Maven Surefire Plugin

A continuación se presenta una explicación del **Plugin Maven Surefire**, que es un componente clave en los proyectos basados en Maven para ejecutar pruebas unitarias durante el proceso de construcción. Esta explicación cubre su propósito, configuración, uso y opciones de personalización.

¿Qué es el Plugin Maven Surefire?

El **Plugin Maven Surefire** es un plugin en Apache Maven diseñado para ejecutar pruebas unitarias durante el ciclo de vida de la construcción. Se integra sin problemas con la fase `test` de Maven y se activa automáticamente cuando se ejecutan comandos como `mvn test`, `mvn package` o `mvn install`. El plugin admite marcos de pruebas populares como JUnit (versiones 3, 4 y 5) y TestNG, y genera informes de pruebas para ayudar a los desarrolladores a evaluar los resultados de las pruebas.

Características Clave

- Ejecuta pruebas en un proceso JVM separado para aislar.
- Soporta múltiples marcos de pruebas (JUnit, TestNG, etc.).
- Genera informes de pruebas en formatos como XML y texto plano.
- Ofrece flexibilidad para omitir pruebas, ejecutar pruebas específicas o personalizar la ejecución.

Configuración Básica en `pom.xml`

El Plugin Surefire está incluido por defecto en el ciclo de vida de construcción de Maven, por lo que no necesita configurarlo para un uso básico. Sin embargo, puede declararlo explícitamente en su archivo `pom.xml` para especificar una versión o personalizar su comportamiento.

Configuración Mínima

Si no agrega ninguna configuración, Maven usa el plugin con la configuración predeterminada: - Las pruebas se encuentran en `src/test/java`. - Los archivos de prueba siguen patrones de nombres como `**/*Test.java`, `**/Test*.java` o `**/*Tests.java`.

Declaración Explicita

Para personalizar el plugin o asegurarse de una versión específica, agréguelo a la sección `<build><plugins>` de su `pom.xml`. Aquí hay un ejemplo:

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.0.0-M5</version> <!-- Use la versión más reciente -->
        </plugin>
    </plugins>
</build>
```

Ejecutar Pruebas con Surefire

El plugin está vinculado a la fase `test` del ciclo de vida de Maven. Aquí se explica cómo usarlo:

Ejecutar Todas las Pruebas

Para ejecutar todas las pruebas unitarias, ejecute:

```
mvn test
```

Ejecutar Pruebas en una Construcción Mayor

Las pruebas se ejecutan automáticamente cuando se ejecutan comandos que incluyen la fase `test`, como:

```
mvn package
mvn install
```

Omitir Pruebas

Puede omitir la ejecución de pruebas usando banderas de línea de comandos: - **Omitir la ejecución de pruebas**: `-DskipTests mvn package -DskipTests` - **Omitir la compilación y ejecución de pruebas**: `-Dmaven.test.skip=true mvn package -Dmaven.test.skip=true`

Personalizar el Plugin Surefire

Puede ajustar el comportamiento del plugin agregando una sección `<configuration>` en el `pom.xml`. Aquí hay algunas personalizaciones comunes:

Incluir o Excluir Pruebas Específicas

Especifique qué pruebas ejecutar o omitir usando patrones:

```
<configuration>
    <includes>
        <include>**/My*Test.java</include>
    </includes>
    <excludes>
        <exclude>**/SlowTest.java</exclude>
    </excludes>
</configuration>
```

Ejecutar Pruebas en Paralelo

Acelere la ejecución ejecutando pruebas de manera concurrente:

```
<configuration>
    <parallel>methods</parallel>
    <threadCount>2</threadCount>
</configuration>
```

Nota: Asegúrese de que sus pruebas sean seguras para hilos antes de habilitar esto.

Pasar Propiedades del Sistema

Establezca propiedades para la JVM de prueba:

```
<configuration>
    <systemPropertyVariables>
        <propertyName>propertyValue</propertyName>
    </systemPropertyVariables>
</configuration>
```

Generar Informes

Por defecto, los informes se guardan en `target/surefire-reports`. Para un informe HTML, use el `maven-surefire-report-plugin`:

```
<reporting>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-report-plugin</artifactId>
            <version>3.0.0-M5</version>
        </plugin>
    </plugins>
</reporting>
```

Ejecute `mvn surefire-report:report` para generar el informe HTML.

Manejo de Fallos en las Pruebas

Fallar la Construcción en Caso de Fallo en la Prueba

Por defecto, una prueba fallida causa que la construcción falle. Para ignorar los fallos y continuar:

```
mvn test -Dmaven.test.failure.ignore=true
```

Volver a Ejecutar Pruebas Fallidas

Maneje pruebas inestables volviendo a intentar los fallos:

```
<configuration>
    <rerunFailingTestsCount>2</rerunFailingTestsCount>
</configuration>
```

Esto vuelve a ejecutar las pruebas fallidas hasta 2 veces.

Usar Surefire con Marcos de Pruebas

El plugin admite varios marcos de pruebas con una configuración mínima:

JUnit 4

No se necesita ninguna configuración adicional; Surefire detecta automáticamente las pruebas de JUnit 4.

JUnit 5

Agregue la dependencia de JUnit 5:

```
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-engine</artifactId>
        <version>5.8.1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Use la versión 2.22.0 o posterior de Surefire para un soporte completo.

TestNG

Agregue la dependencia de TestNG:

```
<dependencies>
    <dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>7.4.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Surefire ejecutará automáticamente las pruebas de TestNG.

Notas Clave

- **Ubicación de la Prueba**: Las pruebas deben estar en `src/test/java`, con recursos en `src/test/resources`.
- **Convenciones de Nombres**: Los patrones predeterminados incluyen `**/*Test.java`, `**/Test*.java`, etc., pero estos pueden personalizarse.

- **Versión del Plugin**: Use una versión reciente (por ejemplo, 3.0.0-M5) para las últimas características y soporte de marco.

Resumen

El **Plugin Maven Surefire** es esencial para ejecutar pruebas unitarias en proyectos Maven. Para usarlo: 1. Confíe en su comportamiento predeterminado o declárelo explícitamente en `pom.xml` para personalizar. 2. Ejecute pruebas con `mvn test` o como parte de construcciones más grandes como `mvn package`. 3. Personalícelo con opciones como patrones de inclusión/exclusión, ejecución paralela o propiedades del sistema. 4. Maneje fallos omitiendo pruebas, ignorándolos o volviendo a ejecutar los inestables. 5. Asegure la compatibilidad con su marco de pruebas (JUnit, TestNG) agregando las dependencias correctas.

¡Este plugin proporciona una manera robusta y flexible de gestionar las pruebas unitarias en sus proyectos Maven!