

Downloading Sample UI Test Framework

In this lesson, we will download and understand the framework structure.

WE'LL COVER THE FOLLOWING ^

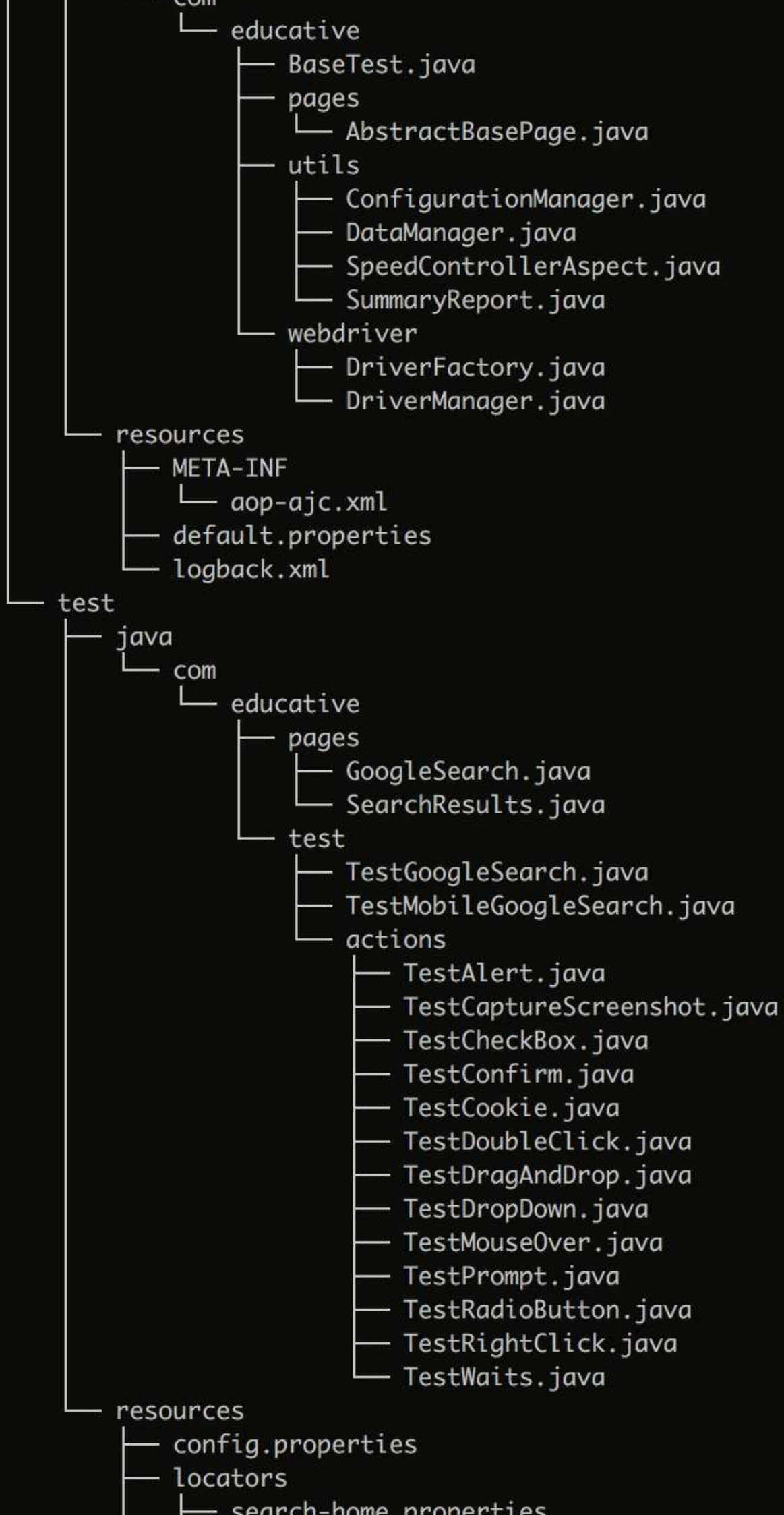
- Downloading framework code
- Project structure
 - src/main/java
 - src/main/resources
- Sample test written using the framework
 - src/test/java
 - src/test/resources

Downloading framework code

A sample framework developed using the discussed concepts can be downloaded from [here](#).

Project structure

```
$ tree .
.
├── build.gradle
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
│       └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
├── settings.gradle
└── src
    ├── main
    │   ├── java
    │   │   └── com
```



```
└─ search-home.properties
└─ search-results.properties
└─ testng.xml
```

21 directories, 38 files

src/main/java #

- `com/educative/test/BaseTest.java` - A class extended by all test classes
- `com/educative/pages/AbstractBasePage.java` - A class extended by all page objects
- `com/educative/utils/ConfigurationManager.java` - Managing the configuration
- `src/main/java/com/educative/utils/DataManager.java` - For managing static test data for different environments
- `src/main/java/com/educative/utils/SpeedControllerAspect.java` - An aspect class for controlling the speed of Selenium execution
- `src/main/java/com/educative/utils/SummaryReport.java` - An implementation of `IReporter` listener to generate a summary report of the test suite run
- `com/educative/webdriver/DriverFactory.java` - Creating WebDriver instance
- `com/educative/webdriver/DriverManager.java` - Creating and destroying WebDriver instance

src/main/resources #

- `default.properties` - A file holding all default configuration
- `logback.xml` - The configuration for logback

Sample test written using the framework

src/test/java #

- `com/educative/test/TestGoogleSearch.java` – A sample test for google search
- `com/educative/test/actions/TestRightClick.java` – A test for performing a right-click action

- `com/educative/test/actions/TestCookie.java` – A test for handling browser cookie
- `com/educative/test/actions/TestAlert.java` – A test for handling alerts
- `com/educative/test/actions/TestPrompt.java` – A test for handling prompts
- `com/educative/test/actions/TestMouseOver.java` – A test for handling mouse hover
- `com/educative/test/actions/TestConfirm.java` – A test for handling confirmation dialogue
- `com/educative/test/actions/TestRadioButton.java` – A test for working with radio buttons
- `com/educative/test/actions/TestDoubleClick.java` – A test for performing a double-click action
- `com/educative/test/actions/TestCheckBox.java` – A test for handling a check-box
- `com/educative/test/actions/TestCaptureScreenshot.java` – A test for taking screenshots
- `com/educative/test/actions/TestDragAndDrop.java` – A test for performing a drag and drop
- `com/educative/test/actions/TestDropDown.java` – A test for selecting from a drop-down menu
- `com/educative/test/actions/TestWaits.java` – A test for simulating various waits
- `com/educative/test/actions/TestCaptureScreenshot.java` – A test for showing screenshot capabilities

`src/test/resources` #

- `config.properties` - The configuration that overrides the default

configuration file that comes bundled with the framework

- `locators/search-home.properties` – A file for storing search homepage web element name and its locator
 - `locators/search-results.properties` – A file for storing search results page web element name and its locator
 - `testng.xml` - A suite of TestNG tests
-

Now, we are familiar with the framework code structure. In the next lesson, we will learn to build, run, and experiment with the downloaded tests.