

Ejecución de una sola prueba o método con Maven

Ejecutar una sola prueba o método con Maven es sencillo y se puede lograr mediante comandos de Maven. De forma predeterminada, Maven ejecuta todas las pruebas del proyecto, pero se puede especificar una sola clase de prueba o incluso un solo método de prueba para su ejecución. En este artículo, aprenderemos a ejecutar una sola prueba o método con Maven, con un ejemplo relacionado.

Configuración del proyecto:

- Cree un proyecto Maven usando maven-archetype-quickstart o créelo usando cualquier IDE. También aquí usamos STS.
- Agregue dependencias de JUnit 5 a pom.xml.
- Escriba casos de prueba en una clase de prueba.
- Ejecute una sola clase de prueba con mvn -Dtest=ClassName test.
- Ejecute un solo método de prueba con mvn -Dtest=ClassName#methodName test

Implementación para ejecutar una sola prueba o método con Maven

A continuación se muestran los pasos de implementación para ejecutar una sola prueba o método con Maven.

Paso 1: Crear un proyecto Maven

Primero, crea un proyecto Maven simple con tu herramienta favorita. Aquí, usamos STS IDE y usamos las dependencias necesarias.

Dependencias:

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

<project xmlns="https://maven.apache.org/POM/4.0.0"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="https://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>3.2.6</version>

    <relativePath/> <!-- lookup parent from repository -->

  </parent>

  <groupId>com.app</groupId>

  <artifactId>mavencommends</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>

<name>mavencommends</name>

<description>Spring Reactive</description>

<properties>
  <java.version>1.8</java.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>

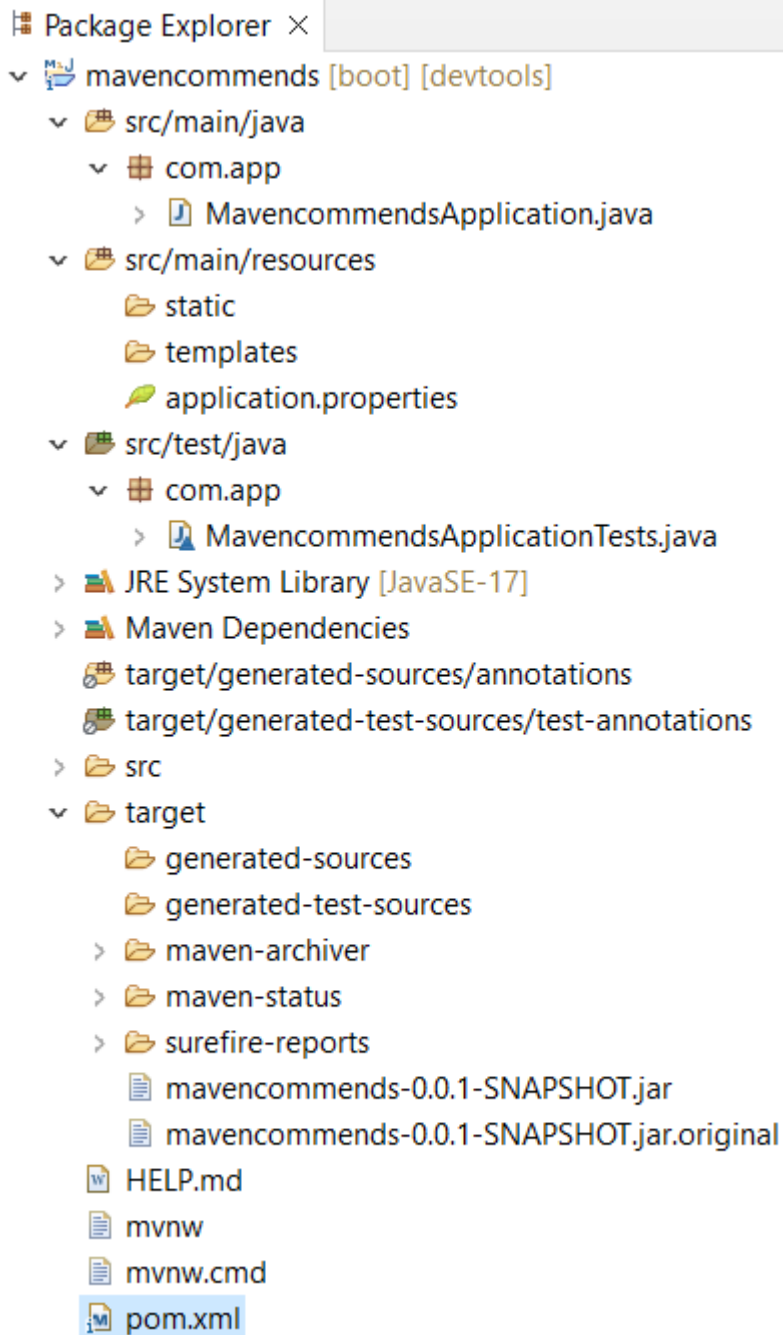
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

</plugins>

</build>

</project>

Estructura del proyecto:



Paso 2: Escribe la clase de prueba

Ahora, dirígete a la carpeta **src/test/java**. Allí encontrarás una clase de prueba predeterminada en ese paquete. Escribe la lógica de la prueba en esa clase. Esta clase contiene dos métodos de prueba: `testMethod1` y `testMethod2`.

MavencommendsApplicationTests.java:

```
package com.app;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class MavencommendsApplicationTests {

    @Test
    void contextLoads() {
    }

    @Test
    public void testMethod1() {
        assertTrue(true);
    }

    @Test
    public void testMethod2() {
        assertTrue(true);
    }

}
```

Paso 3: Configurar dependencias

Asegúrese de que su **pom.xml** esté configurado para usar JUnit 5. Si no lo tiene, le proporcionamos esa dependencia, copie en su pom.xml y luego actualice el proyecto.

<dependencies>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-api</artifactId>

<version>5.7.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>5.7.1</version>

<scope>test</scope>

</dependency>

</dependencies>

Paso 4: Ejecutar toda la clase de prueba

El siguiente comando ejecutará todos los métodos de prueba de la clase MavencommendsApplicationTests. Aquí, ejecutamos todos los métodos de prueba de la clase de prueba.

mvn -Dtest=MavencommendsApplicationTests prueba

Producción:

```
D:\Geeks\mavencommends>mvn -Dtest=MavencommendsApplicationTests test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.app:mavencommends >-----
[INFO] Building mavencommends 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ mavencommends
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO] Copying 0 resource from src\main\resources to target\classes
[INFO]
```

Éxito de compilación de Maven:

```

[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.876 s -- in
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.440 s
[INFO] Finished at: 2024-06-02T13:16:49+05:30
[INFO] -----

```

Paso 5: Ejecutar un método de prueba único

Ahora ejecutamos un solo método en la clase de prueba. Para ejecutar un método de prueba específico, use el siguiente comando. Aquí ejecutamos un método de prueba específico de la clase de prueba usando el siguiente comando.

`mvn -Dtest=MavencommendsApplicationTests#testMethod1 prueba`

Producción:

```

D:\Geeks\mavencommends>mvn -Dtest=MavencommendsApplicationTests#testMethod1 test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.app:mavencommends >-----
[INFO] Building mavencommends 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ mavencommends
[INFO] Copying 1 resource from src\main\resources to target\classes
[INFO] Copying 0 resource from src\main\resources to target\classes

```

Compilación exitosa del método de prueba único de Maven:

```

[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.392 s -- in
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.325 s
[INFO] Finished at: 2024-06-02T13:20:34+05:30
[INFO] -----

```