

# Integrazione in Spring Boot

L'integrazione di Selenium in un progetto Spring Boot consente di automatizzare i test per le applicazioni web basate su Spring. Ecco un approccio generale su come puoi farlo:

## 1. Aggiungi dipendenze Maven o Gradle:

- Se stai usando Maven, puoi aggiungere la dipendenza Selenium al tuo file `pom.xml` . Assicurati di includere anche la dipendenza di Spring Boot Test.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version> <!-- Versione più recente disponibile -->
    <scope>test</scope>
  </dependency>
</dependencies>
```

- Se stai usando Gradle, puoi aggiungere le dipendenze nel tuo file `build.gradle` .

## 2. Configura il WebDriver:

- Puoi configurare il WebDriver nel tuo codice Spring Boot, ad esempio all'interno di una classe di configurazione o di un componente. Assicurati di scaricare il driver corrispondente per il browser che intendi utilizzare (ad esempio, ChromeDriver per Google Chrome).

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class WebDriverConfig {

    @Bean
    public WebDriver webDriver() {
        System.setProperty("webdriver.chrome.driver", "percorso/del/chromedriver");
        return new ChromeDriver();
    }
}
```

- Puoi anche utilizzare le annotazioni `@Before` e `@After` di JUnit o TestNG per inizializzare e chiudere il WebDriver prima e dopo l'esecuzione dei test.

testing di Spring Boot. Ad esempio, puoi utilizzare JUnit o TestNG come framework di test.

```
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import static org.junit.jupiter.api.Assertions.assertEquals;

@SpringBootTest
public class SeleniumSpringBootTestIntegrationTest {

    @Autowired
    private WebDriver webDriver;

    @Test
    public void testExample() {
        // Tuo codice di test con Selenium
        webDriver.get("https://www.example.com");

        // Esempio di asserzione con JUnit
        assertEquals("Titolo della pagina", webDriver.getTitle());
    }
}
```

○ L'annotazione `@SpringBootTest` consente di caricare l'applicazione Spring

#### 4. **Esegui i test:**

- Puoi eseguire i test come parte del tuo processo di build o manualmente tramite il tuo IDE.

Ricorda di configurare correttamente il percorso del tuo driver nel codice, e assicurati che il WebDriver sia disponibile durante l'esecuzione dei test. Integrare Selenium con Spring Boot può essere utile per testare interazioni complesse su applicazioni web Spring Boot, garantendo che il tuo frontend e il backend funzionino correttamente insieme.