3. Browsers handlings

Browser handling

When dealing with browsers, a good approach will be to separate the browser instantiate in a separate class, where to handle the browser capabilities, even if one or multiple browsers are used. This will permit the easy change of the browser used in tests.

One browser instance at a time

One way to do that is to define the declaration of each browser separately, and just call the desired browser from the main settings. Like:

```
public class Browser {

 public enum BrowserType {FIREFOX, IE, CHROME, HTMLUNIT}

 public static void createBrowser(BrowserType browser)
 {
  switch(browser)
  {
   case FIREFOX:
    try {
    Page.driver = new FirefoxDriver();
    Page.driver.manage().timeouts().implicitlyWait(Setup.EXPLICIT_WAIT,
TimeUnit.SECONDS);
    Page.maximize();
    }
    catch (Exception e) {
    e.printStackTrace();
    }
   break;
   case CHROME:
    try {
     Page.driver = new ChromeDriver();
     Page.driver.manage().timeouts().implicitlyWait(Setup.EXPLICIT_WAIT,
TimeUnit.SECONDS);
     Page.maximize();
     }
    catch (Exception e) {
     e.printStackTrace();
     }
   break;
   case IE:
    try {
     Page.driver = new InternetExplorerDriver();
     Page.driver.manage().timeouts().implicitlyWait(Setup.EXPLICIT_WAIT,
TimeUnit.SECONDS);
     Page.maximize();
     }
    catch (Exception e) {
     e.printStackTrace();
     }
   break;
  case HTMLUNIT:
   try {
    Page.driver = new HtmlUnitDriver();
    Page.driver.manage().timeouts().implicitlyWait(Setup.EXPLICIT_WAIT,
TimeUnit.SECONDS);
    Page.maximize();
    }
  catch (Exception e) {
   e.printStackTrace();
   }
  break;
  }
 }
}
```

When calling the browser, you can call it like, if we want to use Firefox for our tests:

```
Browser.createBrowser(Browser.BrowserType.FIREFOX);
```

Another approach of handling browsers would be to set the browser in a text setting file, and extract it from the file as we do for the environment values.

Text file:

- BROWSER=Firefox

Multiple browser instances one time

When trying to execute the tests on multiple browsers, the best approach to do that is to run the tests using Selenium GRID.

Before setting it up, there are also other ways of doing that, but are not that much recommended:

1. Put a loop around the tests. You have a list of different WebDrivers that the tests will run on:
   ```
   WebDriver[] drivers = new WebDriver[]{firefoxDriver, chromeDriver};
   for (WebDriver driver:drivers)
   { ...test goes here..... }
   ```
2. Usinh JUnit, you can use @DataProvider :
   ```
   @DataProvider(name = "drivers")
   public provideDrivers(){
    ...create drivers here...
   return new Object[][]{{firefoxDriver},{chromeDriver},....}; }

   @Test(dataProvider = "drivers")
   public runTest(WebDriver driver){
   ...do stuff with driver here... }
   ```