

Paso 1: Crear un proyecto Maven

Cree un proyecto Maven en su IDE (por ejemplo, IntelliJ IDEA, Eclipse) o utilice el siguiente comando Maven para crearlo a través de la terminal:

```
mvn archetype:generate -DgroupId=com.example -DartifactId=selenium-parallel-test -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

Paso 2: Agregar dependencias pom.xml

Una vez creado el proyecto Maven, abra el pom.xml archivo y agregue dependencias para TestNG y Selenium WebDriver.

<dependencies>

<!-- TestNG Dependency -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version> <!-- Replace with the latest version -->

<scope>test</scope>

</dependency>

<!-- Selenium WebDriver Dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.0.0</version> <!-- Replace with the latest version -->

</dependency>

<!-- Selenium WebDriver for ChromeDriver -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-chrome-driver</artifactId>

<version>4.0.0</version> <!-- Replace with the latest version -->

</dependency>

<!-- Selenium WebDriver for FirefoxDriver -->

```

<dependency>

  <groupId>org.seleniumhq.selenium</groupId>

  <artifactId>selenium-firefox-driver</artifactId>

  <version>4.0.0</version> <!-- Replace with the latest version -->

</dependency>

</dependencies>

```

Paso 3: Crear la clase de prueba

En este paso, creamos una clase de prueba TestNG llamada LoginTest.java para probar la funcionalidad de inicio de sesión en SauceDemo usando los navegadores Chrome y Firefox en paralelo.

```
package ParallelTesting;
```

```

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Test;

```

```
public class LoginTest {
```

```
    @Test
```

```
    public void testLoginInChrome() {
```

```

        WebDriver driver = new ChromeDriver(); // ChromeDriver setup
        driver.get("https://www.saucedemo.com/");
        driver.findElement(By.id("user-name")).sendKeys("standard_user");
        driver.findElement(By.id("password")).sendKeys("secret_sauce");
        driver.findElement(By.id("login-button")).click();
        driver.quit();
    }

```

```
    @Test
```

```
    public void testLoginInFirefox() {
```

```

    WebDriver driver = new FirefoxDriver(); // FirefoxDriver setup

    driver.get("https://www.saucedemo.com/");

    driver.findElement(By.id("user-name")).sendKeys("standard_user");

    driver.findElement(By.id("password")).sendKeys("secret_sauce");

    driver.findElement(By.id("login-button")).click();

    driver.quit();
}
}

```

Paso 4: Crear el archivo testng.xml

Este archivo define cómo TestNG ejecutará las pruebas en paralelo. En este caso, ejecutamos pruebas para Chrome y Firefox simultáneamente.

```

<?xml version="1.0" encoding="UTF-8"?>

<suite name="Parallel Testing Suite" parallel="methods" thread-count="2">

    <test name="Chrome Test">

        <classes>

            <class name="ParallelTesting.LoginTest" />

        </classes>

    </test>

    <test name="Firefox Test">

        <classes>

            <class name="ParallelTesting.LoginTest" />

        </classes>

    </test>

</suite>

```

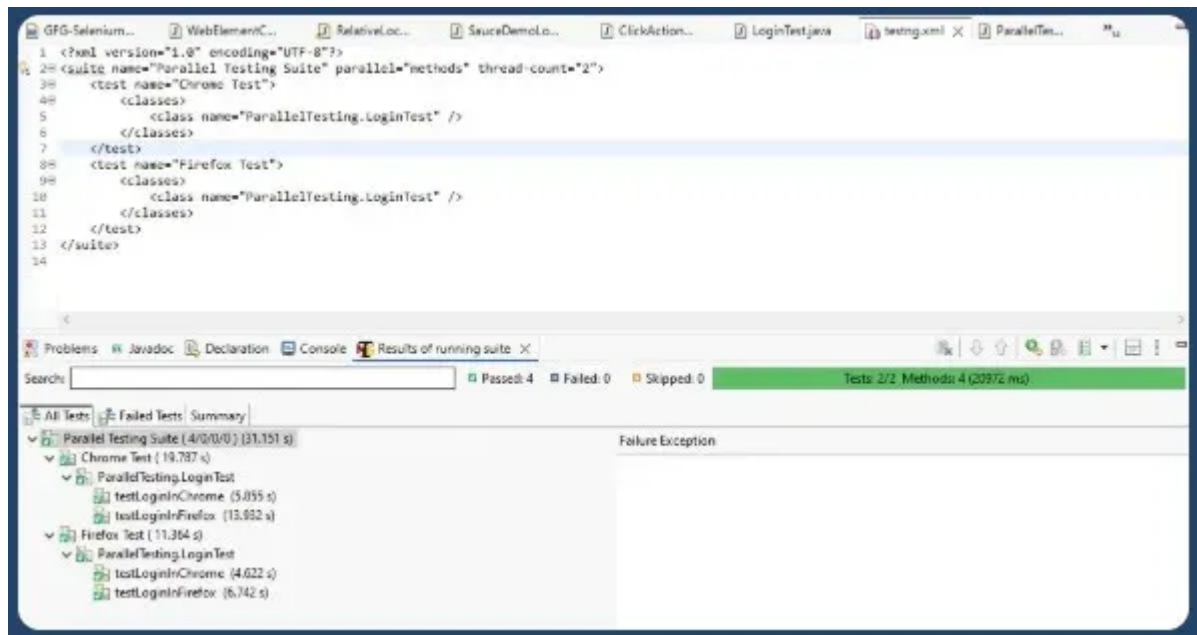
Paso 5: Ejecutar las pruebas

Desde la línea de comandos usando Maven : Abra la terminal y navegue hasta el directorio de su proyecto y ejecute el siguiente comando Maven:

```
mvn test
```

Paso 6: Ver resultados

Una vez ejecutadas las pruebas, puede ver los resultados en la consola de Eclipse o en el informe HTML de TestNG. Encontrará el informe en la test-output carpeta dentro del directorio de su proyecto.



Informe de prueba de pruebas paralelas