Valentino Milanov

**Fetch Coding Exercise - SDET**

Contents

[**1 Overview** 2](#_Toc173919264)

[**2 Project Structure** 3](#_Toc173919265)

[**3 Maven Configuration** 3](#_Toc173919266)

[**3.1 Key Properties** 3](#_Toc173919267)

[**4 Logging Configuration** 4](#_Toc173919268)

[**5 Page Objects** 4](#_Toc173919269)

[**5.1 BasePage** 4](#_Toc173919270)

[**5.2 SdetchallengePage** 4](#_Toc173919271)

[**6 Test Base** 5](#_Toc173919272)

[**6.1 TestBase** 5](#_Toc173919273)

[**7 Test Cases** 5](#_Toc173919274)

[**7.1 BalanceScaleTest** 5](#_Toc173919275)

[**8 Running the Tests** 5](#_Toc173919276)

**1 Overview**

This project is a Java Maven-based Selenium automation framework designed to solve a balance scale challenge. The project utilizes Selenium WebDriver for browser automation, JUnit for testing, and SLF4J with Logback for logging. Below, you will find an overview of the project structure, configuration, and usage.

**2 Project Structure**

sdetchallenge/

├── src/

│ ├── main/

│ │ └── java/

│ │ └── com/

│ │ └── fetch/

│ │ └── sdetchallenge/

│ │ └── pageobjects/

│ │ └── base/

│ │ └── pages/

│ ├── test/

│ │ └── java/

│ │ └── com/

│ │ └── fetch/

│ │ └── sdetchallenge/

│ │ └── tests/

│ │ └── base/

│ │ └── BalanceScaleTest.java

│ └── resources/

│ └── driver/

│ └── chromedriver.exe

│ └── logback.xml

└── pom.xml

**3 Maven Configuration**

The project uses Maven for dependency management and build automation. Key dependencies include:

* **Selenium WebDriver**: For interacting with web browsers.
* **JUnit**: For writing and executing tests.
* **SLF4J**: For logging abstraction.
* **Logback**: For logging implementation.

**3.1 Key Properties**

* **Java Version**: 17
* **Selenium Version**: 4.4.0
* **JUnit Version**: 5.10.2
* **Logback Version**: 1.4.6
* **SLF4J Version**: 2.0.7

**Full pom.xml File:** For a detailed view of all dependencies and plugins, refer to the pom.xml file in the repository.

**4 Logging Configuration**

Logging is managed using Logback. The logging configuration is set up to output logs to the console with the following settings:

* **Log Level**: Trace
* **Log Pattern**: Includes timestamp, log level, logger name, and message for the console and file appenders..

**Full logback.xml File:** To view the complete logging configuration, check the logback.xml file in the repository.

**5 Page Objects**

**5.1 BasePage**

* **Purpose**: Provides common functionalities for interacting with web elements, such as waiting for elements to become invisible and handling browser alerts.
* **Key Methods**:
  + waitFoElemetToDissapear(By selector): Waits for an element to become invisible.
  + isAlertPresent(WebDriver driver): Checks if an alert is present.
  + getAlertMessageandAccept(WebDriver driver): Retrieves and accepts an alert message.

**5.2 SdetchallengePage**

* **Purpose**: Represents the SDET Challenge page and provides methods to interact with its elements and perform actions related to the challenge.
* **Key Methods**:
  + clickOnResetButton(): Clicks the reset button on the page.
  + clickOnWeighButton(): Clicks the weigh button and waits for the result to appear.
  + fillBowlsWithBarNumbers(List<Integer> leftBars, List<Integer> rightBars): Fills bowls with specified numbers of gold bars.
  + findFakeBar(): Finds the fake gold bar using measurements.

**6 Test Base**

**6.1 TestBase**

* **Purpose**: Sets up and tears down the WebDriver for test execution.
* **Key Methods**:
  + setUp(): Initializes the ChromeDriver and configures its options.
  + tearDown(): Closes the WebDriver after tests.
  + openSdetchallengePage(): Opens the SDET Challenge page and returns an instance of SdetchallengePage.

**7 Test Cases**

**7.1 BalanceScaleTest**

* **Purpose**: Verifies the functionality of the balance scale challenge.
* **Test Method**:
  + findFakeGoldBar(): Opens the challenge page, finds the fake gold bar, selects it, and verifies the alert message.

**8 Running the Tests**

1. **Set Up**: Ensure that all dependencies are installed and configured as per the pom.xml file.
2. **Execute Tests**: Run the tests using Maven with the command:

mvn test

1. **Review Results**: Check the console output and logs for test results and information.