Table of Contents

[Task 1 – API: 2](#_Toc145008604)

[Task 2 – Web: 5](#_Toc145008605)

# Task 1 – API:

**Tools used:** Postman

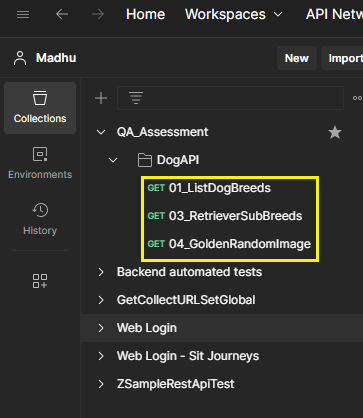
**Justification:** To test the given api I’ve used Postman which is an open source tool.

**Executable File path in gitHub:**

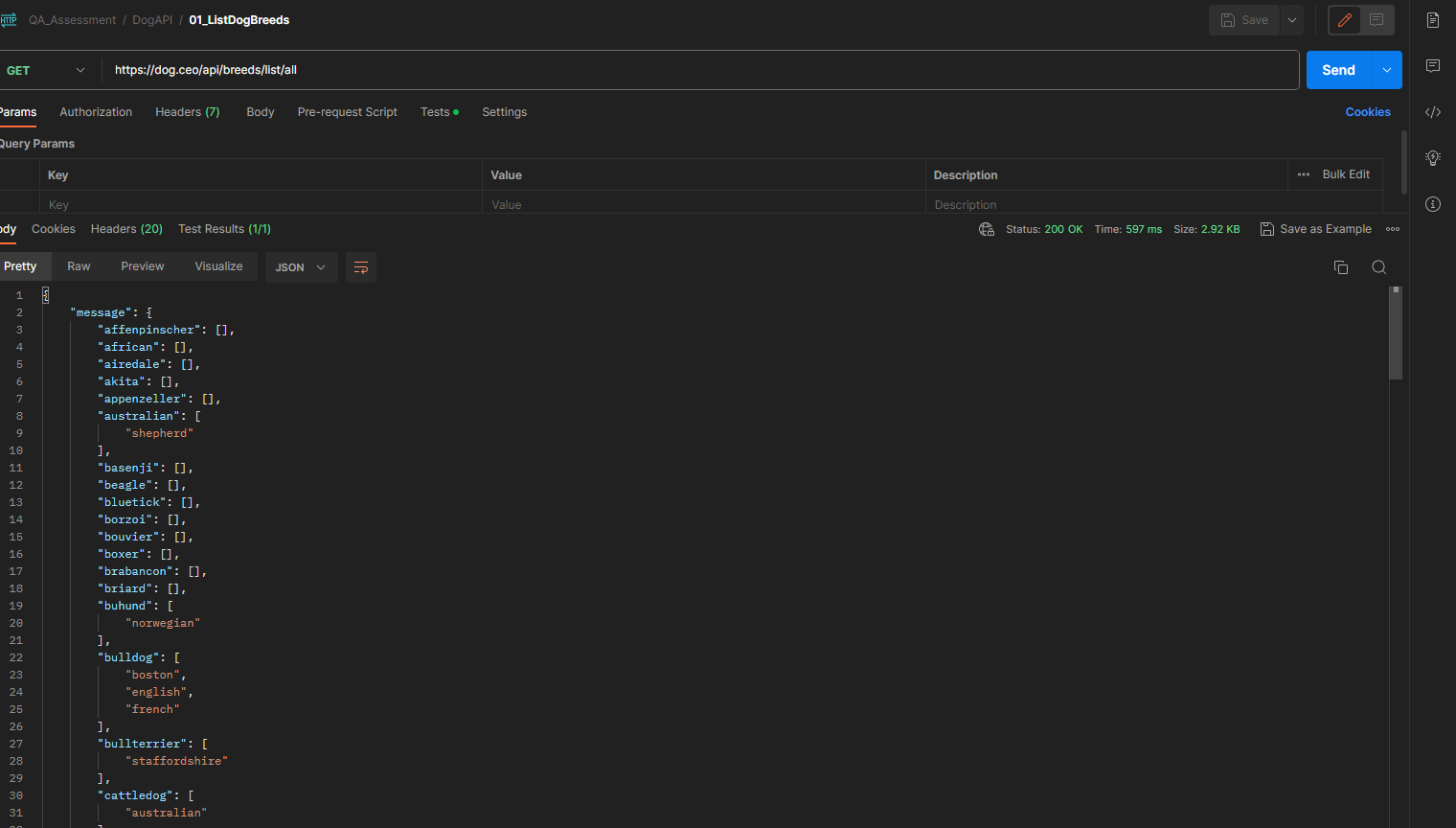
MG\_OMS\_Assessment/QA\_Assessment\_Task1/ QA\_Assessment.postman\_collection.json

**Details:**

The json collection contains all api requests as per test scenarios.



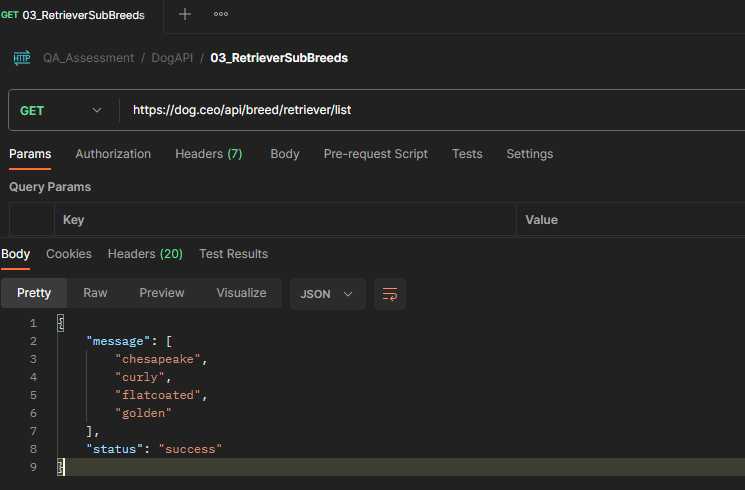
* 01\_ListDogBreeds – GET request, to retrieve all the breeds in json response.



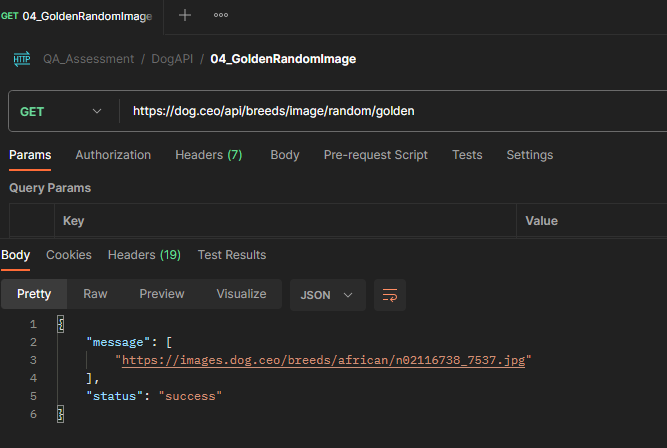
* 01\_ListDogBreeds> Tests:

|  |  |
| --- | --- |
| **Positive Scenario** | **Negative Scenario** |
| **Request and Test Result** | |
|  |  |
| **Respective response** |  |
|  |  |

* 03\_RetriverSubBreeds – GET request, retrieves all sub-breeds



* 04\_GoldenRandomImage – GET request to produce a random image / link for the sub-breed “golden”.



# Task 2 – Web:

**Framework used:** Cucumber, Testng, Selenium, Junit, Extent Reports

**Justification:** To test the given test scenarios I’ve used above mentioned frameworks:

Cucumber and Testng – To achieve Hybrid approach in developing test cases

Selenium – To access and validate the Web page and its elements

Extent Reports – To capture the test results, and test evidences

**Runner File path in gitHub:**

MG\_OMS\_Assessment/QA\_Assessment\_Task2/QA\_Assessment/src/test/java/Runners/Run.java

**Details:**

* Feature file name: WebUserTables.feature
  + Contains the steps to run the automation to test the given scenarios.
* Step Definition file name: StepDefs.java
  + Contains the step definitions for the step mentioned in feature file
* Runner file name: Run.java
  + Specifies the tags, and plugins in test case execution.
* extent.properties and spark-config.xml –
  + Contains customized settings for storing reports, test evidence and results.