Software Quality Assurance (SQA) Report

22-26 Aug 2016

fgAPIServer (FutureGateway API Server)

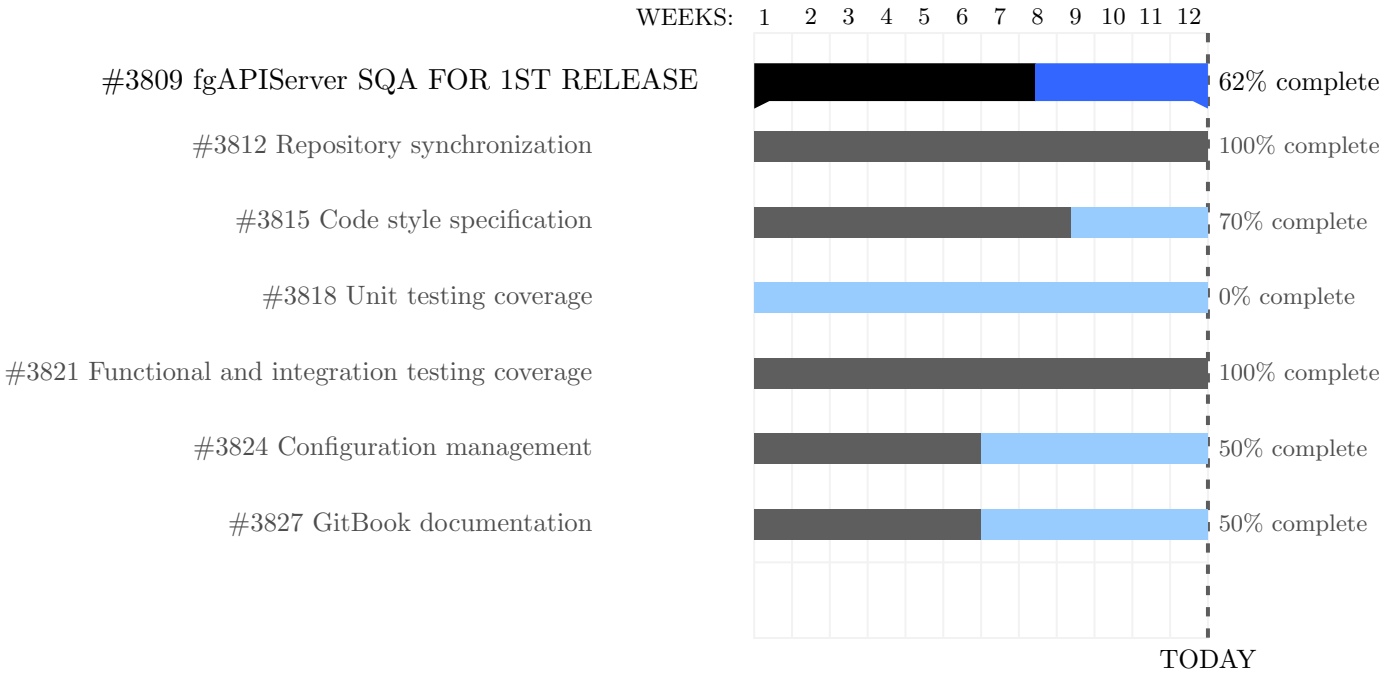
SQA Progress Status **NOT COMPLETE**

62% done

GitHub repository	COMPLETE
Code style adherence	NOT COMPLETE
Code coverage	0%
Functional/integration testing	COMPLETE
GitBook documentation	NOT COMPLETE
Automated deployment	NOT COMPLETE

Part I

Task Progress for the 1st Release



1 Repository synchronization

Products contributing to INDIGO-DataCloud project must have their code avaiable under GitHub's `indigo-dc` organization.

Repository exists under `indigo-dc` GitHub organization:

- https://github.com/indigo-dc/fgAPIServer
- https://github.com/indigo-dc/APIServerDaemon

2 Code Style

Products contributing to INDIGO-DataCloud project are expected to be adhered to a community or de-facto standard code style definition. Exceptions can be made to the selected standard. Custom style guides are accepted but nonetheless not recommended.

Code style definition	PEP 8 – Style Guide for Python Code
Community/de-facto standard	Yes
Exceptions	0
Richness	73 Errors 63 Warnings 10 link

Code style definition	Code Conventions for the JavaTM Programming Language
Community/de-facto standard	Yes
Exceptions	0
Richness	63 Errors 63 Warnings 0 link

2.1 Build status

Last build status on Jenkins CI fgapiserverdaemon-codestyle. fgapiserver-codestyle.

2.2 Observations

- Product `APIServerDaemon` *is not* compliant with Java Sun Style guide.

3 Unit Testing

Code coverage will be tracked for the INDIGO-DataCloud related products and must not decrease during the project's duration. Recommended threshold is 70%.

3.1 Observations

- Cannot define a Jenkins job to check the only test written so far `./tests/test_mklogtoken.py` since this test is malformed, meaning that it tries to import modules as it were a Python package but it is not.

4 Functional/Integration testing

Functional testing must cover at least the basic functionalities that the product was requested to fulfill within the INDIGO-DataCloud project scope. Integration testing must cover the interactions with other components. Both types of testing will be automated whenever feasible by integrating them in the project's continuous integration service.

No automated execution, reports are being provided.

4.1 Test coverage

1. API functionalities being tested urlhttps://github.com/indigo-dc/FutureGateway-APIs/blob/master/apiary.apib (fgAPIS-erver).

2. Obtaining/using a token (fgAPIServer).
3. Submit a task (APIServerDaemon).

4.2 Reports

Functional report/s available:

- owncloud report

4.3 Observations

- Team has provided reports of manual functional tests being performed. Automated functionality testing has not been provided.

5 GitBook documentation

Product-related documentation must be uploaded to GitBook's `indigo-dc` central repository. Types of documentation includes a) Developer b) Deployment and Administration c) Command-line Interface (CLI) and Application Program Interface (API) d) User Documentation. All these types may not be applied for every product. Those products that offer functionalities out of the scope of INDIGO-DataCloud project needs may not provide all the spectrum, but links to the official documentation.

Documentation available under `indigo-dc` GitBook organization:

https://www.gitbook.com/book/ricsxn/futuregateway/details

5.1 Types of documentation currently provided

Installation Usage

5.2 Observations

- Points to improve in the documentation:
  - Documentation is still not available at indigo-dc
  - Align documentation sections to the ones defined in [1], in case you need to add different sections, do it at the end.
  - README (or somewhere else)
    - How does this interface with the INDIGO PaaS ?
    - Provide example of authentication with the INDIGO IAM
  - Deployment guide
    - Only covers Ubuntu 14.04, Centos7 must be documented as well

    - Installation from source is the only form of deployment documented. This must be completed to cover installation using INDIGO repository.
    - There is a typo in the URL pointed at wget https://github.com/FutureGateway/PortalSetup/r
    - what is the usefulness of the fgTools to the user communities and/or site admins ? how and when to use them ?
    - TCP ports that need to be exposed and to which services
  - User guide
    - How to register and manage users ?
    - How to define the universe of resources available to execute the tasks and their properties
    - What does it mean GridEngine? is the SGE batch system or is an engine to submit to grid sites ?
    - Where is the documentation about the adaptors? add pointers to the docs.
  - Regarding PortalSetup
    - Not clear if it works with CentOS7 (*EL6/7 Tested with CentOS6*) and directory CentOS7 is not there (*Please notice that OS specific intallation scripts are collected inside dedicated directories.*)
    - Rephrase *Debian Tested with Ubuntu* statement.

6 Configuration Management

Those products released by INDIGO-DataCloud project that need to be deployed by the end user must rely on a maintained open-source configuration management tool to provide an automated means to install and configure the product. The recommended tool is `Ansible`.

Tool	bash script
Deployment coverage	installation/configuration
Manifest link	None

Product does not currently have an automated deployment at INDIGO-DataCloud's Jenkins CI.

6.1 Observations

- Product does not provide an automated deployment solution based on Ansible or Puppet tools. For the time being, manual deployment is available following the documentation at GitBook.
- Deployment guidelines only cover Ubuntu14.04, not for CentOS7 distributions, which is also required by the project.

Part II

How to read this document

1 Summary (front) page

Both the overall product's SQA adherence and per-task status codes are explained below:

COMPLETE	Task has been successfully completed and fulfills the project's SQA requirements, listed in Deliverable D3.1 and Extensions to Software Quality Assurance documents.
NOT COMPLETE	Task has not been completed, yet some missing required bits have not been provided.
IN PROGRESS	Task has not been completed, but can proceed as it is.
WP3 PENDING	Task has some pending work from WP3 side, meaning that the product team already submitted the required data but it has not been yet consumed by WP3.

2 Task Progress

2.1 Code style

Code style definition	Name and link of the standard to which the product is adhered.
Community/de-facto standard	Whether the adopted standard is community-wide accepted.
Exceptions	Number of exceptions from the standard definition.
Richness	Number of rules defined in the adopted standard. Additionally (whenever available) the number of errors, number of warnings documented in the standard will be displayed as well as the link to the latest definition.

2.2 Unit testing

This section will display the a) trend graph with the evolution of the code coverage over time and b) the Cobertura report, with the coverage results of different methods. Both are taken from the project's Jenkins continuous integration service.
Note: resultant coverage value is the lowest of the ones for the different methods: packages, files, classes, lines, conditionals.

2.3 Functional/Integration testing

2.4 GitBook documentation

Whenever the documentation of the product is available at the project's GitBook repository, both the a) **link** to the documentation index and b) **type of documentation** provided will be displayed in the report.

2.5 Configuration Management

Whenever the product has an recipe to be deployed automatically the following information will be available:

Tool	Configuration management tool used.
Manifest link	URL pointing to the manifest/s.
Deployment level	Whether `installation`, `configuration` or both.
Build status	Current build status for the project's supported OS distributions.