REST API Documentations

This page discuss how to generate the REST API documentations. We are considering to adapt the swagger (https://github.com/wordnik/swagger-core) spec and swagger-UI (https://github.com/wordnik/swagger-ui) to generate REST API documentation automatically.

About Swagger

Swagger Core

"The goal of Swagger is to define a standard, language-agnostic interface to REST APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined via Swagger, a consumer can understand and interact with the remote service with a minimal amount of implementation logic. Similar to what interfaces have done for lower-level programming, Swager removes the guesswork in calling the service.

Swagger-core is the Java/Scala implementation of Swagger. It supports *JAX-RS*, *plain Servlets*, and *Play Framework*.

Check out Swagger-Spec for additional information about the Swagger project, including additional libraries with support for SpringMVC, other languages and more." (excerpted from http://github.com/wordnik/swagger-core)

Swagger-UI

Swagger-UI is the collection of HTML, javascript and CSS to read json data formatted with Swagger-Spec.

You can test this at http://swagger.wordnik.com/

For BrownDog

In order for us to use Swagger-UI for our REST API documentation front end, our codes (DTS, DAP) needs to generate the json data in Swagger-Spec. There are two ways (https://github.com/wordnik/swagger-core/wiki/Adding-Swagger-to-your-API)

1. Server integration
   a. generating the swagger json on-the-fly. our REST api includes ".../api" endpoint to generate the json.
   b. It means that you need to add api documentation at your code by using annotation
2. No server integration
   a. generating the swagger json file manually or
   b. generating the json automatically in compilation (build) by using 3rd party tools
      i. it can use same annotations mentioned #1 above, javadocs or JAX-RS annotation
3. Brown Dog API - utilizing Swagger: https://bd-api-dev.ncsa.illinois.edu/docs/

Testing

Currently, BrownDog uses mostly Java and Scala to create REST API. We will test these languages and preferred-library first. The test results and notes will be posted here.

Java+RestEasy

Tested with Cyberintegrator-serivce (https://opensource.ncsa.illinois.edu/stash/projects/CBI/repos/cyberintegrator-3/browse/cyberintegrator-service)

Method 1 (server integration)

- Used the sample provided by swagger-core project (https://github.com/wordnik/swagger-core/tree/master/samples/java-resteasy)
- Problem:
  - Current swagger doesn't support this "auto-scan" capability. Currently Cyberintegrator uses "auto-scan" capability of RestEasy (it scans the resources in the source code and adds to the Application automatically).
  - Tested http://igaffa.blogspot.com/2013/07/swagger-with-resteasy-in-jboss.html but some parts of the codes are not clear to implement

Method 2 (generating json)

swagger-maven-plugin (https://github.com/kongchen/swagger-maven-plugin)

This is the maven plugin that generate the json files and html file by using swagger java annotations in the code

- It generates the json files such as service.json, workflows.json, datasets.json by using swagger java annotations in the code. Also, it generates single api html file.
- Loaded the each json files into swagger-UI and worked.
- Problem:
    - Swagger-UI with "service.json" supposes to connect to other jsons like workflows.json and datasets.json. But it didn't work. It may need to be tested with the pom.xml configuration for the maven plugin

enunciate (http://enunciate.codehaus.org/)

This is a tool that generate swagger-UI, htmls, etc by using JAX-RS java annotations. It can be ran as a command-line tool or maven-plugin