Oracle®

Pathways

Developer Guide

Version 10g Release 3 (10.3)
Revised: November 2008

Contents

1. About the Pathways API

Copyright

Oracle Pathways Developer Guide, 10g Release 3 (10.3)

CHAPTER 1

About the Pathways API

The Pathways API allows a client application to query and manage Pathways objects through simple HTTP requests.

The Pathways API is a REST-based API. All requests are executed through HTTP using specific URL patterns and standard parameters.

The Pathways API supports the representation of request and response payloads in two formats: JSON (JavaScript Object Notation) or XML. HTTP requests can designate this preference in the Accept: header (Accept: application/xml or Accept: application/json). Additionally, GET type requests allow this preference to be passed as a query string argument (format=XML or format=JSON). The JSON format might be easier to work with in the context of a JavaScript call, while XML is usually preferable from application server code. For example, the responses below are returned from GET /pathways/api/privileges?format=XML and GET /pathways/api/privileges?format=JSON respectively.

XML response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PPrivileges xmlns:soc="http://social.bea.com/pathways">
 <access>HAS_PORTAL_ACTIVITY_ACCESS_PATHWAYS</access
 ><access>CAN_VIEW_TAGS</access><
 access>HAS_APPLICATION_WRITE</access>
</soc:PPrivileges>
```

JSON response:

```
{"soc$PPrivileges":{"@xmlns":{"soc":"http:\/\/social.bea.com\/pathways"},
"access":[{"access":"HAS_PORTAL_ACTIVITY_ACCESS_PATHWAYS"},
{"access":"CAN_VIEW_TAGS"},
{access":"HAS_APPLICATION_WRITE"}]}}
```

The base URL for all requests is the URL of the Remote Server object defined for the Web Service in the portal, plus “/pathways/api/”. (For details on configuring a Remote Server, see the portal online help.)

- For a complete list of URL patterns for specific requests, see *Pathways API Interfaces* on page 8.
- For examples of common requests, see the following topics:
  - *Retrieving Data Using the Pathways API* on page 20
  - *Executing Searches Using the Pathways API* on page 17
  - *Creating and Managing Saved Searches Using the Pathways API* on page 28
  - *Creating and Managing Tags Using the Pathways API* on page 24
  - *Creating and Managing Views Using the Pathways API* on page 30

Note: All HTTP requests to the Pathways API must include security credentials. There are two ways of providing these credentials:

- To use the standard REST API URL in the form of http://server/pathways/api/items, you must provide a portal login token. The login token can be borrowed from a portal-to-portlet request or retrieved using the IDK. For details on configuring the portal to send a login token, see *Configuring Web Services that Use the Pathways API* on page 31. For details on using the IDK to access a login token, see the *Oracle WebCenter Interaction Web Service Development Guide*.
- To use the /direct/ URL pattern (http://server/pathways/api/direct/items), you must provide username/password credentials in a traditional basic authentication format. For details on configuring the portal to send basic authentication information, see *Configuring Web Services that Use the Pathways API* on page 31.

For details on Pathways objects, see the Pathways documentation.

Pathways API Interfaces

The Pathways API includes a collection of interfaces for accessing and managing items, users, views, tags, and saved searches.

Function	URL (Interface)	HTTP Method	Method Parameters
Retrieve the version of the Pathways API.	/pathways/api/version	GET	
Retrieve suggested tags.	/pathways/api/suggest/[tag-prefix]	GET	
Retrieve tags created after a specific date.	/pathways/api/tags/created-after	GET	`date=[encoded date]`
Retrieve tags applied to a specific item.	/pathways/api/item/[id]/tags	GET	

Remove a tag from a specific item.	/pathways/api/item/[id]/tag	DELETE	`<soc:PTagList xmlns:soc="http://social.oracle.com/pathways">`
			`<tag>`
			`<value>TagToRemove</value>`
			`</tag>`
			`</soc:PTagList>`

Add a tag to a specific item.	/pathways/api/item/[id]/tag	POST	`<soc:PTagList xmlns:soc="http://social.oracle.com/pathways">`
			`<tag>`
			`<value>TagToAdd</value>`
			`</tag>`
			`</soc:PTagList>`

Retrieve a tag cloud for a search	/pathways/api/tags/[viewid] [viewid] must be one of the following: • “my” for the current user's tags • “ev” for everyone's tags • a custom view ID	GET	Search query. For a list of query parameters, see *Pathways API Search Query Parameters* on page 14.

Function	URL (Interface)	HTTP Method	Method Parameters
Retrieve the list of saved searches.	/pathways/api/savedsearches	GET	
Retrieve the details of an existing saved search.	/pathways/api/savedsearch/[id]	GET	
Remove an existing saved search	/pathways/api/savedsearch/[id]	DELETE	

Function	URL (Interface)	HTTP Method	Method Parameters
Edit an existing saved search.	/pathways/api/savedsearch/[id]	PUT	<soc:PSavedSearchesList xmlns:soc="http://social.bea.com/pathways"> <savedSearch> <created>2007-10-09T16:58:16.530-07:00</created> <dailyEmail>false</dailyEmail> <ownerId>PTPORTAL:1:1</ownerId> <queryString>Tabloid News</queryString> <rssFormat>false</rssFormat> </savedSearch> </soc:PSavedSearchesList>
Create a new saved search.	/pathways/api/savedsearch	POST	<soc:PSavedSearchesList xmlns:soc="http://social.bea.com/pathways"> <savedSearch> <created>2007-10-09T16:58:16.530-07:00</created> <dailyEmail>false</dailyEmail> <ownerId>PTPORTAL:1:1</ownerId> <queryString>Tabloid

Function	URL (Interface)	HTTP Method	Method Parameters

			`News</queryString>`
			`<rssFormat>false</rssFormat>`
			`</savedSearch>`
			`</soc:PSavedSearchesList>`

Get the list of existing views.	/pathways/api/views	GET	
Get the details of a specific view.	/pathways/api/view/[id]	GET	
Remove an existing view.	/pathways/api/view/[id]	DELETE	

Edit an existing view.	/pathways/api/view/[id]	PUT	`<soc:PViewDetail`
			`xmlns:soc="http://social.oracle.com/pathways">`
			`<members>PTPORTAL:1:203</members>`
			`<name>Marketing view</name>`
			`<published>true</published>`
			`<subscribers>PTPORTAL:1:203,PTPORTAL:1:204,PTPORTAL:1:1</subscribers>`
			`</soc:PViewDetail>`

Create a new view.	/pathways/api/view	POST	Expressed in XML payload.`name=[view name]members=[comma separated list of user/group ids]subscribers=[comma separated list`

Function	URL (Interface)	HTTP Method	Method Parameters
			`of user/group ids]`
Get a list of items.	/pathways/api/items	GET	Search query. For a list of query parameters, see *Pathways API Search Query Parameters* on page 14.
Get details of a specific item.	/pathways/api/item/[ptsource]/[id]	GET	
Retrieve the favorite state for an item or contact state for a user.	/pathways/api/isfavorite/[id]	GET	

Modify the favorite state for an item or contact state for a user.	/pathways/api/isfavorite/[id]	PUT	`<soc:PIsItemFavorite`
			`xmlns:soc="http://social.oracle.com/pathways">`
			`<result>true</result>`
			`</soc:PIsItemFavorite>`

Get a list of experts.	/pathways/api/people	GET	Search query. For a list of query parameters, see *Pathways API Search Query Parameters* on page 14.
Get details of a specific expert.	/pathways/api/people/[id]	GET	
Get access privileges for a specific user.	/pathways/api/privileges	GET	

PUT and DELETE methods can be overloaded by using a POST method with specific headers to resolve issues with browser support for asynchronous PUT and DELETES.

- `X-HTTP-Method-Override: PUT` can be used in a POST request header to signal that a PUT operation is intended.
- `X-HTTP-Method-Override: DELETE` can be used in a POST request header to signal that a DELETE operation is intended.

All requests should include the parameter `format=XML`.

Pathways API Search Query Parameters

The Pathways API provides a set of standard query parameters to execute search queries.

Query Parameter	Purpose	Valid Values	Default Value
order	Defines the order of the search results.	• "ASC" — Ascending • "DES" — Descending	“DES” (Descending)
orderBy	Designates result field for sorting.	• "RANK" • "CONTENTTYPE" • "CONTENTURL" • "DESCRIPTION" • "DETAILSURL" • "LAST_MODIFIED • "NAME" • "PATH" • "PROJECT" • "PROJECTURL" • "TAGS"	“RANK” (The search score for each item.)
tagCount	Defines the maximum number of tags to return. (Applies only to tag queries.)	int	50

Query Parameter	**Purpose**	**Valid Values**	**Default Value**
startAt	Offsets starting result within complete result set.	0 > int < total result count	1
resultCount	Defines the maximum number of results to return.	int > 0	10
query	Establishes the search query itself.	Any terms with the existing search syntax	(everything)
viewId	Limits result set to an existing View.	Existing View ID	(everyone)
resultType	Limits result set to items of a particular type.	• "ALL" • "DOCS_ONLY" • "USERS_ONLY"	"ALL" (both documents and users)
onlyFavorites	Limits result set to Favorites/Contacts.	"true" or "false"	"false"
kdFolders	Limits result set to items in specific Knowledge Directory folders.	Comma delimited list of folder IDs	(all — no constraint)
adminFolders	Limits result set to items in specific Administrative folders.	Comma delimited list of folder IDs	(all — no constraint)
collabProjects	Limits result set to items in specific Collaboration projects.	Comma delimited list of project IDs	(all — no constraint)
newerThan	Limits search result to items created since a specific date.	URL-encoded short form date ("11/7/2007")	(all — no constraint)

Query Parameter	Purpose	Valid Values	Default Value
objectTypes	Restricts search to specific object types.	Comma delimited list of any of the following strings: • AdminFolders • AuthenticationSources • CollabDiscussion • CollabDocument • CollabEvent • CollabMessage • CollabProject • CollabTask • Communities • CommunityPages • CommunityTemplates • ContentSources • ContentTypes • Crawlers • DocumentFolders • ExperienceDefinitions • ExternalOperations • FederatedSearches • Filters • Groups • Invitations • Jobs • PageTemplates • PortalDocuments • Portlets • PortletTemplates • ProfileSources • Properties • PublisherItems • RemoteServers	(all — no constraint)

Query Parameter	Purpose	Valid Values	Default Value
		• SnapshotQueries • Users • WebServices • FromUserPrefs (object types from Pathways user preferences) • FromAdminPrefs (object types pulled from Pathways admin preferences)	
propertyIDs	Designates indexed properties by ID to include with each search result.	Comma delimited property IDs (for example, “210,215” or “PT210,PT215”)	ID, Name, Description, Folder, URL, Last Modified, Activity Rank
propertyNames	Designates indexed properties by name to include with each search result.	Comma delimited property names (for example, “Office,Sales+Region”)	ID, Name, Description, Folder, URL, Last Modified, Activity Rank

Executing Searches Using the Pathways API

To search for items, users, and tags, use a search interface with standard query parameters.

The Pathways API allows you to search for items in Oracle WebCenter Framework Interaction and Oracle WebCenter Collaboration, as well as for users (experts). All search queries use a standard set of query parameters. For a detailed list of parameters, see *Pathways API Search Query Parameters* on page 14. You can also retrieve specific items or lists of items without using a search query; for details, see *Retrieving Data Using the Pathways API* on page 20. To create and manage saved searches, see *Creating and Managing Saved Searches Using the Pathways API* on page 28.

- To search for items, use GET /pathways/api/items. You can search for items in the portal Directory and Administration, Collaboration, and Publisher. To restrict the search to specific types of items, use the objectTypes query parameter.
- To search for experts, use GET /pathways/api/people.
- To retrieve a tag cloud for search, use GET /pathways/api/tags.

For example, the request below submits a search for items containing the text “Planetary Science” with the result set constrained to two items:

```
GET
/pathways/api/items?format=XML&query=planetary+science&resultCount=2
```

The response contains the search results, as shown in the example below:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PResultList xmlns:soc="http://social.bea.com/pathways">
    <item>
        <activityRank>0.0</activityRank>
        <classId>18</classId>
       <description>...featured a hit piece on Dr. S. Fred Singer,
 the esteemed Professor Emeritus of environmental &lt;span
class=&quot;ptHighlight&quot;&gt;science&lt;/span&gt; at the
University of Virginia...featured a hit piece on Dr. S. Fred
Singer, the esteemed Professor Emeritus of environmental &lt;span
 class=&quot;ptHighlight&quot;&gt;science&lt;/span&gt; at the
University of Virginia. In a segment disgracefully entitled
&amp;quot;Welcome to 'The Denial Machine...scientists that disagree
 with what your saying. The IPCC, NASA, NOAA, the National Academy
 of &lt;span
class=&quot;ptHighlight&quot;&gt;Sciences&lt;/span&gt;, the
American Association for the Advancement of Science, the American
 Geophysical Union, the American Meteorological...cyclicity in
varves and turbidites from the Arabian Sea: hypothesis of tidal
origin (Global and &lt;span
class=&quot;ptHighlight&quot;&gt;Planetary&lt;/span&gt; Change,
Volume 34, Issues 3-4, Pages 313-325, November 2002) - W. H.
Bergera, U...M. Kauffman, Ph.D. Organic Chemistry, MIT, USA Joel
 Schwartz, B.S. Chemistry, M.S. &lt;span
class=&quot;ptHighlight&quot;&gt;Planetary&lt;/span&gt; &lt;span
 class=&quot;ptHighlight&quot;&gt;Science&lt;/span&gt;, California
 Institute of Technology, USA John Brignell, Ph.D. Professor
Emeritus, Department of Electronics &amp;amp;
Computer...</description>
        <folderClassId>17</folderClassId>
        <folderId>200</folderId>
        <folderName>\Knowledge Directory\Test</folderName>

<iconLink>plumtree/portal/public/img/sml{52a5a5a0-84c9-11d2-a0c5-0060979c42d8}.gif</iconLink>

        <isFavorite>false</isFavorite>
        <lastModified>2008-03-24T05:38:00-07:00</lastModified>
       <name>ABC's Global Warming Hit Piece &quot;Welcome to 'The
 Denial Machine'&quot; | NewsBusters.org</name>
        <objectId>8267</objectId>
        <path>\Knowledge Directory\Test</path>
        <rank>1</rank>
        <resultScore>2.84252</resultScore>
        <source>PTPORTAL</source>
        <type>Document</type>

<URL>http://smahendr03.amer.bea.com:8080/portal/server.pt?open=18&amp;objID=8267&amp;mode=2&amp;in_hi_userid=1</URL>


<viewDetailURL>http://smahendr03.amer.bea.com:8080/portal/server.pt?objID=8267&amp;open=18&amp;mode=3&amp;subspace=ySubspaceKey&amp;space=Opener</viewDetailURL>

    </item>
    <item>
        <activityRank>0.0</activityRank>
        <classId>18</classId>
        <description>...Coverage of &lt;span
class=&quot;ptHighlight&quot;&gt;science&lt;/span&gt; by
SignOnSanDiego.com and The San Diego Union-Tribune...Pension Crisis
 Special Reports Video Multimedia Photo Galleries Topics Education
 Features Health | Fitness Military Politics &lt;span
class=&quot;ptHighlight&quot;&gt;Science&lt;/span&gt; Solutions
Opinion Columnists Steve Breen Forums Weblogs Communities U-T
South County U-T East...asteroid to determine whether it is on a
 potentially disastrous collision course with Earth, The &lt;span
 class=&quot;ptHighlight&quot;&gt;Planetary&lt;/span&gt; Society
 announced Tuesday. WASHINGTON, 2:01 p.m. Feb. 25 (AP) Limiting
fish catch to...Museum of Man San Diego Natural History Museum
SDSU College of Engineering SDSU College of &lt;span
class=&quot;ptHighlight&quot;&gt;Sciences&lt;/span&gt; San Diego
 Supercomputer Center San Diego Zoological Society Scripps
Institution of Oceanography Scripps Research
Institute...</description>
        <folderClassId>17</folderClassId>
        <folderId>200</folderId>
        <folderName>\Knowledge Directory\Test</folderName>
```

```
<iconLink>plumtree/portal/public/img/sml{52a5a5a0-84c9-11d2-a0c5-0060979c42d8}.gif</iconLink>

        <isFavorite>false</isFavorite>
        <lastModified>2008-03-24T05:41:00-07:00</lastModified>
        <name>SignOnSanDiego.com &gt; News &gt; Science</name>
        <objectId>9238</objectId>
        <path>\Knowledge Directory\Test</path>
        <rank>2</rank>
        <resultScore>2.68074</resultScore>
        <source>PTPORTAL</source>
        <type>Document</type>

<URL>http://smahendr03.amer.bea.com:8080/portal/server.pt?open=18&amp;objID=9238&amp;mode=2&amp;in_hi_userid=1</URL>


<viewDetailURL>http://smahendr03.amer.bea.com:8080/portal/server.pt?objID=9238&amp;open=18&amp;mode=3&amp;subspace=mySubspaceKey&amp;space=Open</viewDetailURL>

    </item>
    <totalCount>102</totalCount>
</soc:PResultList>
```

Retrieving Data Using the Pathways API

To retrieve data about items, views, saved searches and users (experts), use one of the interfaces provided by the Pathways API.

The Pathways API provides a collection of interfaces to retrieve data.

- To retrieve details about a specific item in the portal, Collaboration, or Publisher, use GET /pathways/api/item/[ptsource]/[id]. For example, the following request retrieves details on a portal item with an object ID of 201: `GET /pathways/api/item/PTPORTAL/18_201?format=XML`.

  The response contains detailed information about the item:

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PItemDetail xmlns:soc="http://social.bea.com/pathways">
      <type>PTPORTAL</type>
      <classId>18</classId>
      <objectId>201</objectId>
      <name>APOD: 2007 October 12 - The Whale and the Hockey
  Stick</name>
  ```

```
   <description>A different astronomy and space science related
 image is featured each day, along with a brief
explanation.</description>
    <lastModified>2008-02-25T02:33:00-08:00</lastModified>

<URL>http://smahendr03.amer.bea.com:8080/portal/server.pt?open=18&amp;objID=201&amp;mode=2&amp;in_hi_userid=1</URL>


<viewDetailURL>http://smahendr03.amer.bea.com:8080/portal/server.pt?objID=201&amp;open=18&amp;mode=3&amp;subspace=mySubspaceKey&amp;space=Opener</viewDetailURL>

    <path>\Knowledge Directory\Test</path>
    <folderId>200</folderId>
    <folderName>\Knowledge Directory\Test</folderName>
    <property>
        <id>PT8</id>
        <name>Content Language</name>
        <value>en</value>
    </property>
    <property>
        <id>PT6</id>
        <name>Content Type ID</name>
        <value>106</value>
    </property>
    <property>
        <id>PT2</id>
        <name>Description</name>
        <value>A different astronomy and space science related
 image is featured each day, along with a brief
explanation.</value>
    </property>
    <property>
        <id>PT1</id>
        <name>Name</name>
       <value>APOD: 2007 October 12 - The Whale and the Hockey
 Stick</value>
    </property>
    <property>
        <id>PT3</id>
        <name>Object Created</name>
        <value>4232743.0</value>
    </property>
    <property>
        <id>PT4</id>
        <name>Object Last Modified</name>
        <value>4287993.0</value>
    </property>
    <property>
        <id>PT5</id>
        <name>Open Document URL</name>
        <value>http://apod.nasa.gov/apod/ap071012.html</value>

    </property>
</soc:PItemDetail>
```

You can also find out whether a specific item has been marked as a favorite using GET /pathways/api/isfavorite/[id]. For details on searching for items, see *Executing Searches Using the Pathways API* on page 17.

- To retrieve details about an existing view by ID, use GET /pathways/api/view/[id]. For example, `GET /pathways/api/view/8aa289731303557b0113086b1ece0003?format=XML`. (To retrieve a list of existing saved searches, use GET /pathways/api/views?format=XML.)

  The response contains detailed information about the view:

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PViewDetail xmlns:soc="http://social.bea.com/pathways">
      <created>2007-06-07T16:00:44.367-07:00</created>
      <id>8aa289731303557b0113086b1ece0003</id>
      <lastModified>2008-04-03T08:30:35.807-07:00</lastModified>

      <members>PTPORTAL:1:203</members>
      <name>Marketing view</name>
      <ownerId>PTPORTAL:1:1</ownerId>
      <published>true</published>

  <subscribers>PTPORTAL:1:203,PTPORTAL:1:204,PTPORTAL:1:1</subscribers>
  </soc:PViewDetail>
  ```

  For details on creating new views, see *Creating and Managing Views Using the Pathways API* on page 30.

- To retrieve details about an existing saved search by ID, use GET /pathways/api/savedsearch/[id]. For example, `GET /pathways/api/savedsearch/8aa289731587336f011587345c110001?format=XML`. (To retrieve a list of existing saved searches, use GET /pathways/api/savedsearches?format=XML.)

  The response contains detailed information about the saved search:

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PSavedSearchesList
  ```

```
xmlns:soc="http://social.bea.com/pathways">
    <savedSearch>
        <created>2007-10-09T16:58:16.593-07:00</created>
        <dailyEmail>true</dailyEmail>
        <id>8aa289731587336f011587345c110001</id>
        <language>en</language>

<lastModified>2008-02-22T16:07:32.583-08:00</lastModified>
        <ownerId>PTPORTAL:1:1</ownerId>
        <queryString>news</queryString>
        <rssFormat>false</rssFormat>
        <timeZoneId>America/Los_Angeles</timeZoneId>
    </savedSearch>
</soc:PSavedSearchesList>
```

For details on creating new saved searches, see *Creating and Managing Saved Searches Using the Pathways API* on page 28.

- To retrieve details about an existing user (expert) by ID, use GET /pathways/api/people/[id]. For example, `GET /pathways/api/people/PTPORTAL/1_201?format=XML`.

  The response contains detailed information about the user:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PItemDetail xmlns:soc="http://social.bea.com/pathways">
    <type>PTPORTAL</type>
    <name>user1</name>
    <description></description>
    <lastModified>2008-01-14T07:34:00-08:00</lastModified>

<URL>http://smahendr03.amer.bea.com:8080/portal/server.pt?open=1&amp;objID=201&amp;mode=2&amp;in_hi_userid=1</URL>

    <path>\Admin Objects Directory\Test Resources</path>
    <folderId>306</folderId>
    <folderName>\Admin Objects Directory\Test
Resources</folderName>
    <property>
        <id>PT201</id>
        <name>Secret Property</name>
        <value></value>
    </property>
    <property>
        <id>PT156</id>
        <name>Company</name>
        <value></value>
    </property>
    <property>
        <id>PT152</id>
        <name>Phone Number</name>
        <value></value>
    </property>
    <property>
        <id>PT26</id>
        <name>Email Address</name>
        <value></value>
    </property>
</soc:PItemDetail>
```

You can retrieve the current user's access privileges using GET /pathways/api/privileges. For example, `GET /pathways/api/privileges?format=XML`. The response lists all available privileges:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PPrivileges xmlns:soc="http://social.bea.com/pathways">
    <access>HAS_PORTAL_ACTIVITY_ACCESS_PATHWAYS</access>
    <access>CAN_VIEW_TAGS</access>
    <access>CAN_VIEW_PEOPLE</access>
    <access>CAN_ADD_TAGS</access>
    <access>CAN_CREATE_VIEWS</access>
    <access>CAN_PRESCRIBE_VIEWS</access>
    <access>HAS_MY_VIEW</access>
    <access>HAS_EVERYONE_VIEW</access>
    <access>CAN_ADD_FAVORITES_CONTACTS</access>
    <access>CAN_SET_USER_PREFS</access>
    <access>CAN_ACCESS_ENHANCED_UI</access>
    <access>CAN_SAVE_SEARCHES</access>
    <access>HAS_APPLICATION_WRITE</access>
</soc:PPrivileges>
```

You can also find out whether a specific expert has been marked as a contact of the current user, use GET /pathways/api/isfavorite/[id].

Creating and Managing Tags Using the Pathways API

To create new tags and assign tags to specific items, use the /pathways/api/item/[id]/tag interface.

The Pathways API provides a collection of interfaces to access and manage tags.

- To retrieve tags that contain a specific string, use GET /pathways/api/suggest/[tag-prefix]. For example, the following request retrieves tags that start with “port”: `GET /pathways/api/suggest/port?format=xml`.
- To retrieve a weighted tag cloud, use GET /pathways/api/tags/[viewid] where [viewid] is “my” for the current user's tags, “ev” for everyone's tags, or a custom view ID.
- To retrieve tags applied to a specific item, use GET /pathways/api/item/[id]/tags. For example, the following request retrieves tags associated with the portal item with ID 1985: `GET /pathways/api/item/PTPORTAL/18_1985/tags?format=XML`.

  The response provides a list of tags:

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PTagList xmlns:soc="http://social.bea.com/pathways">
      <tag>
          <count>1</count>
          <isAppliedByCurrentUser>true</isAppliedByCurrentUser>
          <normalizedCount>9</normalizedCount>
          <value>phenomenon</value>
      </tag>
      <tag>
          <count>1</count>
          <isAppliedByCurrentUser>true</isAppliedByCurrentUser>
          <normalizedCount>9</normalizedCount>
          <value>sudden</value>
      </tag>
  </soc:PTagList>
  ```

- To add a tag from a specific item, use POST /pathways/api/item/[id]/tag as shown in the example below.

  The following request creates tags ‘foo’ and ‘bar’ and assigns them to portal document 210:

  ```
  POST /pathways/api/item/PTPORTAL/1_210/tags
  CSP-Session-Token: 1|1207243836|0d6PFr/Z+mRrarbn9uhZJyYL2uI=
  Host: smahendr03.amer.bea.com:11990
  Accept-Language: en-us
  ACCEPT: application/xml

  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PTagList xmlns:soc="http://social.bea.com/pathways">
      <tag>
          <value>foo</value>
      </tag>
      <tag>
          <value>bar</value>
  ```

```
    </tag>
</soc:PTagList>
```

The response confirms that the request was successful:

```
HTTP/1.1 201 Created

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PTagList xmlns:soc="http://social.bea.com/pathways">
    <tag>
        <value>foo</value>
    </tag>
    <tag>
        <value>bar</value>
    </tag>
</soc:PTagList>
```

This same request could be used with the HTTP DELETE method to remove these tags from the item. The successful response code would be 202 Accepted instead of 201 Created.

- To delete a tag from multiple items within a specific view, use DELETE /pathways/api/tag/[viewid]/[tag] where [viewid] is “my” for the current user's tags, “ev” for everyone's tags, or a custom view ID; and [tag] is the tag value.

  Users with WRITE_APPLICATION capability can delete tags that they applied or tags applied by other users. For example, an administrator could remove all tags corresponding to "acquisition" for everyone (the "ev" view) or, more narrowly, for another group view, or for a specifically identified user (“custom/[ptsource]/[userid]”).

  For example, the following request deletes tag ‘foo’ from all items in view ‘bar’: `DELETE /pathways/api/tag/bar/foo`.

  The response confirms that the request was successful:

  ```
  HTTP/1.1 202 Accepted

  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PTagViewStatus xmlns:soc="http://social.bea.com/pathways">

      <value>foo</value>
      <appliedItem>PTPORTAL:18:201</appliedItem>
      <appliedItem>PTPORTAL:18:203</appliedItem>
  </soc:PTagViewStatus>
  ```

  In the example above, ‘foo’ was removed from Knowledge Directory items with object IDs of 201 and 203.

- To rename a tag applied to multiple items within a specific view, use PUT /pathways/api/tag/[viewid]/[tag] where [viewid] is “my” for the current user's tags, “ev” for everyone's tags, or a custom view ID; and [tag] is the tag value.

  Important: The "appliedItem" elements are returned in the response to the 'rename' request, but they should not be passed when making the request. If you add the “appliedItem” element to the request, an error is produced indicating how the call should be invoked.

  Users with WRITE_APPLICATION capability can rename tags that they applied or tags applied by other users. For example, an administrator could rename all “HR” tags to “human resources” for everyone (the "ev" view) or, more narrowly, for another group view, or for a specifically identified user (“custom/[ptsource]/[userid]”).

  For example, the following request renames tag ‘foo’ to ‘bar’ from all items in view ‘ev’:

  ```
  PUT /pathways/api/tag/ev/foo
  CSP-Session-Token: 1|1207243836|0d6PFr/Z+mRrarbn9uhZJyYL2uI=
  Host: smahendr03.amer.bea.com:11990
  Accept-Language: en-us
  ACCEPT: application/xml

  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PTagViewStatus xmlns:soc="http://social.bea.com/pathways">

      <value>bar</value>
  </soc:PTagViewStatus>
  ```

  The response confirms that the request was successful:

  ```
  HTTP/1.1 202 Accepted

  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PTagViewStatus xmlns:soc="http://social.bea.com/pathways">

      <value>bar</value>
      <appliedItem>PTPORTAL:18:201</appliedItem>
      <appliedItem>PTPORTAL:18:203</appliedItem>
  </soc:PTagViewStatus>
  ```

  In the example above, the tag ‘bar’ replaced the tag ‘foo’ for Knowledge Directory items with object IDs of 201 and 203.

Creating and Managing Saved Searches Using the Pathways API

To create new saved searches and modify existing saved searches, use the /pathways/api/savedsearch interfaces.

The Pathways API provides a collection of interfaces to access and manage saved searches.

- To retrieve the list of existing saved searches, use GET /pathways/api/savedsearches.
- To retrieve details about a specific saved search, use GET /pathways/api/savedsearch/[id]. For details, see *Retrieving Data Using the Pathways API* on page 20.
- To create a new saved search, use POST /pathways/api/savedsearch/new. For example, the request below creates a new saved search for “gravity”.

```
POST /pathways/api/savedsearch/new
CSP-Session-Token: 1|1207329770|lKh5QpON3TQykMZ3a2x4ygh6Drs=
Accept-Language: en-us
ACCEPT: text/xml,application/xml

<?xml version="1.0" encoding="UTF-8"?>
<soc:PSavedSearchesList
xmlns:soc="http://social.bea.com/pathways">
  <savedSearch>
<created>2007-10-09T16:58:16.593-07:00</created>
<dailyEmail>true</dailyEmail>
<ownerId>PTPORTAL:1:1</ownerId>
<queryString>Gravity</queryString>
<rssFormat>false</rssFormat>
  </savedSearch>
</soc:PSavedSearchesList>
```

  The response confirms that the saved search was created and includes the details passed in the request.

```
HTTP/1.1 201 Created
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PSavedSearchesList
xmlns:soc="http://social.bea.com/pathways">
    <savedSearch>
        <created>2007-10-09T16:58:16.593-07:00</created>
        <dailyEmail>true</dailyEmail>
        <ownerId>PTPORTAL:1:1</ownerId>
        <queryString>Gravity</queryString>
        <rssFormat>false</rssFormat>
    </savedSearch>
</soc:PSavedSearchesList>
```

- To modify an existing saved search, use PUT /pathways/api/savedsearch/[id] as shown in the example request below.

  ```
  PUT /pathways/api/savedsearch/8aa28973157e6c3901157e74bab70001
  CSP-Session-Token: 1|1207331374|2/XMqYsZj6DwW5b220VWnB2o4S8=
  Accept-Language: en-us
  ACCEPT: text/xml,application/xml

  <?xml version="1.0" encoding="UTF-8"?>
  <soc:PSavedSearchesList
  xmlns:soc="http://social.bea.com/pathways">
    <savedSearch>
  <created>2007-10-09T16:58:16.593-07:00</created>
  <dailyEmail>true</dailyEmail>
  <ownerId>PTPORTAL:1:1</ownerId>
  <queryString>New Search Target</queryString>
  <rssFormat>false</rssFormat>
    </savedSearch>
  </soc:PSavedSearchesList>
  ```

  If the update was successful, the response code will be HTTP 202 Accepted. As with creating a new saved search, the response also includes the details passed in the request.

- To delete an existing saved search, use DELETE /pathways/api/savedsearch/[id]. There is no need to include details about the saved search, as shown in the example request below.

  ```
  DELETE
  /pathways/api/savedsearch/8aa28973191a6c1701191a6d903f0001
  CSP-Session-Token: 1|1207330372|Ybc8ZmBc8vd7svvaa39CzZ5lp18=
  Accept-Language: en-us
  ACCEPT: text/xml,application/xml
  ```

  If the delete was successful, the response code will be HTTP 202 Accepted. The response also includes details about the saved search that was deleted, including the creation date.

Creating and Managing Views Using the Pathways API

To create new views and update existing views, use the /pathways/api/view interfaces.

The Pathways API provides a collection of interfaces to access and manage views.

- To retrieve the list of existing views, use GET /pathways/api/views.
- To retrieve details about a specific views, use GET /pathways/api/view/[id]. For details, see *Retrieving Data Using the Pathways API* on page 20.
- To create a new view, use POST /pathways/api/view/new. For example, the request below adds a new view called “Customer View”.

```
POST /pathways/api/view
CSP-Session-Token: 1|1207244890|EBNGlMeU4zFYKJIFbhIavCVk4/U=
Accept-Language: en-us
ACCEPT: text/xml,application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PViewDetail xmlns:soc="http://social.bea.com/pathways">
    <members>PTPORTAL:1:200,PTPORTAL:1:1</members>
    <name>Customer View</name>
    <ownerId>PTPORTAL:1:1</ownerId>
    <published>true</published>

<subscribers>PTPORTAL:1:201,PTPORTAL:1:1,PTPORTAL:2:201</subscribers>
</soc:PViewDetail>
```

The response confirms that the new view was created and includes the details passed in the request.

```
HTTP/1.1 201 Created
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soc:PViewDetail xmlns:soc="http://social.bea.com/pathways">
    <created>2008-04-03T10:50:55.477-07:00</created>
    <id>8aa289731910767e01191569a5f30003</id>
    <lastModified>2008-04-03T10:50:55.507-07:00</lastModified>

    <members>PTPORTAL:1:200,PTPORTAL:1:1</members>
    <name>Customer View</name>
    <ownerId>PTPORTAL:1:1</ownerId>
    <published>true</published>
```

```
<subscribers>PTPORTAL:2:201,PTPORTAL:1:201,PTPORTAL:1:1</subscribers>
</soc:PViewDetail>
```

- To modify an existing view, use PUT /pathways/api/view/[id] as shown in the example request below.

  ```
  PUT /pathways/api/view/8aa289731303557b0113086b1ece0003
  CSP-Session-Token: 1|1207332050|KHt3F4KlgmlBuntcrEKQpFcQGq8=
  Accept-Language: en-us
  ACCEPT: text/xml,application/xml

  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <soc:PViewDetail xmlns:soc="http://social.bea.com/pathways">
  <members>PTPORTAL:1:203</members>
   <name>Marketing view</name>
   <published>false</published>
  <subscribers>PTPORTAL:1:203,PTPORTAL:1:204</subscribers>
  </soc:PViewDetail>
  ```

  If the update was successful, the response code will be HTTP 202 Accepted. As with creating a new view, the response also includes the details passed in the request.

- To delete an existing view, use DELETE /pathways/api/view/[id]. There is no need to include details about the view, as shown in the example request below.

  ```
  DELETE /pathways/api/view/8aa289731910767e01191569a5f30003
  CSP-Session-Token: 1|1207330830|loGCMTzUMaJMBW4w22ACHoLUH0Y=
  Accept-Language: en-us
  ACCEPT: text/xml,application/xml
  ```

  If the delete was successful, the response code will be HTTP 202 Accepted. The response also includes details about the view that was deleted, including the creation date.

Configuring Web Services that Use the Pathways API

To send the portal login token or basic authentication information to a remote application that uses the Pathways API, register the remote application as a Web Service object in Oracle WebCenter Framework Interaction.

To use the Pathways API, a remote application must have access to the portal login token or a portal user’s basic authentication information. To send this information to the remote application, create a Remote Server and a Web Service in Oracle WebCenter Framework Interaction for the remote application and configure the following settings. (For detailed information on Remote Server and Web Service configuration settings, see the portal online help.)

- On the **Main Settings** page, enter the location of the remote application.
- On the **HTTP Configuration** page of the Web Service editor, add the root folder(s) for the remote application to the list of **Gateway URL Prefixes**. Gateway URL prefixes must include a trailing slash ("http://MyServer/") and are case sensitive. For details on the gateway, see the Developer Guide for Oracle WebCenter Framework Interaction.

  Note: All folders or pages that communicate with the Pathways API must be gatewayed. This includes the path to the service and any pages that are accessed by the service.

- To send the portal login token, on the **Advanced Settings** page, select the **Send Login Token** option. Configure the Login Token duration or leave it at the default. (Alternatively, you can choose to send the user’s basic authentication information as described next.)
- To send the user's basic authentication information, on the **Authentication Settings** page, select the **User’s Basic Authentication Information** option. (Alternatively, you can choose to send the portal login token as described in the previous bullet.)