

Testen

- [Testen](#)
 - [Unit Tests](#)
 - [Integration Test](#)
 - [GUI Test](#)
 - [Wichtige Java Software zum Testen](#)
 - [Sonar Integration](#)

Testen



Diese Seite ist eine Link-Sammlung. Die Seite über Nuclos Unit Tests finden Sie [hier](#).

Unit Tests

- Gute Artikel: [Artikelreihe bei javacodegeeks, /JUnit, TestNG, and Mockito/](#)
- Kurzeinführung JUnit: [junit.pdf](#)

Integration Test

GUI Test

- [Testing Swing GUIs \(aus Java Extreme Programming\)](#)
- [Structuring Swing Code Testable](#)
- [Code/Utilities für Swing Unit Tests](#)
- [Java Swingtests \(mit FEST\)](#)
- [Test First on Swing](#)

Testframeworks

Es gibt zwar eine Anzahl von Testframeworks (u.a. jfcunit, swingunit, fest, apache jelly, jemmy), sie zeichnen sich jedoch *alle* dadurch aus, dass sie *nic* *ht* mehr weiterentwickelt werden. Nur jemmy3 lebt noch 'ein bisschen'.

- [Migration Jelly 2 -> 3](#)
- [WinTester](#) ist ein freies Tool von Google (inklusive Eclipse Plugin), dass GUI Tests ermöglicht
[Swing Tutorial](#) und [API zum Widgetfinden](#)

Wichtige Java Software zum Testen

Test Support in Spring

- [Kapitel über Tests in der Referenz](#)
- [@Configurable in Tests](#)
- [Javadoc Spring 4.0](#)
- [Unit Test mittels ReflectionTestUtils oder SpringJUnit4ClassRunner](#)
- [Unit Test mittels AbstractTestNGSpringContextTests](#)
- [Unit Test mittels Mockito und @Autowired](#)
- [Unit Test mittels @Autowired: hier](#) und [hier](#)
- [Mockito und Spring](#) und [Springockito](#)

TestNG

- [TestNG Home Page](#)
- [Eclipse Plugin Update Site: http://beust.com/eclipse](#) und [die Bedienung des Plugins](#)
- [Manual](#)
- [Javadoc](#)
- Gute Artikel: [TestNG Tutorial](#), [Mkyong Tutorials](#)
- Refcards: [Quick Guide](#)
- Weitere Artikellinks: [Misc](#)

Mockito

- [Mockito Home Page](#)
- Gute Artikel: [Vogella über Mockito](#)
- Refcards: [Refcardz](#)
- [Javadoc](#) (insbesondere die tolle Übersicht in [org.mockito.Mockito](#))
- [Beschreibung des Spy Patterns](#)
- [Einfache Beispiele für den Einsatz von Mockito](#)
- [Beispiele für das Mocken von Stored Procedure Aufrufen und Resultsets](#)

jMock

- [jMock Home Page](#)

Powermock

- [Powermock Home Page](#)

Weitere Libraries

- [Testing Frameworks](#), [Best Testing Frameworks](#)
- [XMLUnit](#) und dessen [User Guide](#)

Sonar Integration

[Wie SonarCube die Unit Test Coverage ermittelt](#)

sonar-maven-plugin

Damit diese Kombination funktioniert, benötigt man ein Profile mit den benötigten Angaben zum Sonar Server und zur Datenbank. Normalerweise reicht es, die folgenden Zeilen in der `~/.m2/settings.xml` Datei hinzu zu fügen bzw. die Datei entsprechend anzupassen.

```
<settings>
[... ]
<profiles>
[... ]
  <profile>
    <id>sonar</id>
    <properties>
      <sonar.host.url>http://****:9000</sonar.host.url>
      <sonar.login>****</sonar.login>
      <sonar.password>****</sonar.password>
      <sonar.jdbc.url>jdbc:postgresql://****:5432/sonar</sonar.jdbc.url>
      <sonar.jdbc.username>****</sonar.jdbc.username>
      <sonar.jdbc.password>****</sonar.jdbc.password>
    </properties>
  </profile>
</profiles>

<!-- activeProfiles
| List of profiles that are active for all builds.
|
-->
<activeProfiles>
  <activeProfile>sonar</activeProfile>
</activeProfiles>

</settings>
```

Details auch im [internen Wiki](#).

Wichtige Artikel und Hilfen zur Konfiguration

- [Creating Code Coverage Reports](#)
- [Separating Code Coverage With Maven, Sonar and Jacoco](#)
- [Tracking Integration Test Coverage with Maven and SonarQube](#)
- [Sonar Unit and Integration Test Coverage with Maven](#)
- [Integration Testing with Maven](#)
- [Separating Integration and Unit Tests with Maven, Sonar, Failsafe, and JaCoCo](#)
- [Sonar: Code Coverage by Integration/Unit Tests](#)

Weitere Artikel zur Konfiguration

- [Configure jacoco agent](#)
- [sonar-maven-plugin database configuration](#)
- [Mvn lifecycles](#)
- [Dokumentation jacoco-maven-plugin](#)
- [Dokumentation maven-failsafe-plugin](#)
- [Dokumentation JaCoCo agent](#)
- [Dokumentation Sonar Plugin für Jenkins \(auch hier\)](#)
- [Code Coverage Tools für Sonar](#)
- [Sonar Code Coverage by Unit Tests for Java Project](#)
- [Sonar Code Coverage by Integration Tests for Java Project](#)
- [Sonar JaCoCo Plugin](#)
- [Sonar für Java](#)
- [Sonar FAQ](#)
- [Sonar Default Links](#)
- [Sonar: excluding files](#)
- [Sonar Jacoco excludes](#)

Spezielle Probleme mit Mvn (sub) Modulen

- [Test Ergebnis Aggregation in Mvn](#)
- [Multi-module Konfiguration für Mvn/JaCoCo/SonarQube](#)
- [TestRunner and Mvn](#)