3/26/2020

SELENNIUM AND CUCUMBER

Submitted by: Justine Jose

S8 INMCA

Roll No : 38

Submitted to: Ms. Sona Maria Sebastian

Asst. Professor

Department of MCA

Amal Jyothi College of Engineering

SELENIUM

SELENIUM is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done using the Selenium tool is usually referred to as Selenium Testing.

- Test scripts can be written in any of these programming languages: Java, Python, C#, PHP, Ruby, Perl & .Net
- Tests can be carried out in any of these OS: Windows, Mac or Linux
- Tests can be carried out using any browser: Mozilla Firefox, Internet Explorer, Google Chrome, Safari or Opera
- It can be integrated with tools such as Testing & JUnit for managing test cases and generating reports
- It can be integrated with Maven, Jenkins & Docker to achieve Continuous Testing

- We can use Selenium only to test web applications. We cannot test desktop applications or any other software
- There is no guaranteed support available for Selenium. We need to leverage on the available customer communities
- It is not possible to perform testing on images. We need to integrate Selenium with Sikuli for image based testing
- There is no native reporting facility. But we can overcome that issue by integrating it with frameworks like TestNG or Junit

Selenium Installation Steps

- *Install Java SDK*
- *Install Eclipse*
- *Install selenium Driver file*
- *Configure eclipse ide with webdriver*
    - *Launch eclipse .exe*
    - *Create a new project file -> new -> java project*
    - *Give project name, location select an external jre, select your project option finish*
    - *Right click new project -> new package -> package name -> finish*
    - *Create a class*

- *New project ->properties -> java build path -> libraries -> add external JARS*
- *Select file outside library folder -> apply and close*
- *Add all JAR file inside and outside the library*

CUCUMBER

Cucumber is a tool that supports Behaviour Driven Development (BDD). It offers a way to write tests that anybody can understand, regardless of their technical knowledge. In BDD, users (business analysts, product owners) first write scenarios or acceptance tests that describes the behaviour of the system from the customer's perspective, for review and sign-off by the product owners before developers write their codes. Cucumber use Ruby programming language.

Behaviour Driven Development (BDD)

BDD is a software development technique that has evolved from Test Driven Development which is an approach or programming

practice where the developer's writ new code only when the automated test is fail

BDD approach involves the usage of shared language that enhance communication between various tech and non-tech teams. Tests are more focused and based on the system behaviour.
I

Advantages of Cucumber

1. *It is helpful to involve business stakeholders who can't easily read code*
2. *Cucumber Testing focuses on end-user experience*
3. *Style of writing tests allow for easier reuse of code in the tests*
4. *Quick and easy set up and execution*
5. *Efficient tool for testing*

How does cucumber testing work?

*When you run **Cucumber**, it reads in your specifications from plain-language text files called features, examines them for*

*scenarios to **test**, and runs the scenarios against your system. Each scenario is a list of steps for **Cucumber** to **work** through. involve using an automation library/framework*

Cucumber installation steps

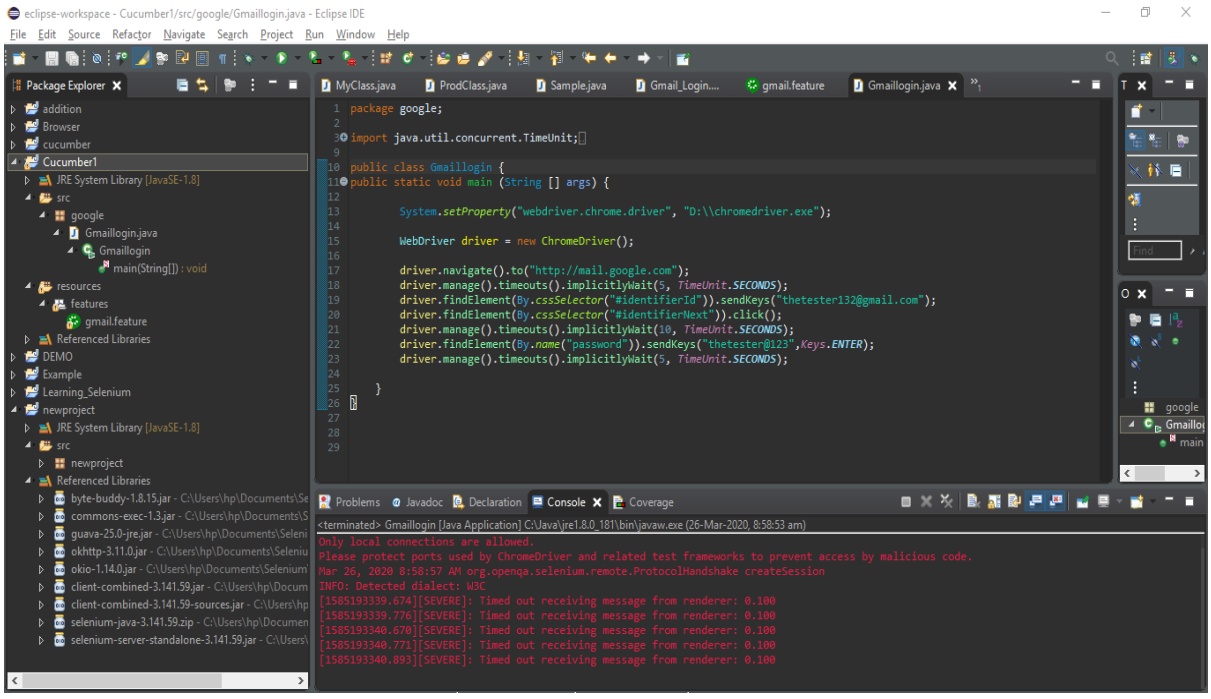
- *Install ruby and DevKit*
- *Open the download file and accept license*
- *Select your installation directory*
- *Install Cucumber*
    - *Type in ruby cmd "gem install cucumber" To verify cucumber is installed type "cucumber –version"*
- *Install RubyMine*
    - *In setup window open -> next -> select directory ->next -> create desktop shortcut ->next -> install -> finish -> ok*

      Evaluate for free -> Evaluate -> Accept license.

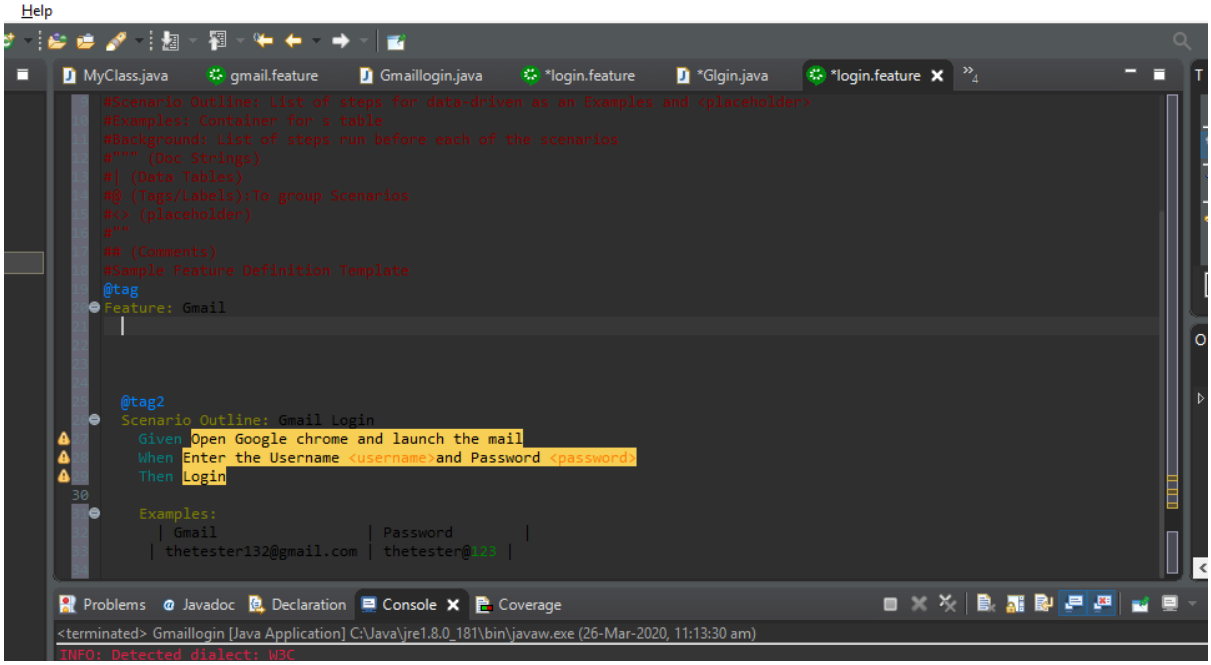
- *Install Webdriver*

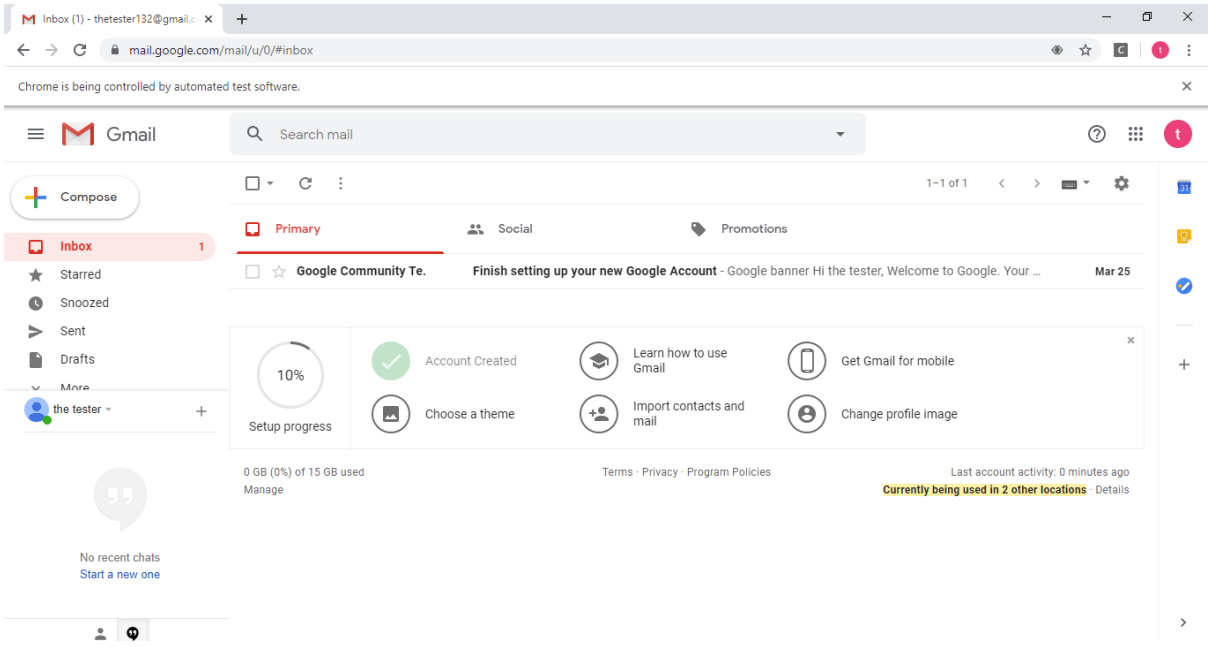
- *Start cmd with ruby cmd and install command "gem install water-webdriver".*

OUTPUT



Help


MyClass.java
gmail.feature
Gmaillogin.java
*login.feature
*Glgin.java
*login.feature
#Scenario Outline: List of steps for data-driven as an Examples and <placeholder>
#Examples: Container for s table
#Background: List of steps run before each of the scenarios
#""" (Doc Strings)
#| (Data Tables)
#@ (Tags/Labels):To group Scenarios
#<> (placeholder)
#""
(Comments)
#Sample Feature Definition Template
@tag
Feature: Gmail
@tag2
Scenario Outline: Gmail Login
Given Open Google chrome and launch the mail
When Enter the Username <username>and Password <password>
Then Login
Examples:
| Gmail | Password |
| thetester132@gmail.com | thetester@123 |
Problems
Javadoc
Declaration
Console
Coverage
<terminated> Gmaillogin [Java Application] C:\Java\jre1.8.0_181\bin\javaw.exe (26-Mar-2020, 11:13:30 am)
INFO: Detected dialect: W3C


Inbox (1) - thetester132@gmail.c
mail.google.com/mail/u/0/#inbox
Chrome is being controlled by automated test software.
Gmail
Search mail
Compose
Inbox 1
Starred
Snoozed
Sent
Drafts
More
the tester
No recent chats
Start a new one
1–1 of 1
Primary
Social
Promotions
Google Community Te.
Finish setting up your new Google Account - Google banner Hi the tester, Welcome to Google. Your ...
Mar 25
10%
Setup progress
Account Created
Learn how to use Gmail
Get Gmail for mobile
Choose a theme
Import contacts and mail
Change profile image
0 GB (0%) of 15 GB used
Manage
Terms · Privacy · Program Policies
Last account activity: 0 minutes ago
Currently being used in 2 other locations · Details