**Tools used: Python (PyCharm IDE), Selenium package.**

**Sources:** [**Selenium with Python**](https://selenium-python.readthedocs.io/) [**Developers - IPinfo.io**](https://ipinfo.io/developers)

**Question#1 QA Planning Exercise**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Description | String 1 | String 2 | String 3 | Expected Output |
| Basic functionality of method | “dkmjeircuj874357mfdujm3eu934jjjj2398” | “jjjj” | “1984” | dkmjeircuj874357mfdujm3eu93419842398 |
| Behaviour of multiple instances | “dkmjjjjjcuj874357mfdujm3eu934jjjj2398”  “jjjjjjjjjjjjxjjjjjj” | “jjjj” | “1984” | dkm1984jcuj874357mfdujm3eu93419842398  198419841984x1984jj |
| Partial instance | ‘jjjeircuj874357mfdujm3eu934jjxj239j” | “jjjj” | ‘1984” | jjjeircuj874357mfdujm3eu934jjxj239j |
| Upper vs Lower (if .upper method not used) | “dkmjjjjcuj874357mfdujm3eu934JJJJ2398” | “jjjj” | “1984” | dkm1984cuj874357mfdujm3eu934JJJJ2398 |
| Un-matching String length replacement | “asd123qwe” | “qwe” | “19844545” | asd12319844545 |
| Symbols in string | “asd-12-3qwe”  “-as/d123qwe”  “-asd123qwe”  “asd-12-3qwe”  “1asd-12-3qwe” | “asd”  “qwe”  “asd”  “asd”  “asd” | “1984”  “1984@”  “1984”  “-=1984”  “ “ | asd-12-3qwe  -as/d1231984@  -1984123qwe  -=1984-12-3qwe  1 -12-3qwe |
| Empty string | “asd-12-3qwe” | “asd” | “” | -12-3qwe |
|  |  |  |  |  |

**Question#2 Web Automation Exercise**

**Scenario#1**: Validate main page title matches to ”וואלה!שופס - אתר הקניות הגדול בישראל”

**Scenario#2**: Check search functionality on the main page - locate the search box, insert search value and click on the submit button. Expected result - results page loaded and presented results per search value.

**Scenario#3:** Check cart functionality - locate cart button, navigate to result page and validate result. Expected result: Cart page loads and present data.

S**cenario#4:** Check favourites functionality - locate favourites button, navigate to results page, locate sign-in with Google account and follow to Google registration page with result validation. Expected result: Google page API works properly and accessible, pages loading and representing data.

**Question#3 API Automation Exercise**

General API test cases: Create - post; Read - get; Update - put/patch; Delete - delete.

For this exercise: Tested the Read - get functionality.

**Scenario#1(A)**: request and get status code from <https://ipinfo.io/161.185.160.93/geo>. Expected result: 200.

**Scenario#2(A)**: Test json data response.

Expected result:

{

"ip": "161.185.160.93",

"city": "New York City",

"region": "New York",

"country": "US",

"loc": "40.7152,-73.9877",

"org": "AS22252 The City of New York",

"postal": "10002",

"timezone": "America/New\_York",

"readme": "https://ipinfo.io/missingauth"

}

**Scenario#3(A)**: Filtered response test.

Expected result: <https://ipinfo.io/8.8.8.8/org> returns: AS15169 Google LLC

<https://ipinfo.io/8.8.8.8/city> returns: Mountain View

<https://ipinfo.io/8.8.8.8/country> returns: US

**Scenario#3(M)**: Requestor Authentication test using terminal with different methods (curl -u $TOKEN: ipinfo.io; curl -H "Authorization: Bearer $TOKEN" ipinfo.io; $ curl ipinfo.io?token=$TOKEN)

1. Invalid Token / params
2. 2 Tokens simultaneously

Expected result: Authentication methods are working, TOKENs validated, invalid detected and rejected. Request without API address ⇒ return my address data.

**Scenario#4(M)**: Testing responses per plan using terminal commands :

1. Free plan curl
2. Business plan
3. Enterprise plan
4. Invalid Token / params
5. 2 Tokens simultaneously

Expected Result: data depth presented per plan.

**Scenario#5(M)**: Rate Limits test - 50k. Expected result: get 429 if over limit.

**Scenario#4(M):** Test exceptions of the negative tests. Expected result: Exceptions are relevant and informative.