		
Description	This request is used to update a single or multiple groups at one time.		
Type	POST		
URL Format	.../admin/groups		
Java API	Method Name		Class Name
	updateGroups		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	groupinfolist	XML	Object containing information for single or multiple groups. See "Request/Response parameter formats."
Returns	Returns a groupInfoList object containing a list of all successfully updated groups.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/admin/groups?op=update&groupInfo= <groupInfoList> <groupInfo id="22"> <name>Production</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList></pre>		
	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <groupInfoList> <groupInfo id="22"> <name>Production</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass></pre>		

	<pre> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList> </pre> <p>Example with multiple groups:</p> <p>URL:</p> <pre> http://serverName:serverPort/rest/service/admin/groups?op=update&groupInfo= <groupInfoList> <groupInfo id="21"> <name>Department</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> <groupInfo id="20"> <name>Editoria</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <groupInfoList> <groupInfo id="21"> <name>Department</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> <groupInfo id="20"> <name>Editoria</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserClassId>2</defaultUserClassId> <defaultUserClass>NoPrivileges</defaultUserClass> <ldapProfileId>10</ldapProfileId> <ldapProfile>Profile Name</ldapProfile> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <memberList> <nameId id="1"> <name>Admin</name> </nameId> </memberList> </groupInfo> </groupInfoList> </pre>
--	--

deleteGroup

Function	Delete a group.	
Description	This request is used to delete the group with the given id or name.	
Type	POST	
URL Format	.../admin/group/[groupIdORgroupName]	
Java API	Method Name	Class Name

	<code>deleteGroup</code>	<code>com.quark.qpp.service.facade</code> <code>Class AdminFacade</code>
Parameters	Name	Type
	<code>op=delete</code>	string
	Description	
	The op parameter allows you to specify which operation is to be performed. (Mandatory)	
Returns	A status of 200 is returned in response to a successful request execution.	
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/groups/[groupIdOR groupName]?op=delete</code>	

createContentTypes

Function	Create content types.		
Description	This request is used to create one or more content types and its child content types according to the parent content type ID.		
Type	POST		
URL Format	<code>.../admin/contenttypes</code>		
Java API	Method Name		Class Name
	<code>createContentTypes</code>		<code>com.quark.qpp.service.facade</code> <code>Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>contenttypeinfolist=<xml/></code>		Object containing a list of the content types to be created. After the creation of a content type, the child content types defined will be considered for creation. See the <code>contentTypeInfoList</code> parameter in "Request/Response parameter formats".
Returns	Returns a <code>contentTypeInfoList</code> object containing containing the successfully created content types..		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/contenttypes?op=create&contenttypeinfolist= <contentTypeInfoList></code> <code><contentTypeInfo> <name>Pru Child1</name></code> <code><extensible>true</extensible></code> <code><parentContentTypeId>10010</parentContentTypeId></code> <code><contentTypeInfoList> <contentTypeInfo> <name>Child of Project 1</name> <extensible>true</extensible> </contentTypeInfo></code> <code></contentTypeInfo> </contentTypeInfoList></code> Result: <code><?xml version="1.0" encoding="UTF-8" standalone="yes"?></code> <code><contentTypeInfoList> <contentTypeInfo id="10011"> <name>Pru Child1</name></code> <code><contentTypeHierarchy>System;Asset;Prudencial;Pru</code>		


```

Child1;</contentTypeHierarchy>
<parentContentTypeId>10010</parentContentTypeId>
<systemDefined>>false</systemDefined>
<childrenAvailable>>true</childrenAvailable>
<extensible>>true</extensible> <created>2013-11-
25T07:21:00Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-11-
25T07:21:00Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="false" attributeId="1">Id
</contentTypeAttributeMappingInfo>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="true" attributeId="2">Name
</contentTypeAttributeMappingInfo> .....
</contentTypeAttributeMappingInfoList> <contentTypeInfoList>
<contentTypeInfo id="10012"> <name>Child of Project 1</name>
<contentTypeHierarchy>System;Asset;Prudencial;Pru Child1;Child
of Project 1;</contentTypeHierarchy>
<parentContentTypeId>10011</parentContentTypeId>
<systemDefined>>false</systemDefined>
<extensible>>true</extensible> <created>2013-11-
25T07:21:00Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-11-
25T07:21:00Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="false" attributeId="1">Id
</contentTypeAttributeMappingInfo>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="true" attributeId="2">Name
</contentTypeAttributeMappingInfo> .....
</contentTypeAttributeMappingInfoList> </contentTypeInfo>
</contentTypeInfoList>

```

Example with multiple content types:

URL:

```

http://serverName:serverPort/rest/service/admin/contenttypes?p=
create&contentTypeInfoList= <contentTypeInfoList>
<contentTypeInfo> <name>Pru 2</name>
<parentContentTypeId>10010</parentContentTypeId>
<childrenAvailable>>true</childrenAvailable>
<extensible>>true</extensible>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="false" attributeId="1">Id
</contentTypeAttributeMappingInfo> .....
</contentTypeAttributeMappingInfoList> <contentTypeInfoList>
<contentTypeInfo> <name>Child 1 of Pru 2</name>
<parentContentTypeId>10011</parentContentTypeId>
<extensible>>true</extensible>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo userMapped="true"
valueMandatory="false" attributeId="509">Master List
</contentTypeAttributeMappingInfo> .....
</contentTypeAttributeMappingInfoList> </contentTypeInfo><>:

```

Function	Create content types through content type hierarchy
Description	This request is used to create one or more content types and its child content types according to the specified hierarchy

Type	POST		
URL Format	.../admin/contenttypes		
Java API	Method Name		Class Name
	createContentTypes		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contentTypeInfoList=<xml/>		Object containing a list of the content types to be created. After the creation of a content type, the child content types defined will be considered for creation. See the contentTypeInfoList parameter in "Request/Response parameter formats".
Returns	Returns a <code>contentTypeInfoList</code> object containing the successfully created content types.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/contenttypes?op=create&contentTypeInfoList= <contentTypeInfoList> <contentTypeInfo> <name>Prudencial</name> <contentTypeHierarchy>System;Asset;Prudencial;</contentTypeHierarchy> <extensible>>true</extensible> <contentTypeAttributeMappingInfoList> <contentTypeAttributeMappingInfo valueMandatory="true">Pru_Id</contentTypeAttributeMappingInfo> <contentTypeAttributeMappingInfo userMapped="true" valueMandatory="false" attributeId="501">Master Text </contentTypeAttributeMappingInfo> </contentTypeAttributeMappingInfoList> </contentTypeInfo> </contentTypeInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <contentTypeInfoList> <contentTypeInfo id="10010"> <name>Prudencial</name> <contentTypeHierarchy>System;Asset;Prudencial;</contentTypeHierarchy> <parentContentTypeId>1100</parentContentTypeId> <systemDefined>>false</systemDefined> <extensible>>true</extensible> <contentTypeAttributeMappingInfoList> <contentTypeAttributeMappingInfo userMapped="false" valueMandatory="false" attributeId="1">Id </contentTypeAttributeMappingInfo> <contentTypeAttributeMappingInfo userMapped="false" valueMandatory="true" attributeId="2">Name </contentTypeAttributeMappingInfo> </contentTypeAttributeMappingInfoList> </contentTypeInfo> </contentTypeInfoList></pre>		

getContenttype

Function	Get content type.
----------	-------------------

Description	This request returns complete information on the specified content type. This will also include the child content types.		
Type	Get		
URL Format	.../admin/contenttypes/{contentTypeIdorName}		
Java API	Method Name		Class Name
	contentType		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	contentTypeIdOrName	string	The ID or Name of the content type to be retrieved.
Returns	Returns a <code>contentTypeInfoList</code> object containing information on the specified content type inclusive of its children .		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/contenttypes/10000</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <contentTypeInfoList> <contentTypeInfo id="10000"> <name>Master</name> <contentTypeHierarchy>System;Article Component;Picture Component;Master;</contentTypeHierarchy> <parentContentTypeId>1300</parentContentTypeId> <systemDefined>>false</systemDefined> <extensible>>false</extensible> <created>2013-11- 22T07:04:29Z</created> <creator>1</creator> <creatorName>Admin</creatorName> <lastModified>2013-11- 22T07:04:29Z</lastModified> <lastModifier>1</lastModifier> <lastModifierName>Admin</lastModifierName> <contentTypeAttributeMappingInfoList> <contentTypeAttributeMappingInfo userMapped="false" valueMandatory="false" attributeId="1">Id </contentTypeAttributeMappingInfo> <contentTypeAttributeMappingInfo userMapped="false" valueMandatory="false" attributeId="51">First page </contentTypeAttributeMappingInfo> </contentTypeAttributeMappingInfoList> </contentTypeInfo> </contentTypeInfoList></pre>		

getAllContentTypes

Function	Get all content types.		
Description	This request returns complete information of all content types in the system in a hierarchical structure.		
Type	GET		
URL Format	.../admin/contenttypes		
Java API	Method Name		Class Name
	getAllContentTypes		com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters		

Returns	Returns a <code>contentTypeInfoList</code> object containing an array of content types (<code>contentTypeInfo</code>).
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/contenttypes</pre> <p>Result:</p> <pre><contentTypeInfoList> <contentTypeInfo id="1502"> <name>Web</name> <contentTypeHierarchy>System;Layout;Web;</contentTypeHierarchy > <parentContentTypeId>1500</parentContentTypeId> <systemDefined>true</systemDefined> <extensible>false</extensible> <created>2012-04- 01T10:00:00Z</created> <creator>1</creator> <creatorName>Admin</creatorName> <lastModified>2012-04- 01T10:00:00Z</lastModified> <lastModifier>1</lastModifier> <lastModifierName>Admin</lastModifierName> <contentTypeAttributeMappingInfoList> <contentTypeAttributeMappingInfo userMapped="false" valueMandatory="false" attributeId="1">Id</contentTypeAttributeMappingInfo> <contentTypeAttributeMappingInfo userMapped="false" valueMandatory="false" attributeId="43">Number of pages</contentTypeAttributeMappingInfo> </contentTypeAttributeMappingInfoList> </contentTypeInfo> </contentTypeInfoList></pre>

updateContentTypes

Function	Update content types.		
Description	This request is used to update content types and their children according to the specified id or contentTypeHierarchy.		
Type	POST		
URL Format	.../admin/contenttypes		
Java API	Method Name		Class Name
	updateContentTypes		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contentTypeInfoList=<xml/>		Object containing a list of the successfully updated content types .
Returns	Returns a <code>contentTypeInfoList</code> object containing a list of all the successfully updated content types..		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/contenttypes?p=update&contentTypeInfoList= <contentTypeInfoList> <contentTypeInfo id="10015"> <contentTypeAttributeMappingInfoList> <contentTypeAttributeMappingInfo> <attributeName>Id</attributeName></pre>		

```
<valueMandatory>true</valueMandatory>
</contentTypeAttributeMappingInfo>
</contentTypeAttributeMappingInfoList> </contentTypeInfo>
</contentTypeInfoList>
```

Result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contentTypeInfoList> <contentTypeInfo id="10015">
<name>Prudential</name>
<contentTypeHierarchy>System;Asset;Prudential;</contentTypeHie
rarchy> <parentContentTypeId>1100</parentContentTypeId>
<systemDefined>>false</systemDefined>
<childrenAvailable>true</childrenAvailable>
<extensible>true</extensible> <created>2013-11-
25T07:35:37Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-11-
25T07:35:37Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="false" attributeId="1">Id
</contentTypeAttributeMappingInfo> .....
</contentTypeAttributeMappingInfoList> <contentTypeInfoList>
<contentTypeInfo id="10016"> <name>Pru Nest1</name>
<contentTypeHierarchy>System;Asset;Prudential;Pru
Nest1;</contentTypeHierarchy>
<parentContentTypeId>10015</parentContentTypeId>
<systemDefined>>false</systemDefined>
<extensible>true</extensible>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo userMapped="false"
valueMandatory="false" attributeId="1">Id
</contentTypeAttributeMappingInfo> .....
</contentTypeAttributeMappingInfoList> </contentTypeInfo>
</contentTypeInfoList>
```

Example with multiple content types:

URL:

```
http://serverName:serverPort/rest/service/admin/users?op=creat
e&userinfolist= <?xml version="1.0" encoding="UTF-8"?>
<userInfoList> <userInfo> <name>SampleQPPGroup1</name>
<nativeTrustee>true</nativeTrustee> <enabled>true</enabled>
<defaultUserClassId>1</defaultUserClassId>
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <deleted>>false</deleted>
</userInfo> <userInfo> <name>SampleQPPGroup2</name>
<nativeTrustee>true</nativeTrustee> <enabled>true</enabled>
<defaultUserClassId>1</defaultUserClassId>
<defaultUserClass>Administrator</defaultUserClass>
<ldapProfileId>0</ldapProfileId> <deleted>>false</deleted>
</userInfo> </userInfoList>
```

Result:

```
http://serverName:serverPort/rest/service/admin/contenttypes?o
p=update&contenttypeinfolist= <contentTypeInfoList>
<contentTypeInfo> <name>project aaa new name</name>
<contentTypeAttributeMappingInfoList>
<contentTypeAttributeMappingInfo> <attributeName>Master
List</attributeName> <valueMandatory>true</valueMandatory>
</contentTypeAttributeMappingInfo>
</contentTypeAttributeMappingInfoList>
<contentTypeHierarchy>System;Asset;Prudential;project aaa new
name;</contentTypeHierarchy> </contentTypeInfo>
<contentTypeInfo> <name>Pru Nest 1</name>
```

<pre> <contentTypeAttributeMappingInfoList> <contentTypeAttributeMappingInfo> <attributeName>Master Day</attributeName> <valueMandatory>>true</valueMandatory> </contentTypeAttributeMappingInfo> </contentTypeAttributeMappingInfoList> <contentTypeHierarchy>System;Asset;Prudential;Pru Nest 1 New Name;</contentTypeHierarchy> </contentTypeInfo> </contentTypeInfoList> </pre>
--

deleteContentType

Function	Delete a content type.		
Description	This request is used to delete the specified content type.		
Type	POST		
URL Format	.../admin/contenttypes/{contentTypeIdorName}		
Java API	Method Name		Class Name
	deleteContentType		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contentTypeIdorName	string	Id or Name of the content type to be deleted.
	attributes= [comma-separated list of attribute names or IDs]	string	Attributes Ids or Names.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/admin/contenttypes/10015?op=delete:		

createAttributes

Function	Create Attributes.		
Description	This request is used to create a single or multiple attributes at one time.		
Type	POST		
URL Format	.../admin/attributes		
Java API	Method Name		Class Name
	createAttributes		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed.

			(Mandatory)
	attributeinfolist	string	List of attributes to create.
Returns	Returns a <code>attributeInfoList</code> object containing a list of successfully created attributes.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/attributes?op=create&attributeinfolist= <attributeInfoList> <attributeInfo> <name>Collections Sequence</name> <valueType>TEXT</valueType> <queryable>true</queryable> <creatorName>Admin</creatorName> <displayable>true</displayable> <valuePreferencesChangeable>true</valuePreferencesChangeable> <attributeValuePreferenceInfo> <defaultValue>From Creation to Publishing</defaultValue> </attributeValuePreferenceInfo> </attributeInfo> </attributeInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeInfoList> <attributeInfo id="519"> <name>Collections Sequence</name> <valueTypeId>1</valueTypeId> <valueType>TEXT</valueType> <userDefined>true</userDefined> <queryable>true</queryable> <modificationLevel>USER_MODIFIABLE</modificationLevel> <created>2013-12-05T05:52:57Z</created> <creator>1</creator> <creatorName>Admin</creatorName> <lastModified>2013-12-05T05:52:57Z</lastModified> <lastModifier>1</lastModifier> <lastModifierName>Admin</lastModifierName> <displayable>true</displayable> <limitedAccess>false</limitedAccess> <limitedAccessChangeable>true</limitedAccessChangeable> <constraintsChangeable>true</constraintsChangeable> <valuePreferencesChangeable>true</valuePreferencesChangeable> <attributeValuePreferenceInfo> <defaultValue>From Creation to Publishing</defaultValue> <textSearchEnabled>false</textSearchEnabled> </attributeValuePreferenceInfo> </attributeInfo> </attributeInfoList></pre> <p>Example with multiple attributes:</p> <p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/attributes?op=create&attributeinfolist= <attributeInfoList> <attributeInfo> <name>Collections Sequence 1</name> <valueType>TEXT</valueType> <queryable>true</queryable> <creatorName>Admin</creatorName> <displayable>true</displayable> <valuePreferencesChangeable>true</valuePreferencesChangeable> <attributeValuePreferenceInfo> <defaultValue>From Creation to Publishing</defaultValue> </attributeValuePreferenceInfo> </attributeInfo> <attributeInfo> <name>Collections Sequence 2</name> <valueType>TEXT</valueType> <queryable>true</queryable> <creatorName>Admin</creatorName> <displayable>true</displayable> <valuePreferencesChangeable>true</valuePreferencesChangeable> <attributeValuePreferenceInfo> <defaultValue>From Creation to Publishing</defaultValue> </attributeValuePreferenceInfo> </attributeInfo> </attributeInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeInfoList> <attributeInfo id="519"> <name>Collections Sequence 1</name> <valueTypeId>1</valueTypeId></pre>		

```

<valueType>TEXT</valueType> <userDefined>true</userDefined>
<queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<created>2013-12-05T05:52:57Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-12-
05T05:52:57Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<displayable>true</displayable>
<limitedAccess>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo> <defaultValue>From Creation to
Publishing</defaultValue>
<textSearchEnabled>false</textSearchEnabled>
</attributeValuePreferenceInfo> </attributeInfo>
<attributeInfo id="520"> <name>Collections Sequence 2</name>
<valueTypeId>1</valueTypeId> <valueType>TEXT</valueType>
<userDefined>true</userDefined> <queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<created>2013-12-05T05:52:57Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-12-
05T05:52:57Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<displayable>true</displayable>
<limitedAccess>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo> <defaultValue>From Creation to
Publishing</defaultValue>
<textSearchEnabled>false</textSearchEnabled>
</attributeValuePreferenceInfo> </attributeInfo>
</attributeInfoList>

```

setAttributes

Function	Map ContentType with attributes.		
Description	This request is used to update the attribute mapping of content types and their children with the specified map of attributes.		
Type	POST		
URL Format	.../admin/contenttypes		
Java API	Method Name	Class Name	
	setAttributes	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	op=setattributes	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contentTypes= [comma-separated list of Content Types]	string	Content type IDs or Names.
	attributes= [comma-separated list of attribute names or	string	Attributes IDs or Names.

	IDs]		
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/contenttypes?op=setattributes&contenttypes=1000&attributes= 180,305:</code>		

getAttribute

Function	Get an attribute with a specific ID or Name.		
Description	This request returns an attribute with the specified ID or Name.		
Type	GET		
URL Format	<code>.../admin/attributes/{attributeIdORattributeName}</code>		
Java API	Method Name	Class Name	
	<code>getAttribute</code>	<code>com.quark.qpp.service.facade</code> <code>Class AdminFacade</code>	
Parameters	This request requires no input parameters.		
Returns	Returns an <code>attributeInfoList</code> object in XML format.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/attributes/collection</code> Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeInfoList> <attributeInfo id="55"> <name>Collection</name> <valueTypeId>7</valueTypeId> <valueType>DOMAIN</valueType> <userDefined>>false</userDefined> <queryable>>false</queryable> <modificationLevel>USER_MODIFIABLE</modificationLevel> <creator>1</creator> <creatorName>Admin</creatorName> <lastModifier>1</lastModifier> <lastModifierName>Admin</lastModifierName> <displayable>>true</displayable> <limitedAccess>>false</limitedAccess> <limitedAccessChangeable>>false</limitedAccessChangeable> <constraintsChangeable>>false</constraintsChangeable> <valuePreferencesChangeable>>false</valuePreferencesChangeable> <attributeValuePreferenceInfo> <domainId>19</domainId> <domainName>Collections</domainName> </attributeValuePreferenceInfo> </attributeInfo> </attributeInfoList></pre>		

getAllAttributes

Function	Get all attributes and attribute IDs.		
Description	This request returns all existing attributes with their metadata of QPP Server.		
Type	GET		
URL Format	<code>.../admin/attributes</code>		
Java API	Method Name	Class Name	

	<code>getAllAttributes</code>	<code>com.quark.qpp.service.facade</code> <code>Class AdminFacade</code>
Parameters	This request requires no input parameters.	
Returns	Returns an <code>attributeInfoList</code> object containing an array of attributes that exist in the QPP server.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/attributes <attributeInfoList> <attributeInfo id="180"> <name>App Studio Issue URN</name> </attributeInfo> <attributeInfo id="305"> <name>Article Component id</name> </attributeInfo> </attributeInfoList></pre> <p>Result:</p> <p>An <code>attributeInfoList</code> object containing an array of attributes that exist in the QPP server.</p>	

updateAttributes

Function	Update attributes.		
Description	This request is used to update the metadata of a single or multiple attributes.		
Type	POST		
URL Format	<code>.../admin/attributes</code>		
Java API	Method Name		Class Name
	<code>updateAttributes</code>		<code>com.quark.qpp.service.facade</code> <code>Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=update</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>attributeinfolist</code>	string	List of attributes to be updated.
Returns	Returns an <code>attributeInfoList</code> object containing a list of successfully updated attributes.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/attributes?op= update&attributevaluelist= <attributeInfoList> <attributeInfo id="507"> <name>Release Version</name> <valueTypeId>5</valueTypeId> <valueType>MEASUREMENT</valueType> <userDefined>true</userDefined> <queryable>true</queryable> <modificationLevel>USER_MODIFIABLE</modificationLevel> <displayable>true</displayable> <limitedAccess>false</limitedAccess> <limitedAccessChangeable>true</limitedAccessChangeable> <constraintsChangeable>true</constraintsChangeable> <valuePreferencesChangeable>true</valuePreferencesChangeable> <attributeValuePreferenceInfo> <defaultValue>4</defaultValue> <maximumPossibleValue>7</maximumPossibleValue> <minimumPossibleValue>2</minimumPossibleValue> <measurementUnit>10</measurementUnit> </attributeValuePreferenceInfo> </attributeInfo></pre>		

```
</attributeInfoList>
```

Result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<attributeInfoList> <attributeInfo id="507"> <name>Release
Version</name> <valueTypeId>5</valueTypeId>
<valueType>MEASUREMENT</valueType>
<userDefined>true</userDefined> <queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<created>2013-12-05T05:14:07Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-12-
05T10:13:29Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<displayable>true</displayable>
<limitedAccess>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo>
<defaultValue>4.0</defaultValue>
<maximumPossibleValue>7.0</maximumPossibleValue>
<minimumPossibleValue>2.0</minimumPossibleValue>
<measurementUnit>10</measurementUnit>
</attributeValuePreferenceInfo> </attributeInfo>
</attributeInfoList>
```

Example with multiple attributes to update:**URL:**

```
http://serverName:serverPort/rest/service/admin/attributes?op=
update&attributevaluelist= <attributeInfoList> <attributeInfo
id="502"> <name>Article Release 1</name>
<valueTypeId>2</valueTypeId> <valueType>DATE</valueType>
<userDefined>true</userDefined> <queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<displayable>true</displayable>
<limitedAccess>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo> <defaultValue>2013-12-
06</defaultValue>
<maximumPossibleValue>2013-12-31</maximumPossibleValue>
<minimumPossibleValue>2013-12-02</minimumPossibleValue>
</attributeValuePreferenceInfo> </attributeInfo>
<attributeInfo id="503"> <name>Article Release 2</name>
<valueTypeId>2</valueTypeId> <valueType>DATE</valueType>
<userDefined>true</userDefined> <queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<displayable>true</displayable>
<limitedAccess>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo> <defaultValue>2013-12-
06</defaultValue>
<maximumPossibleValue>2013-12-31</maximumPossibleValue>
<minimumPossibleValue>2013-12-02</minimumPossibleValue>
</attributeValuePreferenceInfo> </attributeInfo>
</attributeInfoList>
```

Result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<attributeInfoList> <attributeInfo id="502"> <name>Article
Release 1</name> <valueTypeId>2</valueTypeId>
```

```

<valueType>DATE</valueType> <userDefined>>true</userDefined>
<queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<created>2013-12-05T05:11:51Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-12-
05T10:17:38Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<displayable>true</displayable>
<limitedAccess>>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo> <defaultValue>2013-12-
06</defaultValue>
<maximumPossibleValue>2013-12-31</maximumPossibleValue>
<minimumPossibleValue>2013-12-02</minimumPossibleValue>
</attributeValuePreferenceInfo> </attributeInfo>
<attributeInfo id="503"> <name>Article Release 2</name>
<valueTypeId>2</valueTypeId> <valueType>DATE</valueType>
<userDefined>true</userDefined> <queryable>true</queryable>
<modificationLevel>USER_MODIFIABLE</modificationLevel>
<created>2013-12-05T05:11:51Z</created> <creator>1</creator>
<creatorName>Admin</creatorName> <lastModified>2013-12-
05T10:17:38Z</lastModified> <lastModifier>1</lastModifier>
<lastModifierName>Admin</lastModifierName>
<displayable>true</displayable>
<limitedAccess>>false</limitedAccess>
<limitedAccessChangeable>true</limitedAccessChangeable>
<constraintsChangeable>true</constraintsChangeable>
<valuePreferencesChangeable>true</valuePreferencesChangeable>
<attributeValuePreferenceInfo> <defaultValue>2013-12-
06</defaultValue>
<maximumPossibleValue>2013-12-31</maximumPossibleValue>
<minimumPossibleValue>2013-12-02</minimumPossibleValue>
</attributeValuePreferenceInfo> </attributeInfo>
</attributeInfoList>

```

deleteAttribute

Function	Delete an Attribute.		
Description	This request is used to delete an attribute with the specified id or name.		
Type	POST		
URL Format	.../admin/attributes/{attributeIdORattributeName}		
Java API	Method Name		Class Name
	deleteAttribute		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributeIdORName	long or string	The Name or ID of the attribute to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL:		

<code>http://serverName:serverPort/rest/service/admin/attributes/[attributeIdorName]?op=delete</code>

createForms

Function	Create forms.		
Description	This request is used to create a new form in the system.		
Type	POST		
URL Format	.../admin/forms		
Java API	Method Name		Class Name
	<code>createForms</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>forminfolist</code>	string	List of forms to be created.
Returns	Returns a <code>formInfoList</code> object containing a list of successfully created forms. A status of 200 is returned in response to a successful request execution.		
Examples	URL:		
	<pre> http://serverName:serverPort/rest/service/admin/forms?op=create&forminfolist= <formInfoList> <formInfo> <contentTypeName>Project</contentTypeName> <workflowName>Quark</workflowName> <formAttributeList> <formAttribute> <name>Collection Path</name> <xPos>0</xPos> <yPos>175</yPos> <totalWidth>450</totalWidth> <controlWidth>300</controlWidth> <labelWidth>150</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList> </pre>		
	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="32"> <contentTypeid>1102</contentTypeid> <contentTypeName>Project</contentTypeName> <workflowId>10</workflowId> <workflowName>Quark</workflowName> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="57"> <name>Collection Path</name> <xPos>0</xPos> <yPos>175</yPos> <totalWidth>450</totalWidth> <controlWidth>300</controlWidth> <labelWidth>150</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList> </pre>		

getAllForms

Function	Get all forms.
Description	This request is used to get all forms saved in the system.
Type	GET
URL Format	.../admin/forms

Java API	Method Name		Class Name	
	getAllForms		com.quark.qpp.service.facade Class AdminFacade	
Parameters	This request requires no input parameters			
Returns	Returns a <code>formInfoList</code> object containing a list of all the forms.			
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/admin/forms</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="58"> <contentTypeid>1413</contentTypeid> <contentTypeName>Project</contentTypeName> <workflowId>0</workflowId> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="57"> <name>Collection Path</name> <xPos>0</xPos> <yPos>0</yPos> <totalWidth>450</totalWidth> <controlWidth>300</controlWidth> <labelWidth>150</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList></pre>			

Function	Get forms with a specific workflow		
Description	This request returns all forms specific to given workflow.		
Type	GET		
URL Format	<code>.../admin/forms</code>		
Java API	Method Name		Class Name
	getAllForms		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	<code>workflow</code>	string	ID or name of the workflow.
Returns	Returns a <code>formInfoList</code> object containing a list of all the forms specific to a given workflow.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/admin/forms?workflow=Quark</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="53"> <contentTypeid>1100</contentTypeid> <contentTypeName>Asset</contentTypeName> <workflowId>11</workflowId> <workflowName>Quark</workflowName> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="34"> <name>Checkedout Application</name> <xPos>0</xPos> <yPos>0</yPos> <totalWidth>450</totalWidth> <controlWidth>314</controlWidth> <labelWidth>154</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList></pre>		

Function	Get forms with a specific content type.
-----------------	---

Description	This request is used to get all forms specific to a given content type.		
Type	GET		
URL Format	.../admin/forms		
Java API	Method Name		Class Name
	getAllForms		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	contenttype	string	ID or name of the workflow.
Returns	Returns a <code>formInfoList</code> object containing a list of all the forms specific to a given content type.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/forms?workflow=Quark&contenttype=Document</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="58"> <contentTypeid>1413</contentTypeid> <contentTypeName>Project</contentTypeName> <workflowid>0</workflowid> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="57"> <name>Collection Path</name> <xPos>0</xPos> <yPos>0</yPos> <totalWidth>450</totalWidth> <controlWidth>300</controlWidth> <labelWidth>150</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="54"> <contentTypeid>1101</contentTypeid> <contentTypeName>Document</contentTypeName> <workflowid>11</workflowid> <workflowName>Quark</workflowName> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="13"> <name>Is checked out</name> <xPos>0</xPos> <yPos>184.97</yPos> <totalWidth>457</totalWidth> <controlWidth>304.66666666666663</controlWidth> <labelWidth>154</labelWidth> <valueType>6</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList></pre>		

updateForms

Function	Update forms.		
Description	This request is used to updates metadata of a single or multiple forms at one time.		
Type	POST		
URL Format	.../admin/forms		
Java API	Method Name		Class Name
	updateForms		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify

			which operation is to be performed. (Mandatory)
	<code>forminfolist</code>	string	List of forms to be updated.
Returns	Returns a <code>formInfoList</code> object containing a list of all the forms specific to a given workflow.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/forms?op=update&forminfolist= <formInfoList> <formInfo> <contentTypeName>Document</contentTypeName> <workflowName>Quark</workflowName> <formAttributeList> <formAttribute id="13"> <name>Is checked out</name> <xPos>0</xPos> <yPos>184.97</yPos> <totalWidth>457</totalWidth> <controlWidth>304.66666666666663</controlWidth> <labelWidth>154</labelWidth> <valueType>6</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="11"> <contentTypeId>1101</contentTypeId> <contentTypeName>Document</contentTypeName> <workflowId>10</workflowId> <workflowName>Quark</workflowName> <defaultView>false</defaultView> <formAttributeList> <formAttribute id="13"> <name>Is checked out</name> <xPos>0</xPos> <yPos>184.97</yPos> <totalWidth>457</totalWidth> <controlWidth>304.66666666666663</controlWidth> <labelWidth>154</labelWidth> <valueType>6</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList></pre> <p>Example with multiple forms to update</p> <p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/forms?op=update&forminfolist= <formInfoList> <formInfo id="91"> <contentTypeId>10004</contentTypeId> <contentTypeName>Archive</contentTypeName> <workflowId>0</workflowId> <defaultView>false</defaultView> <formAttributeList> <formAttribute id="501"></pre>		

	<pre> <name>Master Text</name> <xPos>0</xPos> <yPos>263.5</yPos> <totalWidth>457</totalWidth> <controlWidth>304.66666666666663</controlWidth> <labelWidth>154</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> <formInfo id="92"> <contentTypeId>1403</contentTypeId> <contentTypeName>Book</contentTypeName> <workflowId>0</workflowId> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="3"> <name>Created</name> <xPos>0</xPos> <yPos>60.53</yPos> <totalWidth>457</totalWidth> <controlWidth>304.66666666666663</controlWidth> <labelWidth>154</labelWidth> <valueType>8</valueType> </formAttribute> </formAttributeList> </formInfo> </formInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <formInfoList> <formInfo id="91"> <contentTypeId>10004</contentTypeId> <contentTypeName>Archive</contentTypeName> <workflowId>0</workflowId> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="501"> <name>Master Text</name> <xPos>0</xPos> <yPos>263.5</yPos> <totalWidth>457</totalWidth> <controlWidth>304.66666666666663</controlWidth> <labelWidth>154</labelWidth> <valueType>1</valueType> </formAttribute> </formAttributeList> </formInfo> <formInfo id="92"> <contentTypeId>1403</contentTypeId> <contentTypeName>Book</contentTypeName> <workflowId>0</workflowId> <defaultView>>false</defaultView> <formAttributeList> <formAttribute id="3"> <name>Created</name> </pre>
--	---

deleteForm

Function	Delete a form.		
Description	This request is used to delete the specified form.		
Type	POST		
URL Format	<code>.../admin/forms</code>		
Java API	Method Name	Class Name	
	<code>deleteForm</code>	<code>com.quark.qpp.service.facade</code> Class <code>AdminFacade</code>	
Parameters	Name	Type	Description
	<code>op=delete</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/forms?op=delete&workflow=Quark&contentType=Document</code>		

createRoles

Function	Create roles.		
Description	This request is used to create one or more roles/userclass in a single request.		
Type	POST		
URL Format	<code>.../admin/roles</code>		
Java API	Method Name	Class Name	
	<code>createRoles</code>	<code>com.quark.qpp.service.facade</code> Class <code>AdminFacade</code>	
Parameters	Name	Type	Description
	<code>op=create</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>roleinfolist =<xml/></code>		Object containing a list of the roles to be created. See the <code>roleInfoList</code> parameter in <i>"Request/Response parameter formats"</i> .
Returns	Returns a <code>roleInfoList</code> object containing containing a list of the successfully created roles.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/roles?op=create&roleinfolist= <roleInfoList> <roleInfo id="11" name="Updated Name of Role"> <applicationPrivList> <privilege id="190000017"> <name>Remove Link with Collection Template</name> </privilege> <privilege id="999991"> <name>Live Preview</name> </privilege> </applicationPrivList> <contentTypePrivList> <contentTypePriv contentTypeId="1100"</code>		

getAllRoles

Function	Get all userclasses/roles.		
Description	This request returns all the userclasses with its metadata.		
Type	GET		
URL Format	<code>../admin/roles</code>		
Java API	Method Name	Class Name	
	<code>getAllRoles</code>	<code>com.quark.qpp.service.facade Class AdminFacade</code>	
Parameters	This request requires no input parameters.		
Returns	Returns a <code>roleInfoList</code> object containing an array of all roles that exist on the QPP Server.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/roles</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <roleInfoList> <roleInfo name="Administrator" id="1"/> <roleInfo name="UserClass2" id="10"/> </roleInfoList></pre>		

updateRoles

Function	Update roles.		
Description	This request is used to update the metadata of a specified single or multiple roles.		
Type	POST		
URL Format	<code>.../admin/roles</code>		
Java API	Method Name	Class Name	
	<code>updateRoles</code>	<code>com.quark.qpp.service.facade Class AdminFacade</code>	
Parameters	Name	Type	Description
	<code>op=update</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>roleinfolist=<xml/></code>		Object containing the metadata of the roles to be updated.
Returns	Returns a <code>roleInfoList</code> object containing a list of all the successfully updated roles.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/admin/roles?op=update&roleinfolist= <roleInfoList> <roleInfo id="11" name="Updated Name of Role"> <applicationPrivList> <privilege id="190000017"> <name>Remove Link with Collection Template</name> </privilege> <privilege id="999991"> <name>Live Preview</name> </privilege> </applicationPrivList> <contentTypePrivList> <contentTypePriv contentTypeId="1100" contentType="Asset"</pre>		

```

contentTypeHierarchy="System;Asset;"/> <privilege
id="10000015"> <name>Get Original Assets</name> </privilege>
..... </contentTypePriv> ..... </contentTypePrivList>
</roleInfo> </roleInfoList>
    
```

Result:

```

<roleInfoList> <roleInfo id="11" name="Updated Name of Role">
<applicationPrivList> <privilege id="190000017"> <name>Remove
Link with Collection Template</name> </privilege> <privilege
id="999991"> <name>Live Preview</name> </privilege> .....
</applicationPrivList> <contentTypePrivList> <contentTypePriv
contentTypeid="1100" contentType="Asset"
contentTypeHierarchy="System;Asset;"/> <privilege
id="10000015"> <name>Get Original Assets</name> </privilege>
..... </contentTypePriv> ..... </contentTypePrivList>
</roleInfo> </roleInfoList>
    
```

Example with multiple roles

URL:

```

http://serverName:serverPort/rest/service/admin/roles?op=updat
e&roleinfolist= <roleInfoList> <roleInfo id="11" name="Updated
Name of Role 1"> <applicationPrivList> <privilege
id="190000017"> <name>Remove Link with Collection
Template</name> </privilege> <privilege id="999991">
<name>Live Preview</name> </privilege> .....
</applicationPrivList> <contentTypePrivList> <contentTypePriv
contentTypeid="1100" contentType="Asset"
contentTypeHierarchy="System;Asset;"/> <privilege
id="10000015"> <name>Get Original Assets</name> </privilege>
..... </contentTypePriv> ..... </contentTypePrivList>
</roleInfo> <roleInfo id="12" name="Updated Name of Role 2">
<applicationPrivList> <privilege id="190000017"> <name>Remove
Link with Collection Template</name> </privilege> <privilege
id="999991"> <name>Live Preview</name> </privilege> .....
</applicationPrivList> <contentTypePrivList> <contentTypePriv
contentTypeid="1100" contentType="Asset"
contentTypeHierarchy="System;Asset;"/> <privilege
id="10000015"> <name>Get Original Assets</name> </privilege>
..... </contentTypePriv> ..... </contentTypePrivList>
</roleInfo> </roleInfoList>
    
```

Result:

```

<roleInfoList> <roleInfo id="11" name="Updated Name of Role
1"> <applicationPrivList> <privilege id="190000017">
<name>Remove Link with Collection Template</name> </privilege>
<privilege id="999991"> <name>Live Preview</name> </privilege>
..... </applicationPrivList> <contentTypePrivList>
<contentTypePriv contentTypeid="1100" contentType="Asset"
contentTypeHierarchy="System;Asset;"/> <privilege
id="10000015"> <name>Get Original Assets</name> </privilege>
..... </contentTypePriv> ..... </contentTypePrivList>
</roleInfo> <roleInfo id="12" name="Updated Name of Role 2">
<applicationPrivList> <privilege id="190000017"> <name>Remove
Link with Collection Template</name> </privilege> <privilege
id="999991"> <name>Live Preview</name> </privilege> .....
</applicationPrivList> <contentTypePrivList> <contentTypePriv
contentTypeid="1100" contentType="Asset"
contentTypeHierarchy="System;Asset;"/> <privilege
id="10000015"> <name>Get Original Assets</name> </privilege>
..... </contentTypePriv> ..... </contentTypePrivList>
</roleInfo> </roleInfoList>
    
```

deleteRole

Function	Delete a role.		
Description	This request is used to delete the specified role.		
Type	POST		
URL Format	.../admin/roles/{roleIdorName}		
Java API	Method Name		Class Name
	deleteRole		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	roleIdorName	string	Id or Name of the role to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <pre>http://serverName:serverPort/rest/service/admin/roles/23?op=delete</pre>		

createRelationType

Function	Create Relation Types.		
Description	This request is used to create a single or multiple Relation Types at one time.		
Type	POST		
URL Format	.../admin/relationtypes		
Java API	Method Name		Class Name
	createRelationTypes		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	relationtypeinfolist	string	List of relation types to create.
Returns	Returns a <code>relationTypeInfoList</code> object containing a list of successfully created relation types.		
Examples	URL: <pre>http://serverName:serverPort/rest/service/admin/relationtypes?op=create&relationtypeinfolist= <relationTypeInfoList> <relationTypeInfo> <name>Private Relations</name> <systemDefined>false</systemDefined> <checkoutMandatory>true</checkoutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="true" id="508"> Master DT </relationAttribute> <relationAttribute valueMandatory="false" userMapped="true" id="509"> Master List </relationAttribute></pre>		

	<pre> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <relationTypeInfoList> <relationTypeInfo id="1010"> <name>Private Relations</name> <systemDefined>>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="true" id="508"> Master DT </relationAttribute> <relationAttribute valueMandatory="false" userMapped="true" id="509"> Master List </relationAttribute> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList> </pre> <p>Example with multiple relation types</p> <p>URL:</p> <pre> http://serverName:serverPort/rest/service/admin/relationtypes? op=create&relationtypeinfoList= <relationTypeInfoList> <relationTypeInfo> <name>InDesign 1</name> <systemDefined>>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="false" id="410"> Item Name </relationAttribute> <relationAttribute valueMandatory="true" userMapped="false" id="419"> InDesign Object UID </relationAttribute> </relationAttributeList> </relationTypeInfo> <relationTypeInfo> <name>InDesign 2</name> <systemDefined>>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="false" id="410"> Item Name </relationAttribute> <relationAttribute valueMandatory="true" userMapped="false" id="419"> InDesign Object UID </relationAttribute> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <relationTypeInfoList> <relationTypeInfo id="1012"> <name>InDesign 1</name> <systemDefined>>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="true" id="410"> Item Name </relationAttribute> <relationAttribute valueMandatory="true" userMapped="true" id="419"> InDesign Object UID </relationAttributeList> </relationAttributeList> </relationTypeInfo> <relationTypeInfo id="1013"> <name>InDesign 2</name> <systemDefined>>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="true" id="410"> Item Name </relationAttribute> <relationAttribute valueMandatory="true" userMapped="true" id="419"> InDesign Object UID </relationAttributeList> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList> </pre>
--	--

getRelationType

Function	Get Relation Type.
-----------------	--------------------

Description	This request is used to get the metadata of a specified relation type.		
Type	GET		
URL Format	.../admin/relationtypes/{relationTypeIdOrName}		
Java API	Method Name		Class Name
	getRelationType		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	relationTypeIdOrName	string	The ID or name of the relation type that is to be retrieved.
Returns	Returns a <code>relationTypeInfoList</code> object containing the metadata of the specified relation type.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/relationtypes/1		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <relationTypeInfoList> <relationTypeInfo id="1"> <name>Primary Attachment</name> <systemDefined>true</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="false" id="416"> Article Component Reference Asset Minor version </relationAttribute> <relationAttribute valueMandatory="false" userMapped="false" id="415"> Article Component Reference Asset Major Version </relationAttribute> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList></pre>		

getAllRelationTypes

Function	Get all relationship types.		
Description	This request returns all the relation types with their attributes.		
Type	GET		
URL Format	.../admin/relationtypes		
Java API	Method Name		Class Name
	getAllRelationTypes		com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.		
Returns	Returns a <code>relationTypeInfoList</code> object containing an array of all relation types that exist on the QPP Server.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/relationtypes		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <relationTypeInfoList> <relationTypeInfo id="1"> <name>Primary Attachment</name> <systemDefined>true</systemDefined> <checkOutMandatory>true</checkOutMandatory> <relationAttributeList> <relationAttribute</pre>		


```

valueMandatory="false" userMapped="false" id="416"> Article
Component Reference Asset Minor version</relationAttribute>
<relationAttribute valueMandatory="false" userMapped="false"
id="414"> Article Component Reference Asset
Id</relationAttribute> ..... </relationAttributeList>
</relationTypeInfo> <relationTypeInfo id="2"> <name>Secondary
Attachment</name> <systemDefined>true</systemDefined>
<checkOutMandatory>true</checkOutMandatory>
<relationAttributeList> <relationAttribute
valueMandatory="false" userMapped="false" id="415"> Article
Component Reference Asset Major Version</relationAttribute>
<relationAttribute valueMandatory="false" userMapped="false"
id="416"> Article Component Reference Asset Minor
version</relationAttribute> <relationAttribute
valueMandatory="false" userMapped="false" id="413"> Article
Id</relationAttribute> ..... </relationAttributeList>
</relationTypeInfo> .... </relationTypeInfoList>
    
```

updateRelationTypes

Function	Update relation types.		
Description	This request is used to update the metadata of user defined relation types.		
Type	POST		
URL Format	.../admin/relationtypes		
Java API	Method Name		Class Name
	updateRelationTypes		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	relationtypeinfolist	string	List of relation types to be updated.
Returns	Returns a <code>relationTypeInfoList</code> object containing a list of successfully updated relation types.		
Examples	URL:		
	<pre> http://serverName:serverPort/rest/service/admin/relationtypes? op=update&relationtypeinfolist= <relationTypeInfoList> <relationTypeInfo id="1000"> <name>Custom Relations Updated</name> <systemDefined>>false</systemDefined> <checkOutMandatory>>false</checkOutMandatory> <relationAttributeList> <relationAttribute valueMandatory="false" userMapped="true" id="507"> Release Points </relationAttribute> <relationAttribute valueMandatory="false" userMapped="true" id="506"> Release Inches Decimal </relationAttribute> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList> </pre>		
	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <relationTypeInfoList> <relationTypeInfo id="1000"> <name>Custom Relations Updated</name> <systemDefined>>false</systemDefined> <checkOutMandatory>>false</checkOutMandatory> </pre>		

	<pre><relationAttributeList> <relationAttribute valueMandatory="false" userMapped="true" id="507"> Release Points </relationAttribute> <relationAttribute valueMandatory="false" userMapped="true" id="506"> Release Inches Decimal </relationAttribute> </relationAttributeList> </relationTypeInfo> </relationTypeInfoList></pre> <p>Example with multiple relation types to update</p> <p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/relationtypes? op=update&relationtypeinfolist= <relationTypeInfoList> <relationTypeInfo> <name>demo relation 1</name> <checkOutMandatory>true</checkOutMandatory> </relationTypeInfo> <relationTypeInfo> <name>demo relation 2</name> <checkOutMandatory>true</checkOutMandatory> </relationTypeInfo> </relationTypeInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <relationTypeInfoList> <relationTypeInfo id="1011"> <name>demo relation 1</name> <systemDefined>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> </relationTypeInfo> <relationTypeInfo id="1012"> <name>demo relation 2</name> <systemDefined>false</systemDefined> <checkOutMandatory>true</checkOutMandatory> </relationTypeInfo> </relationTypeInfoList></pre>
--	--

deleteRelationType

Function	Delete a Relation Type.		
Description	This request is used to delete a relation type with the specified id or name.		
Type	POST		
URL Format	.../admin/relationtypes/{relationTypeIdOrName}		
Java API	Method Name		Class Name
	deleteRelationType		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	relationTypeORName	long or string	List of forms to be updated.The Name or ID of the relation type to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/relationtypes/1000?op=delete</pre>		

createDomains

Function	Create Attribute Domains.
-----------------	---------------------------

Description	This request is used to create a single or multiple attribute domains at one time.		
Type	POST		
URL Format	.../admin/domains		
Java API	Method Name		Class Name
	createDomains		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributedomaininfolist	string	List of attribute domains to create.
Returns	Returns a <code>attributeDomainInfoList</code> object containing a list of successfully created attribute domains.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/admin/domains?op=create&attributedomaininfolist= <attributeDomainInfoList> <attributeDomainInfo> <name>Domain 1</name> <systemDefined>>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>true</itemsSorted> <expandable>true</expandable> <warnBeforeExpansion>>false</warnBeforeExpansion> <domainValueList> <nameId> <name>Value 1</name> </nameId> </domainValueList> </attributeDomainInfo> </attributeDomainInfoList></pre>		
	Result:		
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeDomainInfoList> <attributeDomainInfo id="504"> <name>Domain 1</name> <systemDefined>>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>true</itemsSorted> <expandable>true</expandable> <warnBeforeExpansion>>false</warnBeforeExpansion> <domainValueList> <nameId id="1"> <name>Value 1</name> </nameId> </domainValueList> </attributeDomainInfo> </attributeDomainInfoList></pre>			
Example with multiple domains			
URL:			
<pre>http://serverName:serverPort/rest/service/admin/domains?op=create&attributedomaininfolist= <attributeDomainInfoList> <attributeDomainInfo> <name>Domain 1</name> <systemDefined>>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>true</itemsSorted> <expandable>true</expandable> <warnBeforeExpansion>>false</warnBeforeExpansion> <domainValueList> <nameId> <name>Value 1</name> </nameId> </domainValueList> </attributeDomainInfo> <attributeDomainInfo> <name>Domain 2</name> <systemDefined>>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>true</itemsSorted> <expandable>true</expandable> <warnBeforeExpansion>true</warnBeforeExpansion> <domainValueList> <nameId> <name>Value 1</name> </nameId> </domainValueList> </attributeDomainInfo> </attributeDomainInfoList></pre>			

	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeDomainInfoList> <attributeDomainInfo id="502"> <name>Domain 1</name> <systemDefined>>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>true</itemsSorted> <expandable>true</expandable> <warnBeforeExpansion>>false</warnBeforeExpansion> <domainValueList> <nameId id="1"> <name>Value 1</name> </nameId> </domainValueList> </attributeDomainInfo> <attributeDomainInfo id="503"> <name>Domain 2</name> <systemDefined>>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>true</itemsSorted> <expandable>true</expandable> <warnBeforeExpansion>true</warnBeforeExpansion> <domainValueList> <nameId id="2"> <name>Value 1</name> </nameId> </domainValueList> </attributeDomainInfo> </attributeDomainInfoList></pre>
--	--

getDomain

Function	Get Attribute Domain.		
Description	This request is used to get the metadata of a specified attribute domain.		
Type	GET		
URL Formatadmin/domains/{domainIdOrName}		
Java API	Method Name		Class Name
	getDomain		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	domainTypeIdOrName	string	The ID or name of the domain that is to be retrieved.
Returns	Returns an <code>attributeDomainInfoList</code> object containing the metadata of the specified domain.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/domains/3		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeDomainInfoList> <attributeDomainInfo id="3"> <name>Users</name> <systemDefined>true</systemDefined> <userModifiable>>false</userModifiable> <itemsSorted>true</itemsSorted> <expandable>>false</expandable> <warnBeforeExpansion>>false</warnBeforeExpansion> <domainValueList> <nameId id="1"> <name>Admin</name> </nameId> <valueHierarchy>Desktop</valueHierarchy> </domainValueList> </attributeDomainInfo> </attributeDomainInfoList></pre>		

getAllDomains

Function	Get all attribute domains.
Description	This request is used to get the metadata of all the domains that exist in the system.

Type	GET	
URL Format	.../admin/domains	
Java API	Method Name	Class Name
	getAllDomains	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns an <code>attributeDomainInfoList</code> object containing the metadata of the all the attribute domains.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/domains</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <attributeDomainInfoList> <attributeDomainInfo id="3"> <name>Users</name> <systemDefined>true</systemDefined> <userModifiable>false</userModifiable> <itemsSorted>true</itemsSorted> <expandable>false</expandable> <warnBeforeExpansion>false</warnBeforeExpansion> <domainValueList> <nameId id="1"> <name>Admin</name> </nameId> <valueHierarchy>Desktop</valueHierarchy> </domainValueList> </attributeDomainInfo> </attributeDomainInfoList></pre>	

updateDomains

Function	Update attribute domains.		
Description	This request is used to update the metadata of a single or multiple attribute domains.		
Type	POST		
URL Format	.../admin/domains		
Java API	Method Name	Class Name	
	updateDomains	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributedomaininfolist	string	List of domains to be updated.
Returns	Returns an <code>attributeDomainInfoList</code> object containing a list of successfully updated domains.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/domains?op=update&attributedomaininfolist= <attributeDomainInfoList> <attributeDomainInfo id="503"> <name>Domain 1</name> <systemDefined>false</systemDefined> <userModifiable>true</userModifiable> <itemsSorted>false</itemsSorted></pre>		

```

<expandable>>false</expandable>
<warnBeforeExpansion>>false</warnBeforeExpansion>
<domainValueList> <nameId id="1"> <name>Value 1</name>
</nameId> </domainValueList> </attributeDomainInfo>
</attributeDomainInfoList>

```

Result:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<attributeDomainInfoList> <attributeDomainInfo id="503">
<name>Domain 1</name> <systemDefined>>false</systemDefined>
<userModifiable>>true</userModifiable>
<itemsSorted>>false</itemsSorted>
<expandable>>false</expandable>
<warnBeforeExpansion>>false</warnBeforeExpansion>
<domainValueList> <nameId id="1"> <name>Value 1</name>
</nameId> </domainValueList> </attributeDomainInfo>
</attributeDomainInfoList>

```

Example with multiple attribute domains to update**URL:**

```

http://serverName:serverPort/rest/service/admin/domains?op=update&attributedomaininfolist= <attributeDomainInfoList>
<attributeDomainInfo id="502"> <name>Domain 1</name>
<systemDefined>>false</systemDefined>
<userModifiable>>true</userModifiable>
<itemsSorted>>true</itemsSorted> <expandable>>true</expandable>
<warnBeforeExpansion>>false</warnBeforeExpansion>
<domainValueList> <nameId id="1"> <name>Value 1</name>
</nameId> </domainValueList> </attributeDomainInfo>
<attributeDomainInfo id="503"> <name>Domain 2</name>
<systemDefined>>false</systemDefined>
<userModifiable>>true</userModifiable>
<itemsSorted>>true</itemsSorted> <expandable>>true</expandable>
<warnBeforeExpansion>>true</warnBeforeExpansion>
<domainValueList> <nameId id="2"> <name>Value 1</name>
</nameId> </domainValueList> </attributeDomainInfo>
</attributeDomainInfoList>

```

Result:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<attributeDomainInfoList> <attributeDomainInfo id="502">
<name>Domain 1</name> <systemDefined>>false</systemDefined>
<userModifiable>>true</userModifiable>
<itemsSorted>>true</itemsSorted> <expandable>>true</expandable>
<warnBeforeExpansion>>false</warnBeforeExpansion>
<domainValueList> <nameId id="1"> <name>Value 1</name>
</nameId> </domainValueList> </attributeDomainInfo>
<attributeDomainInfo id="503"> <name>Domain 2</name>
<systemDefined>>false</systemDefined>
<userModifiable>>true</userModifiable>
<itemsSorted>>true</itemsSorted> <expandable>>true</expandable>
<warnBeforeExpansion>>true</warnBeforeExpansion>
<domainValueList> <nameId id="2"> <name>Value 1</name>
</nameId> </domainValueList> </attributeDomainInfo>
</attributeDomainInfoList>

```

deleteDomain

Function	Delete an Attribute Domain.
Description	This request is used to delete an attribute domain with the specified id or name.

Type	POST		
URL Format	.../admin/domains/{domainIdorName}		
Java API	Method Name		Class Name
	deleteDomain		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	domainIdORName	long or string	The Name or ID of the domain to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/admin/domains/503?op=delete		

createLdapProfiles

Function	Create LDAP profiles.		
Description	This request is used to create one or more LDAP profiles.		
Type	POST		
URL Format	.../admin/ldaprofiles		
Java API	Method Name		Class Name
	createLdapProfiles		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	ldaprofileinfolist =<xml/>		Object containing a list of the LDAP profiles to be created.
Returns	Returns a <code>LdapProfileInfoList</code> object containing containing the successfully created LDAP profiles.		
Examples	URL: http://serverName:serverPort/rest/service/admin/ldaprofiles?op=create&ldaprofileinfolist= <ldapProfileInfoList> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ldapProfileInfoList> <ldapProfileInfo> <name>QPP LDAP Profile</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentificationCriteria> <userIdentificationCriteria> ... </userIdentificationCriteria>		

```

<groupIdentificationCriteria> ...
</groupIdentificationCriteria>
<groupMemberAttrName>member</groupMemberAttrName>
<membershipIdentifyingAttrName>memberof</membershipIdentifying
AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup>
... </filterToGetAllUsersOfGroup>
<filterToGetAllSubGroupsOfGroup> ...
</filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser>
... </filterToGetAllGroupsOfUser>
</ldapObjectsIdentificationCriteria> <ldapAttributeInfoList>
<ldapAttributeInfo id="5"> <name>USER_EMAIL</name>
<value>mail</value> </ ldapAttributeInfo > .....
</ldapAttributeInfoList> </ldapProfileInfo>
</ldapProfileInfoList> </ldapProfileInfoList>

```

Result:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ldapProfileInfoList> <ldapProfileInfo id="10"> <name>QPP LDAP
Profile</name> <realm>INDIA.QUARK.COM</realm>
<userName>qppadmin@quark.com</userName>
<authenticationType>KERBEROS</authenticationType>
<ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme>
<serverName>india-dc1.india.quark.com</serverName>
<port>389</port> </ldapServerInfo> </ldapServerInfoList>
<ldapObjectsIdentificationCriteria>
<userIdentificationCriteria> ... </userIdentificationCriteria>
<groupIdentificationCriteria> ...
</groupIdentificationCriteria>
<groupMemberAttrName>member</groupMemberAttrName>
<membershipIdentifyingAttrName>memberof</membershipIdentifying
AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup>
... </filterToGetAllUsersOfGroup>
<filterToGetAllSubGroupsOfGroup> ...
</filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser>
... </filterToGetAllGroupsOfUser>
</ldapObjectsIdentificationCriteria> <ldapAttributeInfoList>
<ldapAttributeInfo id="5"> <name>USER_EMAIL</name>
<value>mail</value> </ ldapAttributeInfo > .....
</ldapAttributeInfoList> </ldapProfileInfo>
</ldapProfileInfoList>

```

Example with multiple LDAP profiles**URL:**

```

http://serverName:serverPort/rest/service/admin/ldapprofiles?p=
create&ldapprofileinfolist= <ldapProfileInfoList> <?xml
version="1.0" encoding="UTF-8" standalone="yes"?>
<ldapProfileInfoList> <ldapProfileInfo> <name>QPP LDAP Profile
India</name> <realm>INDIA.QUARK.COM</realm>
<userName>qppadmin@quark.com</userName>
<authenticationType>KERBEROS</authenticationType>
<ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme>
<serverName>india-dc1.india.quark.com</serverName>
<port>389</port> </ldapServerInfo> </ldapServerInfoList>
<ldapObjectsIdentificationCriteria>
<userIdentificationCriteria> ... </userIdentificationCriteria>
<groupIdentificationCriteria> ...
</groupIdentificationCriteria>
<groupMemberAttrName>member</groupMemberAttrName>
<membershipIdentifyingAttrName>memberof</membershipIdentifying
AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup>
... </filterToGetAllUsersOfGroup>
<filterToGetAllSubGroupsOfGroup> ...
</filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser>
... </filterToGetAllGroupsOfUser>

```


	<pre> </ldapObjectsIdentificationCriteria> <ldapAttributeInfoList> <ldapAttributeInfo id="5"> <name>USER_EMAIL</name> <value>mail</value> </ ldapAttributeInfo > </ldapAttributeInfoList> </ldapProfileInfo> <ldapProfileInfo> <name>QPP LDAP Profile US</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentificationCriteria> <userIdentificationCriteria> ... </userIdentificationCriteria> <groupIdentificationCriteria> ... </groupIdentificationCriteria> <groupMemberAttrName>member</groupMemberAttrName> <membershipIdentifyingAttrName>memberof</membershipIdentifying AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup> ... </filterToGetAllUsersOfGroup> <filterToGetAllSubGroupsOfGroup> ... </filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser> ... </filterToGetAllGroupsOfUser> </ldapObjectsIdentificationCriteria> <ldapAttributeInfoList> <ldapAttributeInfo id="5"> <name>USER_EMAIL</name> <value>mail</value> </ ldapAttributeInfo > </ldapAttributeInfoList> </ldapProfileInfo> </ldapProfileInfoList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ldapProfileInfoList> <ldapProfileInfo id="10"> <name>QPP LDAP Profile India</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentificationCriteria> <userIdentificationCriteria> ... </userIdentificationCriteria> <groupIdentificationCriteria> ... </groupIdentificationCriteria> <groupMemberAttrName>member</groupMemberAttrName> <membershipIdentifyingAttrName>memberof</membershipIdentifying AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup> ... </filterToGetAllUsersOfGroup> <filterToGetAllSubGroupsOfGroup> ... </filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser> ... </filterToGetAllGroupsOfUser> </ldapObjectsIdentificationCriteria> <ldapAttributeInfoList> <ldapAttributeInfo id="5"> <name>USER_EMAIL</name> <value>mail</value> </ ldapAttributeInfo > </ldapAttributeInfoList> </ldapProfileInfo><ldapProfileInfo id="11"> <name>QPP LDAP Profile US</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentificationCriteria> <userIdentificationCriteria> ... </userIdentificationCriteria> <groupIdentificationCriteria> ... </groupIdentificationCriteria> <groupMemberAttrName>member</groupMemberAttrName> <membershipIdentifyingAttrName>memberof</membershipIdentifying </pre>
--	---

	<pre> AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup> ... </filterToGetAllUsersOfGroup> <filterToGetAllSubGroupsOfGroup> ... </filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser> ... </filterToGetAllGroupsOfUser> </ldapObjectsIdentificationCriteria> <ldapAttributeInfoList> <ldapAttributeInfo id="5"> <name>USER_EMAIL</name> <value>mail</value> </ ldapAttributeInfo > </ldapAttributeInfoList> </ldapProfileInfo> </ldapProfileInfoList> </pre>
--	---

getLdapProfile

Function	Get an LDAP profile.		
Description	This request returns the metadata of a specified LDAP profile.		
Type	Get		
URL Format	.../admin/ldaprofiles/{ldapProfileIdOrName}		
Java API	Method Name		Class Name
	getLdapProfile		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	ldapProfileIdOrName	string	The ID or Name of the LDAP profile to be retrieved.
Returns	Returns a <code>ldapProfileInfoList</code> object containing information on the specified LDAP profile.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/ldaprofiles/10		
Examples	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ldapProfileInfoList> <ldapProfileInfo id="10"> <name>QPP LDAP Profile</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentificationCriteria> <userIdentificationCriteria> ... </userIdentificationCriteria> <groupIdentificationCriteria> ... </groupIdentificationCriteria> <groupMemberAttrName>member</groupMemberAttrName> <membershipIdentifyingAttrName>memberof</membershipIdentifying AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup> ... </filterToGetAllUsersOfGroup> <filterToGetAllSubGroupsOfGroup> ... </filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser> ... </filterToGetAllGroupsOfUser> </ldapObjectsIdentificationCriteria> <ldapAttributeInfoList> <ldapAttributeInfo id="5"> <name>USER_EMAIL</name> <value>mail</value> </ ldapAttributeInfo > </ldapAttributeInfoList> </ldapProfileInfo> </ldapProfileInfoList> </pre>		

getAllLdapProfiles

Function	Get all LDAP profiles.	
Description	This request returns all the LDAP profiles saved on the system.	
Type	GET	
URL Format	.../admin/ldaprofiles	
Java API	Method Name	Class Name
	getAllLdapProfiles	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters	
Returns	Returns a ldapProfileInfoList object containing a list of all the LDAP profiles that exist on the QPP Server.	
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/admin/workflows</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ldapProfileInfoList> <ldapProfileInfo id="10"> <name>QPP LDAP Profile</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> <ldapObjectsIdentificationCriteria> <userIdentificationCriteria> ... </userIdentificationCriteria> <groupIdentificationCriteria> ... </groupIdentificationCriteria> <groupMemberAttrName>member</groupMemberAttrName> <membershipIdentifyingAttrName>memberof</membershipIdentifying AttrName> <baseDn>BDN</baseDn> <filterToGetAllUsersOfGroup> ... </filterToGetAllUsersOfGroup> <filterToGetAllSubGroupsOfGroup> ... </filterToGetAllSubGroupsOfGroup> <filterToGetAllGroupsOfUser> ... </filterToGetAllGroupsOfUser> </ldapObjectsIdentificationCriteria> <ldapAttributeInfoList> <ldapAttributeInfo id="5"> <name>USER_EMAIL</name> <value>mail</value> </ ldapAttributeInfo > </ldapAttributeInfoList> </ldapProfileInfo> </ldapProfileInfoList></pre>	

updateLdapProfiles

Function	Update LDAP profiles.	
Description	This request is used to update the metadata of a specified single or multiple LDAP profiles.	
Type	POST	
URL Format	.../admin/ldaprofiles	
Java API	Method Name	Class Name
	updateLdapProfiles	com.quark.qpp.service.facade Class AdminFacade

	Name	Type	Description
Parameters	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	ldaprofileinfolist=<xml/>		Object containing a list of the successfully updated LDAP profiles.
Returns	Returns a <code>LdapProfileInfoList</code> object containing a list of all the successfully updated LDAP profiles.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/workflows?op=update&workflowinfolist= <workflowInfoList> <workflowInfo id="22"> <name>DefaultWorkflow New</name> </workflowInfo> </workflowInfoList></pre> <p>Result:</p> <pre>http://serverName:serverPort/rest/service/admin/ldaprofiles?p=update&attributevaluelist= <ldapProfileInfoList> <?xml version="1.0" encoding="UTF-8" standalone="yes"?><ldapProfileInfoList> <ldapProfileInfo> <name>QPP LDAP Profile India</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> </ldapProfileInfo><ldapProfileInfo> <name>QPP LDAP Profile US</name> <realm>INDIA.QUARK.COM</realm> <userName>qppadmin@quark.com</userName> <authenticationType>KERBEROS</authenticationType> <ldapServerInfoList> <ldapServerInfo> <scheme>LDAP</scheme> <serverName>india-dc1.india.quark.com</serverName> <port>389</port> </ldapServerInfo> </ldapServerInfoList> </ldapProfileInfo> </ldapProfileInfoList></pre>		

deleteLdapProfile

Function	Delete an LDAP profile.		
Description	This request is used to delete the specified LDAP profile from the system. There shouldn't be any LDAP user in the system associated with this profile while executing this request.		
Type	POST		
URL Format	.../admin/ldaprofiles/{profileId}		
Java API	Method Name		Class Name
	deleteLdapProfile		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	ldapProfileIdOrName	string	Id or Name of the LDAP profile to be deleted.

Returns	A status of 200 is returned in response to a successful request execution.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/ldapprofiles/10?p=delete</pre>

getLdapUsers

Function	Get Ldap users.		
Description	This request returns LDAP users using a saved ldap query id/name or a combination of valid ldapprofile id/name, base string & search filter.		
Type	GET		
URL Format	../admin/ldapusers		
Java API	Method Name		Class Name
	getLdapUsers		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	savedldapquery	string	Specify the name of the LDAP Query. (Mandatory)
	ldapProfile	string	Specify the name of the LDAP profile. (Mandatory)
	baseString	string	Specify the base string to fetch LDAP users.(Mandatory)
	searchFilter	string	Specify the filter to filter the users. (Mandatory)
	defaultUserClass	string	The Quark Publishing Platform Server's UserRole. After fetching users from LDAP Server users will be created with this UserRole. (Mandatory)
	pagesize	string	
Returns	Returns a <code>userInfoList</code> object containing the metadata of the specified users.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/users?savedldapquery=LDAP Search&ldapProfile=QPP LDAP Profile&baseString=ou=users,ou=qmhi,dc=india,dc=quark,dc=com&searchFilter=sn=abc&defaultUserClass=UserClass2</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="52"> <name>abc</name> <domainId>0</domainId> <user>true</user> <deleted>false</deleted> <overriddenUserId>0</overriddenUserId> </userInfo> <userInfo id="53"> <name>abc</name> <domainId>0</domainId> </userInfo> </userInfoList></pre>		

importUsersFromLdap

Function	Import users from LDAP using LDAP profile.		
Description	This request is used to fetch users metadata from LDAP Server and push it to the Quark Publishing Platform Server.		
Type	POST		
URL Format	.../admin/users		
Java API	Method Name	Class Name	
	importUsersFromLdap	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	op=importusers	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	ldapProfile		Specify the name of the LDAP profile. (Mandatory)
	baseString		Specify the base string to fetch LDAP users. (Mandatory)
	searchFilter		Specify the filter to filter the users. (Mandatory)
	defaultUserClass		The Quark Publishing Platform Server's UserRole. After fetching users from LDAP Server users will be created with this UserRole. (Mandatory)
Returns	Returns a <code>userInfoList</code> object containing metadata of newly fetched\created users in the form of an array.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/users?op=importusers&ldapProfile=QPP LDAP Profile&baseString=ou=users,ou=qmhi,dc=india,dc=quark,dc=com&searchFilter=sn=abc&defaultUserClass=UserClass2</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="52"> <name>abc</name> <domainId>0</domainId> <user>true</user> <deleted>false</deleted> <overriddenUserId>0</overriddenUserId> </userInfo> <userInfo id="53"> <name>abc</name> <domainId>0</domainId> </userInfo> </userInfoList></pre>		

Function	Import Users from LDAP using Saved Query		
Description	This request is used to fetch users metadata from the LDAP Server and push it to the Quark Publishing Platform Server.		
Type	POST		
URL Format	.../admin/users		
Java API	Method Name	Class Name	
	importUsersFromLdap	com.quark.qpp.service.facade	

			Class AdminFacade
Parameters	Name	Type	Description
	op=importusers	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	savedLdapQuery		Specify the name of the saved LDAP Users search. (Mandatory)
Returns	Returns a <code>userInfoList</code> object containing metadata of newly fetched\created users in the form of an array.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/users?op=importusers &savedLdapQuery=SavedSearch</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <userInfoList> <userInfo id="52"> <name>abc</name> <domainId>0</domainId> <user>true</user> <deleted>false</deleted> <overriddenUserId>0</overriddenUserId> </userInfo> <userInfo id="53"> <name>abc</name> <domainId>0</domainId> </userInfo> </userInfoList></pre>		

getLdapGroups

Function	Get LDAP groups.		
Description	This request returns LDAP groups using a saved LDAP query or the combination of basestring, ldaprofile and searchfilter.		
Type	GET		
URL Format	../admin/ldapgroups		
Java API	Method Name	Class Name	
	getLdapGroups	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	savedldapquery=[savedLdapQueryId] or [Name]	long	
	ldaprofile=[ldapProfileId] or [Name]	long	
	basestring=[dc=xyz,dc=abc,dc=com]	long	
	searchfilter	string	
Returns	Returns a list of LDAP groups on the basis of a saved LDAP query or the combination of LDAP profile, base string and searchfilter.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/ldapgroups?</pre>		

	<pre>savedldapquery=MySearch</pre> <p>Result:</p> <pre><groupInfoList> <groupInfo id="0"> <name>QPP Server</name> <emailAddress>QPP Server</emailAddress> <nativeTrustee>>false</nativeTrustee> <defaultUserId>0</defaultUserId> <ldapProfileId>764</ldapProfileId> <ldapProfile>QPP LDAP Profile</ldapProfile> <ldapDistinguishedName> CN=QPP Server, OU=DistributionLists, OU=QMHI, DC=india, DC=quark, DC=com </ldapDistinguishedName> <memberList/> </groupInfo> </groupInfoList></pre>
--	---

importGroupsFromLdap

Function	Import LDAP groups.		
Description	This request imports groups from the LDAP server along with all of its subgroups and users into Platform Server. An empty string will be the default password for all users imported.		
Type	GET		
URL Format	../admin/groups		
Java API	Method Name	Class Name	
	importGroupsFromLdap	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	savedldapquery=[savedLdapQueryId] or [Name]	long	
	ldaprofile=[ldapProfileId] or [Name]	long	
	basestring=[dc=xyz,dc=abc,dc=com]	long	
	searchfilter	string	
Returns	Returns an <code>importedGroupInfoList</code> object containing a list of LDAP groups on the basis of a saved LDAP query or the combination of LDAP profile, base string and searchfilter.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/ldapgroups? op=importgroups&savedldapquery=MyQuery</pre> <p>Result:</p> <pre><importedGroupInfoList> <importedGroupInfo> <groupInfo id="3681"> <name>QPP Server</name> <emailAddress>QPP Server</emailAddress> <nativeTrustee>>false</nativeTrustee> </groupInfo> <subGroupsInfoList> <groupInfo id="3682"> <name>QPP Server R&D</name> </groupInfo> </subGroupsInfoList> <userInfoList> <userInfo id="3687"> <name>kshankarraman</name></pre>		


```

..... . </userInfo> <userInfo id="3688"> <name>Menon</name>
..... </userInfo> </userInfoList> </importedGroupInfo>
</importedGroupInfoList>
    
```

synchronizeLdapUsersGroups

Function	Synchronize all LDAP users and groups.		
Description	This request synchronizes all users and groups that were imported from LDAP Server to Platform Server.		
Type	GET		
URL Format	../admin/ldaprofiles		
Java API	Method Name		Class Name
	synchronizeLdapUsersGroups		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=synchronize	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/admin/ldaprofiles?op=synchronize		

synchronizeLdapUsersGroupsBasedOnProfile

Function	Synchronize all LDAP users and groups.		
Description	This request synchronizes all users and groups that were imported from LDAP Server to Platform Server using the given LDAP profile..		
Type	GET		
URL Format	../admin/ldaprofiles/{ldaprofileIdOrName}		
Java API	Method Name		Class Name
	synchronizeLdapUsersGroupsBasedOnProfile		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=synchronize	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/admin/ldaprofiles/Quark India? op=synchronize		

createLdapSearches

Function	Create LDAP search definitions.		
Description	This request creates creates LDAP search definitions.		
Type	POST		
URL Format	<code>../admin/ldapsearchdefinitions/</code>		
Java API	Method Name		Class Name
	<code>createLdapSearches</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
	<code>ldapsearchinfolist=<x ml/></code>		See "Request/Response parameter formats."
Returns	Returns a <code>ldapsearchinfolist</code> object in XML format that contains the metadata of all successfully created LDAP search definitions.		
Examples	<p>URL:</p> <pre>https://serverName:serverPort/rest/service/admin/ ldapsearchdefinitions?op=create&ldapsearchinfolist= <ldapSearchInfoList> <ldapSearchInfo> <name>Demo groups LDAP search</name> <ldapSearchCriteria> <ldapProfile>Quark India</ldapProfile> <searchFilter>cn=qpp cli*</searchFilter> <baseString>dc=india,dc=quark,dc=com</baseString> <searchType>GROUPS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> </ldapSearchInfoList></pre> <p>Result:</p> <pre><ldapSearchInfoList> <ldapSearchInfo id="123"> <name>Demo groups LDAP search</name> <ldapSearchCriteria> <ldapProfile id="18">Quark India</ldapProfile> <searchFilter>cn=qpp cli*</searchFilter> <baseString>dc=india,dc=quark,dc=com</baseString> <searchType>GROUPS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> </ldapSearchInfoList></pre>		

getLdapSearch

Function	Get LDAP search definition.		
Description	This request returns the LDAP search definition for the given search id or name.		
Type	GET		
URL Format	<code>../admin/ldapsearchdefinitions{searchIdOrName}</code>		
Java API	Method Name		Class Name
	<code>getLdapSearch</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	This request requires no input parameters.		
Returns	Returns a <code>ldapSearchInfoList</code> object containing the LDAP search definition for		

	the given search id or name.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/ldapsearchdefinitions/MySearch</pre> <p>Result:</p> <pre><ldapSearchInfoList> <ldapSearchInfo id="10"> <name>MySearch</name> <ldapSearchCriteria> <ldapProfile id="20">shankar</ldapProfile> <searchFilter>sn=*gupta</searchFilter> <baseString>dc=quark,dc=com</baseString> <searchType id="1">USERS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> </ldapSearchInfoList></pre>

getAllLdapSearches

Function	Get all saved LDAP search definitions.	
Description	This request returns all saved LDAP search definitions.	
Type	GET	
URL Format	../admin/ldapsearchdefinitions	
Java API	Method Name	Class Name
	getAllLdapSearches	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns a <code>ldapSearchInfoList</code> object containing a list of <code>ldapSearchInfo</code> objects containing the LDAP search definition.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/ldapsearchdefinitions</pre> <p>Result:</p> <pre><ldapSearchInfoList> <ldapSearchInfo id="438"> <name>LDAP User Search</name> <ldapSearchCriteria> <ldapProfile id="764">QPP LDAP Profile</ldapProfile> <searchFilter>sn=kumar</searchFilter> <baseString>ou=users,ou=qmhi,dc=india,dc=quark,dc=com</baseStr ing> <searchType id="1">USERS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> <ldapSearchInfo id="439"> <name>LDAP Group Search</name> <ldapSearchCriteria> <ldapProfile id="764">QPP LDAP Profile</ldapProfile> <searchFilter>cn=qppserver</searchFilter> <baseString>dc=india,dc=quark,dc=com</baseString> <searchType id="2">GROUPS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> </ldapSearchInfoList></pre>	

updateLdapSearches

Function	Update LDAP search definitions.
Description	This request update the given LDAP search definitions.

Type	POST		
URL Format	../admin/ldapsearchdefinitions/		
Java API	Method Name		Class Name
	updateLdapSearches		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
	ldapsearchinfolist=<x ml/>		See "Request/Response parameter formats."
Returns	Returns a <code>ldapsearchinfolist</code> object in XML format that contains the metadata of all successfully updated LDAP search definitions.		
Examples	<p>URL:</p> <pre>https://serverName:serverPort/rest/service/admin/ ldapsearchdefinitions?op=update&ldapsearchinfolist= <ldapSearchInfoList> <ldapSearchInfo> <name>Demo groups LDAP search</name> <ldapSearchCriteria> <ldapProfile>Quark India</ldapProfile> <searchFilter>cn=qpp cli*</searchFilter> <baseString>dc=india,dc=quark,dc=com</baseString> <searchType>GROUPS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> </ldapSearchInfoList></pre> <p>Result:</p> <pre><ldapSearchInfoList> <ldapSearchInfo id="123"> <name>Demo groups LDAP search</name> <ldapSearchCriteria> <ldapProfile id="10">Quark India</ldapProfile> <searchFilter>cn=qpp cli*</searchFilter> <baseString>dc=india,dc=quark,dc=com</baseString> <searchType id="2">GROUPS_SEARCH</searchType> </ldapSearchCriteria> </ldapSearchInfo> </ldapSearchInfoList></pre>		

deleteLdapSearch

Function	Delete an LDAP search definition.		
Description	This request deletes the given LDAP search definition with the given ID or name.		
Type	POST		
URL Format	../admin/ldapsearchdefinitions/{searchIdOrName}		
Java API	Method Name		Class Name
	deleteLdapSearch		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
Returns	A status of 200 is returned in response to a successful request execution.		

Examples	URL:
	<code>http://serverName:serverPort/rest/service/admin/ldapsearchdefinitions/MySearch?op=delete</code>

getTrustee

Function	Get a trustee (a user or group) with a particular name.		
Description	This request returns all the metadata of a specified trustee.		
Type	GET		
URL Format	<code>.../admin/trustees/[trustee name]</code>		
Java API	Method Name	Class Name	
	<code>getTrustee</code>	<code>com.quark.qpp.service.facade Class AdminFacade</code>	
Parameters	This request requires no input parameters.		
Returns	Returns a <code>trusteeInfo</code> object in XML format of specified user or group.		
Examples	URL:	<code>http://serverName:serverPort/rest/service/admin/trustees/Admin</code>	
	Result:	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <trusteeInfo id="1"> <user>true</user> <deleted>false</deleted> <name>Admin</name> <emailAddress>Editors@company.com</emailAddress> <nativeTrustee>false</nativeTrustee> <defaultUserId>2</defaultUserId> <ldapProfileId>10</ldapProfileId> <ldapDistinguishedName>Dist Name</ldapDistinguishedName> <domainId>0</domainId> </trusteeInfo></pre>	

updateTrusteeRule

Function	Update role for the given trustee.		
Description	This request sets the default role for the given trustee (User or Group).		
Type	POST		
URL Format	<code>.../admin/trustees/{trusteeIdOrName}</code>		
Java API	Method Name	Class Name	
	<code>updateTrusteeRole</code>	<code>com.quark.qpp.service.facade Class AdminFacade</code>	
Parameters	Name	Type	Description
	<code>op=setrole</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This operation will run only for a valid session id.
	<code>role=[RoleId] or [Name]</code>	string	

	<code>overrideuserclass=[True] or [False]</code>	boolean	This flag is applicable only if the trustee Id provided is that of a group. If true, it will override the userclass of all users belonging to the given group.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/trustees/John?op=setole&role=Publishers</code>		

createWorkflows

Function	Create workflows.		
Description	This request is used to create one or more workflows and their statuses.		
Type	POST		
URL Format	<code>.../admin/workflows</code>		
Java API	Method Name		Class Name
	<code>createWorkFlows</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>workflowinfolist</code> = <code><xml/></code>		Object containing a list of the workflows to be created. See the workflowInfoList parameter in "Request/Response parameter formats".
Returns	Returns a <code>workflowInfoList</code> object containing containing the successfully created workflows.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/workflows?op=create&workflowinfolist= <workflowInfoList> <workflowInfo> <name>Publsiing Workflow</name> <statusInfoList> <statusInfo> <name>New Status 001</name> <colorBlue>255</colorBlue> <position>1</position> <redlineTracking>>false</redlineTracking> <checkInLayoutEvaluation>>false</checkInLayoutEvaluation> <outputLayoutEvaluation>>false</outputLayoutEvaluation> <privilegeInfoList> <privilegeInfo> <roleName>Administrator</roleName> <roleId>1</roleId> <overridden>>false</overridden> <contentTypePrivList/> </privilegeInfo> </privilegeInfoList> <attributeConstraintsInfoList> <attributeConstraintsInfo> <attributeName>Name</attributeName> <roleName>Administrator</roleName> <constraintName>PREVENT CHANGE</constraintName> </attributeConstraintsInfo> </attributeConstraintsInfoList> </statusInfo> </statusInfoList> <contentTypeInfoList> <contentTypeInfo> <name>Asset</name> </contentTypeInfo> <contentTypeInfo id="1102"/> </contentTypeInfoList> </workflowInfo> </workflowInfoList></code>		

	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <workflowInfoList> <workflowInfo id="20"> <name>Publishing Workflow</name> <domainId>18</domainId> <created>2013-11- 28T08:58:00Z</created> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2013- 11-28T08:58:00Z</lastModified> <statusInfoList> <statusInfo id="20"> <name>New Status 001</name> <domainId>6</domainId> <colorRed>0</colorRed> <colorGreen>0</colorGreen> <colorBlue>255</colorBlue> <position>1</position> <redlineTracking>>false</redlineTracking> <checkInLayoutEvaluation>>false</checkInLayoutEvaluation> <outputLayoutEvaluation>>false</outputLayoutEvaluation> <privilegeInfoList> <privilegeInfo> <roleName>Administrator</roleName> <roleId>1</roleId> <overridden>>false</overridden> <contentTypePrivList/> </privilegeInfo> </privilegeInfoList> <attributeConstraintsInfoList> <attributeConstraintsInfo> <attributeId>2</attributeId> <attributeName>Name</attributeName> <roleId>1</roleId> <roleName>Administrator</roleName> <constraintId>1</constraintId> <constraintName>PREVENT CHANGE</constraintName> </attributeConstraintsInfo> </attributeConstraintsInfoList> </statusInfo> </statusInfoList> <contentTypeInfoList> <contentTypeInfo id="1102"> <name>Project</name> <contentTypeHierarchy>System;Asset;Project;</contentTypeHierar chy> <domainId>0</domainId> <systemDefined>>true</systemDefined> <childrenAvailable>>true</childrenAvailable> <extensible>>true</extensible> </contentTypeInfo> </contentTypeInfoList> </workflowInfo> </workflowInfoList></pre>
--	---

getWorkflow

Function	Get a workflow with a particular ID or name.	
Description	This request returns all the metadata of a specified workflow.	
Type	GET	
URL Format	.../admin/workflows/[workflow name]	
Java API	Method Name	Class Name
	getWorkflow	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns a workflowInfoList object with the metadata of a specified workflow in XML format.	
Examples	URL:	
	http://serverName:serverPort/rest/service/admin/workflows/workflow1	
Examples	Result:	
	<pre><workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> <domainId>18</domainId> <created>2013-</pre>	

	<pre> 12-23T12:06:33Z</created> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2013- 12-23T12:06:33Z</lastModified> <statusInfoList> <statusInfo id="10"> <name>Status1</name> <domainId>6</domainId> <colorRed>50</colorRed> <colorGreen>150</colorGreen> <colorBlue>10</colorBlue> <position>1</position> <redlineTracking>>false</redlineTracking> <checkInLayoutEvaluation>>false</checkInLayoutEvaluation> <outputLayoutEvaluation>>false</outputLayoutEvaluation> <privilegeInfoList> <privilegeInfo> <roleName>Administrator</roleName> <roleId>1</roleId> <overridden>>false</overridden> <contentTypePrivList/> </privilegeInfo> <privilegeInfo> <roleName>UserClass2</roleName> <roleId>10</roleId> <overridden>>false</overridden> <contentTypePrivList/> </privilegeInfo> </privilegeInfoList> <attributeConstraintsInfoList/> </statusInfo> </statusInfoList> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> <contentTypeHierarchy>System;Asset;</contentTypeHierarchy> <domainId>0</domainId> <systemDefined>>true</systemDefined> <childrenAvailable>>true</childrenAvailable> <extensible>>true</extensible> </contentTypeInfo> </contentTypeInfoList> </workflowInfo> </workflowInfoList> </pre>
--	--

Function	Get a workflow		
Description	This request returns the metadata and status of a specified workflow.		
Type	Get		
URL Format	.../admin/workflows/{workflowIdorName}		
Java API	Method Name		Class Name
	getWorkflow		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	workflowIdOrName	string	The ID or Name of the workflow to be retrieved.
Returns	Returns a workflowInfoList object containing information on the specified workflow.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/workflows/21		
Examples	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <workflowInfoList> <workflowInfo id="21"> <name>New Workflow</name> <domainId>18</domainId> <created>2013-12- 09T11:45:42Z</created> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2013- 12-09T11:45:42Z</lastModified> <statusInfoList> <statusInfo id="24"> <name>New Status 001</name> <domainId>6</domainId> <colorRed>0</colorRed> <colorGreen>0</colorGreen> <colorBlue>255</colorBlue> <position>1</position> <redlineTracking>>false</redlineTracking> </pre>		

	<pre> <checkInLayoutEvaluation>>false</checkInLayoutEvaluation> <outputLayoutEvaluation>>false</outputLayoutEvaluation> <privilegeInfoList> <privilegeInfo> <roleName>Administrator</roleName> <roleId>1</roleId> <overridden>>false</overridden> <contentTypePrivList/> </privilegeInfo> </privilegeInfoList> <attributeConstraintsInfoList> <attributeConstraintsInfo> <attributeId>2</attributeId> <attributeName>Name</attributeName> <roleId>1</roleId> <roleName>Administrator</roleName> <constraintId>1</constraintId> <constraintName>PREVENT CHANGE</constraintName> </attributeConstraintsInfo> </attributeConstraintsInfoList> </statusInfo> </statusInfoList> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> <contentTypeHierarchy>System;Asset;</contentTypeHierarchy> <domainId>0</domainId> <systemDefined>>true</systemDefined> <childrenAvailable>>true</childrenAvailable> <extensible>>true</extensible> </contentTypeInfo> </contentTypeInfoList> </workflowInfo> </workflowInfoList> </pre>
--	---

getAllWorkflows

Function	Get all workflows.	
Description	This request returns all the metadata of all the workflows that exist on the QPP Server.	
Type	GET	
URL Format	.../admin/workflows	
Java API	Method Name	Class Name
	getAllWorkflows	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns a workflowInfoList object containing an array of all the workflows that exist on the QPP Server.	
Examples	URL:	
	http://serverName:serverPort/rest/service/admin/workflows	
Examples	Result:	
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <workflowInfoList> <workflowInfo id="10"> <name>Workflow1</name> ... <statusInfoList> <statusInfo id="10"> <name>Status1</name> ... </statusInfo> <statusInfo id="11"> <name>Status2</name> ... </statusInfo> </statusInfoList> </workflowInfo> <workflowInfo id="11"> <name>Workflow2</name> ... <statusInfoList> <statusInfo id="12"> <name>Status1</name> ... </statusInfo> <statusInfo id="13"> <name>Status2</name> ... </statusInfo> </statusInfoList> </workflowInfo> </workflowInfoList> </pre>	

updateWorkflows

Function	Update workflows.
-----------------	-------------------

Description	This request is used to update the metadata of a specified single or multiple workflows and their statuses.		
Type	POST		
URL Format	.../admin/workflows		
Java API	Method Name		Class Name
	updateWorkflows		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	workflowinfolist=<xml />		Object containing a list of the successfully updated workflows .
Returns	Returns a workflowInfoList object containing a list of all the successfully updated workflows.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/admin/workflows?op=update&workflowinfolist= <workflowInfoList> <workflowInfo id="22"> <name>DefaultWorkflow New</name> </workflowInfo> </workflowInfoList></pre>		
	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <workflowInfoList> <workflowInfo id="22"> <name>DefaultWorkflow New</name> <domainId>18</domainId> <created>2013-11-28T09:01:38Z</created> <createdBy>1</createdBy> <creatorName>Admin</creatorName> <lastModifiedBy>1</lastModifiedBy> <lastModifierName>Admin</lastModifierName> <lastModified>2013-11-28T09:03:26Z</lastModified> <statusInfoList/> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> <contentTypeHierarchy>System;Asset;</contentTypeHierarchy> <domainId>0</domainId> <systemDefined>true</systemDefined> <childrenAvailable>true</childrenAvailable> <extensible>true</extensible> </contentTypeInfo> </contentTypeInfoList> </workflowInfo> </workflowInfoList></pre>		

deleteWorkflow

Function	Delete a workflow.		
Description	This request is used to delete the specified workflow.		
Type	POST		
URL Format	.../admin/workflows/{workflowIdorName}		
Java API	Method Name		Class Name
	deleteWorkFlow		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify

			which operation is to be performed. (Mandatory)
	<code>workflowIdOrName</code>	string	ID or Name of the workflow to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/workflows/22?op=delete</pre>		

createPreferences

Function	Create Preferences.		
Description	This request is used to create a single or multiple preferences at one time.		
Type	POST		
URL Format	<code>.../admin/preferences</code>		
Java API	Method Name		Class Name
	<code>createPreferences</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>preferenceinfolist</code>	string	List of preferences to create.
Returns	Returns a <code>preferenceInfoList</code> object containing a list of successfully created preferences.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/preferences?op=create&preferenceinfolist=<xml/></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <preferenceInfoList> <preferenceInfo id="10005"> <name>Demo user preference 3</name> <category id="1">USER</category> <possibleValuesExist>>false</possibleValuesExist> <defaultValue>Ths is the default value.</defaultValue> <systemDefined>>false</systemDefined> </preferenceInfo> <preferenceInfo id="10006"> <name>Demo preference without possible values</name> <category id="0">SYSTEM</category> <possibleValuesExist>>false</possibleValuesExist> <defaultValue>This is the default new value.</defaultValue> <systemDefined>>false</systemDefined> </preferenceInfo> </preferenceInfoList></pre>		

getPreference

Function	Get a specific preference.
Description	This request returns the metadata of a specified preference.

Type	GET		
URL Format	../admin/preferences/{preferenceIdOrName}		
Java API	Method Name		Class Name
	getPreference		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	preferenceIdOrName	string	The ID or Name of the preference to be retrieved.
Returns	Returns a <code>preferenceInfoList</code> object containing the metadata of the specified preference.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/preferences/Content Display View</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <preferenceInfoList> <preferenceInfo id="2006"> <name>Content Display View</name> <category id="1">USER</category> <possibleValuesExist>true</possibleValuesExist> <defaultValue id="1">Keep Last Setting</defaultValue> <preferenceValueList> <preferenceValue valueId="1" preferenceName="Content Display View" preferenceId="2006">Keep Last Setting</preferenceValue> <preferenceValue valueId="2" preferenceName="Content Display View" preferenceId="2006">WYSIWYG</preferenceValue> <preferenceValue valueId="3" preferenceName="Content Display View" preferenceId="2006">Galley</preferenceValue> <preferenceValue valueId="4" preferenceName="Content Display View" preferenceId="2006">Full Screen</preferenceValue> </preferenceValueList> <systemDefined>true</systemDefined> </preferenceInfo> </preferenceInfoList></pre>		

getAllPreferences

Function	Get all Preferences.		
Description	This request returns all of the existing preference definitions.		
Type	GET		
URL Format	../admin/preferences		
Java API	Method Name		Class Name
	getAllPreferences		com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.		
Returns	Returns a <code>preferenceInfoList</code> object containing metadata of all existing preferences with its possible values.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/preferences</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?></pre>		

```
<preferenceInfoList> <preferenceInfo id="1007"> <name>Display all
Redline Deletions in One Color</name> <category
id="0">SYSTEM</category>
<possibleValuesExist>>false</possibleValuesExist>
<defaultValue>No</defaultValue>
<systemDefined>>true</systemDefined> </preferenceInfo>
<preferenceInfo id="1008"> <name>Redline Deletion Color</name>
<category id="0">SYSTEM</category>
<possibleValuesExist>>false</possibleValuesExist>
<defaultValue>255,0,0</defaultValue>
<systemDefined>>true</systemDefined> </preferenceInfo> ....
</preferenceInfoList>
```

getPreferenceValues

Function	Get Preference values.		
Description	This request is used to get preference values for a given preference corresponding to the logged-in user.		
Type	GET		
URL Formatadmin/preferencevalues		
Java API	Method Name		Class Name
	getPreferenceValues		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	preferences	string	Comma separated list of Ids or Names of Preferences to be retrieved.
Returns	Returns an <code>preferenceValueList</code> object containing the values set for given preferences for the logged-in user.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/preferencevalues?preferences=3018, 1007, 3011, Archive Revisions		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <preferenceValueList> <preferenceValue preferenceName="Display all Redline Deletions in One Color" preferenceId="1007">No</preferenceValue> <preferenceValue valueId="1" preferenceName="Default Action for Double Click on Quark XPress Projects" preferenceId="3011">Do Check Out</preferenceValue> <preferenceValue valueId="3" preferenceName="Delete Attached Files" preferenceId="3018">Ask Before Deleting</preferenceValue> <preferenceValue valueId="3" preferenceName="Archive Revisions" preferenceId="4018">Ask</preferenceValue> </preferenceValueList></pre>		

updatePreferences

Function	Update Preferences.		
Description	This request is used to update the metadata of a specified single or multiple preferences.		
Type	POST		
URL Format	.../admin/preferences		
Java API	Method Name	Class Name	
	updatePreferences	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	preferenceinfolist=<xml/>		Object containing the metadata of the preferences to be updated.
Returns	Returns a <code>preferenceInfoList</code> object containing a list of all the successfully updated preferences.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/preferences?op=update&preferenceinfolist=<xml/></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <preferenceInfoList> <preferenceInfo id="10005"> <name>Demo user preference 3</name> <category id="1">USER</category> <possibleValuesExist>>false</possibleValuesExist> <defaultValue>This is the default value.</defaultValue> <systemDefined>>false</systemDefined> </preferenceInfo> <preferenceInfo id="10006"> <name>Demo preference without possible values</name> <category id="0">SYSTEM</category> <possibleValuesExist>>false</possibleValuesExist> <defaultValue>This is the default new value.</defaultValue> <systemDefined>>false</systemDefined> </preferenceInfo> </preferenceInfoList></pre>		

updatePreferenceValues

Function	Update Preference Values.		
Description	This request updates the preference values corresponding to the logged-in user.		
Type	POST		
URL Format	.../preferenceValues		
Java API	Method Name	Class Name	
	updatePreferenceValues	com.quark.qpp.service.facade Class AdminFacade	
Parameters	Name	Type	Description
	op=update	long	The op parameter allows you to specify

			which operation is to be performed. (Mandatory)
	<code>preferencevaluelist=[xml]</code>	string	Object containing the metadata of the preference values to be updated.
Returns	Returns a <code>preferencevaluelist</code> object containing the preferences values successfully updated.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/preferencevalues? op=update&preferencevaluelist=<xml/></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <preferenceValueList> <preferenceValue valueId="1" preferenceName="user 1" preferenceId="10061">system value 2</preferenceValue> <preferenceValue valueId="2" preferenceName="user 2" preferenceId="10062">system value 3</preferenceValue> </preferenceValueList></pre>		

deletePreference

Function	Delete a preference.		
Description	This request is used to delete the specified preference.		
Type	POST		
URL Format	<code>.../admin/preferences/{preferenceIdorName}</code>		
Java API	Method Name	Class Name	
	<code>deletePreference</code>	<code>com.quark.qpp.service.facade Class AdminFacade</code>	
Parameters	Name	Type	Description
	<code>op=delete</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/preference/{prefe renceIdOrName}?op=delete</pre>		

createRepositories

Function	Create one or more storage repositories.		
Description	This request is used to create new storage repositories.		
Type	POST		
URL Format	<code>.../admin/repositories</code>		
Java API	Method Name	Class Name	
	<code>createRepositories</code>	<code>com.quark.qpp.service.facade</code>	

			Class <code>AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>storagerepositoryinfo list</code>	string	List of storage repositories to create.
Returns	Returns a <code>storageRepositoryInfoList</code> object containing a list of successfully created repositories.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/repositories?op=create&storagerepositoryinfo list= <storageRepositoryInfoList> <storageRepositoryInfo> <name>QPP_Repo1234a73d</name> <connectionInfo><connectionProperties><property name='url' required='false'> \\dinesh- 2k8\Repo1</property></connectionProperties></connectionInfo> <repositoryTypeId>1</repositoryTypeId> <statusId>0</statusId> </storageRepositoryInfo> </storageRepositoryInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <storageRepositoryInfoList> <storageRepositoryInfo id="20"> <name>QPP_Repo1234a73d</name> <connectionInfo><connectionProperties><property name='url' required='false'> \\dinesh- 2k8\Repo1</property></connectionProperties></connectionInfo> <repositoryType>fileRepositoryAdapter</repositoryType> <repositoryTypeId>1</repositoryTypeId> <status>READ_WRITE</status> <statusId>1</statusId> </storageRepositoryInfo> </storageRepositoryInfoList></pre>		

getRepository

Function	Get a specific repository.		
Description	This request returns a storage repository with the given Id or Name.		
Type	GET		
URL Format	../admin/repositories/{repositoryIdOrName}		
Java API	Method Name	Class Name	
	<code>getRepository</code>	<code>com.quark.qpp.service.facade</code> Class <code>AdminFacade</code>	
Parameters	Name	Type	Description
	<code>repositoryIdOrName</code>	string	The ID or Name of the repository to be retrieved.
Returns	Returns a <code>repositoryInfoList</code> object containing the metadata of the specified repository.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/repositories/1</pre> <p>Result:</p>		


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storageRepositoryInfoList> <storageRepositoryInfo id="1">
<name>TestRep1</name>
<connectionInfo><connectionProperties><property name='url'
required='true'>
C:\TestRep1</property></connectionProperties></connectionInfo>
<repositoryType>fileRepositoryAdapter</repositoryType>
<repositoryTypeId>1</repositoryTypeId>
<status>READ_WRITE</status> <statusId>1</statusId>
</storageRepositoryInfo> <storageRepositoryInfo id="10">
<name>TestRepA</name>
<connectionInfo><connectionProperties><property name='url'
required='true'>
C:\TestRepA</property></connectionProperties></connectionInfo>
<repositoryType>fileRepositoryAdapter</repositoryType>
<repositoryTypeId>1</repositoryTypeId>
<status>READ_WRITE</status> <statusId>1</statusId>
</storageRepositoryInfo> </storageRepositoryInfoList>
```

getAllRepositories

Function	Get all storage repositories.	
Description	This request returns all the information regarding all repositories registered with the platform.	
Type	GET	
URL Format	../admin/repositories	
Java API	Method Name	Class Name
	getAllRepositories	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns a <code>storageRepositoryInfoList</code> object containing a list of all the registered repositories.	
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/admin/repositories</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <storageRepositoryInfoList> <storageRepositoryInfo id="1"> <name>TestRep1</name> <connectionInfo><connectionProperties><property name='url' required='true'> C:\TestRep1</property></connectionProperties></connectionInfo> <repositoryType>fileRepositoryAdapter</repositoryType> <repositoryTypeId>1</repositoryTypeId> <status>READ_WRITE</status> <statusId>1</statusId> </storageRepositoryInfo> <storageRepositoryInfo id="10"> <name>TestRepA</name> <connectionInfo><connectionProperties><property name='url' required='true'> C:\TestRepA</property></connectionProperties></connectionInfo> <repositoryType>fileRepositoryAdapter</repositoryType> <repositoryTypeId>1</repositoryTypeId> <status>READ_WRITE</status> <statusId>1</statusId> </storageRepositoryInfo> </storageRepositoryInfoList></pre>	

updateRepositories

Function	Update one or more storage repositories.		
Description	This request is used to update the storage repositories on the system.		
Type	POST		
URL Format	.../admin/repositories		
Java API	Method Name		Class Name
	updateRepositories		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	storagerepositoryinfo list=<xml/>		Object containing the metadata of the storage repositories to be updated.
Returns	Returns a <code>storageRepositoryInfoList</code> object containing a list of all the successfully updated storage repositories.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/repositories?op=update&storagerepositoryinfoList= <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <storageRepositoryInfoList> <storageRepositoryInfo id="24"> <name>QPP_Repo111</name> <connectionInfo><connectionProperties><property name='url' required=' false'> \\dinesh-2k8\Repo1</property></connectionProperties></connectionInfo> <repositoryType>fileRepositoryAdapter</repositoryType> <repositoryTypeId>1</repositoryTypeId> <status>READ_WRITE</status> <statusId>1</statusId> </storageRepositoryInfo> </storageRepositoryInfoList></pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <storageRepositoryInfoList> <storageRepositoryInfo id="24"> <name>QPP_Repo111</name> <connectionInfo><connectionProperties><property name='url' required=' false'> \\dinesh-2k8\Repo1</property></connectionProperties></connectionInfo> <repositoryType>fileRepositoryAdapter</repositoryType> <repositoryTypeId>1</repositoryTypeId> <status>READ_WRITE</status> <statusId>1</statusId> </storageRepositoryInfo> </storageRepositoryInfoList></pre>		

deleteRepository

Function	Delete a storage repository.	
Description	This request is used to delete the specified storage repository.	
Type	POST	
URL Format	.../admin/repositories/{repositoryIdorName}	
Java API	Method Name	Class Name
	deleteRepository	com.quark.qpp.service.facade

	Name	Type	Description
Parameters	<code>op=delete</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/repositories/10?op=delete</code>		

createPrivileges

Function	Create Privileges.		
Description	This request creates the required privilege definition(s).		
Type	POST		
URL Format	<code>.../privileges?op=create</code>		
Java API	Method Name	Class Name	
	<code>createPrivileges</code>	<code>com.quark.qpp.service.facadeClass AdminFacade</code>	
Parameters	Name	Type	Description
	<code>priviledgeDefinitionList =<XML/></code>	string	A privilegeDefinitionList object in XML format. See "Request/Response parameter formats."
Returns	Returns a <code>privilegeDefinitionList</code> object containing the metadata of the newly created privileges.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/privileges?op=create</code>		
	Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeDefinitionList> <privilegeDefinition name="Sample Privilege" id="2500000"> <group id="100000">General</group> <statusBased>true</statusBased> <systemDefined>false</systemDefined> <contentPrivilege>true</contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList></pre>		

getPrivilege

Function	Get Privilege.		
Description	This request returns a specific privilege(s) in system with given id or name.		
Type	GET		
URL Format	<code>.../privileges/{privilegeIdOrName}</code>		

Java API	Method Name	Class Name
	getPrivilege	com.quark.qpp.service.facadeClass AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns specific privilege(s) in system with given ID or name. If privilege name is specified and there exists multiple privileges in different privilege groups, with the given name, list of all such privileges will be returned.	
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/admin/privileges/Override Another Users Checkout</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeDefinitionList> <privilegeDefinition name="Override Another Users Checkout" id="1000003"> <group id="100000">General</group> <statusBased>true</statusBased> <systemDefined>true</systemDefined> <contentPrivilege>true</contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100">...</contentTypeInfo> <contentTypeInfo id="1101">...</contentTypeInfo> <contentTypeInfo id="1102"> <name>Project</name> </contentTypeInfo> <contentTypeInfo id="1103"> <name>Picture</name> </contentTypeInfo> <contentTypeInfo id="1104">...</contentTypeInfo> <contentTypeInfo id="1105">...</contentTypeInfo> </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList></pre>	

getAllPrivileges

Function	Get All Privileges.		
Description	This request returns all privilege definitions in the system.		
Type	GET		
URL Format	.../privileges		
Java API	Method Name	Class Name	
	getAllPrivileges	com.quark.qpp.service.facadeClass AdminFacade	
Parameters	Name	Type	Description
	contenttype=[true false]	boolean	Default value is true
	application=[true false]	boolean	Default value is true
Returns	Returns a <code>privilegeDefinitionList</code> object containing the metadata of all privileges in the system.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/privileges		
	Result:		

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<privilegeDefinitionList> <privilegeDefinition name="Check In
New" id="1000001"> <group id="100000">General</group>
<statusBased>false</statusBased>
<systemDefined>true</systemDefined>
<contentPrivilege>true</contentPrivilege>
<contentTypeInfoList> <contentTypeInfo id="1100">
<name>Asset</name> </contentTypeInfo> <contentTypeInfo
id="1101"> <name>Document</name> </contentTypeInfo>
<contentTypeInfo id="1102">...</contentTypeInfo>
<contentTypeInfo id="1103">...</contentTypeInfo>
<contentTypeInfo id="1104">...</contentTypeInfo> ...
</contentTypeInfoList> </privilegeDefinition> .....
</privilegeDefinitionList>
```

getApplicableApplicationPrivileges

Function	Get applicable application privileges for the logged in user.		
Description	This request returns enabled applicable application privileges for the logged in user. Applicable privileges are a union of all the privileges granted to him as a user as well as to groups of which the user is a member.		
Type	GET		
URL Format	../admin/users/		
Java API	Method Name		Class Name
	getApplicableApplicationPrivileges		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op= applicableapplicationprivileges	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	privileges= [comma-separated list of privilege Names or IDs]	string	The set of privilege values. (Optional)
Returns	Returns the applicationPrivList object containing all of the applicable application privileges.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/users? op=applicableapplicationprivileges		
	Result:		
	<pre><applicationPrivList> <privilege id="14000023"> <name>Delete</name> <privilegeHierarchy> Administration;Storage;Storage Rules;Delete </privilegeHierarchy> </privilege> <privilege id="14000022"> <name>Create/Modify</name> <privilegeHierarchy> Administration;Storage;Storage Rules;Create/Modify </privilegeHierarchy> </privilege> </applicationPrivList> Get enabled applicable application privileges for the logged in user from the given privileges:URL:http://serverName:serverPort/rest/service/admin /users?</pre>		

```

op=applicableapplicationprivileges&privileges=14000023,...,Result: <applicationPrivList> <privilege id="14000023">
<name>Delete</name> <privilegeHierarchy>
Administration;Storage;Storage Rules;Delete
</privilegeHierarchy> </privilege> <privilege id="14000022">
<name>Create/Modify</name> <privilegeHierarchy>
Administration;Storage;Storage Rules;Create/Modify
</privilegeHierarchy> </privilege> .....
</applicationPrivList>
    
```

getApplicableContentPrivileges

Function	Get applicable content privileges for the logged in user.		
Description	This request returns enabled privileges of the given content type applicable for the logged in user. Applicable privileges are a union of all the privileges granted to him as a user as well as to groups of which the user is a member.		
Type	GET		
URL Format	../admin/users/		
Java API	Method Name		Class Name
	getApplicableContentPrivileges		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op= applicablecontentprivileges	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contenttype=[contentType ID or Name or Hierarchy]	string	
	privileges= [comma-separated list of privilege Names or IDs]	string	The set of privilege values. (Optional)
Returns	Returns the <code>contentPrivList</code> object containing all of the applicable content privileges.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/users?op=applicablecontentprivileges&contenttype=Picture</code>		
	Result: <pre> <contentTypePriv> <privilege id="3000006"> <name>Change Workflow of Checked In Asset</name> <privilegeHierarchy> Edit Attributes;Change Workflow of Checked In Asset </privilegeHierarchy> </privilege> <privilege id="10000015"> <name>Get Original Assets</name> <privilegeHierarchy>General;Get Original Assets</privilegeHierarchy> </privilege> </contentTypePriv> Get enabled applicable content privileges for the logged in user from the given privileges:URL:http://serverName:serverPort/rest/service/admin/users? op=applicablecontentprivileges&contenttype=Picture&privileges= </pre>		

	<pre>3000006, 10000015Result: <contentTypePriv> <privilege id="3000006"> <name>Change Workflow of Checked In Asset</name> <privilegeHierarchy> Edit Attributes;Change Workflow of Checked In Asset </privilegeHierarchy> </privilege> <privilege id="10000015"> <name>Get Original Assets</name> <privilegeHierarchy>General;Get Original Assets</privilegeHierarchy> </privilege> </contentTypePriv></pre>
--	---

updatePrivileges

Function	Update Privileges.		
Description	This request updates the required privilege definition(s).		
Type	POST		
URL Format	.../privileges?op=update		
Java API	Method Name	Class Name	
	updatePrivileges	com.quark.qpp.service.facadeClass AdminFacade	
Parameters	Name	Type	Description
	priviledgeDefinitionlist =<XML/>	string	A privilegeDefinitionList object in XML format. See "Request/Response parameter formats."
Returns	Returns a privilegeDefinitionList object containing the metadata of the successfully updated privileges.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/privileges?op=update		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeDefinitionList> <privilegeDefinition name="Sample Privilege" id="2500000"> <group id="100000">General</group> <statusBased>true</statusBased> <systemDefined>false</systemDefined> <contentPrivilege>true</contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> <contentTypeInfo id="1102"> <name>Project</name> </contentTypeInfo> </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList></pre>		

deletePrivilege

Function	Delete Privilege.		
Description	This request deletes the specified privilege Id.		
Type	POST		
URL Format	.../privileges/{privilegeIdOrName}?op=delete		
Java API	Method Name	Class Name	
	deletePrivilege	com.quark.qpp.service.facadeClass	

	AdminFacade
Parameters	This request requires no input parameters.
Returns	A status of 200 is returned in response to a successful request execution.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/privileges/{privilegeIdOrName}?op=delete</pre>

createPrivilegeGroups

Function	Creates New Privilege Group.		
Description	This request creates new privilege group along with its subgroups and privileges in the specified hierarchy.		
Type	POST		
URL Format	.../privilegegroups?op=create		
Java API	Method Name	Class Name	
	createPrivilegeGroups	com.quark.qpp.service.facadeClass AdminFacade	
Parameters	Name	Type	Description
	privilegegroupdefinitionlist =<XML/>	string	A privilegeGroupDefinitionList object in XML format. See "Request/Response parameter formats."
Returns	Returns a privilegeGroupDefinitionList object containing the metadata of the successfuy created privilege groups.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/admin/privilegegroups?op=create</pre>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeGroupDefinitionList> <privilegeGroupDefinition name="Sample Privilege Group"> <privilegeDefinitionList> <privilegeDefinition name="Check In New" id="1000001"> <group id="100000">General</group> <statusBased>false</statusBased> <systemDefined>true</systemDefined> <contentPrivilege>true</contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> <contentTypeInfo id="1100"> <name>CustomeContentType_49933</name> </contentTypeInfo> </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList> <parentGroup id="-1"/> </privilegeGroupDefinition> </privilegeGroupDefinitionList></pre>		

getPrivilegeGroup

Function	Get Privilege Group Definition.
Description	This request returns all existing privilege group definitions.

Type	GET	
URL Format	../privilegegroups/{privilegeGroupIdOrName}	
Java API	Method Name	Class Name
	getPrivilegeGroup	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns information for the specified privilege group based on its id or name specified.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/privilegegroups/General</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeGroupDefinitionList> <privilegeGroupDefinition name="General" id="100000"> <privilegeDefinitionList> <privilegeDefinition name="Check In New" id="1000001"> <group id="100000">General</group> <statusBased>false</statusBased> <systemDefined>true</systemDefined> <contentPrivilege>true</contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> ... </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList></pre>	

getAllPrivilegeGroups

Function	Get All Privilege Group Definitions.	
Description	This request returns all existing privilege group definitions.	
Type	GET	
URL Format	../privilegegroups	
Java API	Method Name	Class Name
	getAllPrivilegeGroups	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns all existing privilege group definitions. Each privilege group contains its information along with its privileges and sub-privilege group definitions.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/admin/privilegegroups</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeGroupDefinition name="Administration" id="99998"> .. </privilegeGroupDefinition> <privilegeGroupDefinition name="Workspace" id="99999"> ... </privilegeGroupDefinition> <privilegeGroupDefinition name="General" id="100000"> <privilegeDefinitionList> <privilegeDefinition name="Check In New" id="1000001"> <group id="100000">General</group> <statusBased>false</statusBased></pre>	

```
<systemDefined>true</systemDefined>
<contentPrivilege>true</contentPrivilege>
<contentTypeInfoList> <contentTypeInfo id="1100">
<name>Asset</name> </contentTypeInfo> ...
</contentTypeInfoList> </privilegeDefinition> .....
</privilegeDefinitionList>
```

updatePrivilegeGroups

Function	Updates Given Privilege Group.		
Description	This request updates the given privilege group.		
Type	POST		
URL Format	.../privilegegroups?op=update		
Java API	Method Name	Class Name	
	updatePrivilegeGroups	com.quark.qpp.service.facadeClass AdminFacade	
Parameters	Name	Type	Description
	privilegegroupdefinitionlist =<XML/>	string	A privilegeGroupDefinitionList object in XML format. See "Request/Response parameter formats."
Returns	Returns a privilegeGroupDefinitionList object containing the metadata of the successfuy updated privilege groups.		
Examples	URL:		
	http://serverName:serverPort/rest/service/admin/privilegegroups?op=update		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <privilegeGroupDefinitionList> <privilegeGroupDefinition name="Sample Privilege Group"> <privilegeDefinitionList> <privilegeDefinition name="Check In New" id="1000001"> <group id="100000">General</group> <statusBased>false</statusBased> <systemDefined>true</systemDefined> <contentPrivilege>true</contentPrivilege> <contentTypeInfoList> <contentTypeInfo id="1100"> <name>Asset</name> </contentTypeInfo> <contentTypeInfo id="1100"> <name>CustomeContentType_49933</name> </contentTypeInfo> </contentTypeInfoList> </privilegeDefinition> </privilegeDefinitionList> <parentGroup id="-1"/> </privilegeGroupDefinition> </privilegeGroupDefinitionList></pre>		

deletePrivilegeGroup

Function	Deletes Given Privilege Group.		
Description	This request deletes the given privilege group.		
Type	POST		
URL Format	.../privilegegroups/{privilegeGroupIdOrName}?op=delete		
Java API	Method Name	Class Name	

	<code>deletePrivilegeGroup</code>	<code>com.quark.qpp.service.facadeClass AdminFacade</code>
Parameters	This request requires no input parameters.	
Returns	A status of 200 is returned in response to a successful request execution.	
Examples	URL: <code>http://serverName:serverPortt/rest/service/admin/privilegegroups/ {privilegeGroupIdOrName}?op=delete</code>	

addDomainValue

Function	Add a domain value.		
Description	This request is used to add a new value in domain under the mentioned parent value. If the parent value Id is -1, then the domain value will be added at root level.		
Type	POST		
URL Format	<code>.../domains/{domainIdOrName}/{newValueName}</code>		
Java API	Method Name		Class Name
	<code>addDomainValue</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=create</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>Parentvalue</code>	string	Id/Name of the domain value under which new domain value is to be added.
	<code>sourceReference</code>	string	Source reference of the new domain value.
	<code>position</code>	string	Position among siblings where the domain value will be added.Can have values like 1, 5,...,(use -1 to add domain value at the end).
Returns	Returns metadata of the newly created domain value.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/domains/501/Samp leQPPDomainVal_a2b</code> Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <domainValue id="1"> <name>SampleQPPDomainVal_a2b</name> <valueHierarchy>Desktop</valueHierarchy> <sourceReference>source</sourceReference> </domainValue></pre>		

getDomainValue

Function	Get Domain Value.
Description	This request returns desired DomainValue information.

Type	GET	
URL Format	.../domains/{domainIdOrName}/{valueId}	
Java API	Method Name	Class Name
	getDomainValue	com.quark.qpp.service.facade Class AdminFacade
Parameters	This request requires no input parameters.	
Returns	Returns metadata of the specified domain value.	
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/domains/Users/10</code>	
	Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <domainValue id="10"> <name>Admin2</name> <valueHierarchy>Desktop</valueHierarchy> </domainValue></pre>	

getDomainValueXml

Function	Get XML Value for Given Domain.		
Description	This request returns xml value of the given domain value.		
Type	GET		
URL Format	.../domains/{domainIdOrName}/{valueId}		
Java API	Method Name	Class Name	
	getDomainValueXml	com.quark.qpp.service.facadeClass SessionFacade	
Parameters	Name	Type	Description
	op=fetchxml	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	Returns xml value of the given domain value.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/domains/{domainIdOrName}/{valueId}?op=fetchxml</code>		
	Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <sampleXMLvalue/></pre>		

setDomainValueXml

Function	Set XML Value for Given Domain.		
Description	This request sets xml for the given domain value.		
Type	POST		
URL Format	.../domains/{domainIdOrName}/{valueId}		

Java API	Method Name		Class Name
	setDomainValueXml		com.quark.qpp.service.facadeClass SessionFacade
Parameters	Name	Type	Description
	op=setxmlvalue	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	valueId=[]	string	Id of the domain value whose xml is to be set.
	xmlvalue=[XML Dat]	string	XML Content.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/domains/{domainIdOrName}/{valueId}?op=setxmlvalue&valueid={Id of Domain Value}&xmlvalue={XML Data}</code>		

searchDomainValues

Function	Search Domain with XML Values with matching Xpath Expression.		
Description	This request finds domain values whose xml matches with the given xpath.		
Type	GET		
URL Format	.../domains/{domainIdOrName}		
Java API	Method Name		Class Name
	searchDomainValues		com.quark.qpp.service.facadeClass SessionFacade
Parameters	Name	Type	Description
	op=searchvalues	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	xpath=[]	string	Expression to be used for finding required domain values.
Returns	Returns a <code>domainValueList</code> object whose xml matches with the given XPath.		
Examples	URL: <code>http://serverName:serverPort/rest/service/admin/domains/{domainIdOrName} ?op=searchvalues&xpath={xpath to match the XML Value for Defined Domain}</code>		

updateDomainValue

Function	Update a domain value.
Description	This request is used to updates the name and source reference of the domain value

	with the given ID.		
Type	POST		
URL Format	.../domains/{domainIdOrName}/{ValueId}		
Java API	Method Name		Class Name
	updateDomainValue		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=update	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	name	string	The new name for the domain value.
	sourcereference	string	The source reference of the domain value to be updated.
Returns	Returns metadata of the successfully updated domain value.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/domains/501/1?op=update &name=NewName&sourcereference=updatedSourceRef</code>		
	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <domainValue id="1"> <name>newName</name> <valueHierarchy>Desktop</valueHierarchy> <sourceReference>updatedSourceRef</sourceReference> </domainValue></pre>		

deleteDomainValue

Function	Delete a domain value.		
Description	This request is used to delete the value of a domain.		
Type	POST		
URL Format	.../domains/{domainIdOrName}/{ValueId}		
Java API	Method Name		Class Name
	deleteDomainValue		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/domains/501/1?op=delete</code>		

getFavorites

Function	Get all Favorites.		
Description	This request returns all of the favorites items of the logged-on user.		
Type	GET		
URL Format	<code>../admin/favorites</code>		
Java API	Method Name		Class Name
	<code>getFavorites</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	This request requires no input parameters.		
Returns	Returns a <code>favoriteItemList</code> object containing metadata of all of the favorites items for the logged-in user.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/favorites</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <favoriteItemList> <favoriteItem position="1" type="QUERY" id="100">Default</favoriteItem> <favoriteItem position="2" type="COLLECTION" id="10">Home/Pictures</favoriteItem> </favoriteItemList></pre>		

setFavorites

Function	Set Favorites.		
Description	This request sets the favorites items for the logged-in user.		
Type	POST		
URL Format	<code>.../favorites</code>		
Java API	Method Name		Class Name
	<code>setFavorites</code>		<code>com.quark.qpp.service.facade Class AdminFacade</code>
Parameters	Name	Type	Description
	<code>op=setfavorites</code>	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>favoriteitemlist=[xml]</code>	string	Object containing the metadata of the preference values to be updated.
Returns	Returns a <code>favoriteItemList</code> object containing the metadata of favorite items successfully set for logged-in user.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/admin/preferencevalues? op=update&preferencevalueuelist=<xml/></code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?></pre>		

	<pre><favoriteItemList> <favoriteItem position="1" type="QUERY" id="100">Default</favoriteItem> <favoriteItem position="2" type="COLLECTION" id="10">Home/Pictures</favoriteItem> </favoriteItemList></pre>
--	---

getWorkflowInitiatingStatuses

Function	Get Workflow initial statuses.		
Description	This request returns a list of "Workflow Initiating Statuses" for the given workflow. If the roleIdOrName parameter is specified then "Workflow Initiating Statuses" will be returned on the basis of overridden status transitions for the given userclass, otherwise the default initiating statuses are returned.		
Type	GET		
URL Format	.../workflows/{workflowIdOrName}		
Java API	Method Name		Class Name
	getWorkflowInitiatingStatuses		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	workflowinitiatingsta tuses =true	boolean	
	role={ roleIdOrName }	string	The ID or Name of the role whose Workflow Initiating Statuses you wish to be retrieved.
Returns	Returns a statusInfoList object containing its metadata.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/workflows/ Workflow1?workflowinitiatingstatuses=true&role=10</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <statusInfoList> <statusInfo id="10"> <name>Status1</name> <domainId>6</ domainId> <colorRed>50</ colorRed> <colorGreen>150</colorGreen> <colorBlue>10</colorBlue> <position>1</position> <redlineTracking>>false</ redlineTracking> <checkInLayoutEvaluation>>false</checkInLayoutEvaluation> <outputLayoutEvaluation>>false</outputLayoutEvaluation> <privilegeInfoList> <privilegeInfo> <roleName>Administrator</roleName> <roleId>1</roleId> <overridden>>false</overridden> <contentTypePrivList/> </privilegeInfo> ... </privilegeInfoList> </statusInfo> </statusInfoList></pre>		

getStatusTransitions

Function	Get Status Transitions.
Description	This request returns the status transitions set for the given workflow. If the statusIdOrName parameter is specified, then status transitions for a particular status are returned. If the roleIdOrName parameter is specified, then the overridden

	transitions for the given role for particular workflow will be returned. Otherwise the default status transitions are returned.		
Type	GET		
URL Format	.../workflows/{workflowIdOrName}		
Java API	Method Name		Class Name
	getStatusTransitions		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	statusTransitions=true	boolean	
	status={statusIdOrName}	string	The ID or Name of the status whose Status Transitions you wish to be retrieved."
	role={roleIdOrName}	string	The ID or Name of the role whose Status Transitions you wish to be retrieved ."
Returns	Returns an <code>statusTransitionInfoList</code> object containing a list of StatusTransitions set for the given workflow or workflow status for specified role.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/workflows/Workflow1?statusTransitions=true&role=20</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <statusTransitionInfoList> <statusTransitionInfo id="10"> <name>Status1</name> <workflowInitiatingStatus>true</workflowInitiatingStatus> <nextPossibleStatusInfoList> <nextPossibleStatusInfo id="11"> <name>Status2</name> </nextPossibleStatusInfo> </nextPossibleStatusInfoList> </statusTransitionInfo> <statusTransitionInfo id="11"> <name>Status2</name> <workflowInitiatingStatus>false</workflowInitiatingStatus> </statusTransitionInfo> </statusTransitionInfoList></pre>		

Publishing

The publishing capabilities of the API allow application developers to publish content and includes the following REST calls:

- [getPublishingChannels](#)
- [getAllPublishingChannels](#)
- [getDeliveryChannels](#)
- [publishAsset](#)
- [publishBasedOnURI](#)
- [publishBasedOnStreamContext](#)
- [createDatadoc](#)
- [getDatadocChannels](#)

- [fetchDatadocAssets](#)
- [executeDatadocChannel](#)
- [updateDatadoc](#)
- [detachDatadocChannel](#)

getPublishingChannels

Function	Get the publishing channels for a particular content type.	
Description	This request returns the metadata of all the available publishing channels for given content type. Null will be returned in case the channel doesn't exist for the given content type.	
Type	GET	
URL Format	.../publish/publishingchannels/[content type name or ID]	
Java API	Method Name	Class Name
	getPublishingChannels	com.quark.qpp.service.facade Class PublishFacade
Parameters	This request requires no input parameters.	
Returns	Returns a publishingChannelList object.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/publish/publishingchannels/Picture</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo name="IM.JPG" id="convertImage"> <parameter>SOURCE_FILE_EXTENSION</parameter> <parameter>OUTPUT_FILE_NAME</parameter> </publishingChannelInfo> <publishingChannelInfo name="Extract Image Metadata" id="extractImageAttributes"> <parameter>SOURCE_FILE_EXTENSION</parameter> <parameter>OUTPUT_FILE_NAME</parameter> </publishingChannelInfo> <publishingChannelInfo name="EPS.JPG" id="epsPreview"> <parameter>SOURCE_FILE_EXTENSION</parameter> <parameter>OUTPUT_FILE_NAME</parameter> </publishingChannelInfo> </publishingChannelList></pre>	

getAllPublishingChannels

Function	Get all publishing channels.	
Description	This request returns the metadata of all existing Publishing Channels with their parameters.	
Type	GET	
URL Format	.../publish/publishingchannels	
Java API	Method Name	Class Name
	getAllPublishingChannels	com.quark.qpp.service.facade Class PublishFacade

Parameters	This request requires no input parameters.
Returns	Returns a <code>publishingChannelList</code> object containing an array of all existing publishing channels.
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/publish/publishingchannels</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo name="PDF" id="qxpPdf"> <parameter>ASSET_ID</parameter> </publishingChannelInfo> <publishingChannelInfo name="JPEG.ZIP" id="qxpImage"> <parameter>ASSET_ID</parameter> </publishingChannelInfo> </publishingChannelList></pre>

getDeliveryChannels

Function	Get all delivery channels.	
Description	This request returns the metadata of all existing Delivery Channels with their parameters.	
Type	GET	
URL Format	.../publish/deliverychannels	
Java API	Method Name	Class Name
	<code>getDeliveryChannels</code>	<code>com.quark.qpp.service.facade Class PublishFacade</code>
Parameters	This request requires no input parameters.	
Returns	Returns a <code>publishingChannelList</code> object containing an array of all existing delivery channels.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/publish/deliverychannels</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo name="Publish to Sharepoint" id="checkInToSharepoint"> <parameter>ASSET_ID</parameter> </publishingChannelInfo> <publishingChannelInfo name="Publish to FileNet" id="checkInToFileNet"> <parameter>ASSET_ID</parameter> </publishingChannelInfo> </publishingChannelList></pre>	

publishAsset

Function	Publish an asset.
Description	This request returns the stream after publishing an asset into a specified format using the publishing channel.
Type	GET

URL Format	<code>.../publish/[asset ID] or /publish//publishbypath/[asset path]</code>		
Java API	Method Name		Class Name
	<code>publishAsset</code>		<code>com.quark.qpp.service.facade Class PublishFacade</code>
Parameters	Name	Type	Description
	<code>publishingchannel =[Publishing Channel] & [Publishing Channel Parameters]</code>	<code>string</code>	The publishing channel used for publishing the content along with publishing parameters (specific to provided publishing channel)(Mandatory)
Returns	Returns the filestream after publishing an asset into required format.		
Examples	URL:		
	<code>http://serverName:serverPort/publish/[asset ID]? publishingchannel=qxpPdf&parameter1=value1?...</code>		
Examples	Result:		
	The filestream after publishing an asset into required format.		

Function	Deliver an asset.		
Description	This request delivers a QPP or Non-QPP asset at any fileserver, sharepoint server, filenet or documentum.		
Type	GET		
URL Format	<code>.../publish/[asset ID] or /publish//publishbypath/[asset path]</code>		
Java API	Method Name		Class Name
	<code>publishAsset</code>		<code>com.quark.qpp.service.facade Class PublishFacade</code>
Parameters	Name	Type	Description
	<code>deliverychannel =[Delivery Channel] & [Delivery Channel Parameters]</code>	<code>string</code>	The delivery channel used to deliver a QPP asset or published content to any of the available delivery channels. (Mandatory)Delivery channel parameters are specific to the channel and all are mandatory.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL:		
	<code>http://serverName:serverPort/publish/[asset ID]? deliverychannel=checkInToFileNet&parameter1=value1?...</code>		

Function	Publish and deliver an asset.		
Description	This request publishes any QPP or Non-QPP asset into the required format and delivers the published content at any fileserver, sharepoint server, filenet or documentum at the same time.		
Type	GET		
URL Format	<code>.../publish/[asset ID]/publish/[asset ID] or</code>		

	<code>/publish//publishbypath/[asset path]</code>		
Java API	Method Name		Class Name
	<code>publishAsset</code>		<code>com.quark.qpp.service.facade Class PublishFacade</code>
Parameters	Name	Type	Description
	<code>publishingchannel =[Publishing Channel] & [Publishing Channel Parameters]</code>	string	The publishing channel used for publishing the content along with publishing parameters (specific to provided publishing channel).
	<code>deliverychannel =[Delivery Channel] & [Delivery Channel Parameters]</code>	string	The delivery channel used to deliver the QPP asset or published content to any of the available delivery channels. Delivery channel parameters are specific to the channel and all are mandatory.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/publish/[asset ID]? publishingchannel=qxpPdf& deliverychannel=checkInToFileNet& parameter1=value1?...</code>		

publishBasedOnURI

Function	Publish a non-Platform resource.		
Description	Takes input in the form of URI.		
Type	GET		
URL Format	<code>.../publish</code>		
Java API	Method Name		Class Name
	<code>publishBasedOnURI</code>		<code>com.quark.qpp.service.facade Class PublishFacade</code>
Parameters	Name	Type	Description
	<code>uri</code>	string	Input a non QPP resource available on any of the URIs. Supported URIs are file, qpp, classpath, SharePoint and http.
	<code>publishingchannel</code>	string	The publishing channel used for publishing the content along with publishing parameters (specific to provided publishing channel).
Returns	Returns the filestream of output after publishing an asset to the required format.		
Examples	URL: <code>http://serverName:serverPort/rest/service/publish?uri=file: \\server1\\abc\\abc.xml40& publishingchannel=busDocPdf</code> Result: The filestream of output after publishing an asset to the required format.		

Function	Deliver a non-Platform resource.		
Description	Include the file in the request body or as file part. Takes input in the form of URI.		
Type	GET		
URL Format	.../publish		
Java API	Method Name		Class Name
	publishBasedOnURI		com.quark.qpp.service.facade Class PublishFacade
Parameters	Name	Type	Description
	uri	string	Input the non QPP resource available on any of the URIs. Supported URIs are file, qpp, classpath, SharePoint and http.
	deliverychannel	string	The delivery channel used to deliver the QPP asset or published content to any of the available delivery channels. Delivery channel parameters are specific to the channel and all are mandatory.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/publish?uri=qpp://assets/40&deliverychannel=FileSystem&LOCATION=C:\DATA\temp		

publishBasedOnStreamContext

Function	Publish a non-platform resource.		
Description	This request will publish a non-platform resource. Takes input in the form of a file stream.		
Type	POST		
URL Format	.../publish		
Java API	Method Name		Class Name
	publishBasedOnStreamContext		com.quark.qpp.service.facade Class PublishFacade
Parameters	Name	Type	Description
	publishingchannel	string	The publishing channel used for publishing the content along with publishing parameters (specific to provided publishing channel).
Returns	Returns the filestream of output after publishing an asset to the required format.		
Examples	URL: http://serverName:serverPort/rest/service/publish?publishingchannel=busDocPdf Result: The filestream of output after publishing an asset to the required format.		

Function	Deliver a non-Platform resource.		
Description	Publish and deliver non-platform resource. Takes input in the form of a file stream.		
Type	GET		
URL Format	.../publish		
Java API	Method Name		Class Name
	publishBasedOnStreamContext		com.quark.qpp.service.facade Class PublishFacade
Parameters	Name	Type	Description
	deliverychannel	string	The delivery channel used to deliver the QPP asset or published content to any of the available delivery channels. Delivery channel parameters are specific to the channel and all are mandatory.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/publish?deliverychannel=FileSystem&LOCATION=C:\DATA\temp		

createDatadoc

Function	Create a Datadoc.		
Description	This request creates a datadoc by executing the given datadoc channel and checks it in to the platform with the given metadata.		
Type	GET		
URL Format	.../datadocchannels		
Java API	Method Name		Class Name
	createDatadoc		com.quark.qpp.service.facadeClass PublishFacade
Parameters	Name	Type	Description
	op=createdatadoc	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	datadocchannel={DataDoc Channel Id}	string	The Datadoc channel. (Mandatory)
	createMinorVersion={true false}	boolean	Indicates whether you should create a Minor version.
	attributevalues=<XML/>	string	See "Request/Response parameter formats."
Returns	Returns the metadata of newly created Datadoc asset.		
Examples	URL: http://serverName:serverPort/rest/service/publish/?op=createdatadoc &datadocchannel=createDataDocAssetFromHttpUrl &CONTENT_URL=http://Server:Port/SampleBusDoc.xml &attributevalues=<?xml version="1.0" encoding="UTF-8"?>		

	<pre> <attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">SampleBusDoc827574.xml </attributeValue> <attributeValue valueId="1" type="popup" name="Routed to" id="25">Admin</attributeValue> <attributeValue valueId="10" type="popup" name="Status" id="24">Status1</attributeValue> <attributeValue valueId="1" type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue valueId="10" type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue valueId="1104" type="popup" name="Content Type" id="62">Business Document</attributeValue> <attributeValue valueId="0" type="text" name="File extension" id="11">xml</attributeValue> <attributeValue valueId="0" type="text" name="Original filename" id="33">SampleBusDoc827.xml</attributeValue> </attributeValueList> </pre> <p>Result:</p> <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="61"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">61</attributeValue> <attributeValue type="text" name="Name" id="2">SampleBusDoc827574.xml</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2014-10- 28T11:39:46Z</attributeValue> <attributeValue type="dateTime" name="Last modified" id="4">2014-10- 28T11:39:46Z</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue type="numeric" name="Major version" id="7">1</attributeValue> <attributeValue type="text" name="File path" id="8">/00/3d/61.1-0.1.1.xml</attributeValue> <attributeValue type="text" name="File extension" id="11">xml</attributeValue> <attributeValue type="numeric" name="File size" id="12">1591</attributeValue> <attributeValue type="boolean" name="Is checked out" id="13">>false</attributeValue> <attributeValue type="dateTime" name="Last content update" id="19">2014-10-28T11:39:46Z</attributeValue> </attributeValueList> </assetInfo> </pre>
--	--

getDatadocChannels

Function	Get all datadoc channels.	
Description	This request returns all channels of type 'datadoc' that exist in the system.	
Type	GET	
URL Format	.../datadocchannels	
Java API	Method Name	Class Name
	getDatadocChannels	com.quark.qpp.service.facade Class PublishFacade
Parameters	This request requires no input parameters.	
Returns	Returns a publishingChannelList object containing the metadata of channels of type datadoc.	
Examples	URL:	

	<p><code>http://serverName:serverPort/rest/service/publish/datadocchannels</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo name="Create Data Document From HTTP URL" id="createDataDocAssetFromHttpUrl"> <parameter>CONTENT_URL</parameter> <parameter>XSL_URI</parameter> </publishingChannelInfo> <publishingChannelInfo name="Fetch Excel Data Object" id="fetchExcelDataObject"> <parameter>ASSET_ID</parameter> <parameter>WORKSHEET_NAME</parameter> <parameter>DATA_OBJECT_TYPE</parameter> <parameter>DATA_OBJECT_NAME</parameter> <parameter>OUTPUT_FORMAT</parameter> <parameter>OUTPUT_FORMAT_PROPERTIES</parameter> </publishingChannelInfo> </publishingChannelList></pre>
--	--

Function	Get DataDoc Channels for Desired Content-Type.	
Description	This request returns channels of type DataDoc which are mapped to the given content type.	
Type	GET	
URL Format	.../datadocchannels/{contentTypeIdOrName}	
Java API	Method Name	Class Name
	<code>getDatadocChannels</code>	<code>com.quark.qpp.service.facade Class PublishFacade</code>
Parameters	This request requires no input parameters.	
Returns	Returns a <code>publishingChannelList</code> object containing the metadata of channels of type 'datadoc' mapped to a specific content type.	
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/publish/datadocchannels/{Contenty-Type Id }</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo name="Fetch Excel Data Object" id="fetchExcelDataObject"> <parameter>ASSET_ID</parameter> <parameter>WORKSHEET_NAME</parameter> <parameter>DATA_OBJECT_TYPE</parameter> <parameter>DATA_OBJECT_NAME</parameter> <parameter>OUTPUT_FORMAT</parameter> <parameter>OUTPUT_FORMAT_PROPERTIES</parameter> </publishingChannelInfo> </publishingChannelList></pre>	

fetchDatadocAssets

Function	Fetch Datadoc Assets.
Description	This request returns the metadata of the datadoc assets which had been created by executing the given channel and parameters.
Type	GET

URL Format	...publish/		
Java API	Method Name		Class Name
	fetchDatadocAssets		com.quark.qpp.service.facadeClass PublishFacade
Parameters	Name	Type	Description
	op=fetchdatadocassets	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	datadocchannel={DataDoc Channel Id}	string	The Datadoc channel. (Mandatory)
Returns	Returns the metadata of the Datadoc asset created through the supplied datadoc channel.		
Examples	URL:		
	<pre>http://serverName:serverPort/rest/service/publish? op=fetchdatadocassets&datadocchannel=createDataDocAssetFromHttpUr l</pre>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="61"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">61</attributeValue> <attributeValue type="text" name="Name" id="2">SampleBusDoc827574.xml</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2014-10- 28T11:39:46Z</attributeValue> <attributeValue type="dateTime" name="Last modified" id="4">2014-10- 28T11:39:46Z</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue type="numeric" name="Major version" id="7">1</attributeValue> <attributeValue type="text" name="File path" id="8">/00/3d/61.1-0.1.1.xml</attributeValue> <attributeValue type="text" name="File extension" id="11">xml</attributeValue> <attributeValue type="numeric" name="File size" id="12">1591</attributeValue> <attributeValue type="boolean" name="Is checked out" id="13">>false</attributeValue> <attributeValue type="dateTime" name="Last content update" id="19">2014-10-28T11:39:46Z</attributeValue> </attributeValueList> </assetInfo></pre>		

executeDatadocChannel

Function	Execute Datadoc Channels.	
Description	This request executes the given datadoc channel and writes the resulted datadoc content to the given outputstream.	
Type	GET	
URL Format	...publish/	
Java API	Method Name	Class Name

	<code>executeDatadocChannel</code>	<code>com.quark.qpp.service.facadeClass PublishFacade</code>	
Parameters	Name	Type	Description
	<code>op=executedatadocchannel</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>datadocchannel={DataDoc Channel Id}</code>	string	The Datadoc channel. (Mandatory)
	<code>channelparameters=<XML/></code>	string	Channel parameters.
Returns	Resultant output is returned in the response body stream.		
Examples	URL: <code>http://serverName:serverPort/rest/service/publish?op=executedatadocchannel&datadocchannel=fetchExcelDataObject&...</code>		

updateDatadoc

Function	Update a Datadoc.		
Description	This request updates the existing datadoc asset by creating its new version. Latest content of the asset is fetched by executing the desired datadoc channel.		
Type	POST		
URL Format	<code>...publish/{assetId}</code>		
Java API	Method Name	Class Name	
	<code>updateDatadoc</code>	<code>com.quark.qpp.service.facadeClass PublishFacade</code>	
Parameters	Name	Type	Description
	<code>op=updatedatadoc</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>datadocchannel={DataDoc Channel Id}</code>	string	The Datadoc channel. (Mandatory)
	<code>createMinorVersion={true false}</code>	boolean	Indicates whether you should create a Minor version.
	<code>attributevalues=<XML/></code>	string	See "Request/Response parameter formats."
Returns	Returns the metadata of the successfully updated Datadoc asset.		
Examples	URL: <code>http://serverName:serverPort/rest/service/publish/?op=createdatadoc &datadocchannel=createDataDocAssetFromHttpRequestUrl &CONTENT_URL=http://Server:Port/SampleBusDoc.xml &attributevalues=<?xml version="1.0" encoding="UTF-8"?><attributeValueList> <attributeValue valueId="0" type="text" name="Name" id="2">SampleBusDoc827574.xml </attributeValue> <attributeValue valueId="1" type="popup" name="Routed to" id="25">Admin</attributeValue> <attributeValue valueId="10" type="popup" name="Status" id="24">Status1</attributeValue></code>		

	<pre> <attributeValue valueId="1" type="popup" name="Collection" id="55">Home</attributeValue> <attributeValue valueId="10" type="popup" name="Workflow" id="54">Workflow1</attributeValue> <attributeValue valueId="1104" type="popup" name="Content Type" id="62">Business Document</attributeValue> <attributeValue valueId="0" type="text" name="File extension" id="11">xml</attributeValue> <attributeValue valueId="0" type="text" name="Original filename" id="33">SampleBusDoc827.xml</attributeValue> </attributeValueList> Result: <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="61"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">61</attributeValue> <attributeValue type="text" name="Name" id="2">SampleBusDoc827574.xml</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2014-10- 28T11:39:46Z</attributeValue> <attributeValue type="dateTime" name="Last modified" id="4">2014-10- 28T11:39:46Z</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue type="numeric" name="Major version" id="7">1</attributeValue> <attributeValue type="text" name="File path" id="8">/00/3d/61.1-0.1.1.xml</attributeValue> <attributeValue type="text" name="File extension" id="11">xml</attributeValue> <attributeValue type="numeric" name="File size" id="12">1591</attributeValue> <attributeValue type="boolean" name="Is checked out" id="13">>false</attributeValue> <attributeValue type="dateTime" name="Last content update" id="19">2014-10-28T11:39:46Z</attributeValue> </attributeValueList> </assetInfo> </pre>
--	---

detachDatadocChannel

Function	Detach a DataDoc asset from its source.		
Description	This request detaches or unlinks the given datadoc asset from its source content and creates a new version of the asset. After detaching/unlinking, the given asset will no more be a datadoc asset and thus can be checked-out and checked-in by the end user like other conventional assets in the Platform.		
Type	GET		
URL Format	.../publish/{assetId}/		
Java API	Method Name	Class Name	
	detachDatadocChannel	com.quark.qpp.service.facadeClass PublishFacade	
Parameters	Name	Type	Description
	op=detachdatadoc	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributevalues=<XML/>	string	The Datadoc channel. (Mandatory)

	>		
	createMinorVersion={true false}	boolean	Indicates whether you should create a Minor version.
Returns	Returns the metadata of newly created asset.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/publish?op=detachdatadoc&attributevalues=<xml/>&createminorversion=false</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="61"> <assetVersion minorVersion="0" majorVersion="1"/> <attributeValueList> <attributeValue type="numeric" name="Id" id="1">61</attributeValue> <attributeValue type="text" name="Name" id="2">SampleBusDoc827574.xml</attributeValue> <attributeValue type="dateTime" name="Created" id="3">2014-10- 28T11:39:46Z</attributeValue> <attributeValue type="dateTime" name="Last modified" id="4">2014-10- 28T11:39:46Z</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Creator" id="5">Admin</attributeValue> <attributeValue multiValued="false" valueId="1" type="popup" name="Last modifier" id="6">Admin</attributeValue> <attributeValue type="numeric" name="Major version" id="7">1</attributeValue> <attributeValue type="text" name="File path" id="8">/00/3d/61.1-0.1.1.xml</attributeValue> <attributeValue type="text" name="File extension" id="11">xml</attributeValue> <attributeValue type="numeric" name="File size" id="12">1591</attributeValue> <attributeValue type="boolean" name="Is checked out" id="13">>false</attributeValue> <attributeValue type="dateTime" name="Last content update" id="19">2014-10-28T11:39:46Z</attributeValue> </attributeValueList> </assetInfo></pre>		

Script execution

The script execution capabilities of the API allow application developers to execute Platform scripts and includes the following REST calls:

- [createScript](#)
- [getScriptContent](#)
- [getAllScripts](#)
- [executeScript](#)
- [deleteScript](#)

createScript

Function	Create a script.
Description	This request creates a new script with the given content and name at the platform server. In the script execution information it is mandatory to mention the trigger type and corresponding eventInfo or scheduleInfo depending upon the trigger

	type. Include the file in the request body or as file part.		
Type	POST		
URL Format	.../scripts/{scriptName}		
Java API	Method Name		Class Name
	createScript		com.quark.qpp.service.facade Class ScriptFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)This operation requires parent asset to be in a Checked-Out state.
	language	long	The scripting language
	scriptexecutioninfo =<xml/>	string	The set of attribute values. See "Request/Response parameter formats."
Returns	Returns an <code>scriptInfo</code> object containing metadata of the successfully created script.		
Examples	URL: <code>http://serverName:serverPort/rest/service/scripts?op=create</code>		
	Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <scriptInfo id="12"> <name>abc.js</name> <language>javascript</language> <createdDate>2015-01-06T10:32:31Z</createdDate> <lastModified>2015-01-06T10:32:31Z</lastModified> <lastExecutionLog>NOT_RUN</lastExecutionLog> <creator id="1">Admin</creator> <lastModifiedBy id="1">Admin</lastModifiedBy> <scriptExecutionInfo triggerType="MANUAL"/> </scriptInfo></pre>		

getScriptContent

Function	Get the content of a script.		
Description	This request returns the content of the required script in text format.		
Type	GET		
URL Format	.../scripts/[script ID or name]		
Java API	Method Name		Class Name
	getScriptContent		com.quark.qpp.service.facade Class ScriptFacade
Parameters	This request requires no input parameters.		
Returns	Returns the actual content of the required script in text format.		
Examples	URL: <code>http://serverName:serverPort/rest/service/scripts/QppFunctions</code>		
	<pre>importPackage(Packages.com.quark.qpp.filetransfergateway.service); importPackage(Packages.java.io); importPackage(Packages.com.quark.qpp.core.asset.service.constants);</pre>		

	<pre>importPackage(Packages.com.quark.qpp.core.query.service.dto); importPackage(Packages.com.quark.qpp.core.query.service.constants); importPackage(Packages.com.quark.qpp.core.content.service.constants); importPackage(com.quark.qpp.common.dto); importPackage(com.quark.qpp.common.constants);</pre> <p>Result: The actual content of the required script in text format.</p>
--	--

getAllScripts

Function	Get all scripts.		
Description	This request returns all scripts stored at the platform server.		
Type	GET		
URL Format	.../scripts/		
Java API	Method Name	Class Name	
	getAllScripts	com.quark.qpp.service.facade Class ScriptFacade	
Parameters	This request requires no input parameters.		
Returns	Returns a <code>scriptInfoList</code> object that contains the metadata of all scripts. See the table in the “Request/Response parameter formats” chapter.		
Examples	URL: <code>http://serverName:serverPort/rest/service/scripts</code>		
	Result: <pre><scriptInfoList> <scriptInfo id="11"> <name>GetSavedSearchResults</name> <language>javascript</language> <createdDate>2015-01-06T10:23:37Z</createdDate> <lastModified>2015-01-06T10:23:37Z</lastModified> <lastExecutionLog>NOT_RUN</lastExecutionLog> <creator id="1">Admin</creator> <lastModifiedBy id="1">Admin</lastModifiedBy> <scriptExecutionInfo triggerType="MANUAL"/> </scriptInfo> ... </scriptInfoList></pre>		

executeScript

Function	Execute a script.		
Description	This request executes a specified script that is already saved on the QPP Server.		
Type	POST		
URL Format	... /scripts/[script ID or name]		
Java API	Method Name	Class Name	
	executeScript	com.quark.qpp.service.facade Class ScriptFacade	
Parameters	Name	Type	Description
	op=execute	string	The op parameter allows you to specify which operation is to be performed.

		(Mandatory) This operation executes the complete script.
Returns	The response depends upon the output of the script.	
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/scripts/GetSavedSearchResults?op=execute</pre> <p>Result:</p> <pre>Total No Of Assets : 4 Name: Tulips Collection: Home Workflow: Workflow1 Status: Status1 Routed To: Admin Name: Duplicate_asset Collection: Home Workflow: Workflow1 Status: Status1 Routed To: Admin Name: 1quark2.jpg Collection: Home Workflow: Workflow1 Status: Status1 Routed To: Admin Name: 15116.qxp Collection: Home Workflow: Workflow1 Status: Status1 Routed To: Admin</pre>	

Function	Execute a script method.		
Description	This request executes a specified method of a specified script that is already saved on the QPP Server.		
Type	POST		
URL Format	.../scripts/[script ID or name]		
Java API	Method Name		Class Name
	executeScript		com.quark.qpp.service.facade Class ScriptFacade
Parameters	Name	Type	Description
	op=execute&	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)This operation executes given the method of the script along with the given parameters.
	methodname=[method name] &	string	Specify the method name to be executed. (Mandatory)
	parameters=x,y,z	long or string	Specify the method's parameters if required. (Optional)
Returns	The response depends upon the output of the script.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/scripts/Sample?op=execute&methodname=abc</pre> <p>Result:</p> <p>The response depends upon the output of the script.</p>		

deleteScript

Function	Delete a script.
Description	This request deletes the specified script.

Type	POST		
URL Format	.../scripts/{scriptName}		
Java API	Method Name		Class Name
	deleteScript		com.quark.qpp.service.facade Class ScriptFacadeClass AssetFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory) This request will lock the asset internally and delete it from system.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/scripts/12?op=delete		

Archive and Restore management

The archive and restore management capabilities of the API allow application developers to archive assets (metadata and its high_res) at the collection/query level and provides extensibility to restore/import the same assets into a different QPP server environment. This includes the following REST calls:

- [archiveCollectionAssets](#)
- [archiveQueryAssets](#)
- [archiveAssets](#)
- [restoreAssets](#)

archiveCollectionAssets

Function	Archive collection assets.		
Description	This request archives all assets belonging to a particular collection, from the given collection ID or complete collection path.		
Type	GET		
URL Format	../archive/collectionassets?..		
Java API	Method Name		Class Name
	archiveCollectionAssets		com.quark.qpp.service.facade Class ArchiveFacade
Parameters	Name	Type	Description
	collection	string	Name/ID of the collection. (Mandatory)
	destinationfolder	string	The path. Any file path that is accessible to the server, it could be on a network. (Mandatory)
	allversions=[true	boolean	Defaults to true. (Optional) If true, the

	<code>false]</code>		server will archive all available versions of all assets.
	<code>archiverelations=[true false]</code>	boolean	Defaults to true. (Optional) If true, the server will maintain all the relations of assets to be archived.
	<code>deleteafterarchive=[true false]</code>	boolean	Defaults to false. (Optional) If true, the server will delete all the assets after archiving it.
Returns	Returns an <code>archiveRestoreInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/archive/collectionassets?collection=Home&destinationfolder=C:\QPP Archive</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <archiveRestoreInfo failureCount="0" successCount="50"> <summary> Archive Summary Log Date: 5/01/13 17:17 PM Archive location: C:\QPP Archive Total assets to be archived: 50 Number of assets successfully archived: 50 Number of assets failed to archive: 0 Per Version Listing of Assets _____ _____ - Asset Versions Successfully Archived: _____ - Home/SampleBusDocdc1.qxp, Version: 1.0 Home/Construct/Construct - Copy.zip, Version: 1.0 Home/Construct/Construct.zip, Version: 1.0 Home/Construct/Construct1/Construct.rar, Version: 1.0 Home/Construct/Construct1/dineshjj.xml, Version: 1.0 </summary> </archiveRestoreInfo></pre>		

archiveQueryAssets

Function	Archive resulted assets of any given saved search.		
Description	This request archives all assets returned in the query result for the given query name.		
Type	GET		
URL Format	<code>../archive/searchresults?..</code>		
Java API	Method Name	Class Name	
	<code>archiveQueryAssets</code>	<code>com.quark.qpp.service.facade Class ArchiveFacade</code>	
Parameters	Name	Type	Description
	<code>searchname</code>	string	Name of the saved search. (Mandatory)
	<code>destinationfolder</code>	string	The path. Any file path that is accessible to server, it could be on network. (Mandatory)
	<code>allversions=[true false]</code>	boolean	Defaults to true.(Optional)If true, the server will archive all available versions of all assets.
	<code>archiverelations=[true false]</code>	boolean	Defaults to true.(Optional)If true, the server will maintain all the relations of

			assets to be archived.
	<code>deleteafterarchive=[true false]</code>	boolean	Defaults to false.(Optional)If true,the server will delete all the assets after archiving it.
Returns	Returns an <code>archiveRestoreInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/archive/searchresults?searchname=Default&destinationfolder=C:\QPP Archive</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <archiveRestoreInfo failureCount="0" successCount="50"> <summary> Archive Summary Log Date: 5/01/13 17:17 PM Archive location: C:\QPP Archive Total assets to be archived: 50 Number of assets successfully archived: 50 Number of assets failed to archive: 0 Per Version Listing of Assets _____ _____-- Asset Versions Successfully Archived: _____-- Home/SampleBusDocdcl.qxp, Version: 1.0 Home/Construct/Construct - Copy.zip, Version: 1.0 Home/Construct/Construct.zip, Version: 1.0 Home/Construct/Construct1/Construct.rar, Version: 1.0 Home/Construct/Construct1/dineshjj.xml, Version: 1.0 </summary> </archiveRestoreInfo></pre>		

archiveAssets

Function	Archive specified assets.		
Description	This request archives an asset from the given asset id or asset path.		
Type	GET		
URL Format	<code>../archive/assets?..</code>		
Java API	Method Name	Class Name	
	<code>archiveAssets</code>	<code>com.quark.qpp.service.facade Class ArchiveFacade</code>	
Parameters	Name	Type	Description
	<code>assets</code>	long or string	The IDs or Collection Path of assets to be archived. (Mandatory) Comma separated IDs of assets (long), or a collection path of assets (string).
	<code>destinationfolder</code>	string	The path. Any file path that is accessible to server, it could be on network. (Mandatory)
	<code>allversions=[true false]</code>	boolean	Defaults to true.(Optional)If true, the server will archive all available versions of all assets.
	<code>archiverelations=[true false]</code>	boolean	Defaults to true.(Optional)If true, the server will maintain all the relations of assets to be archived.
	<code>deleteafterarchive=[t</code>	boolean	Defaults to false.(Optional)If true, the

	<code>true false]</code>		server will delete all the assets after archiving it.
Returns	Returns an <code>archiveRestoreInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/archive/assets?assets=1102,1229&destinationfolder=C:\QPP Archive</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <archiveRestoreInfo failureCount="0" successCount="50"> <summary> Archive Summary Log Date: 5/01/13 17:17 PM Archive location: C:\QPP Archive Total assets to be archived: 50 Number of assets successfully archived: 50 Number of assets failed to archive: 0 Per Version Listing of Assets _____ _____-- Asset Versions Successfully Archived: _____-- Home/SampleBusDocdcl.qxp, Version: 1.0 Home/Construct/Construct - Copy.zip, Version: 1.0 Home/Construct/Construct.zip, Version: 1.0 Home/Construct/Construct1/Construct.rar, Version: 1.0 Home/Construct/Construct1/dineshjj.xml, Version: 1.0 </summary> </archiveRestoreInfo></pre>		

restoreAssets

Function	Restore archived assets into the system.		
Description	This request restores assets from the given folder path. This request requires the same collection hierarchy with same set of workflow, users etc which existed at the time of archiving the assets.		
Type	GET		
URL Format	<code>../restore?...</code>		
Java API	Method Name		Class Name
	<code>restoreAssets</code>		<code>com.quark.qpp.service.facade Class RestoreFacade</code>
Parameters	Name	Type	Description
	<code>archivedpath</code>	string	The full path of the folder which contains the archived assets and its information. (Mandatory)
	<code>lookup</code>	string	Specify the lookup strategy to be used for fetching entities, like content type, relation type and attributes. Use name to restore across different servers, else use id. (Optional, Default value is id)
Returns	Returns an <code>archiveRestoreInfo</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/restore?archivedpath=C:\QPP Archive</pre> <p>Result:</p>		

	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <archiveRestoreInfo failureCount="0" successCount="50"> <summary> Archive Summary Log Date: 5/01/13 17:17 PM Archive location: C:\QPP Archive Total assets to be archived: 50 Number of assets successfully archived: 50 Number of assets failed to archive: 0 Per Version Listing of Assets ----- ----- Asset Versions Successfully Archived: ----- Home/SampleBusDocdcl.qxp, Version: 1.0 Home/Construct/Construct - Copy.zip, Version: 1.0 Home/Construct/Construct.zip, Version: 1.0 Home/Construct/Construct1/Construct.rar, Version: 1.0 Home/Construct/Construct1/dineshjj.xml, Version: 1.0 </summary> </archiveRestoreInfo></pre>
--	---

Office

The Office capabilities of the API allow application developers to fetch metadata from Excel worksheets, fetch information about Excel Data objects (like named range, dynamic range, tables and chartes) defined in a worksheet and get and transform Excel Data. The Office capabilities includes the following REST calls:

- [*getExcelChart*](#)
- [*getExcelChartBasedOnStream*](#)
- [*getExcelChartBasedOnURI*](#)
- [*getExcelTable*](#)
- [*getExcelTableBasedOnStream*](#)
- [*getExcelTableBasedOnURI*](#)
- [*getExcelWorksheets*](#)
- [*getExcelWorksheetsBasedOnURI*](#)
- [*getExcelWorksheetsBasedOnStreamContext*](#)
- [*getChart*](#)
- [*getChartBasedOnURI*](#)
- [*getChartBasedOnStream*](#)
- [*getTable*](#)
- [*getTableBasedOnURI*](#)
- [*getTableBasedOnStreamContext*](#)
- [*getNamedRange*](#)
- [*getNamedRangeBasedOnURI*](#)
- [*getNamedRangeBasedOnStream*](#)
- [*getDynamicRange*](#)
- [*getDynamicRangeBasedOnURI*](#)
- [*getDynamicRangeBasedOnStream*](#)

- *getVisioPagesInfo*
- *getVisioPagesInfoBasedOnURI*
- *getVisioPagesInfoBasedOnStreamContext*
- *getVisioPageOutput*
- *getVisioPageOutputBasedOnURI*
- *getVisioPageOutputBasedOnStreamContext*
- *getPowerPointSlidesInfo*
- *getPowerPointSlidesInfoBasedOnURI*
- *getPowerPointSlidesInfoBasedOnStreamContext*
- *getpowerPointSlideOutput*
- *getpowerPointSlideOutputBasedOnURI*
- *getpowerPointSlideOutputBasedOnStreamContext*
- *getPowerPointSlideRangeOutput*
- *getPowerPointSlideRangeOutputBasedOnURI*
- *getPowerPointSlideRangeOutputBasedOnStreamCongtext*

getExcelChart

Function	Get a Microsoft Excel chart based on asset Id		
Description	This request returns a Microsoft Excel Chart in the desired image format. Default image format is PNG.		
Type	GET		
URL Format	...office/excelchart/{assetId}		
Java API	Method Name		Class Name
	getExcelChart		com.quark.qpp.service.facade.OfficeFacade
Parameters	Name	Type	Description
	assetId	string	The AssetID.
	majorversion=[x]	string	Major version of the asset. (Optional)
	minorversion=[x]	string	Minor version of the asset. (Optional)
	outputformat=[image, json, html]	string	Image output format. (Optional)
	contentdisposition	string	Content disposition value to be set on the http response headers.
	outputStream	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required format based on the given parameters.		
Examples	URL: http://server:port/rest/service/office/excelchart/48?outputformat		

	<pre>=image&imageformat=png</pre> <p>Results:</p> <p>A filestream of required output format based on given parameters.</p>
--	---

Function	Get a Microsoft Excel chart by path		
Description	This request returns a Microsoft Excel Chart in the desired image format. Default image format is PNG.		
Type	GET		
URL Format	<code>...office/excelchartbypath/{assetPath}</code>		
Java API	Method Name		Class Name
	<code>getExcelChart</code>		<code>com.quark.qpp.service.facade.OfficeByPathFacade</code>
Parameters	Name	Type	Description
	<code>assetPath</code>	string	The AssetPath.
	<code>majorversion=[x]</code>	string	Major version of the asset. (Optional)
	<code>minorversion=[x]</code>	string	Minor version of the asset. (Optional)
	<code>outputformat=[image, json, html]</code>	string	Image output format. (Optional)
	<code>contentdisposition</code>	string	Content disposition value to be set on the http response headers.
	<code>outputStream</code>	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required format based on the given parameters.		
Examples	URL:		
	<code>http://server:port/rest/service/office/excelchartbypath/Home/Chart1.xlsx?outputformat=image&imageformat=png</code>		
	Results:		
	A filestream of required output format based on given parameters.		

getExcelChartBasedOnStream

Function	Get a Microsoft Excel chart based on stream		
Description	This request returns a Microsoft Excel Chart in the desired image format. Default image format is PNG.		
Type	POST		
URL Format	<code>...office/excelchart</code>		
Java API	Method Name		Class Name
	<code>getExcelChartBasedOnStream</code>		<code>com.quark.qpp.service.facade.OfficeFacade</code>
Parameters	Name	Type	Description
	<code>inputStream</code>	string	The stream of the input file. (Mandatory)

	<code>outputformat=[image, json, html]</code>	string	Image output format. (Optional)
	<code>contentdisposition</code>	string	Content disposition value to be set on the http response headers.
	<code>outputStream</code>	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required format based on the given parameters.		
Examples	<p>URL:</p> <pre>http://server:port/rest/service/office/excelchart?outputformat=image&imageformat=png</pre> <p>Results:</p> <p>A filestream of required output format based on given parameters.</p>		

getExcelChartBasedOnURI

Function	Get a Microsoft Excel chart based on URI		
Description	This request returns a Microsoft Excel Chart in the desired image format. Default image format is PNG.		
Type	GET		
URL Format	<code>...office/excelchart</code>		
Java API	Method Name		Class Name
	<code>getExcelChartBasedOnURI</code>		<code>com.quark.qpp.service.facade.OfficeByPathFacade</code>
Parameters	Name	Type	Description
	<code>uri</code>	string	The URI of the Microsoft Excel chart
	<code>outputformat=[image, json, html]</code>	string	Image output format. (Optional)
	<code>contentdisposition</code>	string	Content disposition value to be set on the http response headers.
	<code>outputStream</code>	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required format based on the given parameters.		
Examples	<p>URL:</p> <pre>http://server:port/rest/service/office/excelchart?uri=qpp://assetsbypath/Home/Chart1.xlsx&outputformat=image&imageformat=png</pre> <p>Results:</p> <p>A filestream of required output format based on given parameters.</p>		

getExcelTable

Function	Get a Microsoft Excel table based on asset Id		
Description	This request returns a Microsoft Excel table in the desired output format. Default output format is Smart Table.		
Type	GET		
URL Format	...office/exceltable/{assetId}		
Java API	Method Name		Class Name
	getExcelTable		com.quark.qpp.service.facade.OfficeFacade
Parameters	Name	Type	Description
	assetId	string	The AssetID.
	majorversion=[x]	string	Major version of the asset. (Optional)
	minorversion=[x]	string	Minor version of the asset. (Optional)
	outputformat	string	Output format. (Optional)
	contentdisposition	string	Content disposition value to be set on the http response headers.
outputStream	string	Output stream onto which output is to be written (for local API invocatins).	
Returns	Returns a stream of the required output format based on the given parameters.		
Examples	URL:		
	http://server:port/rest/service/office/exceltable/48?outputformat=image&imageformat=png		
Examples	Results:		
	A filestream of required output format based on given parameters.		

Function	Get a Microsoft Excel table by path		
Description	This request returns a Microsoft Excel table in the desired output format. Default output format is Smart Table.		
Type	GET		
URL Format	...office/exceltablebypath/{assetPath}		
Java API	Method Name		Class Name
	getExcelTable		com.quark.qpp.service.facade.OfficeByPathFacade
Parameters	Name	Type	Description
	assetPath	string	The AssetPath.
	majorversion=[x]	string	Major version of the asset. (Optional)
	minorversion=[x]	string	Minor version of the asset. (Optional)
	outputformat	string	Output format. (Optional)
contentdisposition	string	Content disposition value to be set on the http response headers.	

	<code>outputStream</code>	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required output format based on the given parameters.		
Examples	<p>URL:</p> <pre>http://server:port/rest/service/office/exceltablebypath/Home/BudgetDetails.xlsx?outputformat=image&imageformat=png</pre> <p>Results:</p> <p>A filestream of required output format based on given parameters.</p>		

getExcelTableBasedOnStream

Function	Get a Microsoft Excel chtable art based on stream		
Description	This request returns a Microsoft Excel table in the desired output format. Default image format is PNG.		
Type	POST		
URL Format	<code>...office/exceltable</code>		
Java API	Method Name		Class Name
	<code>getExcelTableBasedOnStream</code>		<code>com.quark.qpp.service.facade.OfficeByPathFacade</code>
Parameters	Name	Type	Description
	<code>inputStream</code>	string	The stream of the input file. (Mandatory)
	<code>outputformat</code>	string	Output format. (Optional)
	<code>contentdisposition</code>	string	Content disposition value to be set on the http response headers.
	<code>outputStream</code>	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required output format based on the given parameters.		
Examples	<p>URL</p> <pre>http://server:port/rest/service/office/exceltable?outputformat=image&imageformat=png</pre> <p>Results:</p> <p>A filestream of required output format based on given parameters.</p>		

getExcelTableBasedOnURI

Function	Get a Microsoft Excel table based on URI		
Description	This request returns a Microsoft Excel table in the desired image format. Default output format is Smart Table.		
Type	GET		

URL Format	<code>...office/exceltable</code>		
Java API	Method Name		Class Name
	<code>getExcelTableBasedOnURI</code>		<code>com.quark.qpp.service.facade.OfficeByPathFacade</code>
Parameters	Name	Type	Description
	<code>uri</code>	string	The URI of the Microsoft Exceltable
	<code>outputformat</code>	string	Output format. (Optional)
	<code>contentdisposition</code>	string	Content disposition value to be set on the http response headers.
	<code>outputStream</code>	string	Output stream onto which output is to be written (for local API invocatins).
Returns	Returns a stream of the required output format based on the given parameters.		
Examples	URL:		
	<code>http://server:port/rest/service/office/exceltable?uri=qpp://asset sbypath/Home/BudgetDetails.xlsx&outputformat=image&imageformat=png</code>		
	Results:		
	A filestream of required output format based on given parameters.		

getExcelWorksheets

Function	Get all worksheets in Excel document.		
Description	This request returns all worksheets defined in an Excel document.		
Type	GET		
URL Format	<code>..office/excel/{assetId} OR ../ office/ excelbypath /{ coll name1}/{ coll name 2}/../{asset name}</code>		
Java API	Method Name		Class Name
	<code>getExcelWorksheets</code>		<code>com.quark.qpp.service.facadeClass OfficeFacade</code>
Parameters	Name	Type	Description
	<code>majorversion=?</code>	string	Major version to be fetched. (Optional)
	<code>minorversion=?</code>	string	Minor version to be fetched. (Optional)
	<code>worksheets=?</code>	string	Comma separated worksheet names of specific worksheets to be fetched. (Optional)
Returns	Returns all worksheets or required worksheets defined in an Excel document. Each worksheet encapsulates its metadata (such as worksheet index and name.) Each worksheet also provides information about tables, named ranges and charts defined in the worksheet.		
Examples	URL:		
	<code>http://serverName:serverPort/rest/service/office/excel/10</code>		

Result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<worksheetInfoList>
  <worksheetInfo index="0">
    <name>Products</name>
    <chartInfoList>
      <chartInfo>
        <name>Chart 1</name>
      </chartInfo>
    </chartInfoList>
    <namedRangeInfoList>
      <namedRangeInfo>
        <name>Description</name>
      </namedRangeInfo>
      <namedRangeInfo>
        <name>DispatchRange</name>
      </namedRangeInfo>
    </namedRangeInfoList>
  </worksheetInfo>
  <worksheetInfo index="1">
    <name>Sales</name>
    <chartInfoList>
      <chartInfo>
        <name>Chart 1</name>
      </chartInfo>
    </chartInfoList>
    <tableInfoList>
      <tableInfo>
        <name>SalesTable</name>
      </tableInfo>
    </tableInfoList>
  </worksheetInfo>
</worksheetInfoList>
```

URL:

```
http://servername:serverport/rest/service/office/excel/10?
worksheets=sales,sheet3
```

Result:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<worksheetInfoList>
  <worksheetInfo index="1">
    <name>Sales</name>
    <chartInfoList>
      <chartInfo>
        <name>Chart 1</name>
      </chartInfo>
    </chartInfoList>
    <tableInfoList>
      <tableInfo>
        <name>SalesTable</name>
      </tableInfo>
    </tableInfoList>
  </worksheetInfo>
  <worksheetInfo index="2">
    <name>Sheet3</name>
  </worksheetInfo>
</worksheetInfoList>
```

getExcelWorksheetsBasedOnURI

Function	Get all worksheets based on URI of the Excel document.		
Description	This request returns all worksheets defined in an Excel document.		
Type	GET		
URL Format	...office/excel		
Java API	Method Name		Class Name
	getExcelWorksheetsBasedOnURI		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	URI	string	Excel document uri.
Parameters	worksheets	string	Comma separated worksheet names if specific worksheets are to be fetched. (Optional)
	Returns		
Returns all worksheets or required worksheets defined in an Excel document. Each worksheet encapsulates its metadata (such as worksheet index and name. Each worksheet also provides information about tables, named ranges and charts defined in the worksheet.			
Examples	URL:		
	http://serverName:serverPort/rest/service/office/excel? uri=file:C:\DATA\Documents\MyExcelSamples\DocSample.xlsx		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <worksheetInfoList > <worksheetInfo index="0"> <name>Products</name> <chartInfoList> <chartInfo> <name>Chart 1</name> </chartInfo> </chartInfoList> <namedRangeInfoList> <namedRangeInfo> <name>Description</name> </namedRangeInfo> <namedRangeInfo> <name>DispatchRange</name> </namedRangeInfo> </namedRangeInfoList> </worksheetInfo> <worksheetInfo index="1"> <name>Sales</name> <tableInfoList> <tableInfo> <name>SalesTable</name> </tableInfo> </tableInfoList> </worksheetInfo> </worksheetInfoList></pre>		

getExcelWorksheetsBasedOnStreamContext

Function	Get all worksheets defined in an Excel document based on stream context.		
Description	This request returns all worksheets defined in an Excel document.		
Type	GET		
URL Format	...office/excel		
Java API	Method Name		Class Name
	getExcelWorksheetsBasedOnStreamContext		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	stream	string	Excel document in request.
Parameters	worksheets	string	Comma separated worksheet names if

			specific worksheets are to be fetched. (Optional)
Returns	Returns all worksheets or required worksheets defined in an Excel document. Each worksheet encapsulates its metadata (such as worksheet index and name). Each worksheet also provides information about tables, named ranges and charts defined a worksheet.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/office/excel Stream the excel document as input.</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <worksheetInfoList > <worksheetInfo index="0"> <name>Products</name> <chartInfoList> <chartInfo> <name>Chart 1</name> </chartInfo> </chartInfoList> <namedRangeInfoList> <namedRangeInfo> <name>Description</name> </namedRangeInfo> <namedRangeInfo> <name>DispatchRange</name> </namedRangeInfo> </namedRangeInfoList> </worksheetInfo> <worksheetInfo index="1"> <name>Sales</name> <tableInfoList> <tableInfo> <name>SalesTable</name> </tableInfo> </tableInfoList> </worksheetInfo> </worksheetInfoList></pre>		

getChart

Function	Get an Excel chart in image format.		
Description	This request returns an Excel chart in image format.		
Type	GET		
URL Format	..office/excel/{assetId} OR .../ office/ excelbypath /{ coll name1}/{ coll name 2}/../{asset name}		
Java API	Method Name		Class Name
	getChart		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	majorversion=?	string	Major version of the asset. (Optional)
	minorversion=?	string	Minor version of the asset. (Optional)
	chart	string	Name of the Excel chart to be retrieved.
	worksheet	string	The name of the worksheet where the chart is defined. By default the first chart in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	horizontalresolution	string	Image output horizontal resolution. (Optional)
	verticalresolution	string	Image output vetical resolution. (Optional)
imageormat	string	Image output format. By default the image is returned in png format.	

			(Optional)
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to “attachment” if downloading of chart is required. (Optional)
Returns	Returns an Excel chart in the image format.		
Examples	URL: http://serverName:serverPortt/rest/service/office/excel/36? chart=Chart1&horizontalresolution=100&verticalresolution=100&imageformat=png Result:		

getChartBasedOnURI

Function	Get Excel chart based on the URI of the Excel document.		
Description	This request returns an Excel chart in the image format.		
Type	GET		
URL Format	...office/excel		
Java API	Method Name		Class Name
	<code>getChartBasedOnURI</code>		<code>com.quark.qpp.service.facadeClass OfficeFacade</code>
Parameters	Name	Type	Description
	<code>uri</code>	string	URI of the Excel document.
	<code>chart</code>	string	Name of the Excel chart to be retrieved.
	<code>worksheet</code>	string	The name of the worksheet where the chart is defined. By default the first chart in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>horizontalresolution</code>	string	Image output horizontal resolution. (Optional)
	<code>verticalresolution</code>	string	Image output vetical resolution. (Optional)
	<code>imageormat</code>	string	Image output format. By default the image is returned in png format. (Optional)
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to “attachment” if downloading of chart is required. (Optional)
Returns	Returns an Excel chart in the image format.		
Examples	URL: http://serverName:serverPort/rest/service/office/excel?uri=		

<pre>classpath:SampleExcelFile.xlsx&worksheet=Products&chart=chart1&imageformat=png</pre> <p>Result:</p>
--

getChartBasedOnStream

Function	Get a chart based on the input stream of an Excel document.		
Description	This request returns an Excel chart in the image format. Include the file in the request body or as file part.		
Type	GET		
URL Format	...office/excel		
Java API	Method Name	Class Name	
	<code>getChartBasedOnStream</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>chart</code>	string	Name of the Excel chart to be retrieved.
	<code>worksheet</code>	string	The name of the worksheet where the chart is defined. By default the first chart in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>horizontalresolution</code>	string	Image output horizontal resolution. (Optional)
	<code>verticalresolution</code>	string	Image output vertical resolution. (Optional)
	<code>imageformat</code>	string	Image output format. By default the image is returned in png format. (Optional)
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of chart is required. (Optional)
Returns	Returns an Excel chart in the image format.		
Examples	<p>URL:</p> <pre>http://serverName:serverPort/rest/service/office/excel?worksheet=Products&chart=chart1&imageformat=png</pre> <p>Result:</p>		

getTable

Function	Get an Excel table.
Description	This request returns an Excel table in the desired output format.

Type	GET		
URL Format	..office/excel/[assetId] OR .../ office/ excelbypath /{ coll name1}/{ coll name 2}/../{asset name}		
Java API	Method Name		Class Name
	getTable		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	majorversion=?	string	Major version to be retrieved. (Optional)
	minorversion=?	string	Minor version to be retrieved. (Optional)
	table	string	Name of the table to be retrieved. (Optional)
	worksheets=?	string	The name of the worksheet where the table is defined. By default the first table in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	outputformat	string	Value of the output format in which the table is to be rendered. Possible values are defined in class com.quark.qpp.office.service.constants.ExcelDataOutputFormats . Default value is "smarttable" .
contentDisposition	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of table is required. (Optional)	
Returns	Returns an Excel table in the desired output format.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/office/excel/10? worksheet=Products&table=SalesDetails Result:</p> <pre><?xml version="1.0" encoding="utf-8"?> <table xmlns="http://quark.com/smartcontent/1.0" type=" smarttable "> <tgroup cols="2"> < colspec colnum="1" colname="1" colwidth="64px"/> < colspec colnum="2" colname="2" colwidth="131px"/> <tbody > <row> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Name</t> </p> </entry> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Sales (in millions)</t> </p> </entry> </row> </tbody > </ tgroup > </table></pre>		

getTableBasedOnURI

Function	Get an Excel table based on an Excel document URI.	
Description	This request returns an Excel table in the desired output format.	
Type	GET	
URL Format	...office/excel	
Java API	Method Name	Class Name

	<code>getTableBasedOnURI</code>		<code>com.quark.qpp.service.facadeClass OfficeFacade</code>
Parameters	Name	Type	Description
	<code>uri</code>	string	URI of the Excel document.
	<code>table</code>	string	Name of the table to be retrieved. (Optional)
	<code>worksheets=?</code>	string	The name of the worksheet where the chart is defined. By default the first chart in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the table is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> . Default value is "smarttable".
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of chart is required. (Optional)
Returns	Returns an Excel table in the desired output format.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel? uri = classpath:SampleExcelFile.xlsx&table = SalesDetails</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="utf-8"?> <table xmlns="http://quark.com/smartcontent/1.0" type="smarttable"> <tgroup cols="2"> <colspec colnum="1" colname="1" colwidth="64px"/> <colspec colnum="2" colname="2" colwidth="131px"/> <tbody> <row> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Name</t> </p> </entry> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Sales (in millions)</t> </p> </entry> </row> </tbody> </tgroup> </table></pre>		

getTableBasedOnStreamContext

Function	Get an Excel table based on stream context.		
Description	This request returns an Excel table in the desired output format. Include the file in the request body or as file part.		
Type	GET		
URL Format	<code>...office/excel</code>		
Java API	Method Name	Class Name	
	<code>getTableBasedOnStreamContext</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description

	<code>table</code>	string	Name of the table to be retrieved. (Optional)
	<code>worksheets=?</code>	string	The name of the worksheet where the chart is defined. By default the first chart in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the table is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> . Default value is "smarttable".
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of chart is required. (Optional)
Returns	Returns an Excel table in the desired output format.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel ? worksheet= Products&table = SalesInfo & stream the Excel file in input</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="utf-8"?> <table xmlns="http://quark.com/smartcontent/1.0" type="smarttable"> <tgroup cols="2"> <colspec colnum="1" colname="1" colwidth="64px"/> <colspec colnum="2" colname="2" colwidth="131px"/> <tbody> <row> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Name</t> </p> </entry> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Sales (in millions)</t> </p> </entry> </row> </tbody> </tgroup> </table></pre>		

getNamedRange

Function	Get an Excel Named Range.		
Description	This request returns an Excel named range in the desired output format.		
Type	GET		
URL Format	<code>..office/excel/{assetId} OR .../ office/ excelbypath /{ coll name1}/{ coll name 2}/../{asset name}</code>		
Java API	Method Name	Class Name	
	<code>getNamedRange</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>majorversion=?</code>	string	Major version to be retrieved. (Optional)
	<code>minorversion=?</code>	string	Minor version to be retrieved. (Optional)

	<code>namedrange</code>	string	Name of the Excel named range to be retrieved.
	<code>worksheets=?</code>	string	The name of the worksheet where the named ranged is defined. By default the first named ranged in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the named range is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> .
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of named range is required. (Optional)
Returns	Returns an Excel named range in the desired output format.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel/10 ? worksheet= Products& namedrange = DispatchRange</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <table xmlns="http://quark.com/ smartcontent /1.0" type=" smarttable "> < tgroup cols="1"> < colspec colnum ="1" colname ="1" colwidth ="83x"/> < tbody > <row> <entry align="right" valign ="bottom" colsep ="0" rowsep ="0"> <p> <t>10</t> </p> </entry> </row> </ tbody > </ tgroup > </table></pre>		

getNamedRangeBasedOnURI

Function	Get an Excel Named Range based on the URI of the Excel document.		
Description	This request returns an Excel named range in the desired output format.		
Type	GET		
URL Format	<code>...office/excel</code>		
Java API	Method Name	Class Name	
	<code>getNamedRangeBasedOnURI</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>uri</code>	string	Excel document URI.
	<code>namedrange</code>	string	Name of the Excel named range to be retrieved.
	<code>worksheets=?</code>	string	The name of the worksheet where the named ranged is defined. By default the first named ranged in Excel with the given name is returned if the worksheet is not

			mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the named range is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> .
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of named range is required. (Optional)
Returns	Returns an Excel named range in the desired output format.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel? uri = classpath:SampleExcelFile.xlsx& namedrange = DispatchRange</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <table xmlns="http://quark.com/ smartcontent /1.0" type=" smarttable "> < tgroup cols="1"> < colspec colnum ="1" colname ="1" colwidth ="83x"/> < tbody > <row> <entry align="right" valign ="bottom" colsep ="0" rowsep ="0"> <p> <t>10</t> </p> </entry> </row> </ tbody > </ tgroup > </table></pre>		

getNamedRangeBasedOnStream

Function	Get an Excel Named Range based on stream context.		
Description	This request returns an Excel named range in the desired output format. Include the file in the request body or as file part.		
Type	GET		
URL Format	<code>...office/excel</code>		
Java API	Method Name	Class Name	
	<code>getNamedRangeBasedOnStream</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>namedrange</code>	string	Name of the Excel named range to be retrieved.
	<code>worksheets=?</code>	string	The name of the worksheet where the named ranged is defined. By default the first named ranged in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the named range is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> .

	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to “attachment” if downloading of named range is required. (Optional)
Returns	Returns an Excel named range in the desired output format.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel? worksheet=Sales&namedrange = DispatchRange & stream the Excel file in input</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <table xmlns="http://quark.com/ smartcontent /1.0" type=" smarttable "> < tgroup cols="1"> < colspec colnum ="1" colname ="1" colwidth ="83x"/> < tbody > <row> <entry align="right" valign="bottom" colsep ="0" rowsep ="0"> <p> <t>10</t> </p> </entry> </row> </ tbody > </ tgroup > </table></pre>		

getDynamicRange

Function	Get an Excel Dynamic Range.		
Description	This request r eturns an Excel dynamic range in the desired output format.		
Type	GET		
URL Format	<code>..office/excel/{assetId} OR .../ office/ excelbypath /{ coll name1}/{ coll name 2}/../{asset name}d]</code>		
Java API	Method Name	Class Name	
	<code>getDynamicRange</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>majorversion=?</code>	string	Major version to be retrieved. (Optional)
	<code>minorversion=?</code>	string	Minor version to be retrieved. (Optional)
	<code>dynamicrange</code>	string	Name of the Excel dynamic range to be retrieved.
	<code>worksheets=?</code>	string	The name of the worksheet where the dynamic ranged is defined. By default the first dynamic ranged in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the dynamic range is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> .
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to “attachment” if downloading of dynamic range is required. (Optional)

Returns	Returns an Excel dynamic range in the desired output format.
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel/1?worksheet=Sales&dynamicrange=A2:C25</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <table xmlns="http://quark.com/smartcontent/1.0" type="smarttable"> <tgroup cols="1"> <colspec colnum="1" colname="1" colwidth="104px"/> <tbody> <row> <entry colsep="0" rowsep="0"> <p> </p> </entry> </row> <row> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Product details</t> </p> </entry> </row> </tbody> </tgroup> </table></pre>

getDynamicRangeBasedOnURI

Function	Get an Excel Dynamic Range based on the URI of the Excel document.		
Description	This request returns an Excel dynamic range in the desired output format.		
Type	GET		
URL Format	<code>...office/excel</code>		
Java API	Method Name		Class Name
	<code>getDynamicRangeBasedOnURI</code>		<code>com.quark.qpp.service.facadeClass OfficeFacade</code>
Parameters	Name	Type	Description
	<code>uri</code>	string	Excel document URI.
	<code>dynamicrange</code>	string	Name of the Excel dynamic range to be retrieved.
	<code>worksheets=?</code>	string	The name of the worksheet where the dynamic ranged is defined. By default the first dynamic ranged in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	<code>outputformat</code>	string	Value of the output format in which the dynamic range is to be rendered. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code> .
	<code>contentDisposition</code>	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of named range is required. (Optional)
Returns	Returns an Excel dynamic range in the desired output format.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/excel? uri = classpath:SampleExcelFile.xlsx&worksheet = Sales&dynamicrange =A2:C25</code></p>		

<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <table xmlns="http://quark.com/ smartcontent /1.0" type=" smarttable "> < tgroup cols="1"> < colspec colnum="1" colname="1" colwidth="104px"/> < tbody > <row> <entry colsep="0" rowsep="0"> <p> </p> </entry> </row> <row> <entry valign="bottom" colsep="0" rowsep="0"> <p> <t>Product details</t> </p> </entry> </row> </ tbody > </ tgroup > </table></pre>

getDynamicRangeBasedOnStream

Function	Get an Excel Dynamic Range based on stream context.		
Description	This request returns an Excel dynamic range in the desired output format. Include the file in the request body or as file part.		
Type	GET		
URL Format	...office/excel		
Java API	Method Name		Class Name
	getDynamicRangeBasedOnStream		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	dynamicrange	string	Name of the Excel dynamic range to be retrieved.
	worksheets=?	string	The name of the worksheet where the dynamic ranged is defined. By default the first named ranged in Excel with the given name is returned if the worksheet is not mentioned. (Optional)
	outputformat	string	Value of the output format in which the dynamic range is to be rendered. Possible values are defined in class com.quark.qpp.office.service.constants.ExcelDataOutputFormats .
	contentDisposition	string	Used to set the content-disposition header value in the response. Set to "attachment" if downloading of named range is required. (Optional)
Returns	Returns an Excel dynamic range in the desired output format.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/office/excel? worksheet=Sales&dynamicrange=r1c1:r5c5 & stream the Excel file in input</p>		
	<p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <table xmlns="http://quark.com/ smartcontent /1.0" type=" smarttable "> < tgroup cols="1"> < colspec colnum="1" colname="1" colwidth="104px"/> < tbody > <row> <entry colsep="0" rowsep="0"> <p> </p> </entry> </row> <row> <entry valign</pre>		

	="bottom" colsep="0" rowsep="0"> <p> <t>Product details</t> </p> </entry> </row> </tbody> </tgroup> </table>
--	---

getVisioPagesInfo

Function	Get metadata of all pages of a Visio asset.		
Description	This request returns all pages defined within a Visio document. To retrieve specific pages, specify a valid page name in the pages parameter.		
Type	GET		
URL Format	.../office/visio/{assetId}		
Java API	Method Name		Class Name
	getVisioPagesInfo		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	majorversion and minorversion	string	Specify the required major and minor version for which the pages of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	pageNames= [comma-separated list of page names]	string	The set of page names.
Returns	Returns a VisioPageInfoList object that contains index number and page name of all pages of a given Visio asset.		
Examples	URL: http://serverName:serverPort/rest/service/office/visio/{assetId} Result: A VisioPageInfoList object that contains index number and page name of all pages of a given Visio asset.<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <visioPageInfoList> <visioPageInfo index="1"> <pageName>Page-1</pageName> </visioPageInfo> </visioPageInfoList>		

getVisioPagesInfoBasedOnURI

Function	Get metadata of all pages of a Visio asset based on URI.	
Description	This request returns all pages defined within a Visio document. To retrieve specific pages, specify a valid page name in the pages parameter.	
Type	GET	
URL Format	.../office/visio	
Java API	Method Name	Class Name

	<code>getVisioPagesInfoBasedOnURI</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	Uri	string	The URI of the asset. (Mandatory)
	<code>pageNames= [comma-separated list of page names]</code>	string	The set of page names.
Returns	Returns a <code>VisioPageInfoList</code> object that contains index number and page name of all pages of a given Visio asset.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/visio?uri=qpp://assets/{assetId}</code></p> <p>Result:</p> <p>A <code>VisioPageInfoList</code> object that contains index number and page name of all pages of a given Visio asset. <code><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <visioPageInfoList> <visioPageInfo index="1"> <pageName>Page-1</pageName> </visioPageInfo> </visioPageInfoList></code></p>		

getVisioPagesInfoBasedOnStreamContext

Function	Get metadata of all pages of a Visio asset based on stream.		
Description	This request returns all pages defined within a Visio document. To retrieve specific pages, specify a valid page name in the pages parameter. Include the file in the request body or as a file part. (Mandatory)		
Type	GET		
URL Format	<code>.../office/visio</code>		
Java API	Method Name	Class Name	
	<code>getVisioPagesInfoBasedOnStreamContext</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>pageNames= [comma-separated list of page names]</code>	string	The set of page names.
Returns	Returns a <code>VisioPageInfoList</code> object that contains index number and page name of all pages of a given Visio asset.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/visio</code> Include the file in the request body.</p> <p>Result:</p> <p>A <code>VisioPageInfoList</code> object that contains index number and page name of all pages of a given Visio asset. <code><visioPageInfoList> <visioPageInfo index="1"> <pageName>Page-1</pageName> </visioPageInfo> </visioPageInfoList></code></p>		

getVisioPageOutput

Function	Get page output of a Visio asset.		
Description	This request returns a Visio document page output in the desired output format.		
Type	GET		
URL Format	.../office/visio/{assetId}		
Java API	Method Name		Class Name
	getVisioPageOutput		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	majorversion and minorversion	string	Specify the required major and minor version for which the pages of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	pagename= [comma-separated list of page names]	string	The set of page names.
Returns	Returns a filestream of required output format based on given parameters.		
Examples	URL: http://serverName:serverPort/rest/service/office/visio/{assetId}?outputformat=image&pagename={pageName}&imageformat=PNG Result: A filestream of the generated output of the given asset.		

getVisioPageOutputBasedOnURI

Function	Get metadata of all pages of a Visio asset based on URI.		
Description	This request returns a Visio document page output in the desired output format.		
Type	GET		
URL Format	.../office/visio		
Java API	Method Name		Class Name
	getVisioPagesOutputBasedOnURI		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	Uri	string	The URI of the asset. (Mandatory)
	pagename= [comma-separated list of page names]	string	The set of page names.
Returns	Returns a VisioPageInfoList object that contains index number and page name		

	of all pages of a given Visio asset.
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/visio?uri=qpp://assets/{assetId}&outputformat=image&pagename={pageName}&imageformat=PNG</code></p> <p>Result:</p> <p>A filestream of the generated output of the given asset.</p>

getVisioPageOutputBasedOnStreamContext

Function	Get page output of a Visio asset based on stream.		
Description	This request returns a Visio document page output in the desired output format. Include the file in the request body or as a file part. (Mandatory)		
Type	GET		
URL Format	<code>.../office/visio</code>		
Java API	Method Name	Class Name	
	<code>getVisioPageOutputBasedOnStreamContext</code>	<code>com.quark.qpp.service.facadeClassOfficeFacade</code>	
Parameters	Name	Type	Description
	<code>pagename</code>	<code>string</code>	The page name.
Returns	Returns a filestream of the required output format based on the given parameters.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/visio/outputformat=image&pagename={pageName}&imageformat=PNG</code> Include the file in the request body.</p> <p>Result:</p> <p>A filestream of the generated output of the given asset.</p>		

getPowerPointSlidesInfo

Function	Get metadata of all slides of a PowerPoint asset.		
Description	This request returns all slides defined within a PowerPoint document. To retrieve specific slides, specify a valid slide index in the slideIndexes parameter.		
Type	GET		
URL Format	<code>.../office/powerpoint/[assetId]</code>		
Java API	Method Name	Class Name	
	<code>getPowerPointSlidesInfo</code>	<code>com.quark.qpp.service.facadeClassOfficeFacade</code>	
Parameters	Name	Type	Description

	<code>majorversion</code> and <code>minorversion</code>	string	Specify the required major and minor version for which the pages of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	<code>slideindexes=</code> [comma-separated list of slide indexes]	string	The set of slide indexes.
Returns	Returns a <code>PowerPointSlidesInfoList</code> object that contains index numbers and slide IDs of all slides for a given PowerPoint asset.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/powerpoint{assetId}</code></p> <p>Result:</p> <p>A <code>PowerPointSlidesInfoList</code> object that contains index numbers and slide IDs for all slides of a given PowerPoint asset.</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <powerPointSlideInfoList> <powerPointSlideInfo index="1"> <slideId>256</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="2"> <slideId>257</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="3"> <slideId>258</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="4"> <slideId>259</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="5"> <slideId>260</slideId> </powerPointSlideInfo> </powerPointSlideInfoList></pre>		

getPowerPointSlidesInfoBasedOnURI

Function	Get metadata of all slides of a PowerPoint asset based on URI.		
Description	This request returns all slides defined within a PowerPoint document. To retrieve specific slides, specify a valid slide id in the slideIndexes parameter.		
Type	GET		
URL Format	<code>.../office/powerpoint</code>		
Java API	Method Name	Class Name	
	<code>getPowerPointSlidesInfoBasedOnURI</code>	<code>com.quark.qpp.service.facadeClass OfficeFacade</code>	
Parameters	Name	Type	Description
	<code>Uri</code>	string	The URI of the asset. (Mandatory)
	<code>slideindexes=</code> [comma-separated list of slide index numbers]	string	The set of index number.
Returns	Returns a <code>PowerPointSlidesInfoList</code> object that contains index numbers and slide IDs for all slides of a given PowerPoint asset.		
Examples	URL:		

<p><code>http://serverName:serverPort/rest/service/office/powerpoint?uri=qpp://assets/{assetId}</code></p> <p>Result:</p> <p>A <code>PowerPointSlidesInfoList</code> object that contains index numbers and slide IDs for all slides of a given PowerPoint asset.</p>

getPowerPointSlidesInfoBasedOnStreamContext

Function	Get all slide metadata for a PowerPoint asset based on file stream.		
Description	This request returns all slides defined within a PowerPoint document. To retrieve specific slides, specify a valid index of a slide in the <code>slideIndexes</code> parameter. Include the file in the request body or as a file part. (Mandatory)		
Type	GET		
URL Format	<code>.../office/powerpoint</code>		
Java API	Method Name		Class Name
	<code>getPowerPointSlidesInfoBasedOnStreamContext</code>		<code>com.quark.qpp.service.facadeClassOfficeFacade</code>
Parameters	Name	Type	Description
	<code>slideindexes=</code> [comma-separated list of slide indexes]	string	The set of slide ids.
Returns	Returns a <code>PowerPointSlideInfoList</code> object that contains index numbers and slide IDs for all slides of a given PowerPoint asset.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/office/powerpoint</code> Include the file in the request body.</p> <p>Result:</p> <p>A <code>PowerPointSlideInfoList</code> object that contains index numbers and slide ids of all slides of a given PowerPoint asset.</p> <pre><powerPointSlideInfoList> <powerPointSlideInfo index="1"> <slideId>256</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="2"> <slideId>257</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="3"> <slideId>258</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="4"> <slideId>259</slideId> </powerPointSlideInfo> <powerPointSlideInfo index="5"> <slideId>260</slideId> </powerPointSlideInfo> </powerPointSlideInfoList></pre>		

getPowerPointSlideOutput

Function	Get slide output of a PowerPoint asset.
Description	This request extracts PowerPoint slide output in the specified format.
Type	GET
URL Format	<code>.../office/powerpoint/[assetId]</code>

Java API	Method Name		Class Name
	<code>getPowerPointSlidesOutput</code>		<code>com.quark.qpp.service.facadeClass OfficeFacade</code>
Parameters	Name	Type	Description
	<code>majorversion</code> and <code>minorversion</code>	string	Specify the required major and minor version for which the slide of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	<code>slideid</code>	string	The slide Id.
	<code>outputformat</code>	string	The desired output format. Possible values are defined in class <code>com.quark.qpp.office.service.constants.ExcelDataOutputFormats</code>
Returns	Returns a filestream that contains the required output format based on the given parameters.		
Examples	<p>URL:</p> <p><code>http://server:port/rest/service/office/powerpoint/{assetId}?outputformat=image&slideid={slideId}&imageformat=PNG</code></p> <p>Result:</p> <p>Returns a filestream that contains the required output format based on the given parameters.</p>		

getPowerPointSlideOutputBasedOnURI

Function	Get slide output of a PowerPoint asset based on URI.		
Description	This request extracts the PowerPoint slide output in the desired output format..		
Type	GET		
URL Format	<code>.../office/powerpoint</code>		
Java API	Method Name		Class Name
	<code>getPowerPointSlideOutputBasedOnURI</code>		<code>com.quark.qpp.service.facadeClass OfficeFacade</code>
Parameters	Name	Type	Description
	<code>Uri</code>	string	The URI of the asset. (Mandatory)
	<code>slideid</code>	string	The slide Id.
	<code>outputformat</code>	string	The desired output format.
Returns	Returns a filestream that contains the required output format based on the given parameters.		
Examples	URL:		

<p>http://server:port/rest/service/office/powerpoint ?uri=qpp://assets/{assetId}&outputformat=image&slideid={slideId}&imageformat=PNG</p> <p>Result:</p> <p>Returns a filestream that contains the required output format based on the given parameters.</p>
--

getPowerPointSlideOutputBasedOnStreamContext

Function	Get slide output of a PowerPoint asset based on file stream.		
Description	This request extracts the PowerPoint slide output in the desired output format.. Include the file in the request body or as a file part. (Mandatory)		
Type	GET		
URL Format	.../office/powerpoint		
Java API	Method Name		Class Name
	getPowerPointSlideOutputBasedOnStreamContext		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	slideid	string	The slide Id.
	outputformat	string	The desired output format.
Returns	Returns a filestream that contains the required output format based on the given parameters.		
Examples	<p>URL:</p> <p>http://server:port/rest/service/office/powerpoint?outputformat=image&slideid={slideId}&imageformat=PNG</p> <p>Include the file in the request body.</p> <p>Result:</p> <p>Returns a filestream that contains the required output format based on the given parameters.</p>		

getPowerPointSlideRangeOutput

Function	Get slide range output of a PowerPoint asset.		
Description	This request extracts the PowerPoint slide output in the desired output format.		
Type	GET		
URL Format	.../office/powerpoint/{assetId}		
Java API	Method Name		Class Name
	getPowerPointSlideRangeOutput		com.quark.qpp.service.facadeClass OfficeFacade
Parameters	Name	Type	Description
	majorversion and	string	Specify the required major and minor

	<code>minorversion</code>		version for which the slide of an asset is required. (Optional) These parameters will be used collectively. If neither parameter is passed, the latest version of the asset will be considered. If only one of the parameters is not passed, then the value for that parameter is assumed to be 0.
	<code>sliderange = [comma-separated list of slide indexes]</code>	string	The set of slide indexes.
	<code>outputformat</code>	string	The desired output format.
Returns	Returns a filestream that contains the required output format based on the given parameters.		
Examples	<p>URL:</p> <p><code>http://server:port/rest/service/office/powerpoint/{assetId}?outputformat=image&sliderange=1,3,5&imageformat=PNG</code></p> <p>Result:</p> <p>Returns a filestream that contains the required output format based on the given parameters.</p>		

getPowerPointSlideRangeOutputBasedOnURI

Function	Get slide range output of a PowerPoint asset based on URI.		
Description	This request extracts the PowerPoint slide output in the desired output format..		
Type	GET		
URL Format	<code>.../office/powerpoint</code>		
Java API	Method Name		Class Name
	<code>getPowerPointSliderRangeOutputBasedOnURI</code>		<code>com.quark.qpp.service.facadeClassOfficeFacade</code>
Parameters	Name	Type	Description
	<code>Uri</code>	string	The URI of the asset. (Mandatory)
	<code>sliderange = [comma-separated list of slide indexes]</code>	string	The set of slide indexes.
	<code>outputformat</code>	string	The desired output format.
Returns	Returns a filestream that contains the required output format based on the given parameters.		
Examples	<p>URL:</p> <p><code>http://server:port/rest/service/office/powerpoint?uri=qpp://assets/{assetId}&outputformat=image&sliderange=1,3,5&imageformat=PNG</code></p> <p>Result:</p>		

	Returns a filestream that contains the required output format based on the given parameters.
--	--

getPowerPointSlideRangeOutputBasedOnStreamContext

Function	Get slide range output of a PowerPoint asset based on file stream.		
Description	This request extracts the PowerPoint slide output in the desired output format.. Include the file in the request body or as a file part. (Mandatory)		
Type	GET		
URL Format	.../office/powerpoint		
Java API	Method Name	Class Name	
	<code>getPowerPointSlideRangeOutputBasedOnStreamContext</code>	<code>com.quark.qpp.service.facadeClassOfficeFacade</code>	
Parameters	Name	Type	Description
	<code>sliderange = [comma-separated list of slide indexes]</code>	string	The set of slide indexes.
	<code>outputformat</code>	string	The desired output format.
Returns	Returns a filestream that contains the required output format based on the given parameters.		
Examples	URL: http://server:port/rest/service/office/powerpoint?sliderange=1,3,5&imageformat=PNG Include the file in the request body. Result: Returns a filestream that contains the required output format based on the given parameters.		

Charting

The charting capabilities of the API includes the following REST calls:

- [getChart](#)
- [getChartBasedOnUri](#)
- [getChartBasedOnStream](#)

getChart

Function	Create new chart.	
Description	This request draws a chart for the given asset using the specified template.	
Type	GET	
URL Format	...charts/{assetId} or .../chartsbypath/[asset path]	
Java API	Method Name	Class Name

	<code>getChart</code>		<code>com.quark.qpp.service.facadeClass ChartingFacade</code>
Parameters	Name	Type	Description
	<code>assetId</code>	string	The AssetID.
	<code>majorversion=[x]</code>	string	Major version of the asset. (Optional)
	<code>minorversion=[x]</code>	string	Minor version of the asset. (Optional)
	<code>templateuri=[x]</code>	string	Uri for the template file.
	<code>outputformat=[image, json, html]</code>	string	Image output format. (Optional)
Returns	Returns a stream of the successfully created chart.		
Examples	URL: <code>http://serverName:serverPortt/rest/service/charts/408?templateuri= qpp://assets/409&outputformat=html</code> Result: The file stream of the successfully created chart.		

getChartBasedOnUri

Function	Create new chart By URI.		
Description	This request draws a chart for the given data specified in the URI using the specified template.		
Type	GET		
URL Format	<code>...charts/</code>		
Java API	Method Name	Class Name	
	<code>getChartBasedOnURI</code>	<code>com.quark.qpp.service.facadeClass ChartingFacade</code>	
Parameters	Name	Type	Description
	<code>uri=[x]</code>	string	The URI for the source asset.
	<code>majorversion=[x]</code>	string	Major version of the asset. (Optional)
	<code>minorversion=[x]</code>	string	Minor version of the asset. (Optional)
	<code>templateuri=[x]</code>	string	Uri for the template file.
	<code>outputformat=[image, json, html]</code>	string	Image output format. (Optional)File, ClassPath, Http, Qpp are the supported URIs.
Returns	Returns a stream of the successfully created chart.		
Examples	URL: <code>http://serverName:serverPort/rest/service/charts?uri=qpp://assets/408&templateuri= qpp://assets/409&outputformat=htm</code> Result: The file stream of the successfully created chart.		

getChartBasedOnStream

Function	Create new chart By Stream		
Description	This request draws a chart for the given data input stream using the specified template. Include the file in the request body or as file part.		
Type	GET		
URL Format	.../charts/		
Java API	Method Name	Class Name	
	<code>getChartBasedOnStream</code>	<code>com.quark.qpp.service.facadeClass ChartingFacade</code>	
Parameters	Name	Type	Description
	<code>majorversion=[x]</code>	string	Major version of the asset. (Optional)
	<code>minorversion=[x]</code>	string	Minor version of the asset. (Optional)
	<code>templateuri=[x]</code>	string	Uri for the template file.
	<code>outputformat=[image, json, html]</code>	string	Image output format. (Optional)File, ClassPath, Http, Qpp are the supported URIs
Returns	Returns a stream of the successfully created chart.		
Examples	URL: <code>http://serverName:serverPort/rest/service/charts?templateuri=qpp://assets/409&out putformat=html</code> Result: The file stream of the successfully created chart.		

Configuration

The Configuration service is used to manage the publishing configuration (adding/updating new publishing channels, processes and MIME types) and is responsible for the following:

- Managing the configuration files.
- Capturing updates.
- Reinitializing corresponding bean definitions.

In a multi-server environment, each node in a cluster must have access to a single .ext folder and the shared location must be appended to the classpath system environment variable.

POST requests only alter config files on a configured app server node. Multi-server deployments can leverage a shared file system location for the config files and avoid needing to manually maintain multiple sets of files on various application servers.

The `reInitialize` call can be used to avoid restarts even after changing the configuration files manually on the server .

The configuration capabilities of the API includes the following REST calls:

- *getAllXmlMimeTypes*
- *getXmlMimeType*
- *createXmlMimeType*
- *getMappedMimeTypes*
- *mapMimeTypes*
- *unmapMimeTypes*
- *deleteXmlMimeType*
- *getPublishingProcess*
- *createPublishingProcess*
- *deletePublishingProcess*
- *getAllPublishingChannelDefinitions*
- *getPublishingChannelDefinition*
- *getPublishingChannelDefinitions*
- *createPublishingChannels*
- *createContentChannelMappings*
- *deleteContentChannelMappings*
- *createIndexingChannelMapping*
- *getMappedIndexingChannel*
- *deleteIndexingChannelMapping*
- *deletePublishingChannels*
- *getDeploymentProperties*
- *getDeploymentVersion*
- *reInitialize*

getAllXmlMimeTypes

Function	Get XML MimeTypes	
Description	This request returns all XML MimeTypes that exist on the server.	
Type	GET	
URL Format	<code>.../config/xmlmimetypes</code>	
Java API	Method Name	Class Name
	<code>getAllXmlMimeTypes</code>	<code>com.quark.qpp.service.facade Class ConfigurationFacade</code>
Parameters	This request requires no input parameters.	
Returns	Returns the definitions of all xml mime-types.	
Examples	URL:	

	<p>http://serverName:serverPort/rest/service/config/xmlmimetypes</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <mimeTypeInfoList> <mimeTypeInfo type="application/xml; format=smartrtable" parentMimeType ="application/xml;"> <xpath>/*[local-name()='table']/@type='smartrtable'</xpath> </mimeTypeInfo> <mimeTypeInfo type="application/xml; format=auxiliarydictionary" parentMimeType="format=hyphenationexceptions"> <xpath>/*[local-name()='Dictionary']/@Type='AuxiliaryDictionary'</xpath> </mimeTypeInfo> <mimeTypeInfo type="application/dita+xml; format=busdoc" parentMimeType="application/dita+xml; format=topic"> <pi name="Xpress">productLine="busdoc"</pi> </mimeTypeInfo> <mimeTypeInfo type="application/xml; format=smartdoc" parentMimeType="application/xml; format=smartcontent"> <xpath>/*[local- name()='section']/@type='document'</xpath> </mimeTypeInfo> <mimeTypeInfo type="application/xml; format=smartsection" parentMimeType="application/xml; format= smartcontent"> <xpath>/*[local-name()='section']/@type='section'</xpath> </mimeTypeInfo> </mimeTypeInfoList></pre>
--	--

getXmlMimeType

Function	Get XML MimeType		
Description	This request returns a specified XML MimeType.		
Type	GET		
URL Format	.../config/xmlmimetypes		
Java API	Method Name	Class Name	
	getXmlMimeType	com.quark.qpp.service.facade Class ConfigurationFacade	
Parameters	Name	Type	Description
	Mimetype	string	The desired MimeType. (Mandatory)
Returns	Returns the definition of the desired XML MimeType, including the parent MimeType, xpath and the processing instruction name and value based on which given MimeType is detected.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/config/xmlmimetypes? mimetype=application/xml; format=researchreport</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <mimeTypeInfoList> <mimeTypeInfo type="application/xml; format=researchreport" parentMimeType="application/xml; format= smartcontent"> <xpath>/*[local- name()='section']/@type='researchreport'</xpath> </mimeTypeInfo> </mimeTypeInfoList></pre>		

createXmlMimeType

Function	Create a custom XML MimeType.		
Description	This request creates a MimeType for an XML document.		
Type	POST		
URL Format	.../config/xmlmimetypes		
Java API	Method Name		Class Name
	createXmlMimeType		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	mimetype	string	The MimeType to be created. (Mandatory)
	parentmimetype	string	The parent MimeType. (Mandatory)
	xpath	string	The XPath. (It is mandatory to specify either the XPath or the piName and piValue.)
	piname	string	The processing instruction. (It is mandatory to specify either the XPath or the piName and piValue.)
	pivalue	string	The processing value. (It is mandatory to specify either the XPath or the piName and piValue.)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/xmlmimetypes? op=create&mimetype=application/xml; format=researchreport&parentmimetype=application/xml; format=smartcontent&xpath=/*[local-name()='section']/@type='researchreport'		

getMappedMimeType

Function	Get mapped MimeTypes		
Description	This request returns a list of MimeTypes mapped to the given content type.		
Type	GET		
URL Format	.../config/xmlmimetypes		
Java API	Method Name		Class Name
	getMappedMimeType		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	contenttype	string	The content type. (Mandatory)
Returns	Returns a mimeTypeInfoList object that contains the metadata of all mimeTypes		

	mapped to the given content type.
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/config/xmlmimetypes?contenttype=Research Report</code></p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <mimeTypeInfoList> <mimeTypeInfo type="application/xml; format=researchreport" parentMimeType="application/xml; format= smartcontent"> <xpath>/*[local- name()='section']/@type='researchreport'</xpath> </mimeTypeInfo> </mimeTypeInfoList></pre>

mapMimeType

Function	Map custom XML MimeType with ContentType.		
Description	This request maps a content type to a list of mime types.		
Type	POST		
URL Format	<code>.../config/xmlmimetypes</code>		
Java API	Method Name	Class Name	
	<code>mapMimeType</code>	<code>com.quark.qpp.service.facade</code> <code>Class ConfigurationFacade</code>	
Parameters	Name	Type	Description
	<code>op=mapcontenttype</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>contenttype=[contentTypeIdOrName]</code>	string	The content type to be mapped. (Mandatory)
	<code>mimetypes</code>	string	The mimetypes to be mapped. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <p><code>http://serverName:serverPort/rest/service/config/xmlmimetypes?op=mapcontenttype&contenttype=Research Report&mimetypes=application/xml;format=researchreport</code></p>		

unmapMimeType

Function	Unmap custom XML MimeType with ContentType.		
Description	This request deletes the content type mappings for given mime types.		
Type	POST		
URL Format	<code>.../config/xmlmimetypes</code>		
Java API	Method Name	Class Name	
	<code>unmapMimeType</code>	<code>com.quark.qpp.service.facade</code>	

			Class <code>ConfigurationFacade</code>
Parameters	Name	Type	Description
	<code>op=unmapcontenttype</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>mimetypes</code>	string	The mimetypes to be unmapped. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/xmlmimetypes? ?op=unmapcontenttype&mimetypes=application/xml; format=researchreport		

deleteXmlMimeType

Function	Delete custom XML MimeType.		
Description	This request deletes an xml mime type and if this mime type is mapped to a content type, that mapping is also deleted.		
Type	POST		
URL Format	<code>.../config/xmlmimetypes</code>		
Java API	Method Name		Class Name
	<code>deleteXmlMimeType</code>		<code>com.quark.qpp.service.facade</code> Class <code>ConfigurationFacade</code>
Parameters	Name	Type	Description
	<code>op=delete</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>mimeType</code>	string	The mimeType to be deleted. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/xmlmimetypes? op=delete&mimetype=application/xml; format=researchreport		

getPublishingProcess

Function	Get publishing process		
Description	This request returns the xml configuration of the process with the given process name.		
Type	GET		
URL Format	<code>.../config/publishingprocess/{ProcessId}</code>		
Java API	Method Name		Class Name
	<code>getPublishingProcess</code>		<code>com.quark.qpp.service.facade</code> Class <code>ConfigurationFacade</code>

Parameters	Name	Type	Description
	<code>processName</code>	string	The process name. (Mandatory)
Returns	Returns the xml configuration of the process with the given process name.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingprocess/researchReportToQxpRenderer Result: <pre><?xml version="1.0" encoding="UTF-8"> <beans xmlns="http://www.springframework.org/schema/beans"> <!-- PROCESS DEFINITIONS --> <pub:process id="researchReportToQxpRenderer"> <pub:activity class="com.quark.qpp.publishing.activity.UnzipXaDocumentPackage"> <description>Unzips and registers XA document file along with the dependencies supplied in the zip file.</description> <property name="activityName"> <value>UnzipXaDocumentPackage</value> </property> ... </pub:process> </beans></pre>		

createPublishingProcess

Function	Create publishing process		
Description	This request creates a new process based on the publishingPocessXML.		
Type	POST		
URL Format	<code>.../config/publishingprocess</code>		
Java API	Method Name	Class Name	
	<code>createPublishingProcess</code>	<code>com.quark.qpp.service.facade Class ConfigurationFacade</code>	
Parameters	Name	Type	Description
	<code>processxml= [xml]</code>	string	The publishing process XML (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingprocess?op=create&processxml=<xml>		

deletePublishingProcess

Function	Delete publishing process.		
Description	This request deletes the publishing process with the given process name. Deletes the channels where this process is used. Deletes the corresponding configuration file of the process and updates ProcessConfig-ext.xml.		
Type	POST		
URL Format	<code>.../config/publishingprocess/{ProcessName}</code>		
Java API	Method Name	Class Name	
	<code>deletePublishingProcess</code>	<code>com.quark.qpp.service.facade Class ConfigurationFacade</code>	

	Name	Type	Description
Parameters	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	processName	string	The name of the process to be deleted. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingprocess/researchReportToQxpRenderer?op=delete		

getAllPublishingChannelDefinitions

Function	Get all publishing channels	
Description	This request returns all publishing of type publish, deliver and datadoc.	
Type	GET	
URL Format	.../config/publishingchannels	
Java API	Method Name	Class Name
	getAllPublishingChannelDefinitions	com.quark.qpp.service.facade.Class ConfigurationFacade
Parameters	This request requires no input parameters.	
Returns	Returns a publishingChannelList object that contains the metadata of all publishing channels (Publish, Deliver, Datadoc).	
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels Result: <pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo id="qppPreviewConvert" name="QPP.PREVIEW.IM.JPG" process="qppPreviewConverter" type="publish"> <parameter name="SOURCE_FILE_EXTENSION" ask="true"></parameter> <parameter name="OUTPUT_FILE_NAME" ask="true"></parameter> </publishingChannelInfo> <publishingChannelInfo id="smartDocJpeg" name="QXP.JPEG.ZIP" process="smartDocToQxpRenderer" type="publish"> <parameter name="ASSET_ID" ask="true"></parameter> <parameter name="LAYOUT" ask="true"></parameter> </publishingChannelInfo> </publishingChannelList> </pre>	

getPublishingChannelDefinition

Function	Get publishing channel
Description	This request returns the definition of the publishing channel with the given Id, including the channel Id, name, channel type, the channel's process name and channel parameters.

Type	GET		
URL Format	.../config/publishingchannels/{channelId}		
Java API	Method Name		Class Name
	getPublishingChannelDefinition		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	channelId	string	The channel Id of the desired publishing channel. (Mandatory)
Returns	Returns a <code>publishingChannelList</code> object that contains the metadata of the given publishing channel ID.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/config/publishingchannels/smartDocJpeg</p> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <publishingChannelList> <publishingChannelInfo id="smartDocJpeg" name="QXP.JPEG.ZIP" process="smartDocToQxpRenderer" type="publish"> <parameter name="ASSET_ID" ask="true"></parameter> <parameter name="LAYOUT" ask="true"></parameter> <parameter name="BOX" ask="true"></parameter> <parameter name="SCALE" ask="true"></parameter> <parameter name="MAX_PAGES" ask="false">0</parameter> <parameter name="RESOLVE_QPP_RELATIONS" ask="true">true</parameter> </publishingChannelInfo> </publishingChannelList></pre>		

getPublishingChannelDefinitions

Function	Get the publishing channel for a specific content type.		
Description	This request returns the definition of the publishing channels of type publish, deliver and datadoc that are mapped to a particular content type.		
Type	GET		
URL Format	.../config/publishingchannels		
Java API	Method Name		Class Name
	getPublishingChannelDefinitions		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	contenttype	string	The content type that is mapped to the desired publishing channel. (Mandatory)
Returns	Returns a <code>publishingChannelList</code> object that contains the metadata of all the publishing channels available for the given content type.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/config/publishingchannels?contenttype=Smart Document</p> <p>Result:</p>		

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<publishingChannelList> <publishingChannelInfo id="smartDocJpeg"
name="QXP.JPEG.ZIP" process="smartDocToQxpRenderer" type="publish">
<parameter name="ASSET_ID" ask="true"></parameter> <parameter
name="LAYOUT" ask="true"></parameter> <parameter name="BOX"
ask="true"></parameter> <parameter name="SCALE" ask="true"></parameter> .....
<parameter name="MAX_PAGES" ask="false">0</parameter> <parameter
name="RESOLVE_QPP_RELATIONS" ask="true">true</parameter>
</publishingChannelInfo> </publishingChannelList>
```

createPublishingChannels

Function	Create new publishing channels		
Description	This request creates new publishing channels as specified in the publishingChannelList.		
Type	POST		
URL Format	.../config/publishingchannels		
Java API	Method Name	Class Name	
	createPublishingChannels	com.quark.qpp.service.facade Class ConfigurationFacade	
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	publishingchannellist = [publishingChannelList]	string	The publishing channel list. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	<p>URL:</p> <p>http://serverName:serverPort/rest/service/config/publishingchannels?op=create&publishingchannellist=<publishingChannelList>publishingChannelList</p> <p>Example:<publishingChannelList> <publishingChannelInfo id="researchReportPdf" name="PDF" process="smartDocToQxpRenderer" type="publish"> <parameter name="OUTPUT_STYLE" ask="true">PDF High Quality</parameter> <parameter name="ASSET_ID" ask="true"/> <parameter name="QXP_TEMPLATE_URI" ask="true">classpath:ResearchReportTemplate.qxp</parameter> <parameter name="RENDER_FORMAT">PDF</parameter> <parameter name="EXCEL_DATA_IMAGE_FORMAT">PDF</parameter> <parameter name="ANNOTATE_ERRORS">true</parameter> <parameter name="TRACKED_CHANGES" ask="true">ACCEPT</parameter> <parameter name="XSL_URI" ask="true">classpath:SmartDoc2QXPS.xslt</parameter> <parameter name="APPEND_ERRORS" ask="true">true</parameter> </publishingChannelInfo> <publishingChannelInfo id="researchReportJpeg" name="QXP.JPEG.ZIP" type="publish" process="smartDocToQxpRenderer"> <parameter name="ASSET_ID" ask="true"/> <parameter name="LAYOUT" ask="true"/> <parameter name="BOX" ask="true"/> <parameter name="SCALE"</p>		

```
ask="true"/> <parameter name="RENDER_FORMAT">JPEG</parameter>
<parameter name="XSL_URI"
ask="true">classpath:SmartDoc2QXPS.xslt</parameter> <parameter
name="QXP_TEMPLATE_URI"
ask="true">classpath:SmartDocTemplate.qxp</parameter> <parameter
name="EXCEL_DATA_IMAGE_FORMAT" ask="true">PDF</parameter> <parameter
name="CHART_IMAGE_FORMAT">PDF</parameter> <parameter
name="VISIO_DATA_IMAGE_FORMAT">PDF</parameter> <parameter
name="POWERPOINT_DATA_IMAGE_FORMAT">PDF</parameter> <parameter
name="TRACKED_CHANGES" ask="true">ACCEPT</parameter> <parameter
name="RESOLVE_QPP_RELATIONS" ask="true">true</parameter>
</publishingChannelInfo> </publishingChannelList>
```

createContentChannelMappings

Function	Map publishing channel with content type for publishing.		
Description	This request maps content types to channels and updates the "PublishingConfig-ext.xml" file with the new mappings.		
Type	POST		
URL Format	.../config/publishingchannels		
Java API	Method Name	Class Name	
	createContentChannelMappings	com.quark.qpp.service.facade Class ConfigurationFacade	
Parameters	Name	Type	Description
	op=mapcontenttype	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contenttypes	string	The content types to be mapped. (Mandatory)
	channels[comma-separated list of channels]	string	The publishing channels. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels ?op=mapcontenttype&contenttypes=Research Report &channels=researchReportPdf,researchReportJpeg		

deleteContentChannelMappings

Function	Delete publishing channel and content type mapping.
Description	This request deletes content types to channel mappings and updates the "PublishingConfig-ext.xml" file. If content type is specified, channel mappings are deleted for this content type only. If content type is not specified, then mappings with all content types of the specified channel are deleted.

Type	POST		
URL Format	.../config/publishingchannels		
Java API	Method Name		Class Name
	deleteContentChannelMappings		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	op=unmapcontenttype	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contenttype	string	The content types to be mapped. (Mandatory)
	channel	string	The publishing channel. (Mandatory)
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels ?op=unmapcontenttype&contenttypes=Research Report &channels=researchReportPdf,researchReportJpeg		

createIndexingChannelMapping

Function	Map publishing channel with content type for indexing.		
Description	This request maps content types to indexing channels and updates the "IndexingChannels-ext.xml" file with the new mappings.		
Type	POST		
URL Format	.../config/publishingchannels		
Java API	Method Name		Class Name
	createIndexingChannelMapping		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	op=mapindexingchannel	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	contenttype	string	The content types to be mapped. (Mandatory)
	channels[comma-separated list of channels]	string	The indexing channels. (Mandatory)
	parameterMap	string	
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels ?op=mapindexingchannel&contenttype=Research Report		

	&channel=researchReportJpeg&QXP_TEMPLATE_URI=classpath:ResearchReportTemplate.qxp
--	---

getMappedIndexingChannel

Function	Get mapped indexing channels for a given content type.		
Description	This request returns the definition of the indexing channel mapped to the given content type.		
Type	GET		
URL Format	.../config/publishingchannels		
Java API	Method Name		Class Name
	getMappedIndexingChannel		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	contenttype= [contentTypeOrName]	string	The content type. (Mandatory)
	indexingchannel=[true false]	boolean	
Returns	Returns a <code>publishingChannelList</code> object that contains a list of mapped publishing channels and its metadata.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels? ?contenttype=Research Report&indexingchannel=true		

deleteIndexingChannelMapping

Function	Delete publishing channel and content type mapping for indexing.		
Description	This request deletes content type to indexing channel mapping and updates the "IndexingChannels-ext.xml" file. If content type is specified, channel mapping is deleted for the given content type only. If content type is not specified, then mappings with all content types of the specified channel are deleted.		
Type	POST		
URL Format	.../config/publishingchannels		
Java API	Method Name		Class Name
	deleteIndexingChannelMapping		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	op=unmapindexingchannel	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	channel	string	The channel.
	contenttype	string	The content type.

Returns	A status of 200 is returned in response to a successful request execution.
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels ?op=unmapindexingchannel&channel=researchReportJpeg&contenttype=Research Report

deletePublishingChannels

Function	Delete publishing channel.		
Description	This request deletes the channels specified. The “ChannelConfig-ext.xml” file is updated and any mapping for the specified channels is also deleted.		
Type	POST		
URL Format	.../config/publishingchannels		
Java API	Method Name		Class Name
	deletePublishingChannels		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	Name	Type	Description
	op=delete	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	channelids	string	The channel Ids of the channels to be deleted.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: http://serverName:serverPort/rest/service/config/publishingchannels ?op=delete&channelids=researchReportPdf		

getDeploymentProperties

Function	Get deployment properties		
Description	This request returns all properties specified for the deployment in the “Deployment.properties” file.		
Type	GET		
URL Format	.../config/deploymentproperties		
Java API	Method Name		Class Name
	getDeploymentProperties		com.quark.qpp.service.facade Class ConfigurationFacade
Parameters	This request requires no input parameters.		
Returns	Returns a <code>nameValueList</code> object that contains all the properties.		
Examples	URL: http://serverName:serverPort/rest/service/config/deploymentproperties		

Result:
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <nameValueList> <nameValue name="deployment.version" value=""/> </nameValueList></pre>

getDeploymentVersion

Function	Get deployment version		
Description	This request returns the version of the custom deployment.		
Type	GET		
URL Format	<code>.../config/deploymentversion</code>		
Java API	Method Name	Class Name	
	<code>getDeploymentVersion</code>	<code>com.quark.qpp.service.facade</code> <code>Class ConfigurationFacade</code>	
Parameters	This request requires no input parameters.		
Returns	Returns the text value of the deployment version specified in the "Deployment.properties" file.		
Examples	URL: <code>http://serverName:serverPort/rest/service/config/deploymentversion</code> Result: 1.2		

reinitialize

Function	ReInitialize		
Description	This request reinitializes the mime-type, channel and process definitions using their config files. It will reinitialize the content-mime-type mappings configuration, content type to channel mappings configuration and content type to indexing channel mappings configuration. This request may be used to apply settings assigned directly to config files without restarting the server.		
Type	POST		
URL Format	<code>.../config/</code>		
Java API	Method Name	Class Name	
	<code>reInitialize</code>	<code>com.quark.qpp.service.facade</code> <code>Class ConfigurationFacade</code>	
Parameters	Name	Type	Description
	<code>op=reinitializeconfigs</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>reinitializemimetypes = [true false]</code>	boolean	
	<code>reinitializechannels=[true false]</code>	boolean	

	<code>reinitializeprocesses</code> = <code>[true false]</code>	boolean	
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://serverName:serverPort/rest/service/config/?op=reinitializeconfigs&reinitializemimetypes=true&reinitializechannels=true&reinitializeprocesses=true</code>		

Clipboard

The Clipboard service is used to manage user-specific clipboard data on the Platform server. The server-side clipboard enables clipboard access across multiple devices and securely maintains data by user; accessible only to the user who copied the data. The service also manages clipboard data conversion to other formats.

The clipboard capabilities of the API includes the following REST calls:

- `addClipboardData`
- `addFormattedData`
- `getAllClipboardData`
- [*getAllClipboardDataInfos*](#)
- `getAllClipboardFormats`
- `getClipboardDataInfo`
- `getClipboardChannelsMappings`
- `getFormattedData`
- `getLatestClipboardData`
- `getLatestClipboardDataInfo`
- `getLatestFormattedData`

addClipboardData

Function	Add data to clipboard.		
Description	This request adds data to the logged-in user's clipboard and returns info about the data added .		
Type	POST		
URL Format	<code>.../clipboard</code>		
Java API	Method Name		Class Name
	<code>addClipboardData</code>		<code>com.quark.qpp.service.facade.ClipboardFacade</code>
Parameters	Name	Type	Description

	<code>op=adddata</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>contentType=[contentType]</code> or <code>[Name]</code>	string	The content type Id or Name of the data.
	<code>mimeType=[mimeType]</code>	string	The <code>mimeType</code> of the data.
	<code>inputstream</code>	string	The data stream.
Returns	Returns a <code>clipboardDataInfo</code> object in XML format, containing information about the data added to the user's clipboard. See the table in the "Request/Response parameter formats" chapter.		
Examples	<p>URL:</p> <pre>http://server:port/rest/service/clipboard?op=adddata&mimeType=application/vnd.openxmlformats-officedocument.spreadsheetml.sheet&contentType=Microsoft Excel Table&preview=My Excel table preview...&created by=QppAdmin&create on=12th June 2017</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <clipboardDataInfo dataId="24"> <timestamp>2017-06-12T14:15:34Z</timestamp> <contentType id="110171" name="Microsoft Excel Table"/> <mimeType>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</mimeType> <clipboardMeta> <nameValue name="preview" value="My Excel table preview..."/> <nameValue name="create on" value="12th June 2017"/> <nameValue name="created by" value="QppAdmin"/> </clipboardMeta> </clipboardDataInfo></pre>		

addFormattedData

Function	Add formatted data.		
Description	This request adds formatted-data to the logged-in user's clipboard for the data corresponding to the data Id.		
Type	POST		
URL Format	<code>.../clipboard/{dataId}</code>		
Java API	Method Name	Class Name	
	<code>addFormattedData</code>	<code>com.quark.qpp.service.facade.ClipboardFacade</code>	
Parameters	Name	Type	Description
	<code>op=addformatteddata</code>	string	The <code>op</code> parameter allows you to specify which operation is to be performed.

			(Mandatory)
	<code>inputstream</code>	string	The data stream.
	<code>format</code>	string	The format of the data.
Returns	A status of 200 is returned in response to a successful request execution.		
Examples	URL: <code>http://server:port/rest/service/clipboard/24?op=addformatteddata&format=jpeg</code>		

getAllClipboardChannelsMapping

Function	Get all clipboard channel mappings.	
Description	This request retrieves all of the clipboard format to channel mappings.	
Type	GET	
URL Format	<code>.../clipboard/channelsmapping</code>	
Java API	Method Name	Class Name
	<code>getAllClipboardChannelsMapping</code>	<code>com.quark.qpp.service.facade.ClipboardFacade</code>
Parameters	This request requires no input parameters.	
Returns	Returns a <code>clipboardChannelMappingInfoList</code> object in XML format, containing a list of all clipboard format to channel mappings.	
Examples	URL: <code>http://server:port/rest/service/clipboard/channelsmapping</code> Result: <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <clipboardChannelMappingInfoList> <clipboardChannelMappingInfo id="fetchExcelTable" contentType="Microsoft Excel Table" clipboardFormat="svg"/> <channelParameters> <nameValue name="OUTPUT_FORMAT" value="image"/> <nameValue name="OUTPUT_FORMAT_PROPERTIES" value="imageformate=svg"/> </channelParameters> </clipboardChannelMappingInfoList></pre>	

```

        </channelParameters>
    </clipboardChannelMappingInfo>
    <clipboardChannelMappingInfo id="fetchExcelTable"
    contentType="Microsoft Excel Table" clipboardFormat="png"/>
    <channelParameters>
        <nameValue name="OUTPUT_FORMAT" value="image"/>
        <nameValue name="OUTPUT_FORMAT_PROPERTIES"
    value="imageformat=png"/>
    </channelParameters>
    </clipboardChannelMappingInfo>
    .....
</clipboardChannelMappingInfoList>
    
```

getAllClipboardDataInfos

Function	Get all clipboard data information.		
Description	This request fetches information about all the data present in the logged-in user's clipboard.		
Type	GET		
URL Format	.../clipboard		
Java API	Method Name	Class Name	
	getAllClipboardDataInfos	com.quark.qpp.service.facade.ClipboardFacade	
Parameters	Name	Type	Description
	metadata=true	boolean	(Mandatory)
Returns	Returns a <code>clipboardDataInfoList</code> object in XML format, containing information about all the data present in the user's clipboard.		
Examples	URL:		
	http://server:port/rest/service/clipboard? metadata=true		
Examples	Result:		
	<pre> <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <clipboardDataInfoList> <clipboardDataInfo dataId="24"> <timestamp>2017-06-12T14:15:34Z</timestamp> <contentType id="110171" name="Microsoft Excel Table"/> <mimeType>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</mimeType> <clipboardMeta> <nameValue name="preview" value="My Excel table preview..."/> <nameValue name="create on" value="12th June 2017"/> <nameValue name="created by" value="QppAdmin"/> </clipboardMeta> </clipboardDataInfo> </clipboardDataInfoList> </pre>		

getAllClipboardFormats

Function	Get all clipboard formats
-----------------	---------------------------

Description	This request returns all the data formats supported for clipboard data.		
Type	GET		
URL Format	.../clipboard/formats		
Java API	Method Name		Class Name
	getAllClipboardFormats		com.quark.qpp.service.facade.ClipboardFacade
Parameters	This request requires no input parameters		
Returns	Returns a <code>clipboardFormatInfoList</code> object containing an array of all the clipboard data formats in XML format. See the table in the “Request/Response parameter formats” chapter.		
Examples	URL:		
	<code>http://server:port/rest/service/clipboard/formats</code>		
Examples	Result:		
	<pre><?xml version="1.0" encoding="UTF-8"?> <clipboardFormatInfoList> <clipboardFormatInfo> <name>svg</name> <mimeType>image/svg+xml</mimeType> </clipboardFormatInfo> <clipboardFormatInfo> <name>png</name> <mimeType>image/png</mimeType> </clipboardFormatInfo> <clipboardFormatInfo> <name>jpeg</name> <mimeType>image/jpeg</mimeType> </clipboardFormatInfo> <clipboardFormatInfo> <name>pdf</name> <mimeType>application/pdf</mimeType> </clipboardFormatInfo> <clipboardFormatInfo> <name>cals</name> <mimeType>application/xml</mimeType> </clipboardFormatInfo> </clipboardFormatInfoList></pre>		

getClipboardDataInfo

Function	Get clipboard data information.		
Description	This request fetches information about the logged-in user’s clipboard data using the data Id.		
Type	GET		
URL Format	.../clipboard/{dataId}		
Java API	Method Name		Class Name
	getClipboardDataInfo		com.quark.qpp.service.facade.ClipboardFacade
Parameters	Name	Type	Description
	metadata=true	boolean	(Mandatory)

Returns	Returns a <code>clipboardDataInfo</code> object in XML format, containing information about the clipboard data using the data Id.
Examples	<p>URL:</p> <pre>http://server:port/rest/service/clipboard/24? metadata=true</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <clipboardDataInfo dataId="24"> <timestamp>2017-06-12T14:15:34Z</timestamp> <contentType id="110171" name="Microsoft Excel Table"/> <mimeType>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</mimeType> <clipboardMeta> <nameValue name="preview" value="My Excel table preview..."/> <nameValue name="create on" value="12th June 2017"/> <nameValue name="created by" value="QppAdmin"/> </clipboardMeta> </clipboardDataInfo></pre>

getClipboardChannelsMapping

Function	Get clipboard channel mappings for content type.		
Description	This request returns the clipboard format to channel mappings based on the content type.		
Type	GET		
URL Format	.../clipboard/channelsmapping		
Java API	Method Name	Class Name	
	<code>getClipboardChannelsMapping</code>	<code>com.quark.qpp.service.facade.ClipboardFacade</code>	
Parameters	Name	Type	Description
	<code>contentType=contentType</code> <code>peIdOrName</code>	string	The content type.(Mandatory)
Returns	Returns a <code>clipboardChannelMappingInfoList</code> object in XML format.		
Examples	<p>URL:</p> <pre>http://server:port/rest/service/clipboard/channelsmapping?contentType=Microsoft Excel Table</pre> <p>Result:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <clipboardChannelMappingInfoList> <clipboardChannelMappingInfo id="fetchExcelTable" contentType="Microsoft Excel Table" clipboardFormat="svg"> <channelParameters> <nameValue name="OUTPUT_FORMAT" value="image"/> <nameValue name="OUTPUT_FORMAT_PROPERTIES" value="imageformat=svg"/> </channelParameters> </clipboardChannelMappingInfo> <clipboardChannelMappingInfo id="fetchExcelTable" contentType="Microsoft Excel Table" clipboardFormat="png"> <channelParameters></pre>		


```

<nameValue name="OUTPUT_FORMAT" value="image"/>
<nameValue name="OUTPUT_FORMAT_PROPERTIES"
value="imageformat=png"/>
</channelParameters>
</clipboardChannelMappingInfo>
.....
</clipboardChannelMappingInfoList>

```

getFormattedData

Function	Get formatted data.		
Description	This request retrieves formatted data based on the data Id and format. The formatted data is retrieved by using default Channel Mappings if no publishing channel parameter is specified.		
Type	GET		
URL Format	.../clipboard/{dataId}		
Java API	Method Name	Class Name	
	getFormattedData	com.quark.qpp.service.facade.ClipboardFacade	
Parameters	Name	Type	Description
	format	string	The format in which clipboard data is to be retrieved.
	publishingchannel	string	The publishing channel Id.
	parametersMap	string	The publishing channel parameters.
	contentDisposition		The content disposition value to be set on the http response header.
	outputStream	string	The output stream onto which output is to be written (for local API invocations).
Returns	Returns a stream of formatted data.		
Examples	URL: http://server:port/rest/service/clipboard/24? format=png		
	Result: A filestream of the required output format based on given parameters.		

getLatestClipboardData

Function	Get latest clipboard data.
Description	This request retrieves the latest data that was added to the logged-in user's clipboard.
Type	GET

URL Format	<code>.../clipboard</code>		
Java API	Method Name		Class Name
	<code>getLatestClipboardData</code>		<code>com.quark.qpp.service.facade.ClipboardFacade</code>
Parameters	Name	Type	Description
	<code>contentDisposition</code>	string	The content disposition value to be set on the http response headers.
	<code>outputStream</code>	string	The output stream onto which output is to be written (for local API invocations).
Returns	Returns a stream of copied data into server.		
Examples	URL:		
	<code>http://server:port/rest/service/clipboard</code>		
	Result:		
A filestream of the latest copied data.			

getLatestClipboardDataInfo

Function	Get latest clipboard data information.		
Description	This request fetches information about the latest data that was added to the logged-in user's clipboard.		
Type	GET		
URL Format	<code>.../clipboard</code>		
Java API	Method Name		Class Name
	<code>getLatestClipboardDataInfo</code>		<code>com.quark.qpp.service.facade.ClipboardFacade</code>
Parameters	Name	Type	Description
	<code>metadata=true</code>	boolean	
	<code>latest=true</code>	boolean	
Returns	Returns a <code>clipboardDataInfo</code> object in XML format, containing information about the latest data added to the user's clipboard . See the table in the "Request/Response parameter formats" chapter.		
Examples	URL:		
	<code>http://server:port/rest/service/clipboard?metadata=true&latest=true</code>		
	Result:		
<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <clipboardDataInfo dataId="24"> <timestamp>2017-06-12T14:15:34Z</timestamp> <contentType id="110171" name="Microsoft Excel Table"/> <mimeType>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</mimeType></pre>			

```

<clipboardMeta>
  <nameValue name="preview" value="My Excel table
  preview..."/>
  <nameValue name="create on" value="12th June 2017"/>
  <nameValue name="created by" value="QppAdmin"/>
</clipboardMeta>
</clipboardDataInfo>

```

getLatestFormattedData

Function	Get latest formatted data.		
Description	This request retrieves the latest data added to the logged-in user's clipboard in the specified format. The formatted data is retrieved by using default Channel Mappings if no publishing channel parameter is specified.		
Type	GET		
URL Format	.../clipboard		
Java API	Method Name	Class Name	
	getLatestFormattedData	com.quark.qpp.service.facade.ClipboardFacade	
Parameters	Name	Type	Description
	format	string	The format in which clipboard data is to be retrieved.
	publishingchannel	string	The publishing channel Id.
	parametersMap	string	The publishing channel parameters.
	contentdisposition		The content disposition value to be set on the http response header.
	outputStream	string	The output stream onto which output is to be written (for local API invocations).
Returns	Returns a stream of formatted data.		
Examples	URL: http://server:port/rest/service/clipboard? format=png Result: A filestream of the required output format based on given parameters.		

promoteClipboardDataToAsset

Function	Promote clipboard data to asset
Description	This request promotes the user's clipboard data to Platform's asset.
Type	POST

URL Format	<code>.../clipboard/{dataId}</code>		
Java API	Method Name		Class Name
	<code>promoteClipboardDataToAsset</code>		<code>com.quark.qpp.service.facade.ClipboardFacade</code>
Parameters	Name	Type	Description
	<code>op=promotetoasset</code>	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	<code>createmajorversion</code>	boolean	
	<code>attributevalues</code>	string	
Returns	Returns an <code>assetInfo</code> object in XML format, containing the metadata of the newly created asset. See the table in the “Request/Response parameter formats” chapter.		
Examples	<p>URL:</p> <pre>http://server:port/rest/service/clipboard/{DataId}?loginname=Admin&loginpassword=Admin&op=promotetoasset&attributevalues=<xml/></pre> <p>RESULT:</p> <pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?> <assetInfo id="220"> <assetVersion majorVersion="1" minorVersion="0"/> <attributeValueList> <attributeValue id="1" name="Id" type="numeric">220</attributeValue> </attributeValueList> <articleComponentInfoList/> </assetInfo></pre>		

JSON support in request parameters for POST requests.

Both XML and JSON can be used to pass request parameters.

The following are some examples:

createCollection

Function	Create a collection.		
Description	This request creates a new collection according to the given attributes.		
Type	POST		
URL Format	.../collections/[collection name] or collectionsbypath/[collection path]		
Header	qpp-content-type:json		
Java API	Method Name		Class Name
	createCollection		com.quark.qpp.service.facade Class CollectionFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	targetparent=[collection path or ID]	long or string	The parent collection for the new collection. (Mandatory)There is a choice between collection ID, Name and Path when providing a value.
	attributevalues=<xml/>	string	The set of attribute values.: See "Request/Response parameter formats."(All attributes are Optional)
	contenttype=[content type name]	string	Defaults to GENERAL. The default content type IDs are documented in the JavaDoc SDK at com.quark.qpp.core.content.service.constants.DefaultContentTypes. To get a list of content types, use a call to ../admin/contenttypes.(Optional)There is a choice between content type ID, Name and Hierarchy when providing a value.

JSON SUPPORT IN REQUEST PARAMETERS FOR POST REQUESTS.

Returns	Returns the <code>collectionInfo</code> object of the newly created collection.
Examples	<p>URL:</p> <pre> http://serverName:serverPort/rest/service/collections? op=create&collectioninfolist= {"CollectionInfoList": {"collectionInfo": [{"assetInfoList":null, "attributeValueList": { "attributeValue": [{"id":2, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":"Collection_5800a", "valueId":0}, {"id":59, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":"False", "valueId":0}, {"id":57, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":"Home/Collection_5800a", "valueId":0}, {"id":62, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":null, "valueId":1416}] }, "collectionClass":null, "collectionClassId":1416, "collectionInfoList":null, "collectionPath":"Home/Collection_5800a", "defaultJobJacketId":1, "defaultJobJacketName":null, "defaultJobJacketTicket":"Default Job Ticket", "domainId":19, "groupInfoList":null, "id":0, "jobJacketInfoList":null, "name":"Collection_5800a", "parentColInfoList":null, "revisionControlInfoList":null, "routingInfoList":null, "templateId":0, "userInfoList":null, "workflowInfoList":null }] } } Input:{"CollectionInfoList": {"collectionInfo": [{"assetInfoList":null, "attributeValueList": {"attributeValue": [{"id":2, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":"Collection_5800a", "valueId":0}, {"id":59, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":"False", "valueId":0}, {"id":57, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":"Home/Collection_5800a", "valueId":0}, {"id":62, "multiValued":false, "name":""," "sourceReference":null, "type":""," "value":null, "valueId":1416}] }, "collectionClass":null, "collectionClassId":1416, "collectionInfoList":null, "collectionPath":"Home/Collection_5800a", "defaultJobJacketId":1, "defaultJobJacketName":null, "defaultJobJacketTicket":"Default Job Ticket", "domainId":19, "groupInfoList":null, "id":0, "jobJacketInfoList":null, "name":"Collection_5800a", "parentColInfoList":null, "revisionControlInfoList":null, "routingInfoList":null, "templateId":0, "userInfoList":null, "workflowInfoList":null}] } } </pre>

checkinNew

Function	Check in a new asset.
Description	Include the file in the request body or as file part.
Type	POST
URL Format	.../assetsbypath/[asset path] or /assets/[New Asset Name]
Header	qpp-content-type: json

JSON SUPPORT IN REQUEST PARAMETERS FOR POST REQUESTS.

Java API	Method Name		Class Name
	checkInNew		com.quark.qpp.service.facade Class AssetFacade
Parameters	Name	Type	Description
	op=checkinnew	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributevalues		See “checkinNewRequest/Response parameter formats.” The following attributes are Mandatory:Content type (There is a choice between content type ID, Name and Hierarchy when providing a value for the Content type) attribute. Collection (There is a choice between collection ID, Name and Path when providing a value for the Collection attribute). WorkflowStatusFile extensionOriginal filename
	relations		See “Request/Response parameter formats.” If this parameter is passed, the Platform Server will check-in the asset and create child relations with the new asset being checked in at the same time. (Optional).
	createminorversion =[true false]	boolean	Specify if this version of the asset is to be checked in to the QPP server as a major or a minor version. (Optional)
Returns	Returns an assetInfo object of the newly checked in asset with its metadata.		
Examples	<p>URL:</p> <pre>http://myplatform:61400/rest/service/assets/Platform 11.2 What's New.pptx? op=checkinnew&attributevalues={"AttributeValueList":{"...}} Input:{"AttributeValueList": {"attributeValue":[{ "id":62, "multiValued":false, "name":""," sourceReference":null, "type":""," value":"System;Asset;Project;QuarkXPress Project;", "valueId":0 }, { "id": 54, "multiValued": false, "name": "", "sourceReference": null, "type": "", "value": null, "valueId": 11 }, { "id": 24, "multiValued": false, "name": "", "sourceReference": null, "type": "", "value": null, "valueId": 13 }, { "id": 33, "multiValued": false, "name": "", "sourceReference": null, "type": "", "value": "15116.qxp", "valueId": 0 }, { "id": 25, "multiValued": false, "name": "", "sourceReference": null, "type": "", "value": null, "valueId": 1 }, { "id": 11, "multiValued": false, "name": "", "sourceReference": null, "type": "", "value": "qxp", "valueId": 0 }, { "id": 55, "multiValued": false, "name": "", "sourceReference": null, "type": "", "value": null, "valueId": 1 }] } }</pre>		

createDomains

Function	Create Attribute Domains.		
Description	This request is used to create a single or multiple attribute domains at one time.		
Type	POST		
URL Format	.../admin/domains		
Header	qpp-content-type: json		
Java API	Method Name		Class Name
	createDomains		com.quark.qpp.service.facade Class AdminFacade
Parameters	Name	Type	Description
	op=create	string	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	attributedomaininfolist	string	List of attribute domains to create.
Returns	Returns a <code>attributeDomainInfoList</code> object containing a list of successfully created attribute domains.		
Examples	<p>URL:</p> <pre>http://myplatform:61400/rest/service/admin/domains? op=create&attributedomaininfolist= Input: {AttributeDomainInfoList: {"attributeDomainInfo": [{"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [], "id": 0, "name": "7", "sourceReference": "source7", "valueHierarchy": null }] }, "id": 0, "name": "4", "sourceReference": "source4", "valueHierarchy": null }] }, "id": 0, "name": "1", "sourceReference": "source1", "valueHierarchy": null }, {"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [], "id": 0, "name": "8", "sourceReference": "source8" "valueHierarchy": null }] }, "id": 0, "name": "5", "sourceReference": "source5", "valueHierarchy": null }] }, "id": 0, "name": "2", "sourceReference": "source2", "valueHierarchy": null }, {"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [{"domainValueList": {"domainValue": [], "id": 0, "name": "9", "sourceReference": "source9", "valueHierarchy": null }] }, "id": 0, "name": "6", "sourceReference": "source6", "valueHierarchy": null }] }, "id": 0, "name": "3", "sourceReference": "source3", "valueHierarchy": null }] }, "expandable": true, "hierarchical": true, "id": 0, "itemsSorted": false, "name": "SampleQPP_0cb", "sourceReference": null, "systemDefined": false, "userModifiable": false, "warnBeforeExpansion": true }] } }</pre>		

createSearches

Function	Create searches.
Description	This request creates a single or multiple searches at one time.

JSON SUPPORT IN REQUEST PARAMETERS FOR POST REQUESTS.

Type	POST		
URL Format	.../searchdefinitions		
Header	qpp-content-type: json		
Java API	Method Name		Class Name
	createSearches		com.quark.qpp.service.facade Class QueryDefinitionFacade
Parameters	Name	Type	Description
	op=create	long	The op parameter allows you to specify which operation is to be performed. (Mandatory)
	searchinfolist=[xml]	string	An XML object containing metadata of searches to be created.
Returns	Returns a <code>searchInfoList</code> object containing metadata of newly created searches.		
Examples	<p>URL:</p> <pre>http://myplatform:61400/rest/service/searchdefinitions? op=create&searchinfolist= Input: {"SearchInfoList": {"searchInfo": [{"id": 0,"owner": {"id": 0, "value": null}, "parameterized": false, "queryConditionList": {"queryCondition": []}, "queryContext": {"collections": {"collection": [{"id": 0,"name": "Home","path": null}], "recursive": true }, "context": "ASSET" }, "queryDisplay": {"displayColumnList": {"displayColumn": [{"attributeColumn": true, "columnId": 2, "value": null, "width": 0 }, {"attributeColumn": true, "columnId": 55, "value": null, "width": 0 }, {"attributeColumn": true, "columnId": 54, "value": null, "width": 0 }, {"attributeColumn": true, "columnId": 24, "value": null, "width": 0 }] }, "displayMode": "PLAIN", "exploreMode": "PLAIN", "groupingAttributeList": {"groupingAttribute": [{"attributeId": 55, "order": "ascending", "value": null }] }, "sortInfoList": {"sortInfo": [{"attributeColumn": true, "columnId": 2, "order": "ascending", "value": null }] } }, "searchName": "SampleQuery_90d", "trusteeInfoList": {"trusteeInfo": null} }] } }</pre>		

Change Log

Changes in version 11.0

Changes in version 11.0 included the following:

Additions

The following additions were made to the REST API in version 11.0.

Service	Method Name	Function	Reference
Collection facade	getApplicablePrivileges	Get applicable privileges for the logged in user.	"Collection Management"
	getApplicableRoutingGroups	Get applicable routing groups for the logged in user.	
	getApplicableAttributeConstraints	Get applicable attribute constraints for the logged in user.	
	getApplicableStatusTransition	Get applicable status transitions for the logged in user.	
	getApplicableWorkflowInitialStatuses	Get applicable workflow initial statuses.	
Admin facade	updateUserEnabled	Enables/disables the Platform Server Logon for the given user.	"Administration"
	getGroupsWithGivenUserClass	Get groups with the given role.	
	getUserGroups	Get groups of a user.	
	getLdapGroups	Get LDAP groups.	
	importGroupsFromLdap	Get LDAP groups.	
	synchronizeLdapUsersGroups	Synchronize all LDAP users and groups.	
	synchronizeLdapUsersGroupsBasedOnProfile	Synchronize all LDAP users and groups.	
	createLdapSearches	Create LDAP search definitions.	
	getLdapSearch	Get LDAP search definition.	
	getAllLdapSearches	Get all saved LDAP search definitions.	
	updateLdapSearches	Update LDAP search definitions.	

CHANGE LOG

Service	Method Name	Function	Reference
	deleteLdapSearch	Delete an LDAP search definition.	
	updateTrusteeRule	Update role for the given trustee.	
	getApplicableApplicationPrivileges	Get applicable application privileges for the logged in user.	
	getApplicablecontentPrivileges	Get applicable content privileges for the logged in user.	

Changes

- The `userInfo` XML has been modified to support LDAP groups functionality.
 - The `groupInfo` XML has been modified to support LDAP groups functionality.
 - The `trusteeInfo` XML has been modified to support LDAP groups functionality.
 - The `domainValue` XML has been modified to populate a new element, `valueHierarchy` referring to the hierarchy of a domain value.
- ➔ Population of this unique domain value hierarchy is helpful in asset metadata update when the domain value id is not known and there exists multiple domain values with the same name.
- The `ldapProfileInfo` XML has been modified to support LDAP groups functionality.
 - In order to set the attribute value of a domain type attribute during check in of an asset: asset metadata update, asset checkout etc., the `attributeValue` XML has been enhanced to support the following options in priority:

- Specify domain value via hierarchy:

```
<attributeValueList>
  <attributeValue
    name="Departments">Enterprise;Server</attributeValue>
</attributeValueList>
```

- Specify domain value via name:

```
<attributeValueList>
  <attributeValue
    name="Departments">Server</attributeValue>
</attributeValueList>
```

- Specify domain value via id:

```
<attributeValueList>
  <attributeValue name="Departments"
    valueId="1"></attributeValue>
</attributeValueList>
```

- ➔ If the CDATA has a value mentioned, it will be assumed to be a domain value hierarchy first and if not resolved, then a domain value name. The `valueId`

attribute will only be considered if the CDATA is empty for that <attributeValue> node for a domain type attribute.

Deletions

- The `getOverriddenUserClass(long collectionId)` request has been removed.

In the current model, a users privileges is a union of the following roles:

- Role of the logged in user in the given collection, only if the user is mapped to the given collection.
- Role of collection mapped groups of which the logged in user is a member.

The `getApplicablePrivileges` request should be used to get applicable privileges for the logged in user for the given collection.

Changes in version 11.2

Changes

- Added support for JSON format.

Changes in version 12.0

Changes in version 12.0 included the following:

Additions

The following additions were made to the REST API in version 12.0.

Service	Method Name	Function	Reference
Asset facade	<code>generateTextIndexesOfAllAssets</code>	Generate text indexes for all assets.	"Asset Management"
	<code>generateTextIndexes</code>	Generate text indexes for a specified asset.	
Office facade	<code>getVisioPagesInfo</code>	Get metadata of all pages of a Visio asset.	"Office"
	<code>getVisioPagesInfoBasedOnURI</code>	Get metadata of all pages of a Visio asset based on URI.	
	<code>getVisioPagesInfoBasedOnStreamContext</code>	Get metadata of all pages of a Visio asset based on stream.	
	<code>getVisioPageOutput</code>	Get page output of a Visio asset.	
	<code>getVisioPageOutputBasedOnURI</code>	Get metadata of all pages of a Visio asset based on URI.	
	<code>getVisioPageOutputBasedOnStreamContext</code>	Get page output of a Visio asset based on stream.	
	<code>getPowerPointSlidesInfo</code>	Get metadata of all slides of a	

CHANGE LOG

Service	Method Name	Function	Reference
		PowerPoint asset.	
	getPowerPointSlidesInfoBasedOnURI	Get metadata of all slides of a PowerPoint asset based on URI.	
	getPowerPointSlidesInfoBasedOnStreamContext	Get metadata of all slides of a PowerPoint asset based on file stream.	
	getPowerPointSlideOutput	Get slide output of a PowerPoint asset.	
	getPowerPointSlideOutputBasedOnURI	Get slide output of a PowerPoint asset based on URI.	
	getPowerPointSlideOutputBasedOnStreamContext	Get slide output of a PowerPoint asset based on file stream.	
	getPowerPointSlideRangeOutput	Get slide range output of a PowerPoint asset.	
	getPowerPointSlideRangeOutputBasedOnURI	Get slide range output of a PowerPoint asset based on URI.	
	getPowerPointSlideRangeOutputBasedOnStreamContext	Get slide range output of a PowerPoint asset based on file stream.	
Configurat ion facade	getAllXmlMimeTypes	Get XML MimeTypes	"Configura tion"
	getXmlMimeType	Get XML MimeType	
	createXmlMimeType	Create a custom XML MimeType.	
	getMappedMimeTypes	Get mapped MimeTypes	
	mapMimeTypes	Map custom XML MimeType with ContentType.	
	unmapMimeTypes	Unmap custom XML MimeType with ContentType.	
	deleteXmlMimeType	Delete custom XML MimeType.	
	getPublishingProcess	Get publishing process	
	createPublishingProcess	Create publishing process	
	deletePublishingProcess	Delete publishing process.	
	getAllPublishingChannelDefinitions	Get all publishing channels	
	getPublishingChannelDefinition	Get publishing channel	
	getPublishingChannelDefinitions	Get the publishing channel of a specific content type.	
	createpublishingChannels	Create new publishing channels	
	createContentChannelMappings	Map publishing channel with content type for publishing.	
deleteContentChannelMappings	Delete publishing channel and content type mapping.		
createIndexingChannelMapping	Map publishing channel with content		

Service	Method Name	Function	Reference
	g	type for indexing.	
	getMappedIndexingChannel	Get mapped indexing channels for a given content type.	
	deleteIndexingChannelMapping	Delete publishing channel and content type mapping for indexing.	
	deletePublishingChannels	Delete publishing channel.	
	getDeploymentProperties	Get deployment properties	
	getDeploymentVersion	Get deployment version	
	reInitialize	ReInitialize the MimeType	

Changes in version 13.0

Changes in version 13.0 included the following:

The `PublishingFacade` bean API has been updated which could result in a compile error if using this as a Spring bean.

The method signatures in `PublishFacade` and `PublishByPathFacade` were modified by adding the `HttpServletRequest` argument. This affects the following REST calls:

- `publishAsset`
- `publishBasedOnStreamContext`
- `publishBasedOnURI`
- `executeDatadocChannel`

This argument sets the appropriate content-type header values in the http response for publishing requests.

Although this change impacts no REST URLs, any custom code using these methods by injecting `PublishFacade` and `PublishByPathFacade` as a Spring bean will encounter compile errors. These errors can be avoided by passing the argument as null.

Additions

The following additions were made to the REST API in version 13.0.

Service	Method Name	Function	Reference
Office facade	getExcelChart	Get Microsoft Excel chart	"Office"
	getExcelChartBasedOnStream	Get Microsoft Excel chart based on stream	

CHANGE LOG

Service	Method Name	Function	Reference
	getExcelChartBasedOnURI	Get Microsoft Excel chart based on URI	
	getExcelTable	Get Microsoft Excel table	
	getExcelTableBasedOnStream	Get Microsoft Excel table based on stream	
	getExcelTableBasedOnURI	Get Microsoft Excel table based on URI	
Clipboard facade	addClipboardData	Add data to clipboard	"Clipboard"
	addFormattedData	Add formatted data	
	getAllClipboardChannelsMapping	Get all clipboard format to channel mappings	
	getAllClipboardDataInfos	Get all clipboard data information.	
	getAllClipboardFormats	Get all clipboard formats	
	getClipboardDataInfo	Get clipboard data information	
	getClipboardChannelsMappings	Get clipboard channel mappings for content type	
	getFormattedData	Get formatted data	
	getLatestClipboardData	Get latest clipboard data	
	getLatestClipboardDataInfo	Get latest clipboard data information	
	getLatestFormattedData	Get latest formatted data	

Changes in version 13.1

Changes in version 13.1 included the following:

Additions

The following additions were made to the REST API in version 13.1.

Service	Method Name	Function	Reference
Clipboard facade	promoteClipboardDataToAsset	Promote clipboard data to asset.	"Clipboard"

Contacting Quark

The support portal allows you to log support tickets, track tickets, receive status notifications, chat with a technical support representative, search the Knowledge Base, and access product documentation.

With direct access to documentation across all Quark software - from QuarkXPress and App Studio to Quark Enterprise Solutions - you can find answers to your questions at your convenience. Our support team is also available to help, either through our support portal, or via phone for our maintenance contract customers.

If you are a Quark customer and have a current maintenance or support contract your account has already been created for you using your registered email address. If you do not have a support contract you can purchase a single support incident to get your problem resolved. If you have purchased or registered a supported product, you are eligible for free support for the first 90 days.

In the Americas

For more details, please check out our support website www.quark.com/support

Outside the Americas

For countries outside the Americas, please visit the following sites to access your support account:

Support Website

- France - www.quark.com/fr/support
- Germany - www.quark.com/de/support

Legal notices

© 2022 Quark Software Inc. and its licensors. All rights reserved.

Protected by the following United States Patents: 5,541,991; 5,907,704; 6,005,560; 6,052,514; 6,081,262; 6,633,666 B2; 6,947,959 B1; 6,940,518 B2; 7,116,843; 7,463,793; and other patents pending.

Quark, the Quark logo, Quark Publishing Platform are trademarks or registered trademarks of Quark Software Inc. and its affiliates in the U.S. and/or other countries. All other marks are the property of their respective owners.