1. **Test Case:**

As a tester, I want to make sure no JavaScript errors when you visit

<https://www.mytheresa.com/en-de/men.html>

**Observation Found:**

When I Launch the application Manually page Loaded fine without any errors. But while launching the application Through automation Script , the page throws an error as **Access Denied You don't have permission to access "website" on this server.**

**Approach and Solution:**

1. Use Selenium Tool to Automate the Application
2. To handle or avoid the JS error throwing the page , I will use USERAGENT .

**Solution1** : I will use the below code to overcome the Access denied issue across the multiple browsers.

user\_agent = ('Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_10\_1) '

'AppleWebKit/537.36 (KHTML, like Gecko) '

'Chrome/39.0.2171.95 Safari/537.36')

**Solution2** : I will use ChromOptionsClass or Fire Fox preferences with add arguments method to control the browser properties and try to overcome the access denied issue.

1. Once I Overcome the Access denied issue , to ensure that there are no JS errors

**Solution1**: I will user driver.manage().logs()get(Logtype.BROWSER) which returns the LogEntires and with logEntries.getAll().size() which tell the no of JS errors present .I will ensure that if the size is 0 , there is no JS errors present

**Solution2:** I will use Desired Capabilities or Fire Fox preferences to control and handle the JS errors in browser using the logging preferences . Similarly with the respective browser capabilities we can detect the JS errors and verify it

1. For verifying the all the valid links in the application , below are the steps followed

**Step1 .** Fetch the list of all the links inside theapplication driver.findElements(By.tagName("a")

**Step2 :** Iterate the List items and using getAttribute(“href”) and store the URL in URL class.

**Step3:** Create an object of URL class and open the connection to the URL using httpconnection abstract class.

**Step4:** Using URL object , call the OpenConnection()and connect to all URL servers.

**Step5:** Validate if the Status codes range between 200 to 399 , The Test case pass and if Status code >=400 ,the Test case fails**.**

1. **Test Case**

As a product owner, I want to see how many open pull requests there are for our

product. You can use https://github.com/appwrite/appwrite/pulls as an example

product

1. Output is a list of PR in CSV format with PR name, created date and author

**Approach and Solution:**

**Pre-requisite:**

**Add Open CSV dependency Library in the maven POM.XML file**

**Solution:**

1. Launch the GitHub URL and Take Open request list size using the <a> tag using the xpath as //div[contains(@id,'issue\_')]. Note The list is present in the multiple pages .
2. Instantiate the CSV Writer class as below

**CSVWriter write = new Scripter(“Empty CSVFilePath”)**

1. To write the date into Csv file , we need to have Multiple arrays and List which holds the list of String arrays as below

**List<String[]> - Which holds the list of String array.**

1. Each String array should contain the column values as below

**String array1[] = { “PRname”,”CreateDate”,”Authorname”} -**

**For Eg : List<String[]> listobject;**

**Listobject.get(1)[0]= “PRname”**

**1- indicates the Row1**

1. We need to Iterate through the list and add the PRname, Created date and author from the GitHub URL

**For Eg: Hover on Pull request , Get the PRname, created date, Author and add to each row in the list.**

**Data may look like this**

**Row1 Array[] = { { “PRname”,”CreateDate”,”Authorname”}**

**Row2 Array[] = { { “PRname data”,”CreateDate”,”Authorname”}**

1. Now , Using CSVWriter Object , Write the List of String[] into CSV file

**Write.Writeall( listObject); -**  it Writes all the data into CSV file