

Source Code

Cucumber Framework for Amazon App

Amazon.feature

@all

Feature: Test Amazon page I want to use this template for my feature file

Background:

Given I am on Amazon Homepage

Scenario Outline: Testing of SignIn Page

When I click on signin button

Then I will be on Signin Page and Capture the title of the page

And I entered email id as '<emailId>' on the page

Then I clicked on Continue button

And I entered password as '<password>' on the page

And I click on submit button

Examples:

emailId	password
demouser123@gmail.com	demouser@1234

Scenario:

Then I start searching item

Then now i click on searched item

#And i add the item to wishlist for further booking

Then i close the browser

Amazon_steps.java

```
package steps;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class Amazon_steps {

    WebDriver driver;

    @Given("I am on Amazon Homepage")
    public void i_open_the_amazon_page() {
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.amazon.in/");
    }

    @When("I click on signin button")
```

```

public void I_click_on_signin_button() {
    driver.findElement(By.linkText("Sign in")).click();
}

@Then("I will be on Signin Page and Capture the title of the page")
public void i_will_be_on_signin_page_and_capture_the_title_of_the_page() {
    System.out.println("title of the is page is : "+ driver.getTitle());
}

@Then("I entered email id as {string} on the page")
public void i_enter_email_id_as_seleniumdemo_gmail_com_on_the_page(String
emailId) {
    driver.findElement(By.id("ap_email")).sendKeys(emailId);
    //driver.findElement(By.xpath("//input[@id='continue']")).click();

}

@Then("I clicked on Continue button")
public void i_clicked_on_continue_button() {
    driver.findElement(By.xpath("//input[@id='continue']")).click();
}

@Then("I entered password as {string} on the page")
public void i_enter_password_as_on_the_page(String password) {
    driver.findElement(By.id("ap_password")).sendKeys(password);
}

@Then("I click on submit button")
public void i_click_on_submit_button() {
    driver.findElement(By.id("signInSubmit")).click();
}

```

```
}
```

```
@Then("I start searching item")
```

```
public void I_start_searching_item() {
```

```
    driver.findElement(By.id("twotabsearchtextbox")).sendKeys("Mens Shirts");
```

```
    driver.findElement(By.id("twotabsearchtextbox")).submit();
```

```
}
```

```
@Then("now i click on searched item")
```

```
public void now_i_click_on_searched_item() {
```

```
driver.findElement(By.xpath("(//div[@class='a-section'])[2]/descendant::span[11]")).click();
```

```
}
```

```
/*@Then("i add the item to wishlist for further booking")
```

```
public void i_add_the_item_to_wishlist_for_further_booking() {
```

```
    driver.findElement(By.id("wishlistButtonStack")).click();
```

```
}*/
```

```
@Then("i close the browser")
```

```
public void i_close_the_browser() {
```

```
    driver.close();
```

```
}
```

```
}
```

TestRunner.java

```
package runner;
```

```
import org.junit.runner.RunWith;
```

```
import io.cucumber.junit.Cucumber;
```

```
import io.cucumber.junit.CucumberOptions;
```

```
@RunWith(Cucumber.class)
```

```
@CucumberOptions(features =
```

```
"C:\\Users\\Swetha\\eclipse-workspace\\Cucumber-Workspace\\CucumberTesting\\src\\test\\java\\features\\Amazon.feature",
```

```
glue = { "steps" }, dryRun = false, plugin = { "pretty",  
"html:target/cucumberreport.html",
```

```
"com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:",  
"timeline:test-output-thread/"}  
  
)
```

```
public class TestRunner1 {
```

```
// A cucumber runner file is used to build a communication between your feature
```

```
// files and step definition file
```

```
// Whenever we have to run multiple features or multiple steps, then we use
```

```
// Cucumber runner file.

// Whenever have to generate reports , we will use cucumber runner file

/*
 * - cucumber report - Extent report - Cucumber reports in PDF format - HTML
 * Reports - TestNG Reports
 *
 * Various Cucumber options or properties :
 *
 * features : comma separated paths to feature file. dryRun=true : It will check
 * if every steps has an step defination method implemented or not glue : comma
 * separated package names of step def. file plugin = pretty : Tags
 *
 */
}
```

Extent.properties

```
extent.reporter.spark.start=true
extent.reporter.spark.out=test-output/SparkReport/Spark.html
extent.reporter.spark.config=src/test/resources/extent-config.xml

extent.reporter.spark.out=test-output/SparkReport/

screenshot.dir=test-output/
screenshot.rel.path=../
extent.reporter.pdf.start=true
extent.reporter.pdf.out=test output/PdfReport/ExtentPdf.pdf
#basefolder.name=reports
#basefolder.datetimepattern=d-MMM-YY HH-mm-ss
extent.reporter.spark.vieworder=dashboard,test,category,exception,author,device,log
systeminfo.os=Windows
systeminfo.user=swetha
systeminfo.build=1.1
systeminfo.AppName=AutomationPractice
```

Extent-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<extentreports>
  <configuration>
    <!-- report theme -->
    <!-- standard, dark -->
    <theme>dark</theme>
    <!-- document encoding -->
    <!-- defaults to UTF-8 -->
    <encoding>UTF-8</encoding>
    <!-- protocol for script and stylesheets -->
    <!-- defaults to https -->
    <protocol>http</protocol>
    <!-- title of the document -->
    <documentTitle>Extent</documentTitle>
    <!-- report name - displayed at top-nav -->
    <reportName>Grasshopper Report</reportName>
    <!-- location of charts in the test view -->
    <!-- top, bottom -->
    <testViewChartLocation>bottom</testViewChartLocation>
    <!-- custom javascript -->
    <scripts>
      <![CDATA[
$(document).ready(function() {
});
]]>
    </scripts>
    <!-- custom styles -->
    <styles>
      <![CDATA[
]]>
    </styles>
  </configuration>
</extentreports>
```