Testing APIs with Postman

1. Have you used Postman before? \*

Yes, I did use.

2. Have you used Swagger before? \*

No, I did not use.

3. Get endpoint parameters for Login DE \*

Using the provided swagger, please tell us all the parameters for Login DE endpoint, including if they are required, type of

parameter e.g., header and allowed value(s)



4. Create necessary Environments \*

Please create and upload environments based on the following: Kurt Fritze is a DE customer and logs in to their app using their

username (test) and password (test), their booking reference is ADE9452. Sam Smith is a UK customer using an android device

and their booking reference is AR58930. Once complete, upload the environments. Please name the environments with the

following format: Your\_Name \_Market (e. John\_Doe\_UK).

Files submitted: UK

Graphical user interface, text, application, email

Description automatically generated

DE

Graphical user interface, text, application, email

Description automatically generated

5. Create a Collection \*

Using the provided swagger (attached in the email), create a single collection with all available endpoints. Once complete and

you can call each endpoint, upload the collection. If you have made changes to environments, or created globals, please

upload them here as well. (Hint: Login endpoints return "login-token" header on success request). Please name your

collections with Your\_Name.

Files submitted:

Graphical user interface, text, application, email

Description automatically generated

6. Did you use the import functionality to create the collection? \*

Yes, I did for creating the requests

Steps:

1. On the “swagger” editor, generate and copy the “Curl”

2. On the “Postman”, select the collection

3. Click on the “import” button

4. The “Import” modal window is displayed, select the “raw-text” option

5. Past the “curl” in the editor, and click the “Continue” button

6. On the “Import Elements”, click “Import”.

Result: After saving, Postman generates a new “Request” inside the selected collection

7. Reading response results in postman \*

Please provide screenshots of status, time taken, size, headers, and body of a SINGLE RESPONSE in Postman

Files submitted:

RESPONSE: status, time taken, size, and body

Graphical user interface, text, application, email

Description automatically generated

RESPONSE: headers

Graphical user interface, text, application, email

Description automatically generated

8. Analyze and improve testing requirements for the following scenario \*

A new API has been created to standardize Login endpoint for all markets. The request is now POST only with b parameters:

bookingRef, surname, username, and password. A new header parameter "appName" has also been added to ensure correct

processing of the data. Tester tries to login with each market and closes the testing tic What other tests could the tester do to

ensure that API is working correctly?

Test scenarios:

1. Validate the login with different devices (variables: ios and android) - different markets (Ps: To validate this scenario need a clarification to confirm the expected result).
2. Validate that the request payload with invalid **data type** for bookingRef, surname, username, and password.
3. Validate that the request payload with **invalid values** for bookingRef, surname, username, and password
4. Validate that appName header is present– ( Ps: appName is required and is a string?)
5. Validate that the new API is not duplicating the resources in the database ( (Ps: Test Condition based in the issue that the DE market is duplicating resources on the “Available bookings” https request
6. Validate the Json schema
7. Validate auth sending user credentials in the URL
8. Validate that the new API is not log in with the old login market requirements:

* UK market: Login UK market sends requests only with bookingRef and surname.
* DE market - UK market: Login UK market and send a request only with a username and password

Business Question:

1. Will the JSON response format for the common requests be changed to display the same structure and values type?

Example:

1. The healthcheck response payload contains "apiVersion" only for DE Market.
2. On the response payload for the “Weather” HTTPS method, the "speedUnit" parameter will be "MPH" or “KPH(PS: currently each environment is using a unit convention)

9. Test the APIs \*

Test all the APIs using Exploratory Testing / Session Based test techniques. This is one of the more important questions in the

Test so spend plenty of time on it! Document all your findings including your test ideas, unresolved questions,

positive/negative results, headers, payloads, values etc. Upload all your documentation here. Your ability to organize the

findings in a clear and structured way will also be judged.

Files submitted:

Please: Refer to XTSessionReport (1).html doc in the “COCUS Challenge – Postman” folder

Ps:

* I did just some validations as examples, in the real world I would add more bugs and notes for questions.

e.g the duplication of resources on the database, or others' misspelling errors as the “Celsius” word.

Intermediate

Postman

10. Validate headers for all requests (20%) \*

All requests in this API have to return Content-Type (application/json) and Connection (keep-alive) headers. If you think some

other header validations need to be done for some endpoints, please add it. Once complete, upload collection.

Files submitted: Test on the Collection Level

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

11. Do you want to try the advanced section? \*

No Skip to question 14

Ps: I got some denied permission running “npm install -g newman”, I saw some resolutions on the stack flow, but the Mac is not mine so I decided to skip to questions.

But probably executing the steps below I would answer the skipped questions

1 - I installed node js

2 - Installed newman:

* npm install -g newman
* npm install -g newman-report-html

3 – Export the postman collection –

4 - Run the command from the folder the collection is present

* newman run <collection.json> -r html
* A new folder would be generated with the run status report

14. Please write your feedback here:

* In real world, I would create more test validations.
* Please, refer to the json collection file in the folder for more details
* It’s the first time I am using SQS AWS, so I don’t know if I did what you want.
* It's a cool challenge, it was a opportunity to learn, thanks.