Focal Point®

REST RDF API Reference Manual

Release 7.50

Contents

1 REST RDF API reference

Focal Point 6.5.2 and later supports REST Resource Description Framework (RDF) API, where RDF representation for IBM Focal Point is included and available for linked data. The REST RDF API uses resources that are defined by RDF XML API and few additional resources. Using these REST RDF APIs, you can consume or modify the Focal Point resources from external applications.

Focal Point supports content negotiation by using the standard HTTP Accept header values for getting the required resource representations.

Setup

The server name part of the resource URI can be configured in the Login or Balancer URL page.

To specify the server name for the resource URI, in Focal Point, from the User menu, select Administration. On the Administration page, click **Application > Login Page**; under **Login Page Settings**, click the **Edit** icon for **Login or Balancer URL**, specify the URL, and click the **Save** icon.

Note Make sure that the host name or the server name does not change. A change in the host name can lead to broken links in Focal Point integrated systems that link to the Focal Point resources.

Authentication

The requests to the RESTful API must be authenticated by using HTTP basic authentication. Unless you use HTTPS authentication, the user name and password are sent without encryption. In HTTP basic authentication, character encoding is not specified for user names and passwords. User names and passwords can include ASCII characters only. You might be able to use ISO-8859-1 characters if the characters are encoded by the client correctly.

Examples

The RESTful RDF API for IBM Focal Point is a provisional API and subject to change. Most of the RDF examples provided in this document are in the TURTLE format, see
http://www.w3.org/TeamSubmission/turtle/.

Prerequisites

You must have knowledge of RESTful APIs and RDF to work with these APIs and integrate Focal Point with your application.

Content negotiation

REST RDF API supports RDF/XML, TURTLE and N3 formats. You can request any of these formats by using the standard HTTP Accept header content negotiation.When making the REST API calls, you can use these HTTP Accept header values to specify the preferred format:

Accept header value	Preferred format	URL extension
application/rdf+xml	rdf/xml	.rdf
application/x-turtle	rdf/turtle	.ttl
application/rdf+n3	rdf/n3	.n3
application/xml	XML	.xml
application/html	HTML	.html

For example, to view resources in the RDF or XML format, use this URL in a web browser:

`http://fpserver/fp/resources/workspaces/2.rdf`

RDF vocabulary

The following Focal Point RDF vocabularies are used in RDF representations of Focal Point resources:

- Standard vocabulary

- Focal Point core vocabulary

- Focal Point data type vocabulary

- Focal Point auto-generated vocabulary

The Focal Point defined RDF vocabularies use `http://jazz.net/ns/psm` as the base URI for defining the resources and terms. Focal Point RDF resource representations also use auto-generated vocabulary for user-defined terms and terms from the Dublin Core, FOAF, OSLC, OWL, RDF, and XML Schema vocabularies.

The following table lists the vocabulary prefixes that are used in Focal Point RDF representations.

Prefix	Namespace URI	Description
dcterms:	`http://purl.org/dc/terms/`	Dublin Core vocabulary.
foaf:	`http://xmlns.com/foaf/0.1/`	Friend of a Friend (FOAF) vocabulary.

Prefix	Namespace URI	Description
owl:	`http://www.w3.org/2002/07/owl#`	Web Ontology Language (OWL) vocabulary.
rdf:	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`	RDF vocabulary.
rdfs:	`http://www.w3.org/2000/01/rdf-schema#`	RDF Schema vocabulary.
xsd:	`http://www.w3.org/2001/XMLSchema#`	XML Schema (XSD) vocabulary.
oslc:	`http://open-services.net/ns/core#`	OSLC Core vocabulary.
fps:	`http://jazz.net/ns/psm/focalpoint#`	Focal Point core vocabulary.
fpdt:	`http://jazz.net/ns/psm/focalpoint/datatypes#`	Focal Point data type vocabulary.

Standard vocabulary

Focal Point uses a few attributes that are built-in for the primary resources. For good interoperability, RDF representations reuse existing, widely-adopted vocabularies such as the Dublin Core. Focal Point maps many built-in Focal Point attributes to these standard vocabularies.

The following table lists the mapping to some of the standard properties:

Focal Point built-in attribute	Standard URI
Alias	dcterms:identifier
Created Date	dcterms:created
Creator	dcterms:creator
Description	dcterms:description
Last Changed Date	dcterms:modifed
Title	dcterms:title
Online account	foaf:OnlineAccount
User	foaf:Person
Account	foaf:account
User name	foaf:name

Focal Point built-in attribute	Standard URI
User title	foaf:title
User email	foaf:mbox
User login name	foaf:accountName

Focal Point core vocabulary

The Focal Point defined concepts and, built-in attributes that are not mapped to the standard vocabulary are defined in the Focal Point core vocabulary.

Property	Value
Namespace URI	`http://jazz.net/ns/psm/focalpoint#`
Prefix	fpdt: (Focal Point Schema) to distinguish it from fp: which is used in the XML representation
Details	*https://jazz.net/wiki/bin/view/Main/FocalPointDatatypeVocabulary*

Focal Point data type vocabulary

Focal Point provides support for a large number of data types that could be used as the values of attributes. These data types are described in their own vocabulary so that the main Focal Point vocabulary smaller and is easier to understand. For a description of the data types see *https://jazz.net/wiki/bin/view/Main/FPDatatypes*.

Property	Value
Namespace URI	`http://jazz.net/ns/psm/focalpoint/datatypes#`
Prefix	fpdt: (Focal Point data type)
Details	*https://jazz.net/wiki/bin/view/Main/FocalPointDatatypeVocabulary*

Focal Point auto generated vocabulary

In Focal Point, you can define modules, attributes and choice items. By default, these definitions are treated as terms that are defined by the user, and the vocabularies are automatically generated. Auto-generated vocabularies are of three scopes:

Workspace scope

Property	Value
Namespace URI	`http://`*`fpserver`*`/fp/resources/workspaces/wid/ns#`
Prefix	workspace_wid:
Scope	Workspace scope. Vocabulary for all of the user-defined modules in a given Focal Point workspace

Module scope

Property	Value
Namespace URI	`http://`*`fpserver`*`/fp/resources/workspaces/wid/modules/mid/ns#`
Prefix	module_mid_wid:
Scope	Module scope. Vocabulary for all of the user-defined attributes for a given Focal Point module.

Attribute scope

Property	Value
Namespace URI	`http://`*`fpserver`*`/fp/resources/workspaces/wid/modules/mid/attributes/aid/ns#`
Prefix	attribute_aid_mid_wid:
Scope	Attribute scope. Vocabulary for all of the user-defined choice items for a given Focal Point attribute.

Primary resources

The primary resources in Focal Point are service, workspace, module, view, element, and attribute. The REST XML API does not support HTTP GET operation for workspace, module and view resources. REST RDF API extends the REST XML API defined resources to define these additional resources.

These are the types and example REST API URIs for the primary resources:

Type	Example REST API URI	Description
fps:Service	`http://`*`fpserver`*`/fp/resources/`	Service document

Type	Example REST API URI	Description
fps:Workspace	`http://fpserver/fp/resources/workspaces/3`	Workspace
foaf:OnlineAccount	`http://fpserver/fp/resources/users/13`	Online account for a global user
foaf:Person	`http://fpserver/fp/resources/users/13#me`	Global user
fps:Member	`http://fpserver/fp/resources/workspaces/3/members/15`	Membership entry
fps:Module	`http://fpserver/fp/resources/workspaces/3/modules/1`	Module or Element collection
fps:Module	`http://fpserver/fp/resources/workspaces/3/modules/1?view=123`	Element collection of a View
fps:View	`http://fpserver/fp/resources/workspaces/2/modules/2/views/759`	View
fps:Element (natural)	`http://fpserver/fp/resources/workspaces/3/modules/3/elements/3`	Element, natural graph (default)
fps:Element (generic)	`http://fpserver/fp/resources/workspaces/3/modules/3/elements/3?format=genericrdf`	Element, generic graph
fps:Attribute	`http://fpserver/fp/resources/workspaces/3/modules/3/elements/3/attributes/38`	Attribute

Service document (fps:Service)

The service document lists the high level resources for workspace and user module that the current user has access to. Only users with global administrator rights can access the list of users. Referencing the workspace resource lists the modules, views, member resources of the workspace.

The service document contains information where the resources that are listed can be referenced to find subsequent resources. The resource URI for the service document is in the form `http://fpserver/context/resources/`. For example, `http://fpserver1/fp/resources/`.

Examples

```
<http://focalpointserver/fp/resources/>
a fps:Service ;
fps:memberModule <http://focalpointserver/fp/resources/users> ;
fps:memberWorkspace <http://focalpointserver/fp/resources/workspaces/2> ;
fps:rootService <http://focalpointserver/fp/resources/rootservices> ;
dcterms:description "This Service documents lists the top level resources
for Focal Point REST API" ;
dcterms:title "Focal Point REST API Services Document".
```

Focal Point primary data types

Focal Point has standard XML data types that are used for basic data types.

Focal Point data types	Mapped data types
Boolean (Check Box, Lock)	xsd:boolean
Date	xsd:date
Integer	xsd:integer
Float	xsd:decimal
Text (plain)	plain text
Text (rich)	rdf:XMLLiteral
Time (created, last modified)	xsd:dateTime

For information on how rest of the Focal Point data types are represented in RDF, see
https://jazz.net/wiki/bin/view/Main/FPDatatypes.

REST Operations

This topic explains how to use make various HTTP method calls on the REST RDF API defined
resource. For all of the REST operations, the HTTP Accept header value is appropriately set to any
one of RDF formats as mentioned in *'Content negotiation' on page 5*.

Parameters

The following parameters can used when making REST RDF API calls:

- View selection, see

- Modified since, see

- Resource paging, see *'Resource paging' on page 11*

- Selective properties, see *'Selective properties' on page 13*

- Format, see *'Format' on page 13*

View selection

Use the `view` parameter to list only the elements and attributes of the given view.

- **Parameter value**: `view_id` as integer

- **Can be used with**: Module and Element resource

- **Applicable**: HTTP GET, PUT, and POST methods

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1?view=39
```

Modified since

Use the `modifiedSince` parameter to list only the elements that are modified after the entered time value.

- **Parameter value**: `time` as xsd:date or xsd:dateTime

- **Can be used with**: Module (Element collection) resource

- **Applicable**: HTTP GET

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1.rdf?
modifiedSince=2012-06-11T09:56:48.128Z
```

Resource paging

Sometimes, an element collection is too large to be reasonably transmitted in a single HTTP message. In such cases, resource paging can be used to split the response into several pages. The following parameters can be used:

oslc.paging

Use the `oslc.paging` parameter to activate resource paging, so that the result element collection is paginated. By default, `oslc.paging` is turned off.

- **Parameter value**: boolean

- **Can be used with**: Module (Element collection) resource

- **Applicable**: HTTP GET

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1.rdf?oslc.paging=true
```

pageno

If paging is turned on, use the `pageno` parameter to request any valid page by specifying an integer value. By default, `pageno` is set to 0.

- **Parameter value**: integer

- **Can be used with**: Module (Element collection) resource

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1.rdf?
oslc.paging=true&pageno=1
```

oslc.pageSize

If paging is turned on, use the `oslc.pageSize` parameter to determine the number of elements per page.

Note You can change the default page size. From the Focal Point Admin configuration page, click **Application > Data Access > REST Page size** to specify the value.

- **Parameter value**: integer

- **Can be used with**: Module (Element collection) resource

- **Applicable**: HTTP GET

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1.rdf?
oslc.paging=true&pageno=1&oslc.pageSize=100
```

Note When paging is activated, the following resource is included in the response and it gives details such as total elements and URL for the next page.

```
<http://fpserver/fp/resources/workspaces/2/modules/1?
oslc.paging=true&pageno=0&oslc.pageSize=100>
    a oslc:ResponseInfo ;
    oslc:nextPage <http://fpserver/fp/resources/workspaces/2/modules/1?
oslc.paging=true&pageno=1&oslc.pageSize=100> ;
    oslc:totalCount 118 .
```

For details, see *http://open-services.net/resources/tutorials/oslc-primer/resource-paging/* for more details.

Selective properties

Sometimes, only a subset of attributes is required when querying for element or element collection. Use the `oslc.properties` parameter to specify a comma-delimited set of attribute names, only specified attribute values are returned in the result.

- **Parameter value**: List of comma-separated attribute names.

- **Can be used with**: Module (element collection) or Element resources

- **Applicable**: HTTP GET

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1.rdf?
oslc.properties=dcterms:title, fps:id, :version
```

Format

Element has two graphs, natural and generic. Use the `format` parameter to select the generic graph type. The default graph type is natural (`naturalrdf`).

- **Parameter value**: `genericrdf`

- **Can be used with**: Element and Element Collection resources only

- **Applicable**: HTTP GET

For example:

```
http://fpserver/fp/resources/workspaces/2/modules/1/elements/1.rdf?
format=genericrdf
```

HTTP methods

The following topics explain how to use the HTTP methods.

HTTP GET

The HTTP GET method is supported for all Focal Point REST RDF API defined resources. To request for appropriate resource format, see *'Content negotiation' on page 5*. Administrative privilege is need to access an element with a view parameter.

URI: `http://fpserver/fp/resources/workspaces/2/modules/1/elements/53.ttl?view=39&oslc.properties=dcterms:title`

HTTP Accept Header: application/x-turtle (or URL Extension: .ttl)

Result:

```
<http://fpserver/fp/resources/workspaces/2/modules/1/elements/53>
   a workspace_2:Business_Need ;
   fps:genericLink <http://focalpointserver/fp/resources/workspaces/2/modules/1/elements/53?
format=genericrdf> ;
   fps:inModule <http://focalpointserver/fp/resources/workspaces/2/modules/1> ;
   dcterms:identifier "c414ea18-25ea-4bc1-8cc6-830a074a01d5" ;
   dcterms:title "Enable touch screen"^^rdf:XMLLiteral .
```

HTTP PUT

The HTTP PUT method is supported only for the Element resource. The Element RDF/XML document received from an HTTP GET call can be used to update attributes by using a PUT call.

URI: `http://`*`fpserver`*`/fp/resources/workspaces/2/modules/1/elements/53.rdf`

HTTP Accept Header: application/xml (or URl Extension: .xml)

Request body:

```
<rdf:RDF
   xmlns:dcterms="http://purl.org/dc/terms/"   xmlns:fps="http://jazz.net/ns/psm/focalpoint#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:fpdt="http://jazz.net/ns/psm/focalpoint/datatypes#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:module_2_1="http://focalpointserver/fp/resources/workspaces/2/modules/1/ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:oslc="http://open-services.net/ns/core#"
xmlns:workspace_2="http://focalpointserver/fp/resources/workspaces/2/ns#"
   xmlns:oslc_rm="http://open-services.net/ns/rm#" >
   <rdf:Description
rdf:about="http://focalpointserver/fp/resources/workspaces/2/modules/1/elements/53">
   <dcterms:title rdf:parseType="Literal">New Title value</dcterms:title>
   <fps:genericLink
rdf:resource="http://focalpointserver/fp/resources/workspaces/2/modules/1/elements/53?format=g
enericrdf"/>
   <fps:inModule rdf:resource="http://focalpointserver/fp/resources/workspaces/2/modules/1"/>
   <dcterms:identifier>c414ea18-25ea-4bc1-8cc6-830a074a01d5</dcterms:identifier>
   <rdf:type rdf:resource="http://focalpointserver/fp/resources/workspaces/2/ns#Business
Need"/>
   </rdf:Description>
</rdf:RDF>
```

Result:

After a successful PUT call, the title attribute of element 53 is changed to 'New Title value'. Though the example shows updating of only one attribute, it is possible to update more than one attribute in a single PUT call.

HTTP POST

The HTTP POST method is supported only for the Module resource for creating new Elements in the Module. The Element RDF/XML document received from an HTTP GET call can be used to create one or more elements resource using a single PUT call.

URI: `http://`*`fpserver`*`/fp/resources/workspaces/2/modules/1/elements.rdf`

HTTP Accept Header: .xml

Request body:

```
<rdf:RDF
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:fps="http://jazz.net/ns/psm/focalpoint#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:fpdt="http://jazz.net/ns/psm/focalpoint/datatypes#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:module_2_1="http://focalpointserver/fp/resources/workspaces/2/modules/1/ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:oslc="http://open-services.net/ns/core#"
xmlns:workspace_2="http://focalpointserver/fp/resources/workspaces/2/ns#"
    xmlns:oslc rm="http://open-services.net/ns/rm#" >
    <rdf:Description
rdf:about="http://focalpointserver/fp/resources/workspaces/2/modules/1/elements/53">
    <dcterms:title rdf:parseType="Literal">New Title value</dcterms:title>
    <fps:genericLink
rdf:resource="http://focalpointserver/fp/resources/workspaces/2/modules/1/elements/53?format=g
enericrdf"/>
<fps:inModule rdf:resource="http://focalpointserver/fp/resources/workspaces/2/modules/1"/>
    <dcterms:identifier>c414ea18-25ea-4bc1-8cc6-830a074a01d5</dcterms:identifier>
    <rdf:type rdf:resource="http://focalpointserver/fp/resources/workspaces/2/ns#Business
Need"/>
    </rdf:Description>
</rdf:RDF>
```

Result:

HTTP Response:

```
http://fpserver/fp/resources/workspaces/2/modules/1/elements/163
```

After a successful PUT call, a new element is created in module 1 and updated with the attributes values specified in the input RDF/XML document. The URI of the newly created element is returned as response.

More than one element can be created in a single POST call by including more than one element resource in the input RDF/XML to the HTTP request. Make sure that the URI of each Element resource has a different element ID.

HTTP DELETE

The HTTP DELETE method is supported only for the Element resource.

URI: `http://fpserver/fp/resources/workspaces/2/modules/1/elements/155.rdf`

HTTP Accept Header: application/xml (or URL Extensions: .xml)

Result:

After a successful DELETE call, the element 155 is deleted.

www.unicomsi.com

We welcome feedback on our documentation. Please email us at:
tech.authors@unicomsi.com

www.unicomglobal.com