

# Maven Surefire Plugin

Hier ist eine Erklärung des **Maven Surefire Plugins**, das eine Schlüsselkomponente in Maven-basierten Projekten ist, um Einzeltests während des Build-Prozesses auszuführen. Diese Erklärung deckt seinen Zweck, die Konfiguration, die Verwendung und die Anpassungsoptionen ab.

---

## Was ist das Maven Surefire Plugin?

Das **Maven Surefire Plugin** ist ein Plugin in Apache Maven, das entwickelt wurde, um Einzeltests während des Build-Lebenszyklus auszuführen. Es integriert sich nahtlos in die `test`-Phase von Maven und wird automatisch ausgelöst, wenn Sie Befehle wie `mvn test`, `mvn package` oder `mvn install` ausführen. Das Plugin unterstützt beliebte Test-Frameworks wie JUnit (Versionen 3, 4 und 5) und TestNG und erstellt Testberichte, um Entwicklern bei der Bewertung der Testergebnisse zu helfen.

## Wichtige Funktionen

- Führt Tests in einem separaten JVM-Prozess zur Isolation aus.
  - Unterstützt mehrere Test-Frameworks (JUnit, TestNG, usw.).
  - Erstellt Testberichte in Formaten wie XML und einfacher Text.
  - Bietet Flexibilität, um Tests zu überspringen, spezifische Tests auszuführen oder die Ausführung anzupassen.
- 

## Grundlegende Einrichtung in `pom.xml`

Das Surefire Plugin ist standardmäßig in den Build-Lebenszyklus von Maven integriert, sodass Sie es für die grundlegende Verwendung nicht konfigurieren müssen. Sie können es jedoch explizit in Ihrer `pom.xml`-Datei deklarieren, um eine Version anzugeben oder sein Verhalten anzupassen.

## Minimale Konfiguration

Wenn Sie keine Konfiguration hinzufügen, verwendet Maven das Plugin mit den Standard-Einstellungen: - Tests befinden sich in `src/test/java`. - Testdateien folgen Mustern wie `**/*Test.java`, `**/Test*.java` oder `**/*Tests.java`.

## Explizite Deklaration

Um das Plugin anzupassen oder eine bestimmte Version sicherzustellen, fügen Sie es dem Abschnitt `<build><plugins>` Ihrer `pom.xml` hinzu. Hier ist ein Beispiel:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M5</version> <!-- Verwenden Sie die neueste Version -->
    </plugin>
  </plugins>
</build>
```

---

## Tests mit Surefire ausführen

Das Plugin ist an die `test`-Phase des Maven-Lebenszyklus gebunden. Hier ist, wie Sie es verwenden:

### Alle Tests ausführen

Um alle Einzeltests auszuführen, führen Sie aus:

```
mvn test
```

### Tests in einem größeren Build ausführen

Tests werden automatisch ausgeführt, wenn Sie Befehle ausführen, die die `test`-Phase enthalten, wie z.B.:

```
mvn package
```

```
mvn install
```

### Tests überspringen

Sie können die Testausführung mit Befehlszeilen-Flags überspringen: - **Tests überspringen**: `-DskipTests`

`mvn package -DskipTests` - **Testkompilierung und -ausführung überspringen**: `-Dmaven.test.skip=true`

```
mvn package -Dmaven.test.skip=true
```

## Anpassen des Surefire Plugins

Sie können das Verhalten des Plugins anpassen, indem Sie einen `<configuration>`-Abschnitt in der `pom.xml` hinzufügen. Hier sind einige gängige Anpassungen:

### Bestimmte Tests ein- oder ausschließen

Geben Sie an, welche Tests ausgeführt oder übersprungen werden sollen, indem Sie Muster verwenden:

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

### Tests parallel ausführen

Beschleunigen Sie die Ausführung, indem Sie Tests parallel ausführen:

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

*Hinweis:* Stellen Sie sicher, dass Ihre Tests thread-sicher sind, bevor Sie diese Funktion aktivieren.

### Systemeigenschaften übergeben

Legen Sie Eigenschaften für die Test-JVM fest:

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

## Berichte erstellen

Standardmäßig werden Berichte in `target/surefire-reports` gespeichert. Für einen HTML-Bericht verwenden Sie das `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>
```

Führen Sie `mvn surefire-report:report` aus, um den HTML-Bericht zu erstellen.

---

## Umgang mit Testfehlern

### Build bei Testfehler abbrechen

Standardmäßig führt ein fehlgeschlagener Test dazu, dass der Build fehlschlägt. Um Fehler zu ignorieren und fortzufahren:

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

### Fehlgeschlagene Tests erneut ausführen

Behandeln Sie fluktuierende Tests, indem Sie Fehler wiederholen:

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

Dies führt fehlgeschlagene Tests bis zu 2 Mal erneut aus.

---

## Verwendung von Surefire mit Test-Frameworks

Das Plugin unterstützt verschiedene Test-Frameworks mit minimaler Einrichtung:

## JUnit 4

Es ist keine zusätzliche Konfiguration erforderlich; Surefire erkennt JUnit 4-Tests automatisch.

## JUnit 5

Fügen Sie die JUnit 5-Abhängigkeit hinzu:

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

Verwenden Sie Surefire Version 2.22.0 oder höher für volle Unterstützung.

## TestNG

Fügen Sie die TestNG-Abhängigkeit hinzu:

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

Surefire führt TestNG-Tests automatisch aus.

---

## Wichtige Hinweise

- **Teststandort:** Tests sollten sich in `src/test/java` befinden, mit Ressourcen in `src/test/resources`.
- **Benennungskonventionen:** Standardmuster umfassen `**/*Test.java`, `**/Test*.java`, usw., diese können jedoch angepasst werden.

- **Plugin-Version:** Verwenden Sie eine aktuelle Version (z.B. 3.0.0-M5) für die neuesten Funktionen und Framework-Unterstützung.
- 

## Zusammenfassung

Das **Maven Surefire Plugin** ist unerlässlich für das Ausführen von Einzeltests in Maven-Projekten. Um es zu verwenden: 1. Verlassen Sie sich auf sein Standardverhalten oder deklarieren Sie es explizit in `pom.xml` für Anpassungen. 2. Führen Sie Tests mit `mvn test` oder als Teil größerer Builds wie `mvn package` aus. 3. Passen Sie es mit Optionen wie Ein-/Ausschlussmustern, paralleler Ausführung oder Systemeigenschaften an. 4. Behandeln Sie Fehler, indem Sie Tests überspringen, sie ignorieren oder fluktuierende Tests erneut ausführen. 5. Stellen Sie die Kompatibilität mit Ihrem Test-Framework (JUnit, TestNG) sicher, indem Sie die richtigen Abhängigkeiten hinzufügen.

Dieses Plugin bietet eine robuste und flexible Möglichkeit, das Einzeltesten in Ihren Maven-Projekten zu verwalten!