Fetch - Weighing Exercise

# Overview

Given a balance scale and 9 gold bars of the same size and look. You don’t know the exact weight of each bar, but you know they all weigh the same, except for one fake bar. It weighs less than others. You need to find the fake gold bar by only bars and balance scales. You can only place gold bars on scale plates (bowls) and find which scale weighs more or less.

The test suite is designed to interact with a web application that simulates the gold bar weighing process.

Table of Contents

[Overview 1](#_Toc163646053)

[Tools and Technologies Used 2](#_Toc163646054)

[Test Structure 2](#_Toc163646055)

[Setup and Teardown 2](#_Toc163646056)

[Test Execution 2](#_Toc163646057)

[Recursive Algorithm 2](#_Toc163646058)

[Assertions and Verifications 2](#_Toc163646059)

[Installation 3](#_Toc163646060)

[Prerequisites 3](#_Toc163646061)

[Configuration 4](#_Toc163646062)

[Dependencies 5](#_Toc163646063)

[Project Structure 5](#_Toc163646064)

[Test Execution 6](#_Toc163646065)

# Tools and Technologies Used

* Selenium WebDriver for web automation
* JUnit 5 for test execution and assertions
* FirefoxDriver for browser interaction

# Test Structure

### Setup and Teardown

* + BeforeAll setup method initializes the WebDriver environment and browser settings.
  + BeforeEach setup method launches the browser and navigates to the test website.
  + AfterEach teardown method quits the WebDriver instance after each test execution.

### Test Execution

* + findLeastWeightBar() is the main test method responsible for initiating the fake gold bar finding process.
  + It starts by retrieving the gold bar elements from the web page and their total count.
  + Calls the recursive findGoldBarWeight() method passing the total bars count and gold bar elements.

### Recursive Algorithm

* + findGoldBarWeight() is a recursive method that simulates the weighing process using two balancing scales.
  + It divides the gold bars into three buckets based on the total bar count.
  + Enter the values of gold bars into left and right weighing bowls.
  + Click the weigh button to get the weighing result.
  + Based on the weighing result ("<", ">", or "="), it recursively calls itself with a reduced set of gold bars.
  + Stops when only one gold bar is left and returns the fake gold bar.
  + Print the weighing result and the fake bar result

### Assertions and Verifications

* + Uses JUnit 5 assertions to validate that the input values are correctly entered into the weighing bowls.
  + Verifies that the alert message is displayed and accepts it after each weighing operation.
  + Validates the final output by printing the alert message and the weighing result list.

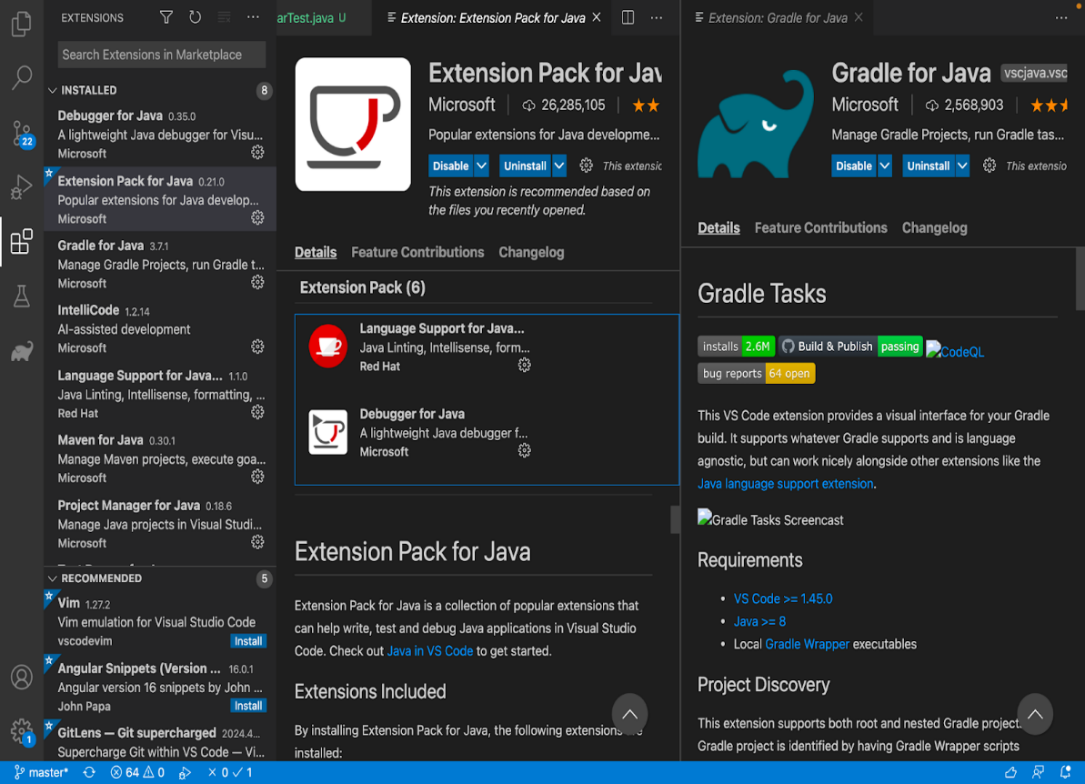
# Installation

### Prerequisites

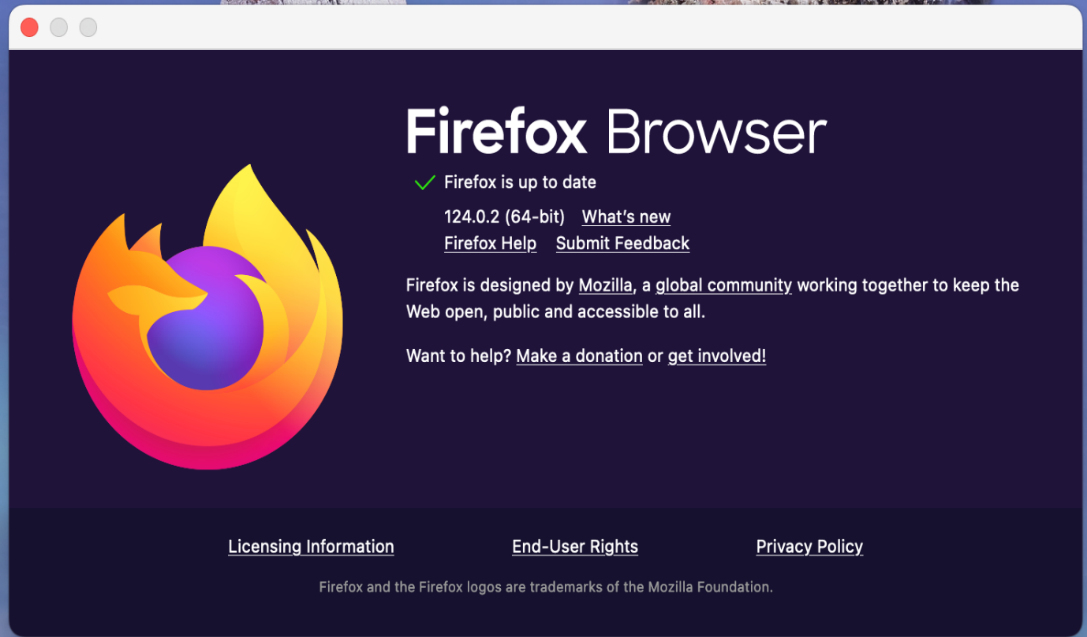
* An **IDE** to open the project. For example, Eclipse, VS Code, IntelliJ
* **JDK 11** - can be installed form the IDE Marketplace
* **Gradle** - can be installed form the IDE Marketplace

For Visual Studio Code:

* Launch Visual Studio Code
* Install Extension Pack for Java and Gradle for Java

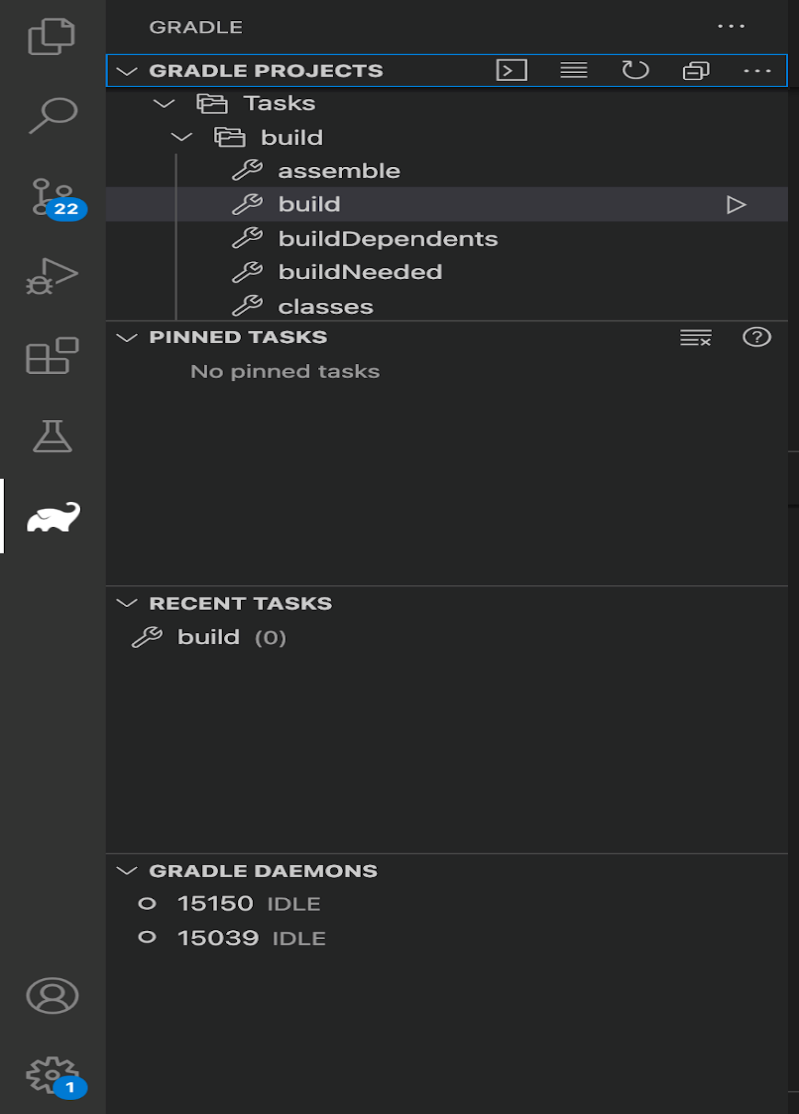


* Verify that latest Firefox browser is installed on the system(v 124.0.2)



# Configuration

* Open project using File > Open
* Navigate to the gradle tab, and click on build > build. This will build the project and fetch required dependencies



* Alternatively, the project can also be built by running the following command in terminal:
  + >> gradle clean build
* To run the test, add your Firefox browser installation path in **setupWebdriverFirefoxDriver** method of the test:

       System.setProperty("webdriver.firefox.bin", "/Applications/Firefox.app/Contents/MacOS/firefox");

# Dependencies

Junit5 and Selenium dependencies are added to the build.gradle file:

testImplementation 'org.junit.jupiter:junit-jupiter:5.7.1'

   testRuntimeOnly 'org.junit.platform:junit-platform-launcher'

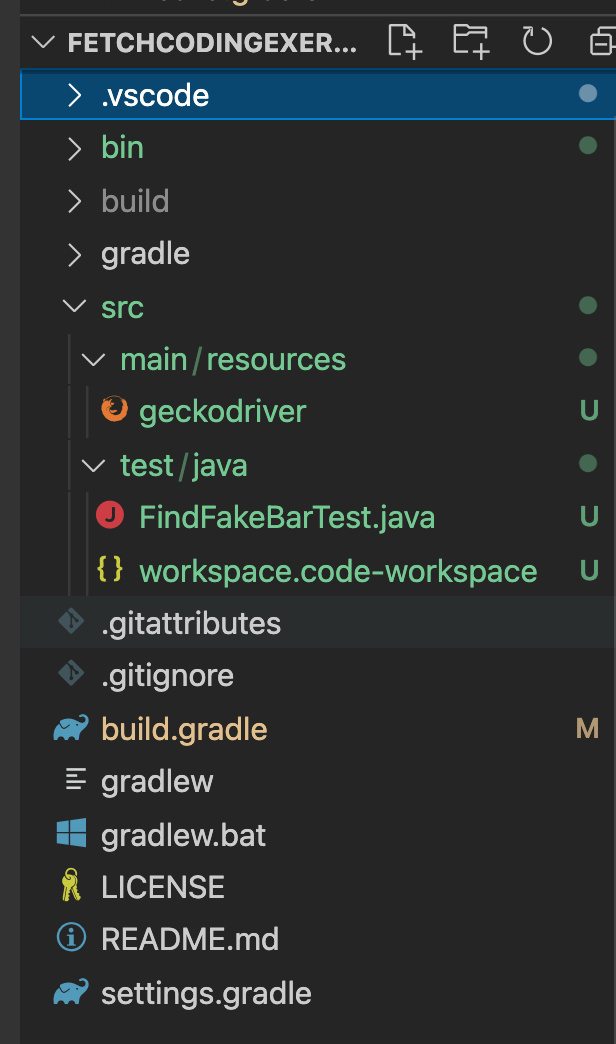
   implementation 'org.seleniumhq.selenium:selenium-java:4.1.0'

# Project Structure

**src/main/resources** - contains geckodriver to run the test on Firefox browser

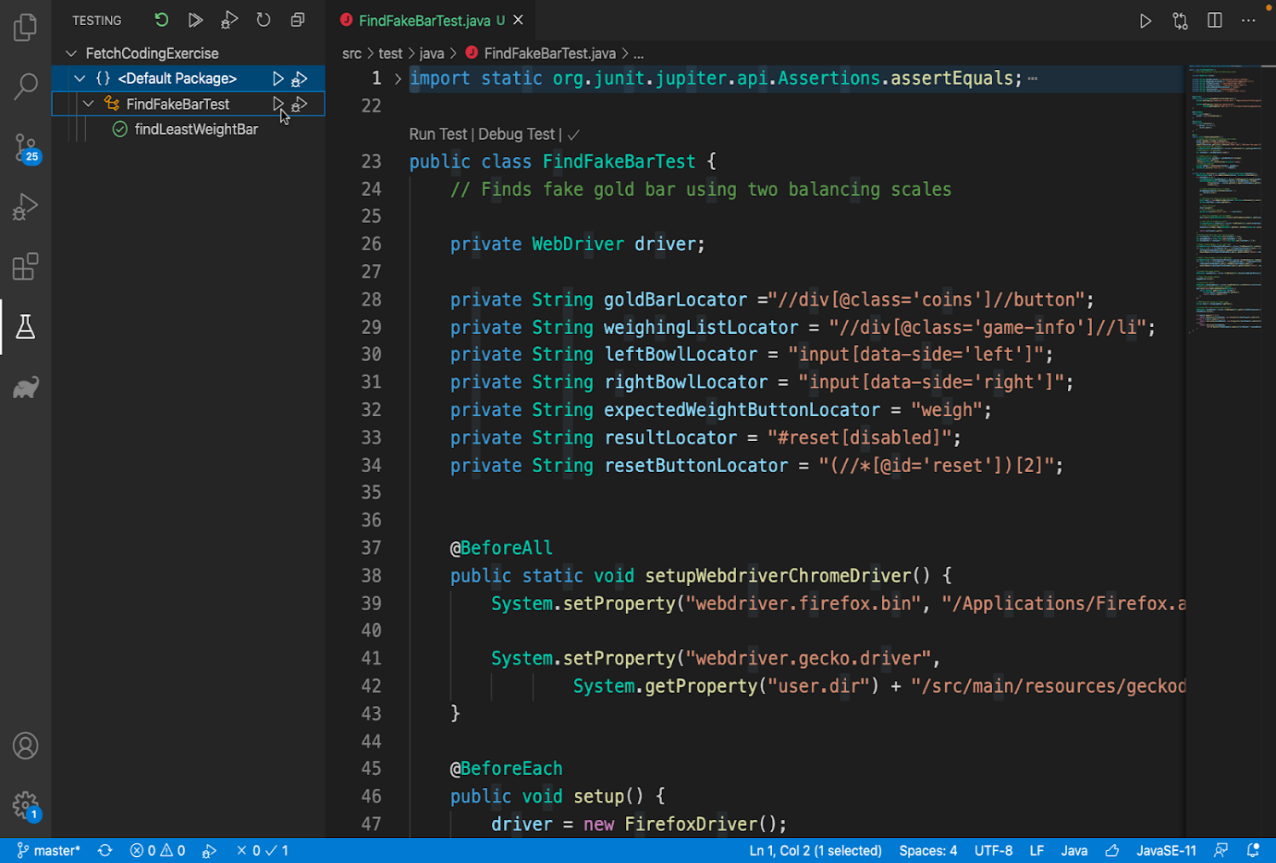
**src/test/java** - contains the test file to find the fake gold bar

**build.gradle** - contains the configuration required to run the project



# Test Execution

Navigate to the test section and click the run button for “FindFakeBarTest” :



The fake gold bar and the weighing results are displayed in the output console :

