DSL Tariffs

Scenario 1: Verify the DSL calculator

1: open www.verivox.de

2: Navigate to the DSL calculator page using Menu navigation DSL → Dsl-vergleich.

3: Enter 030 for my area code # Ihre Vorwahl = your area code AND select the 100 Mbit/s option as bandwidth.

4: Ensure at least 1 Verivox recommended tariffs are loaded and at least 5 tariffs in Ermittelte

for given option is loaded.

5: Ensure that all the tariffs loaded have bandwidth greater or equal than 100.

Scenario 2: Load multiple tariff result pages

1: open www.verivox.de

2: Navigate to the DSL calculator page using Menu navigation DSL → Dsl-vergleich.

3: Enter 030 for my area code # Ihre Vorwahl = your area code AND select the 100 Mbit/s option as bandwidth.

4: Note the number of available tariffs listed in the Ermittelte Tarife section.

5: Ensure at least 1 Verivox recommended tariffs are loaded and at least 5 tariffs in Ermittelte

for given option is loaded.

6: Ensure that all the tariffs loaded have bandwidth greater or equal than 100.

7: Note and store the number of tariffs displayed on a button **"# weitere Tarife laden"**. Click on button **20 weitere Tarife laden** and ensure that only next 20 games have been loaded.

Note: For the last load , ensure that **"# weitere Tarife laden"** button displays number of remaining tariff games.

8: Repeat step 6 until all tariffs have been loaded.

9: Once all tariffs have been displayed ensure number of tariffs displayed is equal to the number of tariffs recorded in step 4.

Address checks API

Scenario 3: Find the cities for a given postcode

1: Request a cities for postcode 10409 and 77716.

2: Ensure response for each request has 200 status code and contains no error.

3: Ensure response array for each request contains all values as string.

4: Ensure for post code 10409 response contain "Berlin" in "Cities "array. And for post code 77716 response contain in "Fischerbach","Haslach","Hofstetten" “Cities “array.

Note: Test implementation for this step stores postcode and their respective city names in key value pair dictionary. Hence in only one test step using two parameters as post codes I am checking the responses for respective requests.

Scenario 4: Find the streets for a given postcode

1: Request Cities from post codes 10409 and 77716.

2: For each of the received city request streets from that respective city.

3: Ensure response has 200 status code, have “Streets” key and contains no error.

3: Ensure all streets from “Streets” array contains street value as string and they are expected streets for the current city in loop (Current City is each city name received in step2).

4: Ensure that streets with Germanic letter retains umlaut characters.

Note: Test implementation for this case receives cities from postcodes 10409 and 77716 and then for each of those cities streets are verified for conditions illustrated in above steps.