

Web Services Provided by eagle-i Nodes

The eagle-i repository provides a comprehensive range of CRUD and query web services through its REST API, described in the [Repository Design Specification and API Manual](#). This API provides interaction at a relatively low level of abstraction, as data input and output are carried out in RDF with no ontology constraints. The primary clients of this API are the eagle-i SWEET, SWIFT/ETL and data management toolkits, and the eagle-i search application.

Other eagle-i software components provide more targeted public web services that do not require RDF knowledge. They are described below.

Resource Provider Web Services

These two web services are packaged with the SWEET application. They are centered on the concept of a *Resource Provider*, which is an organization that houses eagle-i resources - typically a laboratory or a core facility. The information returned by this web service is assembled by issuing targeted SPARQL queries to the eagle-i repository and collating information from the resource provider record and from its linked resources records, such that it be possible to find out at a glance what a resource provider offers. The information can be requested in XML format, so as to integrate it into other content pages. NB. we will soon provide a JSON alternative.

In order for these services to work properly, the following properties should be configured in the eagle-i-apps-credentials property file - see [Configuration Property Guide - pre 2.0MS3.x](#):

```
eaglei.catalyst.user  
eaglei.catalyst.password
```

The value of these properties should be a user created in the repository with no roles. By default, resource provider records will not contain contact information. This can be changed (though only for Core facilities), by adding the anonymousExtended user to the Contact Properties ACL via the repository's administrative console.

Resource Provider Lists - `/sweet/providers`, `/sweet/cores`, `/sweet/labs`

This call returns a list of resource providers, optionally restricted by type (e.g. only Core Laboratories), and includes minimal information about the provider (description, contact people). It is intended to be used as a "table of contents" from which individual resource providers can be selected, or as a list to loop through and visit all records. Note that at the moment this service does not use HTTP content negotiation for determining the returned format, but instead relies on the mime type being passed as an argument.

URL: `/sweet/providers` (GET and POST)

Shorthand URL: `/sweet/cores` (GET and POST)

Same as `/sweet/providers?type=http://vivoweb.org/ontology/core#CoreLaboratory&...`

Shorthand URL: `/sweet/labs` (GET and POST)

Same as `/sweet/providers?type=http://vivoweb.org/ontology/core#Laboratory&...`

Args:

`format--` text/html (default), application/xml

`type--` URI of an optional type restriction; If this argument is absent, all types are returned.

e.g. return only providers of type Core Laboratory: `type=http://vivoweb.org/ontology/core#CoreLaboratory`

Result:

A list of resource providers with their description and minimal information. If `format=application/xml`, the list is formatted according to the following XML Schema:

<https://open.med.harvard.edu/svn/eagle-i-dev/apps/trunk/datatools/catalyst/src/main/resources/resourceProviders.xsd>

If `format=text/html`, the XML is transformed into HTML using the following XSLT stylesheet:

<https://open.med.harvard.edu/svn/eagle-i-dev/apps/trunk/common/ui/static-resources/src/main/webapp/styles/resource-providers.xsl>

One Resource Provider - /sweet/provider

This call returns information about one resource provider, including short descriptions of all the resources that link to it (i.e are "located in", "provided by", etc.) organized by resource type.

URL: /sweet/provider (GET and POST)

Args:

`uri`-- the eagle-i ID (URI) of the resource provider, e.g. `uri=http://alaska.eagle-i.net/i/0000012a-25c2-586d-76e0-f22c80000000`
`format`-- text/html (default), application/xml

Result:

Information about one resource provider.

If `format=application/xml`, the list is formatted according to the following XML Schema:

<https://open.med.harvard.edu/svn/eagle-i-dev/apps/trunk/datatools/catalyst/src/main/resources/resourceProvider.xsd>

If `format=text/html`, the XML is transformed into HTML using the following XSLT stylesheet:

<https://open.med.harvard.edu/svn/eagle-i-dev/apps/trunk/common/ui/static-resources/src/main/webapp/styles/resource-providers.xsl>

Sitemap Generation Web Service

eagle-i resources are published as Linked Open Data on the web. If instructed to do so, search engine crawlers can index the HTML rendering of resources, which makes them available as results in public search engines. A `sitemap.xml` file that lists published resources and that is accessible at the root of your site communicates to search engines that resources are ready to be indexed. This web service provides a means of generating the contents of a sitemap, and is intended to be incorporated in a periodic job that updates the `sitemap.xml` file.

Sitemap Generation - /sweet/sitemap

URL: /sweet/sitemap (GET and POST)

Args:

none

Result:

A dynamically generated list of published resource providers and resources, formatted in XML according to the sitemap schema:<http://www.sitemaps.org/schemas/sitemap/sitemap.xsd>