1.11 Page Object Design Pattern



This section will guide you to:

* Implement the Page Object Model (POM)

This guide has mainly three sub-sections, namely:

1.11.1 Explaining why POM is used

1.11.2 Implementing Page Object Model

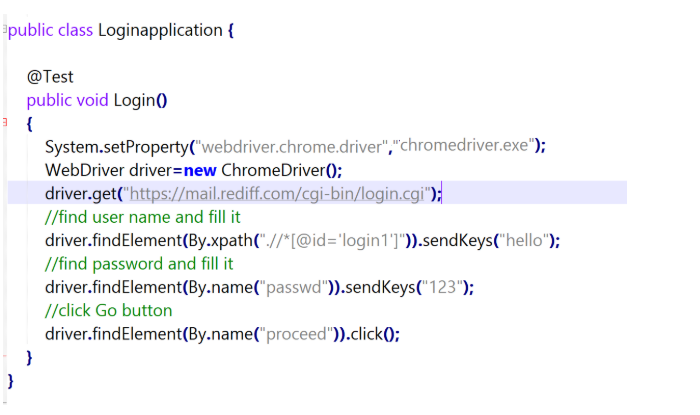
1.11.3 Writing code to implement Page Object Model using a test case

1.11.4 Pushing the code to your GitHub repositories

**Note: Make sure that you provide your specific file location for any terminal path given throughout the assisted practice**

**Step 1.11.1:** Explaining why POM is used

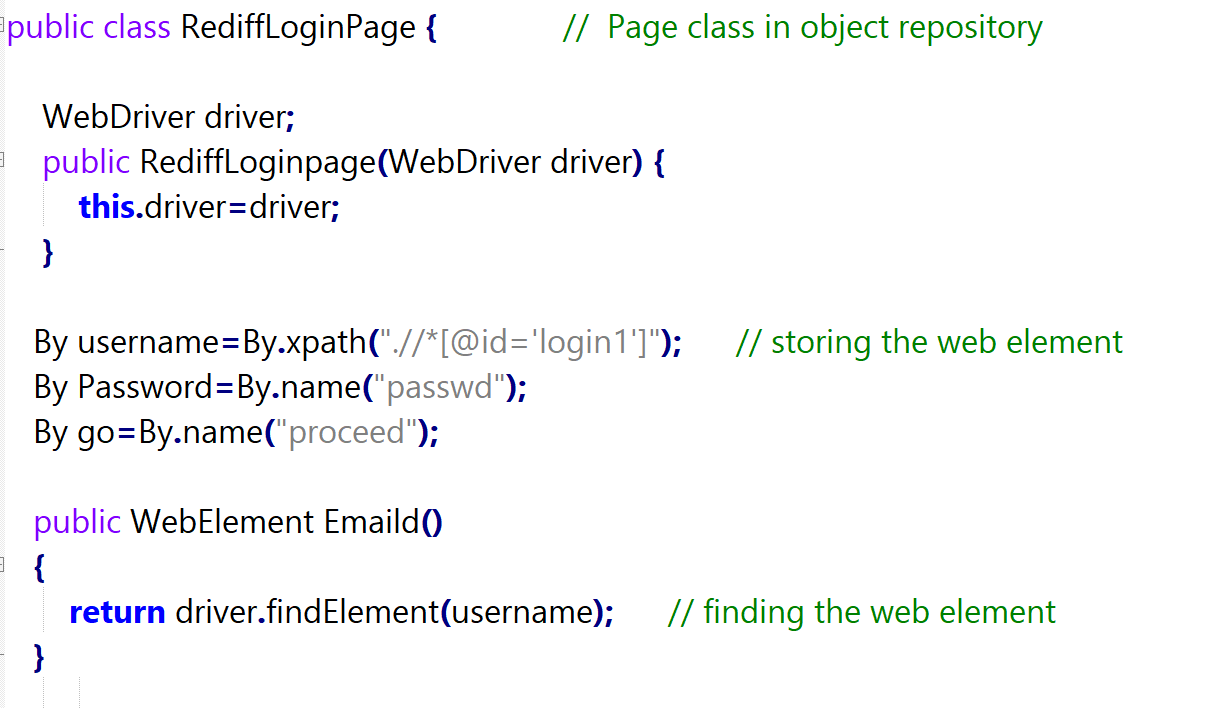
* Consider this simple script to log in to the Rediff mail website.



* As you can see, all we are doing is finding elements and filling values for those elements.
* This is a small script. Script maintenance looks easy. But with time the test suite will grow. As you add more and more lines to your code, things become tough.
* The main problem with script maintenance is that if 10 different scripts are using the same page element, with any change in that element, you need to change all the 10 scripts. This is time-consuming and error-prone.
* A better approach to script maintenance is to create a separate class file, which would find web elements, fill them, or verify them. This class can be reused in all the scripts using that element. In the future, if there is a change in the web element, we need to make the change in just 1 class file and not 10 different scripts.

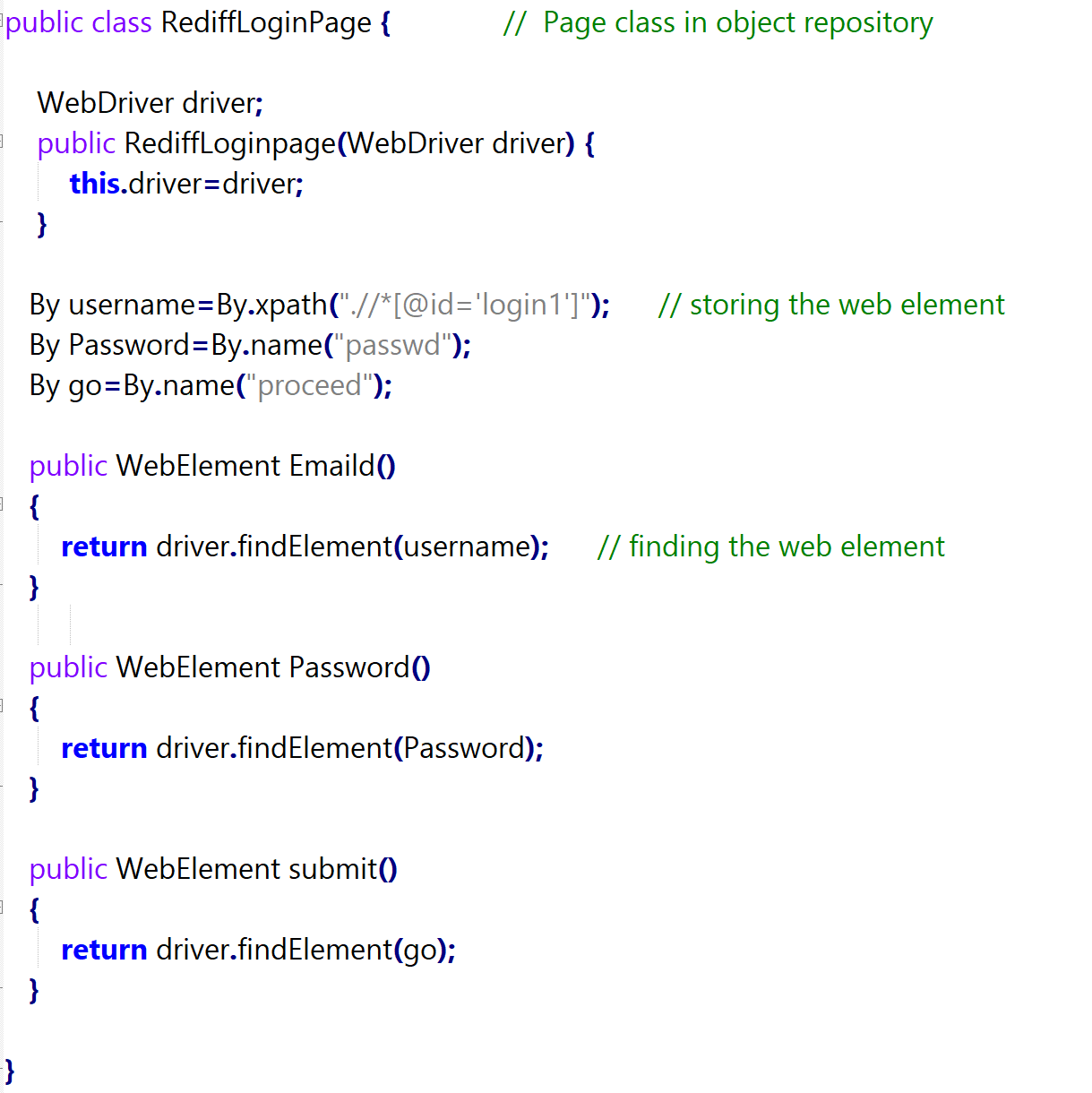
**Step 1.11.2:** Implementing Page Object Model

* This is the basic structure of the Page Object Model (POM), where all Web Elements of the Application Under Test and the method that operate on these Web Elements, are maintained inside a class file.

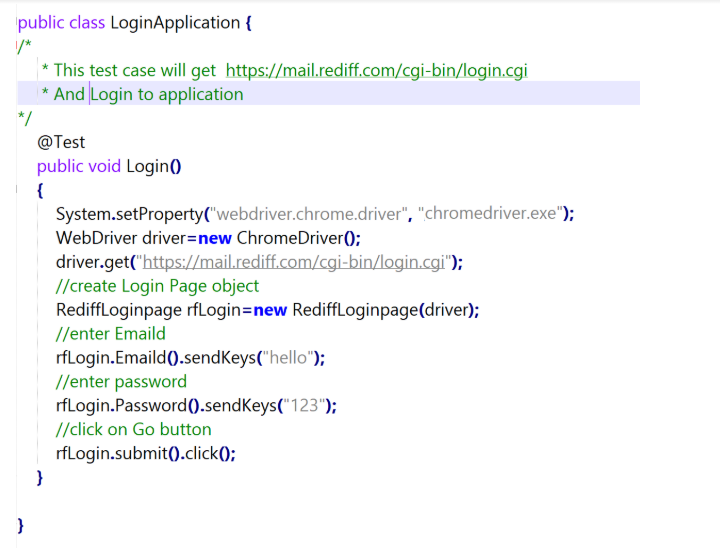


**Step 1.11.3:** Writing code to implement the Page Object Model using a test case

* Test Case: Log in to Rediff mail. Here we will be dealing with Login page POM and Test script.
* Login Page:



* Test Case:



**Step 1.11.4:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master