**API Test Strategy**

The structure of Postman test collection will follow the structure of the API endpoints and at the lower level of each main category it will be divided into Valid and Invalid:

**Test Collection**

* **Auth**
  + **Login**
    - **Login DE**
      * **Valid**
      * **Invalid**
    - **Login UK**
      * **Valid**
      * **Invalid**
* **Healthcheck**
  + **Valid**
  + **Invalid**
* **Booking**
  + **bookings**
    - **Valid**
    - **Invalid**
  + **bookings/Id**
    - **Valid**
    - **Invalid**
  + **bookings/id/weather**
    - **Valid**
    - **Invalid**

The goal is to create several requests that include the maximum combination as possible of conditions. The division between valid and invalid requests takes into consideration the expected status code of the response, where valid would be successful responses (200, 201,…) and invalid would be error responses (400, 401,…).

This division makes it possible to create shared tests at the parent folder level (for instance we can create a test to validate that the response status code is 200 at **Auth\Login\Login DE\Valid** that would be applied to all the requests contained inside that folder). Furthermore, generic tests, such as the validation of the header “correlation-id” on the response, could actually be done at **Test Collection,** so that they would be applied to all the test requests.

The response validation can be done through JSON scheme validation, based on the response models available on Swagger. This should include the validation of the presence of the expected properties in each response, but also from the type of value of each property (string, boolean, …) or even of the specific values for properties that only allow a restricted list of values.

**The Postman Test Collection is available at COCUS\_Assignment\_QA\_Automation\Postman\17.TestAPI\Test\_Collection.json**