Research Service Support For The CANARIE Registry And Monitoring System

Introduction

As part of a long-term strategy to facilitate research service software re-use, CANARIE requires that all services developed under the Research Software program support the CANARIE service registry and service monitor located at https://science.canarie.ca

The service registry allows potential users to discover and learn about your service. The goal is that this content will fully describe the details of the service, how to use it, and how to obtain assistance, in a format that provides a consistent user experience across all services.

The purpose of this document is to describe an interface that CANARIE requires developers to add to their services in order to support these capabilities. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, regardless of case or capitalization (see http://www.ietf.org/rfc/rfc2119.txt.pdf for details).

Types of Services

CANARIE supports three types of services in the registry. These types are:

- Managed Service – a single instance of the service is deployed on the service contributor's cloud or hardware resources. Users of this service access it over the network using a web services API.
- Self-Deployed Service – a single demonstration instance of this service is deployed on the service contributor's cloud or hardware resources. Potential users (ie. platform creators) can use this instance to evaluate the service. Once a user has decided to use the service, they download some sort of installation media (source code, binaries, VM images, …) from the contributor and deploy an instance on their own cloud or hardware resources. As is the case with a Managed Service, a Self-Deployed Service is accessed over the network using a web services API.
- Integrated Service - a catch-all for software services that are included directly into a research platform rather than being accessd over the network via an API. Examples of Integrated Services include Javascript fragments used for visualization and VM images.

The requirements for supporting the CANARIE registry and service monitor vary between these service types, so each type has its own section in this document.

Managed Services

Methodology

The goal of the API enhancements described in this section is to provide useful service information in a consistent way while minimizing the impact on existing services. To support these goals, simple HTML pages can be returned in many instances. In situations where a web service call is required, JSON will be employed.

To minimize the need for human maintenance of the registry, CANARIE would like all affected services to implement the URIs that are used by the registry and monitoring services in a standard subtree. Once your service's base URI, (referred to as <base> for the purposes of this document) is entered into the registry, all other items can be auto-populated.

Authentication

As there is currently no common standard for user authentication among research services, the API described in this document must not require user authentication.

Availability

CANARIE's monitoring service will poll all URIs listed below periodically. If any request fails or times out, the monitor will flag your service as unavailable.

Service Info

URI - <base>/service/info

Return basic identification and provenance information about the service. CANARIE's monitoring service will poll this URI periodically.

When an HTTP GET is performed on this URI, and the Accept header specifies JSON (ie. application/json), the service return the following:

{
"name" : "<name of service>",
"synopsis": "<one or two sentences describing what the service is for>",
"version" : "<service version identifier>",
"institution": "<the name of the institution that contributed this service>",
"releaseTime": "<time at which this version of the service was published>"
"researchSubject":"<the research area to which this service applies>"
"supportEmail": "<email address for support in case of service outage>"
"category": "<the type of service>"
"tags":[<terms describing this service – to be used in searches>]
}

where:

- The values of "name", "synopsis", "researchSubject" and "tags" may be entered either or both official languages.

- The value of "institution" will be taken as a string identifying the name of the organization that contributed the service.
- The value of "version" will be taken as a string and can follow any versioning scheme you deem appropriate.
- "releaseTime" must be formatted as YYYY-MM-DDThh:mm:ssZ (ISO 8601) and must be expressed in UTC rather than in local time.
- "researchSubject" is the area of research that your service applies to. You must use the research subject names published by NSERC. These can be found at http://www.nserc-crsng.gc.ca/help-aide/codes-listedecodes_eng.asp. Use the subject name rather than the subject number and feel free to use the top-level subject names if none of the detailed subject names match the purpose of your service. For services that are applicable to multiple areas of research, return "Multi-discipline".
- "supportEmail" is an email address that users of your service can use to communicate with you when the service is unavailable. In most cases, users will get support through your service support URI (see below) but in cases where the service support URI is not working, users must be able to contact your support staff via email at this address. CANARIE will publish this address on the Research Middleware portal.
- "category" must be one of:
    - Sensor Management/Data Acquisition
    - Data Storage and Retrieval
    - Data Manipulation
    - Data Visualization
    - Resource/Cloud Management
    - Service Registration/Discovery
    - Workflow/Service Scheduling
    - User Management/Authentication
    - Other
- "tags" shall contains a list of terms describing the service, to aid in searches. This structure shall be implemented as a JSON array with strings as elements - for example:

"tags":["astronomy", "FITS"]

The content type for this response should specify "application/json".

If an HTTP GET is performed and the Accept header does not indicate JSON, or there is no Accept header, services shall return an HTML page providing the information listed above in human-readable format. The content type for this response should specify "text/html".

The HTTP HEAD request must also be supported when the Accept header does not indicate JSON.

Service Statistics

URI - <base>/service/stats

Return information about the usage of your service. To aid in determining availability, your service must be designed such that requests to this URI will fail if other service functionality is not available. In situations where your service is not fully functional (if a database engine or other software service on which it relies is not available, for example), your service must return an HTTP error 503 in response to a GET on this URI.

Instead of checking all subsystems every time this request is made, you may choose to implement a separate, periodically scheduled "self test" within your service and base your response to a poll of the <base>/service/stats URI on the most recent results of this test.

The research software ecosystem supports a number of different Managed Services and there is not necessarily a single usage measurement that is applicable to all of them. Instead, CANARIE is asking the service creators to determine what service usage metric makes sense for their particular service. As such, the protocol described in this section allows the service creator to define their own <usage type> field for the status report. The guidelines are:

1. You must include one <usage type> field in the response packet defined in this section.
2. Field names must be meaningful and should convey units when they are not obvious. Consider names such as "CPU Hours" or "File Opens".
3. A previous version of this document specified a required field called "invocations" : "<number of times service has been used since last reset>". You may use this field definition for <usage type>, if appropriate.

When an HTTP GET is performed on this URI, and the Accept header specifies JSON (ie. application/json) services shall return the following:

{

"<usage type>" : "<service usage count since last reset>",
"lastReset": "<the time and date at which the <usage type> field was last reset to zero>",

}

where:

- <usage type> is a meaningful field name that indicates the type of usage, along with units where appropriate, being reported (see above).
- The value of "<usage type>" is a positive integer. This value shall start at zero when the service is first published and shall be incremented every time the service is used, according to whatever usage criteria the service creator has defined. Note that accesses to the API defined in this document should not be included in usage measurements.

- "lastReset" shall be formatted as YYYY-MM-DDThh:mm:ssZ (ISO 8601) and shall be expressed in UTC rather than local time. If you reset the <usage type> value to zero or it wraps around, this value must be updated accordingly.

The content type for this response should specify "application/json".
If an HTTP GET is performed and the Accept header does not indicate JSON, or there is no Accept header, services shall return an HTML page providing the information listed above in human-readable format. The content type for this response should specify "text/html".

The HTTP HEAD request must also be supported when the Accept header does not indicate JSON.

Service Documentation

URI - <base>/service/doc

Make the online documentation for your service available.

When an HTTP GET is performed, the user documentation for your service shall be returned in a human-readable format. This information must include:

- A detailed description of the service and what it does
- Training materials explaining how to use the service
- A complete description of the API
- Sample code illustrating API usage, as appropriate

If the documentation is hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Release Notes

URI - <base>/service/releasenotes

Return release notes describing the current version of your service.

When an HTTP GET is performed, the release notes for the current version of your service shall be returned in a human-readable format. As a minimum, these notes must identify:

- Changes made from previous versions
- Any known issues
- Work-arounds for known issues (where applicable).

If the release notes are hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify “text/html”.

Service Support

URI - <base>/service/support

Provide users with information on how to get support for your service.

When an HTTP GET is performed, instructions for users on how to get support for your service shall be returned in human-readable format. Include help desk contact info, a link to any bug tracking systems and/or forums, etc.

If the service support information is hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify “text/html”.

Service Source Code

URI - <base>/service/source

When an HTTP GET is performed, return link(s) to the source code for your service. This is only applicable if you are making your source code publicly available. If the source code is hosted elsewhere, please return and HTTP redirect in response to this request. If you are not providing access to your service’s source code, return status code 204 (No Content) must be returned.

If you are providing source code, also consider including design documentation. If the source code is hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify “text/html”.

Service Try Me

URI - <base>/service/tryme

Allow users to try out your service online.

This URI shall return a page that allows the user to try your service, possibly with fixed input. Consider displaying the text of the HTTP request and response for reference purposes. CANARIE realizes that it may be prohibitively complex to add this capability to some services. If that is the case, you may provide videos demonstrating the use of your service at this URI.

If the try me content is hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify “text/html”.

Service Licence

URI - <base>/service/licence

Allow users to view your service’s licencing/usage terms.

When an HTTP GET is performed, a page that indicates the licences and usage terms/restrictions associated with your service shall be returned in human-readable format. Licensing information for any third-party components your platform incorporates or makes use of must be included, as applicable.

If the licensing information is hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify “text/html”.

Service Provenance

URI - <base>/service/provenance

Allow users to view the software provenence process used when making a new release of your service publicly available.

When an HTTP GET is performed, a page that indicates what criteria has to be met before a version of your service is deployed shall be returned in human-readable format. This page must include, as a minimum:

- Information about who authorizes releases
- A description of the validation procedures that are completed prior to each release
- A list of documentation that is created as part of the release package
- A statement regarding your policy in dealing with upgrades/patches to any third-party software packages you may be using

If the provenance information is hosted elsewhere, an HTTP redirect may be returned in response to this request. This URI must also support HTTP HEAD requests. The content type for the response should specify “text/html”.

Self-Deployed Services

As there is no single instance of a Self-Deployed service that research platforms depend on for their proper operation, the CANARIE registry does not poll this type of service directly for availability and usage information. Instead, the service’s documentation links are verified periodically. If you are the creator of a Self-Deployed service and you would

like your demonstration instance monitored as a Managed Service, please contact us at support@science.canarie.ca.

For Self-Deployed Services, you will be asked to enter the following information into the registry at science.canarie.ca manually when your service is registered:

- Name – a short, descriptive name for your service
- Synopsis - a 2-3 sentence description of what your service does, to complement the Name
- Version – uniquely identifies the current version of your service. The value of "Version" will be taken as a string and can follow any versioning scheme you deem appropriate.
- Contributor - the name of the institution that contributed the service
- Research Subject - the research area to which this service applies. You must use the research subject names published by NSERC. These can be found at http://www.nserc-crsng.gc.ca/help-aide/codes-listedecodes_eng.asp. Use the subject name rather than the subject number and feel free to use the top-level subject names if none of the detailed subject names match the purpose of your service. For services that are applicable to multiple areas of research, return "Multi-discipline".
- Support Email - email address for users to obtain technical support for your service
- Category - the type of service. Category must be one of:
    - Sensor Management/Data Acquisition
    - Data Storage and Retrieval
    - Data Manipulation
    - Data Visualization
    - Resource/Cloud Management
    - Service Registration/Discovery
    - Workflow/Service Scheduling
    - User Management/Authentication
    - Other
- Tags – a list of terms describing this service to aid registry searching
- Administrators – the email address(es) of person(s) responsible for managing your service within the registry at science.canarie.ca. Specified addresses must be either from institutions that are part of the Canadian Access Federation (CAF) or must registered with the CAF Guest Identity Provider. Contact support@science.canarie.ca for assistance.
- Note: The values of "name", "synopsis", "researchSubject" and "tags" may be entered either or both official languages.

In addition to the static information listed above, you will be asked to provide the following documentation URIs. These URIs will be polled periodically by the registry to verify the availability of your service's documentation. Redirects are permitted. Unlike Managed Services, the documentation URIs for Self-Deployed Service do not need to have a common base.

Service Documentation

When an HTTP GET is performed, the user documentation for your service shall be returned in a human-readable format. This information must include:

- A detailed description of the service and what it does
- Training materials explaining how to use the service
- A complete description of the API, if any
- Sample code illustrating API usage, as appropriate

This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Release Notes

Return release notes describing the current version of your service.

When an HTTP GET is performed, the release notes for the current version of your service shall be returned in a human-readable format. As a minimum, these notes must identify:

- Changes made from previous versions
- Any known issues
- Work-arounds for known issues (where applicable).

This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Support

Provide users with information on how to get support for your service.

When an HTTP GET is performed, instructions for users on how to get support for your service shall be returned in human-readable format. Include help desk contact info, a link to any bug tracking systems and/or forums, etc. This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Source Code

When an HTTP GET is performed, return link(s) to the source code for your service. This is only applicable if you are making your source code publicly available. In this case, this URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

If you are not providing access to your service's source code it is not necessary to provide a URL and it may be left blank in the registry.

If you are providing source code, also consider including design documentation.

Service Try Me
Allow users to try out your service online.

When an HTTP GET is performed, this URI shall return a page that allows the user to try your service, possibly with fixed input. CANARIE realizes that it may be prohibitively complex to add this capability to some services. If that is the case, you may provide videos demonstrating the use of your service at this URI. This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Licence
Allow users to view your service's licencing/usage terms.

When an HTTP GET is performed, a page that indicates the licences and usage terms/restrictions associated with your service shall be returned in human-readable format. Licensing information for any third-party components your service incorporates or makes use of must be included, as applicable. This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Provenance
Allow users to view the software provenence process used when making a new release of your service publicly available.

When an HTTP GET is performed, a page that indicates what criteria has to be met before a version of your service is deployed shall be returned in human-readable format. This page must include, as a minimum:

- Information about who authorizes releases
- A description of the validation procedures that are completed prior to each release
- A list of documentation that is created as part of the release package
- A statement regarding your policy in dealing with upgrades/patches to any third-party software packages you may be using

This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Service Download
When an HTTP GET is performed, a page that allow users to download the distribution package for your service shall be returned in human-readable format. Distribution packages include tarballs, VM images, compiled binaries, etc. If source code is your distribution medium, this link should be the same as the "Service Source Code" link described above. This URI must also support HTTP HEAD requests. The content type for the response should specify "text/html".

Integrated Services

The requirements for Integrated Services are the same as those for Self-Deployed Services at this time.