02/20/2020
Js_Day03

WarmUp:
if you did not attend yesterday' section or haven't create your protractor project ready:
1. go to canvas -> JavaScript -> Modules -> Day02 -> Protractor Project.zip
2. Download the zip folder
3. unzip it to the desktop
4. open visual studio code
5. open "protractor project" folder

install webdriver manager:
webdriver-manager update

start webdriver-manager:
webdriver-manager starts

End-to-End: runs in a single browser
focuses on what user expereince

</html>

<Script>
javascript codes
<Script>

</html>

protractor: it's an end to end test faremwork for AngularJS applications

protractor has 9 extra locators:
1. buttonText, 2. PartialButtonText, 3. cssContaingText, 4. model, 5. binding, 6. exactBinding, 7. repeater, 8. ExactRepeater, 9. deepCSS

we also can use selenium locators

Jasmine: it's BDD framework

Two important components in jasmine:
1. Config: runs the stepdefinitions

framework: define the framework name

seleniumAddress: remote web service

directConnect: directly connects

specs: defines where the test cases are

capabilities: allows us to use different browsers

multiCapabilities: multi- browsers

suites: creating test suites

onPrepare:

implicit wait

waitforangularEnavbled: false

maximize

Reports :HTML (protractor beautiful reports)

2. Spec: Step definition

describe: Test Group (contains your similar test cases)

can be nested

it: Test Case

Run the protractor: after get to the desired directory

protractor filename.js

annotation for describe: controls the excution flow of the test cases

beforeAll(function(){ codes }); runs before all test cases in the group started (similar to @BeforeClass)

afterAll(function(){ codes}); runs after all test cases in the group finished (similar with @AfterClass)

beforeEach(function() {code }); runs before each test cases (similar with @BeforeMethod)

afterEach(function(){ code }); runs after each test cases (similar with @AfterMethod)

browser: global object (same with webdriver)

```
browser.get(URL);
browser.sleep();
```

```
                browser.waitForAngularEnabled(false);


run test suites:
      protractor config.js --suites


Generate HTML reports:
                1. install protractor beautiful reporter
                                  npm install protractor-beautiful-
reporter --save-dev

                2. do one time setup in Onprepare statement:
                                  let HTMLReporter = require("protractor-
beautiful-reporter");
                        jasmine.getEnv().addReporter(
                               new HTMLReporter(
                                                {
                                                            baseDirectory:
"../Reports/VYTrackReports" ,

takeScreenShotsOnlyForFailedSpecs: true,

                                                      }
                                               ).getJasmine2Reporter()
                         );


pop up handlings:
      var alert = browser.switchTo().altert();
      alert.dismiss();
                or
      alter.accept();
      alert.dismiss();

                getText();


multi-windows:
      browser.getWindowHandles();

      browser.switchTo().window();

      browser.switchTo().defaultContent();


frames:
      browser.switchTo().frame();

      browser.switchToParentFrame();
```

Explicity wait:

```
var EC = protractor.ExpectedConditions;

browser.wait(EC.Condition, 10);
```

more methods at:
https://www.protractortest.org/#/api

Explicit wait vs implict wait vs browser.sleep():

Page Object Model: reusable, readable, easy to maintain

folders:

- Specs: where the test cases are located at
- Pages: stored the needed elements
- Config: runs the test case
- Resources: Json files
- Reports: HTML reports