**Steps for Executing Tests in Cucumber:**

Step 1: Install java and Eclispe IDE in personal computer

Step 2: Open Eclipse IDE assign work space and create Maven project.

Step3: Open pom.xml and add dependencies for the following Jars.

1.Selenium Java

2.Cucumber Java

3.Cucumber Junit

4.Junit

Step 4: Create a folder with name features under the src/test/java then create new file with .feature extension and write all the scenarios using Gherkin language.

Step 5: Create package under src/test/java as Stepdefinition then create a new class and write code for executing tests.

Step 6: Create a Runner class file to link feature file(scenarios) and stepdefiniton package(Selenium code) for executing tests.

Step 7: For Executing the Test script follow the below steps

-->Open “TestRunner” class.

-->Right click and select—Run As--JUnit

Step 8: After successful script execution HTML Report will be generated in the target folder

(target-->cucumber-html-report-->index.html)

The following things will happen after execution

1)An instance of Firefox web browser will open

2)It will open the we transfer site on the browser

3)It will detect the agree button preforms click action.

4)It will detect “Add files” option and then browse’s and upload’s the file.

5)It will enter all the input details.

6)It will Detect “Transfer” but and then transfer the file.

7)Then browser will be closed.

8)Automatically reported will be generated in “target” folder

**Note:**

1)Disable the browser security pop-ups before executing script.

2)If you are exporting the project, change the location of file to be uploaded as per PC.

I.e., change the string selection value in the code.

3)If you want transfer file to more users, make changes in Examples of the feature file.

4)After executing script Refresh the target folder to see generated html report

BugReporiting

|  |  |
| --- | --- |
| Bug Id | Upload \_Process\_1 |
| Name | Page Not Found (404) |
| Reporter | Surekha |
| Date | 28/01/2020 |
| Summary | After uploading a valid file for transferring in the we transfer site, it is getting re-directed to page with a message “404 Page Not Found”. |
| Site URL | https://wetransfer.com |
| Screenshot | Path of the screenshot |
| Operating System | Windows 10 |
| Browser | Firefox |
| Severity | Major |
| Priority | High |

**Steps to reproduce**

>Launch site in the Firefox browser.

>Select Add your files and browse the Test file.

>Fill the Email to, Your Email and Message fields

>Click on Transfer.

**Test Case**

|  |  |  |
| --- | --- | --- |
| Email to | Venkatsiva789@gmail.com | Valid |
| Your Email | Surekhagoddli417@gmail.com | Valid |
| Message | Hello | Valid |
| File Uploaded | Path of the test file | Valid |

**Expected Result**

File has to be uploaded and users triggered with a successful transfer message to the mentioned mails.

**Actual Result**

Re-directing to page with “404 page Not Found” message.

**API Testing:**

**GET:**

Positive scenarios:

>Validate the User Id by sending HTTP GET Request to server.

>Verify User Ids with an integer number in between 1 to 999999 are valid.

>Validate the user details retrieved from server for valid GET Request.

I.e,

{

User\_id:1234,

User\_name:”testUser1”,

Telephone:”0123456789”

Permissions[“read”,”write”]

}

>Verify the status code has to be 200 for successful request.

>Verify the time taking for receiving response.

>Verify the size of the response body

>Verify cookies and headers if exist

Negative Scenarios:

>Make the GET Request with invalid user id such as Negative number/Null (0)/string/Decimal number/greater than 999999

>For In-valid request it returns failure response with status code 404.

**GET/all-users {page no}**

Positive Scenarios:

>Make a request to get all the records present in one particular page with page\_no.

>Verify that the success response is having the user details as mentioned below.

{

User\_id:1234,

User\_name:”testUser1”,

Telephone:”0123456789”

Permissions[“read”,”write”]

}

>Verify the status code

>Validate time taking for response, size of response file, Headers and cookies.

Negative Scenarios:

>Make a request with invalid page no i.e. negative, string, decimal, null, number which does not exist.

>Verify the failure response body and status code.

**Post**

>Create and validate the user details by HTTP POST Request.

>Verifying by passing valid values to “user\_name”, “telephone”, “Permissions”

Eg:- {

username:” testUser1”,

telephone:”0123456789”

permissions [“read", "write”]}

>Verify that User\_Id is created for a successful request and validate created User\_id

>Verify that for valid details the new record is created and return success response and status code as 201 Created.

Negative Scenarios:

>Verify that for the invalid POST request it returns failure response with status code.

Invalid Request:

-->Trying to create details without mandatory fields I.e. User\_name and Permissions

-->Creating POST Request with existing user name.

-->Verify that we can perform read or write operations as per the given permissions.

**Delete:**

>Create a request to delete the user details by unique user\_id.

>For request with the Integer number in between 1 to 999999 as User\_id, details have to be deleted and return status code as 200.

>For request with the integer number null/negative number/string/decimal/greater than 999999, the request has to fail and return Failure response.