4.12 Selenium and JDBC



This section will guide you to:

* Integrate JDBC with selenium

**Development Environment:**

* Eclipse IDE for Enterprise Java Developers Version Oxygen.3a Release (4.7.3a)
* JavaDevelopment Kit Version 8
* Selenium Standalone Server Version 3.141.59
* Structured query language server Version SQL Server 2016 SP2

This guide has mainly three subsections, namely:

4.12.1 Creating a Table in Database

4.12.2 Writing the JDBC connection integrating with selenium

4.12.3 Pushing the code to GitHub repositories

**Step 4.12.1:** Creating a table in Database

* Create a table and enter the data in the table in the Database.

**Step 4.12.2:** Writing the JDBC connection integrating with selenium

* Load the driver class

Syntax: class.forName(“Connection URL”);

com.mysql.cj.jdbc.Driver

URL -

* Create a Connection

Connection con = DriverManager.getConnection(“URL”, “UserName”, “Password”);

* Create a statement

Syntax: Statement stmt = con.createStatement();

* Execute SQL query

Syntax: ResultSet rs= stmt.executeQuery(“sql query”);

* Close the connection

Syntax: Con.close();

The code in Eclipse will look like this:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class TestDatabaseWithSelenium {

    private WebDriver driver;

    @BeforeTest

    public void setup() {

        // Set the path to the ChromeDriver executable

        System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

        // Create a new instance of the ChromeDriver

        driver = new ChromeDriver();

    }

    @Test

    public void testVerifyDB() throws ClassNotFoundException, SQLException {

        // Step 1: Load the driver class

        Class.forName("oracle.jdbc.driver.OracleDriver");

        // Step 2: Create the connection object

        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "your\_username", "your\_password");

        // Step 3: Create the statement object

        Statement stmt = con.createStatement();

        // Step 4: Execute the SQL query

        ResultSet rs = stmt.executeQuery("SELECT \* FROM Products");

        // Step 5: Iterate through the result set and perform web testing

        while (rs.next()) {

            int productId = rs.getInt(1);

            String productName = rs.getString(2);

            String productDescription = rs.getString(3);

            // Perform web testing using Selenium

            driver.get("https://www.seleniumhq.org");

            WebElement searchInput = driver.findElement(By.id("q"));

            searchInput.sendKeys(productName);

            searchInput.submit();

            // Print the database record and web page title

            System.out.println("Product ID: " + productId);

            System.out.println("Product Name: " + productName);

            System.out.println("Product Description: " + productDescription);

            System.out.println("Web Page Title: " + driver.getTitle());

            System.out.println("--------------------------------------------");

        }

        // Step 6: Close the connection object

        con.close();

    }

    @AfterTest

    public void teardown() {

        // Close the browser

        driver.quit();

    }

}

Note: Replace the "**path/to/chromedriver**" with the actual path to the **chromedriver** executable on your system.

**Step 4.12.3:** Pushing the code to GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

Cd <folder path>

* Initialize your repository using the following command:

Git init

* Add all the files to Git repository using following command:

Git add.

* Commit the changes using the following command:

Git commit -m “add the comment”

* Push the files to the folder you initially created using the following command:

Git push -u origin master