

edgex-global-pipelines Unit Tests

- Leverage **Jenkins-Spock** - library to help with testing Jenkins pipeline scripts and functions written in Groovy.
 - Based off of Spock Unit Testing Framework
 - Leverages Maven build automation tool
 - Requires JDK (1.8+) and Maven (3.3+)
- Mocking Jenkins dependencies
 - Add dependencies to Project Object Model (pom.xml)
 - Only pipeline steps whose JARs are dependencies will end up being mocked
 - Jenkins components (core and plugins) are distributed as Maven artifacts

edgex-global-pipelines Unit Tests

- Pull Request

- <https://github.com/edgexfoundry/edgex-global-pipelines/pull/69>

- Unit Test Summary

- ~89% Coverage for Groovy code

- Two types of Unit Tests

- Data-Driven Tests

- Exercise the same test code multiple times, with varying inputs and expected results

- Behavior Driven Tests

- Test behavior instead of implementation

```
[WARNING] Tests run: 87, Failures: 0, Errors: 0, Skipped: 2
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:37 min
```

Gaps

- **Jenkins-Spock** currently lacks test coverage reporting
- Does not support Declarative Pipeline
 - Will require a leveraging a different capability to test this
 - JenkinsPipelineUnit testing framework
- No code linting capability included
- Library includes basic documentation
 - Initial implementation relied on trial/error

Next Steps

- Create CI build image to facilitate testing
 - Initial compilation can take several minutes to complete
 - Pull down all required dependencies (pom.xml)
- Add build automation integration
 - edgex-global-pipelines git commit status checks
- Refactor source code to make testing easier
 - Implement consistent environment variable access across all libraries
 - Consolidate and reuse shared methods (DRY)
 - Implement smaller functions where possible (keep the code simple)
 - Return and error conditions should be checked early (avoid deep nesting)