Agave Postman Integration Test Suite

This repository contains the Postman test collections used by the Agave Platform as part of the project's continuous integration pipeline. There are two primary branches: master contains tests that are ready to be run across all production tenants, while dev contains tests still under development.

Environments and Data Variables

An execution of a postman collection is a single execution of all tests in the collection. A test run in general consists of multiple executions of the collection. Postman makes use of environments for storing variables to parameterize tests with values that will be consistent across executions. Environments for running the dev and master collections are are stored in files ending in .postman_environment within the environments directory. Related, postman makes use of data variables for iteration-specific test data. Data in a file ending in .json is used to parameterize tests with values changing with each execution. Data variables should be specified as a JSON list; each object in the list will be used for a single execution. For examples of environments and data variables, respectively, see environments/public.postman_environment and data/newman_data.json in this repository.

> *Note that the environment file can be imported, edited, and exported using the Postman interface.*

Conventions

- Use URLs with the following format: {{BASEURL}}/{{<service>_SERVICE}}/{{<service>_VERSION}}<endpoint_path>
  - example: {{BASEURL}}/{{FILES_SERVICE}}/{{FILES_VERSION}}media/{{USERNAME}}
- Within a collection, tests are run in the order listed. Ensure the order of execution of subcollections (folders within the main collection directory) by prefixing the names of the subcollection directories with numbers (e.g., 01_my_first_subcollection, 02_my_second_subcollection, etc.)
- Generate a token in the first test to be run using credentials stored in the environment. Pass the token to subsequent tests by storing it in the environment (see the Generate an Access Token test in the 00_setup subcollection in the prod collection).

Prerequisites

In order to make full use of this repository, the following technologies must be present on your system:

- Postman
- Newman Collection Runner
- Docker
- jq
- Git
- A running Agave sandbox with all four servers <auth server>, <dbs server>, <core apis server> and <core workers server>.
- At least one host to act as our execution/storage system with ssh and sftp enabled (we'll call this host <Test External Host>)
- A user defined for <Test External Host> able to login with ssh keys.
- Our "testuser" defined in our APIM server and the tenant designation

Project Structure

Clone the Git repository

The git repo for postman/newman tests resides on our gitlab located here https://YOUR_TACC_USERID@gitlab.tacc.utexas.edu/cic/postman.git, after replacing YOUR_TACC_USERID with your id.

> check our ability to limit or allow access to this repo.

Configuration files for test fixtures

The project directory structure is as follows. Explore the **tenants** directory where configuration for newman tests takes place.

Files	description
contributors.txt	contributors list
execute-archive-testrun.sh	script for executing a newman test run and archiving the inputs, outputs and test results
README.md	initial readme for project usage

Directories	description
admin	admin services tests
ansible	ansible playbooks
data-management	special data management tests
jira-issue-regressions	JIRA issues directory
tenants	tenant based test fixture configuration
testrun	archive directory for test run information

Under the tenants directory

The configuration files in this directory hold the properties needed to template the test fixtures before running tests.

config files	description
collections/agave-core-services.postman_collection.json	holds the template for postman tests run by newman
config/newman_data.json	list of properties used to parameterize the environment and test data sent to the above postman tests
data	directory holding files used in the actual postman tests run against the services under test
environments	directory holding postman environment files that hold values for the postman tests. specific to the tenant and user for an environment

Step by Step guide

Step 1 Clone the repository

Clone the git repository for the Agave postman/newman tests. The git repo for postman/newman tests resides on our gitlab located here https://YOUR_TACC_USERID@gitlab.tacc.utexas.edu/cic/postman.git, after replacing YOUR_TACC_USERID with your id.

Step 2 Configuration changes

We will make changes to our configuration files that will define our environment, <Test External Host> and template properties. Review the values for your particular environment and make any necessary changes. The changes listed below are a starting point, others may need modification as well.

- We will need to replace the IP address or host name of the <Test External Host> in our configuration files.
    - testsystem in config/newman_data.json

    - login.host and storage.host in data/systems/compute.json
    - storage.host in data/systems/storage.json
- We will also need to replace the user information for logging into the above systems.
    - systemusername, systemhomedir in config/newman_data.json
    - login.auth.username , storage.homeDir and storage.auth.username in data/systems/compute.json
    - storage.homeDir, storage.auth.username in data/systems/storage.json
- The file for settings changes is environments/dev.sometenant.postman_environment
    - Change the name of the file to the tenant designation for your environment, for example in our environment to test Sandbox we use dev.sandbox.postman_environment.

Step 3 Running

We are now ready to run the newman test suite.

All of the collections in this repository should be run from the command line with the newman runner. To make this as easy as possible, several script have been developed to select a tenant environment, stage data in place, and select individual tests to run on the fly.

The `**agave-core-services**` collection has several requests which require file uploads. Due to an outstanding issue in Postman, the Runner application included in the Postman Application cannot properly run the collection. Whenever running collections, use Newman.

The `**tenants/newman-v2.sh**` script is the primary interface for running the `agave-core-services` Postman Collection. From the `**tenants**` directory, the directory command usage is as follows:

```
$ ./newman-v2.sh --help
Usage: newman-v2.sh [<options>] TENANT_ID
Runs Agave Platform integration tests against the Science APIs and
publishes a
report summary to Slack. Options exist for running directly against the
backend,
running tests against a single service, and suppressing Slack updates.

Options:

-s, --service The name of an Agave Science API service to run the tests on.
If specified, only the tests against this service will be run.
--skip-frontend If present the tests will be run against the backend
services,
bypassing the tenant APIM, with a self-generated JWT.
--notify-slack If present, the results summary will be written to slack.
--add-host Any additional hosts to add into the docker containers.
--dry-run Run the filters and set the environment, but skip actual test run
-h, --help Show this help information.
-v, --verbose Increase script verbosity.
-d, --debug Enable debug logging on the HTTP requests.
```

Examples of running the scripts with different configurations are given below. All commands are run from the `tenants` directory.

```
# Run against the sandbox tenant with verbose output turned on
./newman-v2.sh -v dev.sandbox

# Run against the develop tenant, with dry-run doing template substitution
but does not run tests.
./newman-v2.sh --dry-run dev.develop

# Run against the sandbox tenant with verbose output turned on and only
running job tests
./newman-v2.sh -v -s jobs dev.sandbox
```

Examples of running the above tests then archiving the results. Run from the root of the project.

```
# Run against the sandbox tenant with verbose output turned on and only
running jobs tests
./execute-archive-testrun.sh
dev.tenants.sandbox.agaveapi.co:149.165.157.105  dev.sandbox "-s jobs"
```