Extending the GOCDB's role in the Information Service Space

George Ryall (George.Ryall@stfc.ac.uk)
David Meredith (David.meredith@stfc.ac.uk)

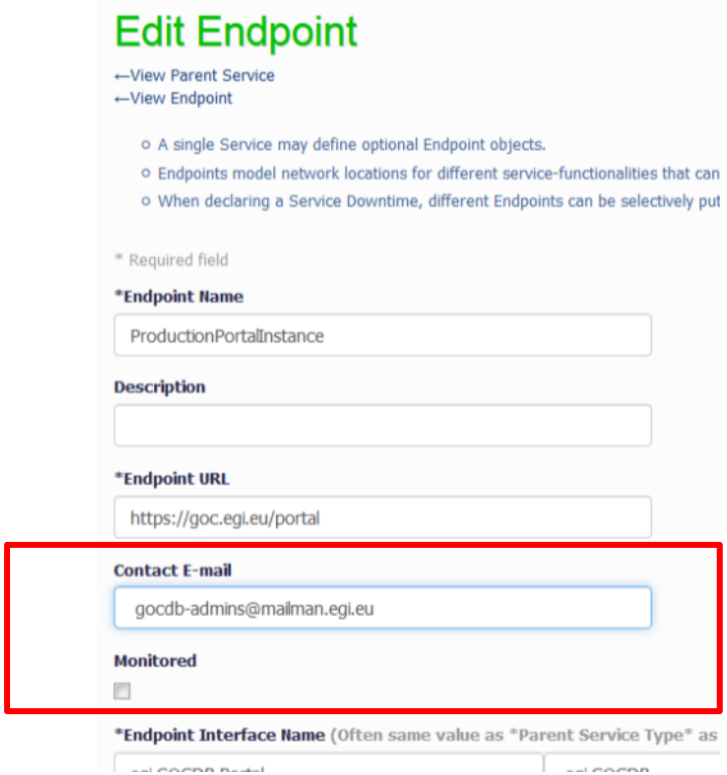
10/05/17 – EGI Conference 2017

Overview

- New service endpoint attributes
  - What?
  - Why?
- Write API
  - Why?
  - Authentication
  - Currently supported methods
  - Worked example
- Future Development

New Service Endpoint Attributes What?

- New boolean ‘monitored’ flag, exposed through read API
- New Email field for adding contact details for the SE



Edit Endpoint

←View Parent Service
←View Endpoint

- A single Service may define optional Endpoint objects.
- Endpoints model network locations for different service-functionalities that can
- When declaring a Service Downtime, different Endpoints can be selectively put

* Required field

***Endpoint Name**

ProductionPortalInstance

Description

***Endpoint URL**

https://goc.egi.eu/portal

Contact E-mail

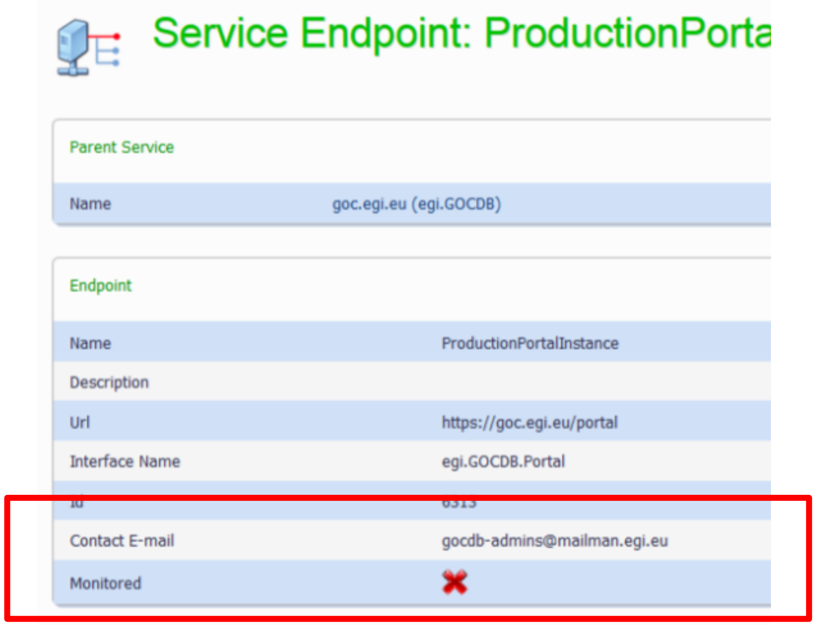
gocdb-admins@mailman.egi.eu

Monitored

□

***Endpoint Interface Name** (Often same value as *Parent Service Type* as

egi.GOCDB.Portal egi.GOCDB



Service Endpoint: ProductionPorta

Parent Service	
Name	goc.egi.eu (egi.GOCDB)

Endpoint	
Name	ProductionPortalInstance
Description	
Url	https://goc.egi.eu/portal
Interface Name	egi.GOCDB.Portal
Id	6313
Contact E-mail	gocdb-admins@mailman.egi.eu
Monitored	✗

New Service Endpoint Attributes Why?

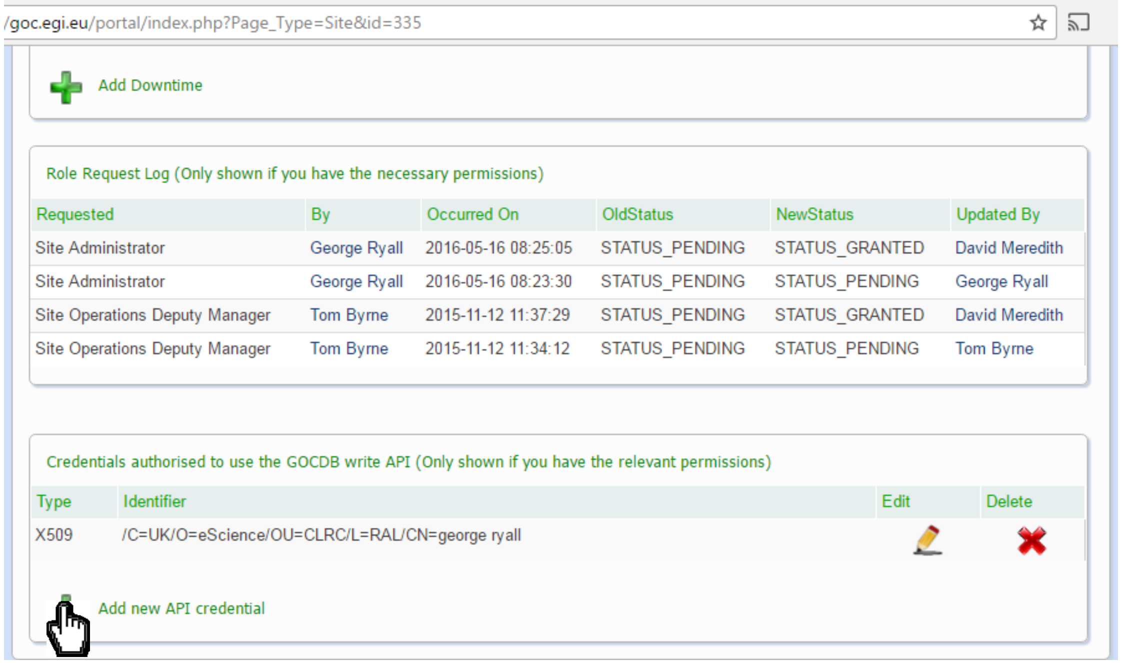
- Enabled ARGO to remove dependency on BDII
- The SE email is NOT currently rendered in the output of the 'get_service' method, as this method has a Level 1 protection level which is considered a public method containing no critical information and no personal email/details (https://wiki.egi.eu/wiki/GOCDB/PI/Technical_Documentation#Protection_levels) If emails are needed, we can either:
  - Elevate the protection level of the get_service method to Level 2 (which means a certificate would be needed to query the method)
  - Relax EGI/GOCDB protection level rules so that Level 1 can include site/service emails but no personal emails, note however there would still be a risk that personal emails would still be added.

Write API
Why?

- Assists WLCG to reduce dependence on BDII by
- Enabling the ability to add/update/delete custom properties defined on Sites/Services/Endpoints using a REST API.
- These custom properties are key-value pairs, allowing additional information to be added to entities without changes to the GOCDB schema.
- Documentation: https://wiki.egi.eu/wiki/GOCDB/Write_API/Technical_Documentation

Write API Authorisation

- Authentication is by X509 certificate, a DN of a host allowed to make changes through the API can be associated with each site.
- This is then allowed to update custom properties for that site, its services, and service endpoints.
- This association is set through the web portal on the site details page.



/goc.egi.eu/portal/index.php?Page_Type=Site&id=335

Role Request Log (Only shown if you have the necessary permissions)

Requested	By	Occurred On	OldStatus	NewStatus	Updated By
Site Administrator	George Ryall	2016-05-16 08:25:05	STATUS_PENDING	STATUS_GRANTED	David Meredith
Site Administrator	George Ryall	2016-05-16 08:23:30	STATUS_PENDING	STATUS_PENDING	George Ryall
Site Operations Deputy Manager	Tom Byrne	2015-11-12 11:37:29	STATUS_PENDING	STATUS_GRANTED	David Meredith
Site Operations Deputy Manager	Tom Byrne	2015-11-12 11:34:12	STATUS_PENDING	STATUS_PENDING	Tom Byrne

Credentials authorised to use the GOCDB write API (Only shown if you have the relevant permissions)

Type	Identifier	Edit	Delete
X509	/C=UK/O=eScience/OU=CLRC/L=RAL/CN=george ryall	✎	✖

Write API -Supported Methods

Method	url	function
POST	<apiurl>/v5/ServiceEndPoint/<End point ID>/ExtensionProperties	Adds the extension properties defined in the request to the SE with the given ID (fails if there are already extension properties)
PUT	<apiurl>/v5/ServiceEndPoint/<End point ID>/ExtensionProperties	replace the extension properties for the SE with the given ID (fails if there are already extension properties) with those in the request (?if non defined?)
DELETE	<apiurl>/v5/ServiceEndPoint/<End point ID>/ExtensionProperties	Removes all the extension properties for the SE with the given ID (fails if there are already extension properties) (?if non defined?)
POST	<apiurl>/v5/ServiceEndPoint/<End point ID>/ExtensionProperties/<name>	Adds an extension property of the named SE with the named name and value in the request (fails if property with that name is already defined)
PUT	<apiurl>/v5/ServiceEndPoint/<End point ID>/ExtensionProperties/<name>	Updates the extension property of the named SE with the named name and value in the request (?If property is not already defined)
DELETE	<apiurl>/v5/ServiceEndPoint/<End point ID>/ExtensionProperties/<name>	Removes the named extension property from the named SE
POST	<apiurl>/v5/Service/<Service ID>/ExtensionProperties	Adds the extension properties defined in the request to the service with the given ID (fails if there are already extension properties)
PUT	<apiurl>/v5/Service/<Service ID>/ExtensionProperties	Replace the extension properties for the service with the given ID (fails if there are already extension properties) with those in the request (?if non defined?)
DELETE	<apiurl>/v5/Service/<Service ID>/ExtensionProperties	Removes all the extension properties for the service with the given ID (fails if there are already extension properties) (?if non defined?)
POST	<apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name>	Adds an extension property of the named service with the named name and value in the request (fails if property with that name is already defined)
PUT	<apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name>	Updates the extension property of the named service with the named name and value in the request (?If property is not already defined)
DELETE	<apiurl>/v5/Service/<Service ID>/ExtensionProperties/<name>	Removes the named extension property from the named service
POST	<apiurl>/v5/Site/<Site ID>/ExtensionProperties	Adds the extension properties defined in the request to the Site with the given ID (fails if there are already extension properties)
PUT	<apiurl>/v5/Site/<Site ID>/ExtensionProperties	Replace the extension properties for the site with the given ID (fails if there are already extension properties) with those in the request (?if non defined?)
DELETE	<apiurl>/v5/Site/<Site ID>/ExtensionProperties	Removes all the extension properties for the site with the given ID (fails if there are already extension properties) (?if non defined?)
POST	<apiurl>/v5/Site/<Site ID>/ExtensionProperties/<name>	Adds an extension property of the named site with the named name and value in the request (fails if property with that name is already defined)
PUT	<apiurl>/v5/Site/<Site ID>/ExtensionProperties/<name>	Updates the extension property of the named site with the named name and value in the request (?If property is not already defined)
DELETE	<apiurl>/v5/Site/<Site ID>/ExtensionProperties/<name>	Removes the named extension property from the named site

Science & Technology
Facilities Council
10 Years of Impact and Inspiration

Write API
Worked Example 1

```
$ curl -i -k --cacert /cygdrive/c/George_local/GOCDB/certs/escience-root.pem --cert
/cygdrive/c/George_local/personalCerts/2016.crt.pem --key /cygdrive/c/George_local/personalCerts/2016.key.pem -X POST
https://goc.egi.eu/gocdbpi/V5/service/782/ExtensionProperties -d '{"TestProp":"someKey"}'
HTTP/1.1 403 Forbidden
Date: Thu, 08 Dec 2016 12:00:44 GMT
Server: Apache/2.2.15 (Red Hat)
X-Powered-By: PHP/5.3.3
Content-Length: 248
Connection: close
Content-Type: application/json

{"Error":{"Code":403,"Message":"The X509 identifier \"\/C=UK\/O=eScience\/OU=CLRC\/L=RAL\/CN=george ryall\" is not authorised
to alter the RAL-LCG2 site","API-Documentation":"https:\/\/wiki.egi.eu\/wiki\/GOCDB\/Write_API\/Technical_Documentation"}}
```

- I try and edit a service which I am not authorised to, so I get a 403 error code and a JSON formatted error message explaining the error

Write API
Worked Example 2

```
$ curl -i -k --cacert /cygdrive/c/George_local/GOCDB/certs/escience-root.pem --cert
/cygdrive/c/George_local/personalCerts/2016.crt.pem --key /cygdrive/c/George_local/personalCerts/2016.key.pem -X POST
https://goc.egi.eu/gocdbpi/V5/service/4180/ExtensionProperties -d '{"TestProp":"someValue"}'
HTTP/1.1 204 No Content
Date: Thu, 08 Dec 2016 12:02:28 GMT
Server: Apache/2.2.15 (Red Hat)
X-Powered-By: PHP/5.3.3
Content-Length: 0
Connection: close
Content-Type:
```

- This time I am authorised to change this service.
- I add an extension property with the key ‘TestProp” and value “someValue” to the service with ID 4180.
- Currently the API returns no content, but it provides a 204 http response code on success.

Write API
Worked Example 3

```
$ curl -i -k --cacert /cygdrive/c/George_local/GOCDB/certs/escience-root.pem --cert
/cygdrive/c/George_local/personalCerts/2016.crt.pem --key /cygdrive/c/George_local/personalCerts/2016.key.pem -X POST
https://goc.egi.eu/gocdbpi/V5/service/4180/ExtensionProperties -d
'{"TestProp":"someDifferentValue","secondTestProp":"someOtherValue"}'
HTTP/1.1 409 Conflict
[…]
Content-Type: application/json
{"Error":{"Code":409,"Message":"A property with name \"TestProp\" already exists for this object, no properties were added.","API-
Documentation":"https:\/\/wiki.egi.eu\/wiki\/GOCDB\/Write_API\/Technical_Documentation"}}
```

```
$ curl -i -k --cacert /cygdrive/c/George_local/GOCDB/certs/escience-root.pem --cert
/cygdrive/c/George_local/personalCerts/2016.crt.pem --key /cygdrive/c/George_local/personalCerts/2016.key.pem -X
PUT https://goc.egi.eu/gocdbpi/V5/service/4180/ExtensionProperties -d '{"TestProp":" someDifferentValue
","secondTestProp":"someOtherValue"}'
HTTP/1.1 204 No Content
[…]
```

- This is a demonstration that the behaviour of ‘PUT’ and ‘POST’ requests is different.
- POST will not override existing properties, PUT will

Write API
Worked Example 4

```
$ curl -i -k --cacert /cygdrive/c/George_local/GOCDB/certs/escience-root.pem --cert
/cygdrive/c/George_local/personalCerts/2016.crt.pem --key /cygdrive/c/George_local/personalCerts/2016.key.pem -X PUT
https://goc.egi.eu/gocdbpi/V5/endpoint/6313/ExtensionProperties/TestProp -d '{"value":"SomeValue"}'
HTTP/1.1 204 No Content
[…]

$ curl -i -k --cacert /cygdrive/c/George_local/GOCDB/certs/escience-root.pem --cert
/cygdrive/c/George_local/personalCerts/2016.crt.pem --key /cygdrive/c/George_local/personalCerts/2016.key.pem -X DELETE
https://goc.egi.eu/gocdbpi/V5/endpoint/6313/ExtensionProperties/TestProp
HTTP/1.1 204 No Content
[…]
```

- Finally I wanted to show that properties can also be added and deleted one at a time

Future Developments

- The development roadmap for GOCDB, up to the end of EGI Engage, can be found here.
- This contains several tasks, but the main one relating to the information space is extending the write API to enable:
  - Creation, update, and deletion of service endpoints
  - Update of details of services

Future Developments

- Additionally over coming months we will be working on:
  - Data freshness verification
  - Completing integration with EGI checkIn service
  - Changes to scheduled downtime rules (subject to continuing discussion)
- Beyond EGI Engage we will be replacing the current UI with a modern web framework
- GOCDB is developed in response to community need. If there are things you think could be improved or new features that you believe would improve it, please contact: gocdb-admins@mailman.egi.eu.

Questions?

All of our source code is on GitHub: https://github.com/gocdb/GOCDB/