

CAFETOWNSEND WEB APPLICATION TESTS:

Automating an Angular Web Page with Page Objects and Cucumber Framework by the help of ngWebDriver.

Project link: <https://github.com/kinoeater/cafetownsend>

I. Introduction

This document mainly serves as the plan for testing all software artifacts as well as the reporting of test results.

II. Automation Framework

Why " NGWebDriver " but not Protractor?

At first I was thinking to automate the test cases by using protractor. Although I had some experience on that tool (there is a small project on my github) and it is a good fit for angular apps, I also want to benefit from page objects methodology and Cucumber framework. That was the point I came up to use NGWebDriver. NGWebDriver is a small open source library of WebDriver locators created by " Paul Hammant " and more for AngularJS (v1.x) and Angular (v2.x +), for Java. It works with Firefox, Chrome and all the other Selenium-WebDriver browsers.

<https://github.com/paul-hammant/ngWebDriver>

Dowside of using NGWebDriver:

Unfortunately it is not a magic tool that make QA people happy with angular app testing. Sometime it has causing synchronisation issues.

Why " Cucumber" and "Page Objects"?

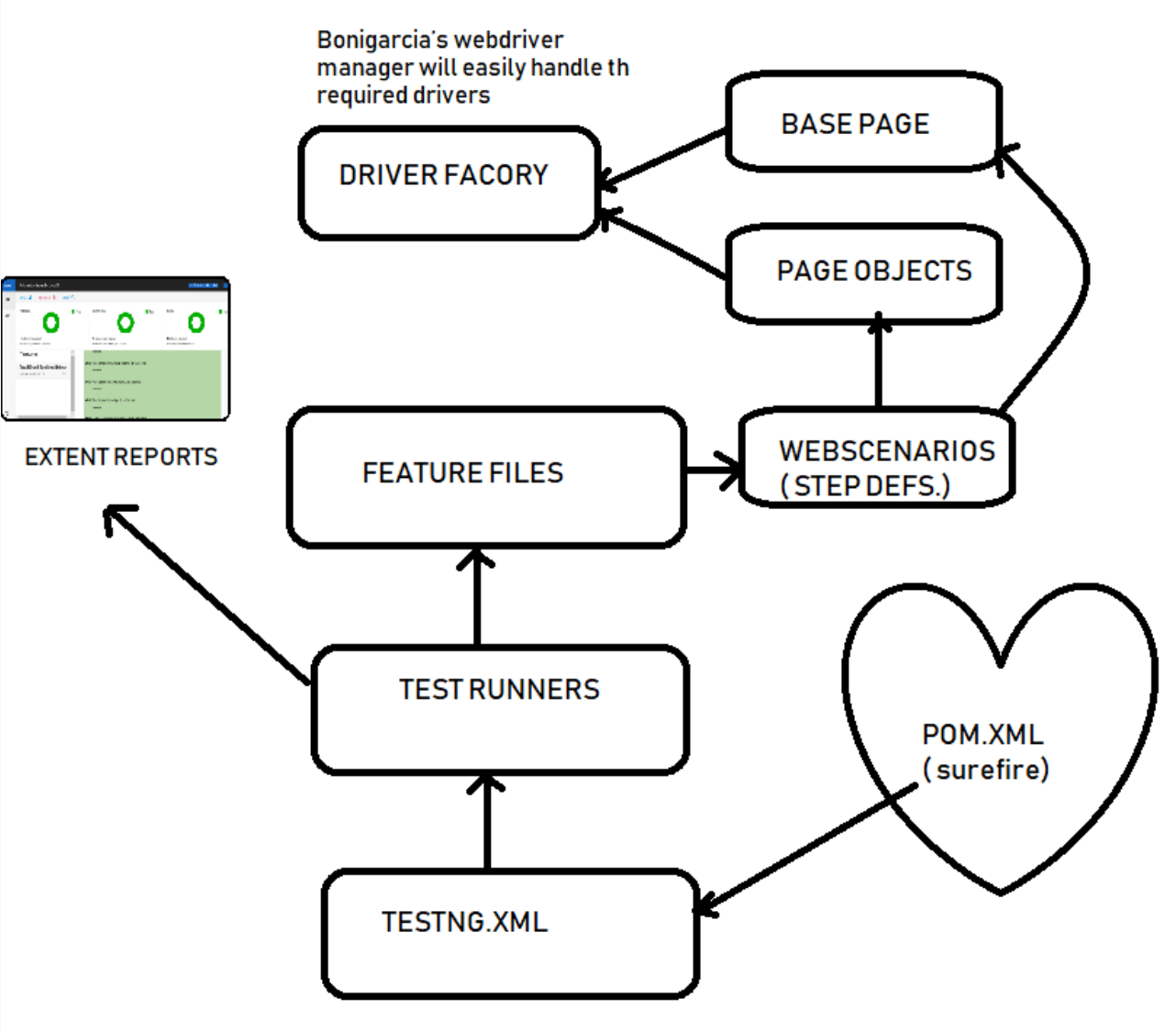
By using NGWebDriver library we can handle angular application easily and create page objects with ByAngular locator helper. Another thing page objects design pattern helps us to enhance test maintenance and reduces code duplication. Maybe the best part of this framework is using gherkin language to create easily readable test cases within the automation. They are are also reusable and easy to edit. Moreover tomorrow if we want to add more tests it very easy, we can pick proper methods from the base page then stick with the step definitions of the gherkin files.

```
Scenario: Successfully creates a new user then deletes it

Given User should be at home page
And User clicks on the Create button
And User should be in new user page
And User enters following details for new user
| First_name | Last_name | Start_date | Email |
| Alfred     | Pennyworth | 1933-03-14 | Pennyworth@dark.com |
When User clicks on the Add button
Then User should see "Alfred Pennyworth" in the employees list
And User deletes "Alfred Pennyworth"
# This one deletes newly added employee, so same credentials can be used next time
And User logs out
```

Cucumber Framework

You can choose firefox or chrome at the gherkin level. You do not need to bother with updating the browser drivers, because Bonigarcia’s webdriver manager will easily handle it. If you want to use different browsers like IE, code can also be modified. Page Object page has its specific web elements in it and Base Page has some specific easy to use methods. More methods can be added if required. It is like a toolbox of the framework, use the gadgets, add new gadgets when you want.

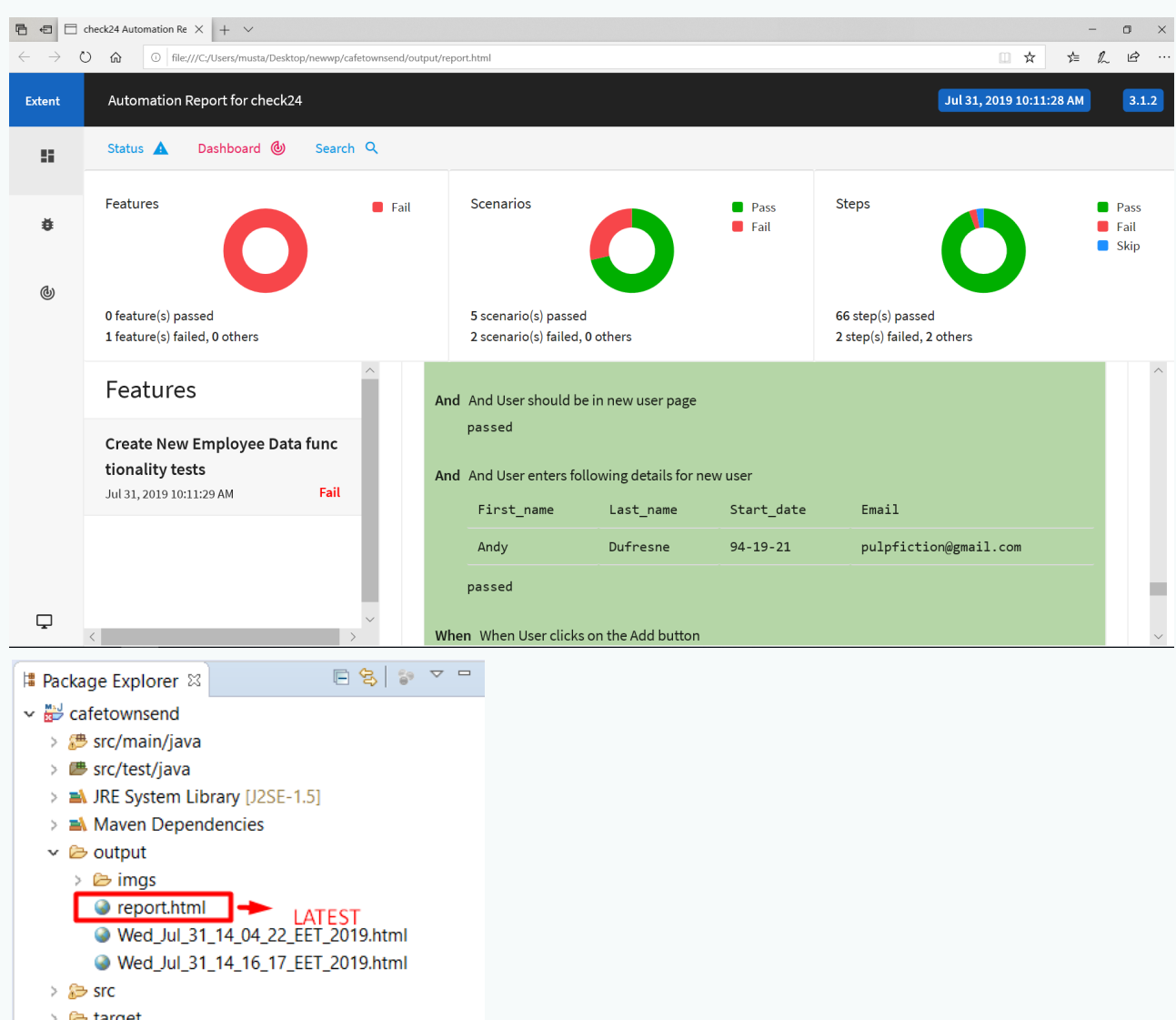


MOBIQUITY QA TASK

CAFETOWNSEND-ANGULAR-RAILS.HEROKUAPP TESTS

What about reporting?

I used Cucumber Extent Report as reporting tool. After each test run latest test report can be seen with "report.html" name under /output directory. Report can be opened simply double clicking on the item.

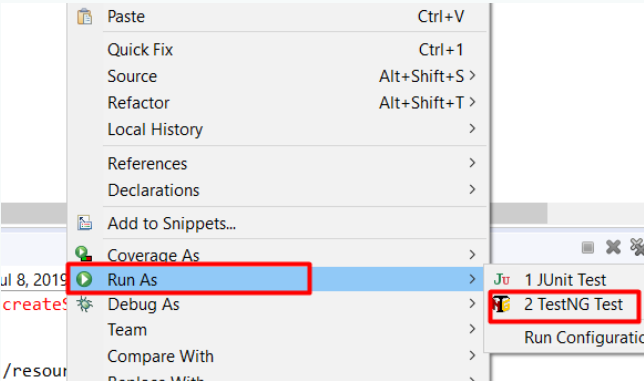


Running the Tests

1. By directly using runner class on IDE

Prerequisites

- *Install Java and Maven to your machine.
- *Install an IDE of your choice, Eclipse or IntelliJ.
- *Install cucumber, gherkin and Test NG plugins to your IDE.



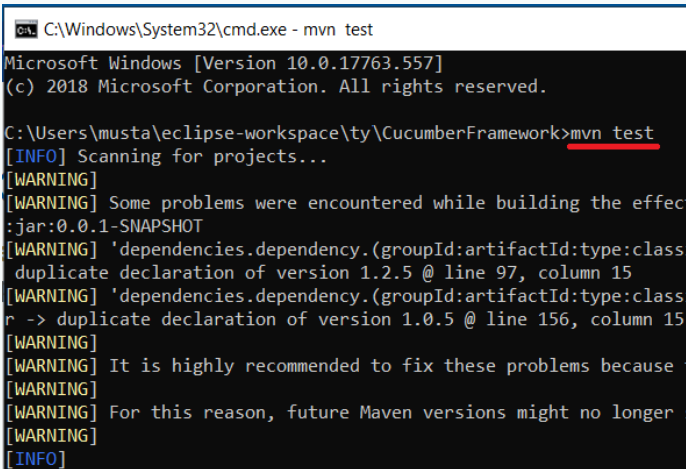
- a) You should open / choose one of the runner classes under /check_24_Framework/src/test/java/runners
- b) Then you should right click on the page or on the class,
- c) Then you should click on TestNG option.

2. By using "mvn test" command on CLI

Prerequisites

- *Install Java and Maven to your machine.

- a) You go to root directory of the maven project from command line (CMD / terminal)
- b) Then type followings
mvn test
- c) Then hit enter



**Surefire plugin is added to the POM, so you can also put the Jenkins in work

III. Manual Test Cases

Black box manual test cases that will run on the code. Rather creating all of the test cases I wrote down the the test cases for login and create new user functionalities.
The URL: <http://cafetownsend-angular-rails.herokuapp.com>

Login functionality tests:

- Verify that user can successfully login with correct credentials.
 1. Verify that user cannot login with incorrect user name.
 2. Verify that user cannot login with incorrect password.
 3. Verify that user cannot login with empty email
 4. Verify that user cannot login with empty password
 5. Verify that user cannot login with empty email and empty password

MOBIQUITY QA TASK

CAFETOWNSEND-ANGULAR-RAILS.HEROKUAPP TESTS

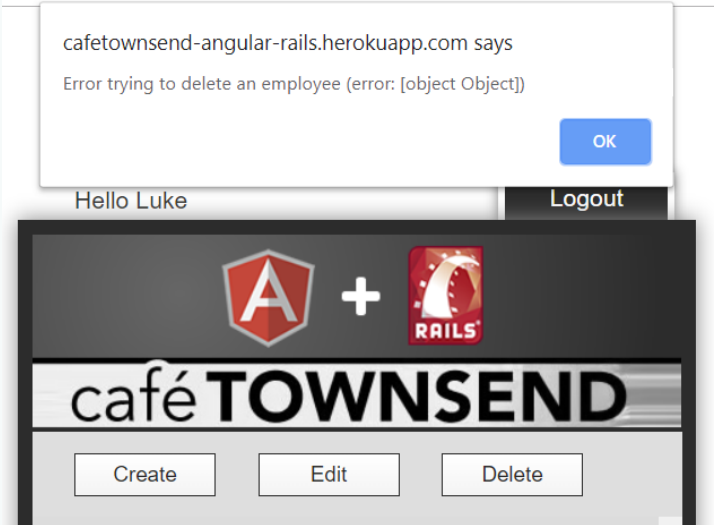
Create new employee functionality tests:

1. Verify that user can successfully creates a new user by entering whole credentials correctly, then deletes newly created user entry.
2. Verify that User leaves the first name area empty, then cannot create the new user entry.
3. Verify that User leaves the last name area empty, then cannot create the new user entry.
4. Verify that User leaves the start date area empty, then cannot create the new user entry.
5. Verify that User leaves the email area empty or enters email without ' @ ' sign, then cannot create the new user entry.
6. Verify that User enters whole credentials correctly, then hits "enter key" on keyboard instead of using Add button, then successfully creates a new user.
7. Verify that User enters incorrect date format, then presented with an alert and cannot create the new user entry.

Note: Although more test cases can be created I limited the numbers for these two features.

CLIST OF ISSUES IDENTIFIED

1. In some cases, while the automation script tries to delete an entry web page throws an error stating that " Error trying to delete an employee..". That also happen if we try to delete entries several times. To handle that kind of errors, exeptions try-catch mecanism can be used.



2. After logging out the app, the password is not deleted automatically although the username area is deleted.

