Tech Assessment - 1 - Introduction

Monday, August 18, 2025

4:21 PM

* **About the Tech Assignment, I completed tasks:** 
  + **Generated an API app server to simulate API functionality for this assignment and deployed locally on Ubuntu Linux for testing.**
    - As 2nd email clarified with Anirban, using LLMs help to create that app.
    - It support API Endpoints: /upload and /validate, and related response & validation functions.
    - Providing an extra endpoint /query API, so I can use SQL query to check the DB which populated data from uploaded CSV for testing purpose.
    - Details more about the App server and setup see related doc section of this assignment, and the app server attached READM.MD file
  + **Built the Python test framework (task #1)**
  + **Developed unit tests (#2, with mocks) and integration test (part of #3)**
  + **For the AWS integration (part of #3), provide a separate strategy and pseudo-code doc for outlining test approach.**
  + **Documented (#4) covering the automation testing approach, assumptions, test results and instructions.**

* **Locations:** 
  + The QA Python tests and related data files I created for unit test and integration test uploaded to the GitHub, see: [My\_Proj3/brainbox\_qa at main · ghu21st/My\_Proj3](https://github.com/ghu21st/My_Proj3/tree/main/brainbox_qa), or see attached zipped package: 'brainbox\_qa.zip'
    - The test result and Junit format report included (test\_report.xml)
    - The test files can be found under /tests folder, including the api main function file: main\_brainbox\_api.py.
  + The App server can be also found on GitHub, at: [My\_Proj3/brainbox\_local\_api at main · ghu21st/My\_Proj3](https://github.com/ghu21st/My_Proj3/tree/main/brainbox_local_api); or see attached zipped package 'brainbox\_local\_api.zip'
  + The QA Python tests and the App Server are separately setup/installed, QA test suite setup & tested on Windows, and the App server deployed and run on a Ubuntu Linux server.

* **Introduction for my Python Test Suite:**
  + **Features of the Test Suite for BrainBox API app server testing:** 
    - **Total test cases: Unit test (23) + Integration test (13) = 36**
    - **Based on Pytest framework**
    - **Integration Test using Fixtures**
      * Set scope to ensure isolation for each function/test case
      * Using multiple fixtures for File handling and API request handling
      * Tear up/tear down for API requests
    - **Unit Test using Mock** 
      * No dependencies with other resources
      * Mock REST API methods (POST)
      * CSV created on-fly by tempFile module based on case needs
    - **Other features:**
      * Retries failed test enabled
      * Added log info for test case run

* **Unit Tests (test\_unit\_\*.py)** 
  + **All Unit functionality tests are based on my own main API call function file - "main\_brainbox\_api.py" that created for AI generated API server for this assignment.** 
    - Positive
    - Negative
  + Mocking for external dependencies

* **Integration Tests (test\_int\_\*.py)**
  + **Real API functionality Test Calls - RESTful APIs, Populated uploaded CSV into SQLite DB tables, local deployed to simulated AWS env**
    - Positive Tests, including API call tests for: Validate CSV -> Upload CSV -> Query DB
    - Negative Tests, including: CSV missing column, wrong data type, empty, file non-exist, etc.

* **Special /query API and tests**
  + it is not part of this tech assignment requirements, I created just for integration test purpose, so I can use SQL query to check the DB which populated data from uploaded CSV. Therefore, the /query API test included in the integration test subset, but is not included in the unit test subset.

**---------------**

* **Python Test Suite Execution:**

. Copy / download the QA test set and files

. Pip install required packages/modules

.. Minimal package: pytest, requests; more refer to qa\_test\_requirements.txt as reference.

. Run the whole test set - Unit test + Integration test (38 cases) and output as junit format xml report command:

C:\My\_Proj3\brainbox\_qa> pytest --junitxml=./test\_report.xml