JUnit 4 Project Documentation

Fork me on GitHub

JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.

```
@Test
public void newArrayListsHaveNoElements() {
    assertThat(new ArrayList().size(), is(0));
}

@Test
public void sizeReturnsNumberOfElements() {
    List instance = new ArrayList();
    instance.add(new Object());
    instance.add(new Object());
```

Annotations

Start by marking your tests with `@Test` .

Let's take a tour »

Welcome

- Download and install
- Getting started
- Release Notes
  - 4.12
  - 4.11
  - 4.10
  - 4.9.1
  - 4.9
- Maintainer Documentation
- I want to help!
- Latest JUnit Questions on StackOverflow
- JavaDocs
- Frequently asked questions
- Wiki
- License

Usage and Idioms

- Assertions
- Test Runners
- Aggregating tests in Suites
- Test Execution Order
- Exception Testing
- Matchers and assertThat
- Ignoring Tests
- Timeout for Tests
- Parameterized Tests
- Assumptions with Assume
- Rules
- Theories
- Test Fixtures
- Categories
- Use with Maven
- Multithreaded code and Concurrency
- Java contract test helpers
- Continuous Testing

Third-party extensions

- Custom Runners
- net.trajano.commons: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-quickcheck - QuickCheck-style parameter suppliers for JUnit theories. Uses junit.contrib's version of the theories machinery, which respects generics on theory parameters.

Built by: maven

built on DEV@cloud