What concerns would you have from a testing perspective?

- *Parameter Combination:* Testing the API with all the possible request parameter combinations gets complex as the number of parameters increasing. For example, below API need to be tested with the username, repository name values as both valid; both invalid; username valid, repository invalid and vice versa.
  "repository_url": "https://api.github.com/repos/{owner}/{repo}

  Hence, a proper test design is needed to reduce the complexity (data driven test).

- *Parameter Selection:* Testing each API parameter with valid/invalid data inputs. This requires good understanding of each parameter and its usage to determine the valid resources. In below API, the valid values for query parameter "state" should be known ahead of preparing the tests.
  https://api.github.com/repos/vmg/redcarpet/issues?state=closed"

  Hence, a proper documentation on the usage should be provided by the service team.

- *API Call Sequence:* As the input values for one API may dependent on the output of the other API call. A good understanding on the API dependencies is required to thoroughly test the API functionality. For example, get the list of repositories for a particular user using first API and then get a particular repository details using second API.

  "user_repositories_url": "https://api.github.com/users/{user}/repos
  "repository_url": "https://api.github.com/repos/{owner}/{repo}

- Request/Response change: API tests need to be revisited in the case the request/response body schema changes for any business reason.

How would you go about tackling the QA for this work?

Test Plan:

- Go through the service documentation
- Prepare test plan with the below details
  - List of APIs under test (Endpoints, Path, query parameter keys and values etc)
  - Input data conditions
  - Expected Response details
  - Expected Response time
- Gather security information (Basic auth details)
- Identify the tools to be used

Test Design:

- Create unit tests to validate the API resources

- Create Integration tests to validate API call dependencies

Test Execution:

- Execute testcases using Junit assertions and postman

What sort of tests would be worth describing or worth automating?

Tests to validate the below scenarios

1. API response code
2. API response content
3. API endpoint
4. Response time assertions
5. API sequence
6. Invalid Input conditions
7. Proper message is displayed in case of errors
8. Authorization (Example: Basic Auth)

What tools would you use?

- Junit framework
- Postman
- Jersey 1.19 Client API