

Name – Kunal Manchanda

## Task 1: Automate API Requests and Verify Responses

**Objective:** Automate requests to the Airalo Partner API and validate the responses.

**Choice of Framework/Language:** Cypress/JavaScript

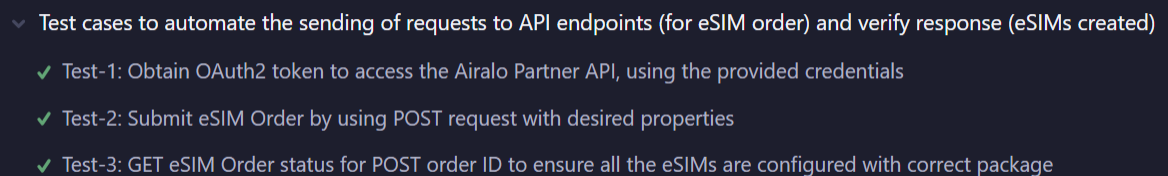
**Reason:** Cypress is an excellent choice for automating API testing in most scenarios due to its ease of use, real-time debugging, and integration capabilities.

### Automation Approach:

- 1) Define variables and constants
- 2) Automate POST API for obtaining OAuth2 tokens to access the Airalo Partner API, using the provided credentials and save the token to be used in subsequent AIRALO APIs
- 3) Automate POST API to order 6 "merhaba-7days-1gb" eSIMs. Save the order ID to verify configured eSIMs through the GET API.
- 4) Automate GET API to list eSIMs related to your specific order Obtained through step-3. Ensure the list contains 6 eSIMs, and that all of them have the "merhaba-7days-1gb" package slug.

### Artefacts:

1. Test case script -attached (**TEST\_FILE.spec.js**)
2. Screenshots
  - a. Successful execution of all test cases.

- 
- A screenshot of a Cypress test runner interface showing a list of test cases. The first item is a collapsed folder icon with the text "Test cases to automate the sending of requests to API endpoints (for eSIM order) and verify response (eSIMs created)". Below it are three expanded test cases, each with a green checkmark icon: "Test-1: Obtain OAuth2 token to access the Airalo Partner API, using the provided credentials", "Test-2: Submit eSIM Order by using POST request with desired properties", and "Test-3: GET eSIM Order status for POST order ID to ensure all the eSIMs are configured with correct package".
- ✓ Test cases to automate the sending of requests to API endpoints (for eSIM order) and verify response (eSIMs created)
    - ✓ Test-1: Obtain OAuth2 token to access the Airalo Partner API, using the provided credentials
    - ✓ Test-2: Submit eSIM Order by using POST request with desired properties
    - ✓ Test-3: GET eSIM Order status for POST order ID to ensure all the eSIMs are configured with correct package

- b. Test-1: Obtain OAuth2 token to access the Airalo Partner API, using the provided credential
      - ✓ Verify token generation using provided client id and secret.
      - ✓ Verify 200 success response

- ✓ Verify property access\_token is available in the response
- ✓ Save the token for subsequent API endpoint use

```

✓ Test-1: Obtain OAuth2 token to access the Airalo Partner API, using the provided credentials
▼ TEST BODY
1  request  ○ POST 200 https://sandbox-partners-api.airalo.com/v2/token
2  - assert  expected 200 to equal 200
3  - assert  expected { Object (token_type, expires_in, ...) } to have property access_token
4  log  Successfully obtained OAuth2 token to access the Airalo Partner API
5  wrap  eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiI5NTgiLCJqdGkiOiI5YWZmYzUyZmQwNzAyYjYyNmUzZjIzZjc5MDlMDg3ZTM0MzU0ZGYwMDg5MzNiZTQ0Njk3MjAyZTVlMjE2ZjBjNDRjY2Y3MjZkNDZkZGZjMSIsImhhdCI6MTc0OTE4MTE3My4zNTc5OTEsIm5iZiI6MTc0OTE4MTE3My4zNTc5OTMsImV4cCI6MTc0OTI2NzU3My4zNTUyMzUzInN1YiI6IiIsInNjb3BlcyI6W119.ysAS6j5NQWnGEm-ZRnuB5ZFVBIX1ebDHuAK0moUzFAzcfkKbZ-UBzWwS1bzggRuUJpNI8UGTpRIwIAghpPcf5li3f9B4IQ2g0j2FZqEgHVWpWwNzYMPkjFRfjLQ67cYxqFHFex7NXYLInUR4uqvK2ZVy9JsgyWAmULQZNJw3LxVLWxTn3t2YUKXjDH0w3HK0DyCw0sEurQmC9fT9xY4GGKAgYk2mY14z1nRFgYRY0kHL9F9QxVZC1tKxD2S48uNrNgvDgrFLdP9e00sAHhHvVw3uyMTXje7sfrLajN2zw1ML5TqatBe4LLaWgMeGHYTDxGtRzqvMMDs_yyfTwfVEyoAxpM2q2T-7mpYWL-JLJHXbcNqZs3NmtXYaWNca6ie0_hK1abs4sUDfLJ0A7A9AAEDhPo2XBQnsVLnk6b7WXBORDf6C-CczmIhj0ETpQVGamDT8uBsmzaxiv9cQ295MqT2MoR6p_QMtaI9fk7qP-XzuWuNnss6kE9KMVDz0Al2yA0TtdP2Ya2UwXxkLYCsFDeM04aFQDM2CHoa3BNaH5sC1sJLMApQ30-9bnmd49PpyTQb07dlWyuzVE3tBiRtZ4JADyeKS7Vr-Ljn3hSYuG6q4LhvXgxVXSgjhTbLAlActUH4-Teg_kTVMT9Lxw-

```

c. Test-2: Submit eSIM Order by using POST request

- ✓ Automate POST API with variable headers and body parameters.
- ✓ POST API order for 6 "merhaba-7days-1gb" eSIMs by using the token extracted in previous test case.
- ✓ Verify 200 OK response
- ✓ Verify response body for the correct information and eSIM properties
- ✓ Save the order ID

```

✓ Test-2: Submit eSIM Order by using POST request with desired properties
▼ TEST BODY
1  request  ○ POST 200 https://sandbox-partners-api.airalo.com/v2/orders
2  - assert  expected 200 to equal 200
3  - assert  expected { Object (id, code, ...) } to have property id
4  - assert  expected { Object (id, code, ...) } to have property package_id
5  - assert  expected { Object (id, code, ...) } to have property sims
6  - assert  expected { Object (id, code, ...) } to have property quantity
7  log  POST CALL of ordering 6 SIMS successful- Recieved 200 OK
8  log  Saved Order ID: 182727

```

d. Test-3: GET eSIM Order status for your order ID to ensure all the eSIM are configured with correct package

- ✓ Automate GET API with variable headers
- ✓ Extract ESIM for order details saved from previous POST API ( in this case Order ID: 182727).
- ✓ Verify 200 OK response
- ✓ Verify ensure all the eSIM are configured with correct parameters e.g. package\_id – merhaba-7days-1gb
- ✓ Verify there are 6 eSIMs under ORDER ID:182727 (6 Matching Objects).

✓ Test-3: GET eSIM Order status for POST order ID to ensure all the eSIMs are configured with correct package

✓ TEST BODY

```
1 request GET 200 https://sandbox-partners-api.airalo.com/v2/sims?include=order
2 -assert expected 200 to equal 200
3 -assert expected { Object (data, links, ...) } to have property data
4 -assert expected [ Array(25) ] to be an array
5 -assert expected 6 to equal 6
6 -assert expected { Object (id, created_at, ...) } to have property simable
7 -assert expected { Object (id, code, ...) } to have property package_id
8 -assert expected merhaba-7days-1gb to equal merhaba-7days-1gb
9 -assert expected { Object (id, created_at, ...) } to have property simable
10 -assert expected { Object (id, code, ...) } to have property package_id
11 -assert expected merhaba-7days-1gb to equal merhaba-7days-1gb
12 -assert expected { Object (id, created_at, ...) } to have property simable
13 -assert expected { Object (id, code, ...) } to have property package_id
14 -assert expected merhaba-7days-1gb to equal merhaba-7days-1gb
15 -assert expected { Object (id, created_at, ...) } to have property simable
16 -assert expected { Object (id, code, ...) } to have property package_id
17 -assert expected merhaba-7days-1gb to equal merhaba-7days-1gb
18 -assert expected { Object (id, created_at, ...) } to have property simable
19 -assert expected { Object (id, code, ...) } to have property package_id
20 -assert expected merhaba-7days-1gb to equal merhaba-7days-1gb
21 -assert expected { Object (id, created_at, ...) } to have property simable
22 -assert expected { Object (id, code, ...) } to have property package_id
23 -assert expected merhaba-7days-1gb to equal merhaba-7days-1gb
24 log GET CALL of ordering 6 SIMS successful- Recieved 200 OK
25 log Count of Total eSIMS simable.id = 182727: 6
26 log Matching Objects Count:, 6
```

**Important Considerations:**

- In real scenario, Ids and secrets should be stored in a Vault. As this is a test sandbox environment, I am calling these parameters directly in the test file.
- We can add more log statements to print full API response or a particular API response object for debugging purpose for error scenarios.
- We can add negative scenarios for API Testing by playing around the configs and passing wrong tokens, header or body parameters.
- In an actual test project, Constants and Variables should be stored in an environment variable or test data file which can be called during the test execution.