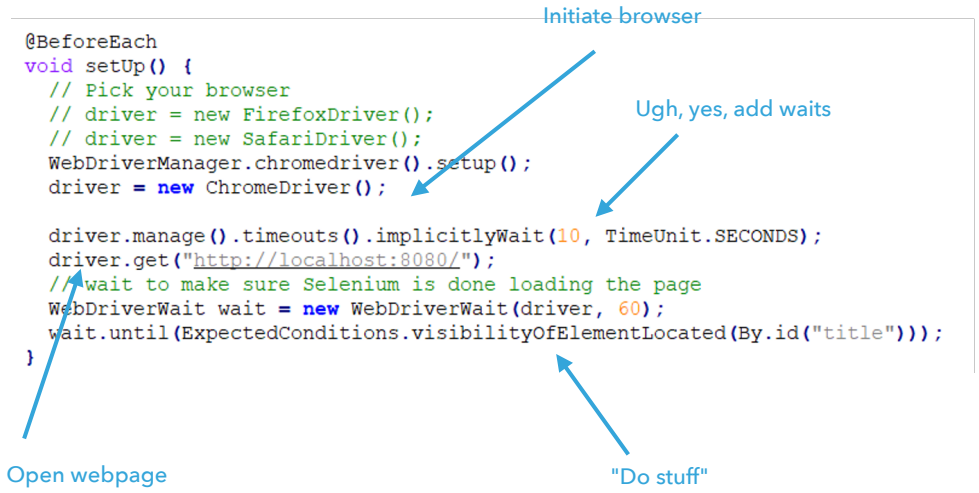
Automated User Acceptance Testing with Selenium WebDriver

- We must make sure that our **product's features** are in accordance **with users needs**.
- We can i**mitate user's possible interaction** with the application to make sure our application is **working correctly**.

- Selenium is an automated testing tool for web applications
- You can easily record users' interaction with web applications
- You can export the tests in various programming languages for reuse
- It is available across various browsers including Chrome and Firefox
- Selenium IDE extension for browsers is available and easy-to-use
- You can also download Selenium libraries and add it to your IDE

https://www.selenium.dev/selenium-ide/




```java
@BeforeEach
void setUp() {
  // Pick your browser
  // driver = new FirefoxDriver();
  // driver = new SafariDriver();
  WebDriverManager.chromedriver().setup();
  driver = new ChromeDriver();

  driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
  driver.get("http://localhost:8080/");
  // wait to make sure Selenium is done loading the page
  WebDriverWait wait = new WebDriverWait(driver, 60);
  wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("title")));
}
```

```
@Test
public void test3() {
  driver.get("http://www.sampleURL.com");
  WebElement username = driver.findElement(By.name("username"));
  WebElement password = driver.findElement(By.name("password"));
  username.sendKeys("admin");
  password.sendKeys("password");

  WebElement sign_in = driver.findElement(By.id("loginBtn"));
  sign_in.click();


  WebDriverWait wait = new WebDriverWait(driver, 60);

  String expectedUrl="http://sampleURL.com/login";
  String actualUrl=driver.getCurrentUrl();

  assertEquals(expectedUrl, actualUrl);
}
```

Objectives

- Maven to manage a java project
- Configuring Selenium Web Driver
- Automated Tests controlling a browser

Using seg3103_playground

- Create **/lab06** directory
  - **Read** the bookstore requirements
  - Extract **BookstoreApp.zip**

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

`src/main/java`	Application/Library sources
`src/main/resources`	Application/Library resources
`src/main/filters`	Resource filter files
`src/main/webapp`	Web application sources
`src/test/java`	Test sources
`src/test/resources`	Test resources
`src/test/filters`	Test resource filter files
`src/it`	Integration Tests (primarily for plugins)
`src/assembly`	Assembly descriptors
`src/site`	Site
`LICENSE.txt`	Project's license
`NOTICE.txt`	Notices and attributions required by libraries that the project depends on
`README.txt`	Project's readme

https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html

Having a common directory layout allows users familiar with one Maven project to immediately feel at home in another Maven project. The advantages are analogous to adopting a site-wide look-and-feel.

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< SEG3103:BookstoreApp >------------------------
[INFO] Building BookstoreApp 0.1.0
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ BookstoreApp
---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /private/tmp/seg3x03/BookstoreApp/src/
main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ BookstoreApp ---
[INFO] Nothing to compile - all classes are up to date
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.491 s
[INFO] Finished at: 2021-06-28T09:03:49-04:00
[INFO] ------------------------------------------------------------------------
```

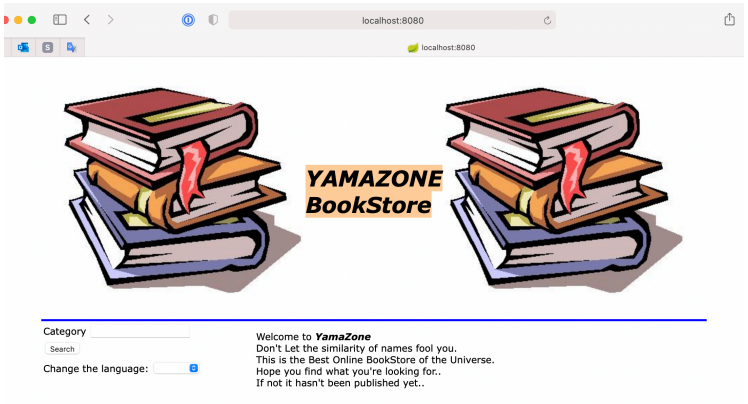
mvn compile

First, make sure you can compile the application.

```
[INFO] 
[INFO] ------------------------< SEG3103:BookstoreApp >------------------------
[INFO] Building BookstoreApp 0.1.0
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] 
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ BookstoreApp
---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /private/tmp/seg3x03/BookstoreApp/src/
main/resources
[INFO] 
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ BookstoreApp ---
[INFO] Nothing to compile - all classes are up to date
[INFO] 
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @
BookstoreApp ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /private/tmp/seg3x03/BookstoreApp/src/
test/resources
```

mvn package -DskipTests

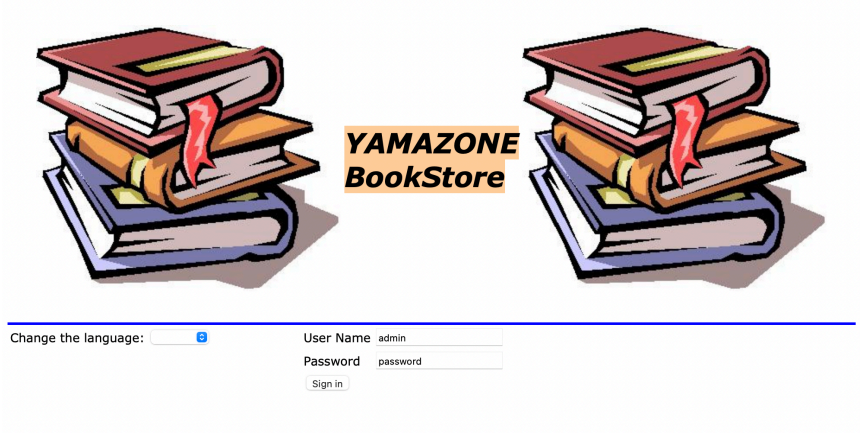
Then make sure you can package the application into a jar



java -jar ./target/BookstoreApp-0.1.0.jar

http://localhost:8080

Now launch the web application



http://localhost:8080/admin

Access to the administration is with username "admin" and password of "password

```
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 13.162 s
- in selenium.ExampleSeleniumTest
[INFO] Running ExampleTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in
ExampleTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  14.898 s
[INFO] Finished at: 2021-06-28T09:15:25-04:00
[INFO] ------------------------------------------------------------------------
```

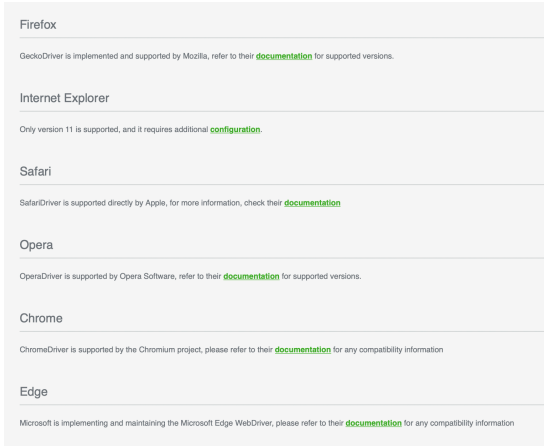
mvn test

Now try and run the tests.

```
[ERROR] test1  Time elapsed: 0.044 s  <<< ERROR!
org.openqa.selenium.WebDriverException:
Cannot find firefox binary in PATH. Make sure firefox is installed.
OS appears to be: MAC
Build info: version: '3.141.59', revision: 'e82be7d358', time:
'2018-11-14T08:17:03'
System info: host: 'Andrews-MacBook-Air.local', ip:
'fe80:0:0:0:cd6:4440:c26a:7fbf%en0', os.name: 'Mac OS X', os.arch:
'aarch64', os.version: '11.2.3', java.version: '17'
Driver info: driver.version: FirefoxDriver
  at
selenium.ExampleSeleniumTest.setUp(ExampleSeleniumTest.java:33)
```

Please read the output...

It probably will not work out of the box and will need tweaking to get working. Here above we see that FireFox isn't installed. There are three browsers configured, Chrome, Safari and FireFox. Others are possible and up to the students to work that out in their own developing environments.



Configure the correct browser

https://www.selenium.dev/downloads/

The code has snippets for Chrome, Firefox and Safari, but other browsers are available

```
//setting the driver executable
WebDriverManager.chromedriver().setup();

//Initiating your chromedriver
WebDriver driver = new ChromeDriver();

//Applied wait time
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

//maximize window
driver.manage().window().maximize();

//open browser with desired URL
driver.get("https://duckduckgo.com/");

//closing the browser
driver.close();
```

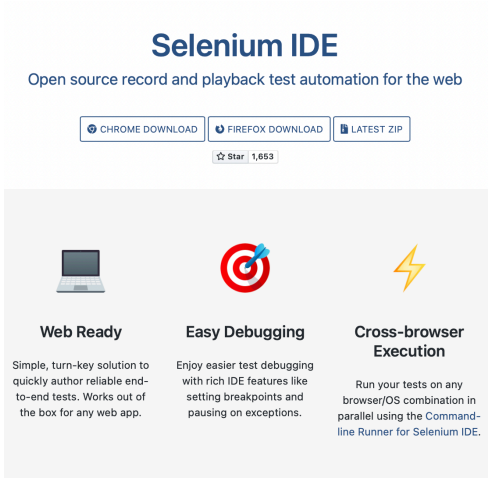

A hello world test.

Selenium WebDriver

If you want to create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver, a collection of language specific bindings to drive a browser - the way it is meant to be driven.

https://www.selenium.dev

Now start writing browser tests!!!



Selenium IDE to explore the app!!!

https://www.selenium.dev/selenium-ide/

The students can also explore the Selenium IDE to explore the bookstore application

Resources

- https://www.selenium.dev/selenium/docs/api/java/overview-summary.html
- https://www.browserstack.com/guide/selenium-with-java-for-automated-test
- https://www.browserstack.com/guide/locators-in-selenium
- https://www.browserstack.com/guide/verify-and-assert-in-selenium
- https://www.guru99.com/first-webdriver-script.html
- https://artoftesting.com/selenium-with-java-example
- https://examples.javacodegeeks.com/enterprise-java/selenium/selenium-tutorial-for-beginners/

Submission

- All work should written under
  - **seg3103_playground/lab06**
- Add at least one additional selenium web driver test
- Show screenshots of
  - Application running
  - Tests passing