JUnit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development.

JUnit is linked as a JAR at compile-time; the framework resides under package JUnit.framework

Installation

To install Junit, you have to download the following JARs and add them to your test classpath:

- `junit.jar`
- `hamcrest-core.jar`

Configuration

Add a dependency to junit:junit in test scope

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.12</version>
  <scope>test</scope>
</dependency>
```

You need to have a JDK installed and a text editor. It is always suggested to use a build tool for building your software and running the tests.

JUnit Usage and Idioms

- Assertions - your bread and butter for unit testing
- Test Runners - how tests should be executed
- Aggregating tests in Suites - how to combine multiple related tests into a test suite
- Test Execution Order - specifying what order to run unit tests
- Exception Testing - how to specify expected exceptions in unit tests
- Matchers and assertThat - how to use Hamcrest matchers and more descriptive assertions

- Ignoring Tests - how to disable test methods or classes
- Timeout for Tests - how to specify maximum execution times for tests
- Parameterized Tests - writing tests that can be executed multiple times with different parameter values
- Assumptions with Assume - similar to assertions, but without making tests fail
- Rules - stop extending abstract test classes and start writing test rules
- Theories - write tests that are more like scientific experiments using randomly generated data
- Test Fixtures - specify set up and clean up methods on a per-method and per-class basis
- Categories - group your tests together for easier test filtering
- Use with Maven - how to use JUnit with your Maven build
- Use with Gradle - how to use JUnit with your Gradle build
- Multithreaded code and Concurrency - basic ideas behind testing concurrent code
- Java contract test helpers
- Continuous Testing

Third-party extensions using Junit

- Custom Runners
- github.com/trajano/commons-testing for UtilityClassTestUtil per #646
- System Rules – A collection of JUnit rules for testing code that uses java.lang.System.
- JUnit Toolbox – Provides runners for parallel testing, a PoolingWait class to ease asynchronous testing, and a WildcardPatternSuite which allow you to specify wildcard patterns instead of explicitly listing all classes when you create a suite class.
- junit-dataprovider – A TestNG like dataprovider (see here) runner for JUnit.
- junit-quickcheck – QuickCheck-style parameter suppliers for JUnit theories. Uses junit.contrib's version of the theories machinery, which respects generics on theory parameters.
- JGiven - Behavior-Driven Development (BDD) for JUnit