

# UNIT TESTING AND JUNIT



1

**Davide Falessi**

# WHAT IS JUNIT?

- Open source Java testing framework used to write and run repeatable automated unit tests
- JUnit is open source ([junit.org](http://junit.org))

# WHAT IS JUNIT?

- JUnit features include:
  - Assertions for testing expected results
  - Test features for sharing common test data
  - Test suites for easily organizing and running tests
  - Graphical and textual test runners
- JUnit is widely used in industry
- JUnit can be used as stand alone Java programs (from the command line) or within an IDE such as Eclipse

# JUNIT TEST

- JUnit can be used to test ...
  - ... an entire object
  - ... part of an object – a method or some interacting methods
  - ... interaction between several objects
- It is primarily for unit and integration testing, not system testing

# WRITING TESTS FOR JUNIT

- Need to use the methods of the `junit.framework.assert` class
  - javadoc gives a complete description of its capabilities
- Each test method checks a condition (**assertion**) and reports to the test runner whether the test failed or succeeded

# ASSERTION TYPES

- *static void assertEquals (boolean expected, boolean actual)*  
Asserts that two booleans are equal
- *static void assertEquals (byte expected, byte actual)*  
Asserts that two bytes are equal
- *static void assertEquals (char expected, char actual)*  
Asserts that two chars are equal
- *static void assertEquals (double expected, double actual, double delta)*  
Asserts that two doubles are equal, within a delta
- *static void assertEquals (float expected, float actual, float delta)*  
Asserts that two floats are equal, within a delta
- *static void assertEquals (int expected, int actual)*  
Asserts that two ints are equal
- For a complete list, see
  - <http://junit.sourceforge.net/javadoc/org/junit/Assert.html>

# SIMPLE EXAMPLE

```
package logic;  
  
public class BasicCalculations {  
  
    public int perfectSum (int a, int b){  
        int result;  
        result = a + b;  
        return (result);  
    }  
  
    public double perfectDivision (int a, int b){  
        double result;  
        result = a / b;  
        return (result);  
    }  
}
```

# SIMPLE EXAMPLE

```
package test;

import logic.BasicCalculations;
import org.junit.Test;
import static org.junit.Assert.*;

public class TestDivision {

    @Test
    public void testDivision() {

        BasicCalculations b = new BasicCalculations();
        double output = b.perfectDivision(11, 10);
        assertEquals((double) 2, output, 1);

    }

}
```

- 1) Right Click, Build Path, Configure, Add Library, Junit
- 2) Right Click, Run as, Junit Test



# CREATING AND RUNNING A TEST SUITE

- Create a test division class.
- Go to test, ->New, Other, Java, Junit, JunitTestSuite, select and run as Junit Test.

# BEST PRACTICES

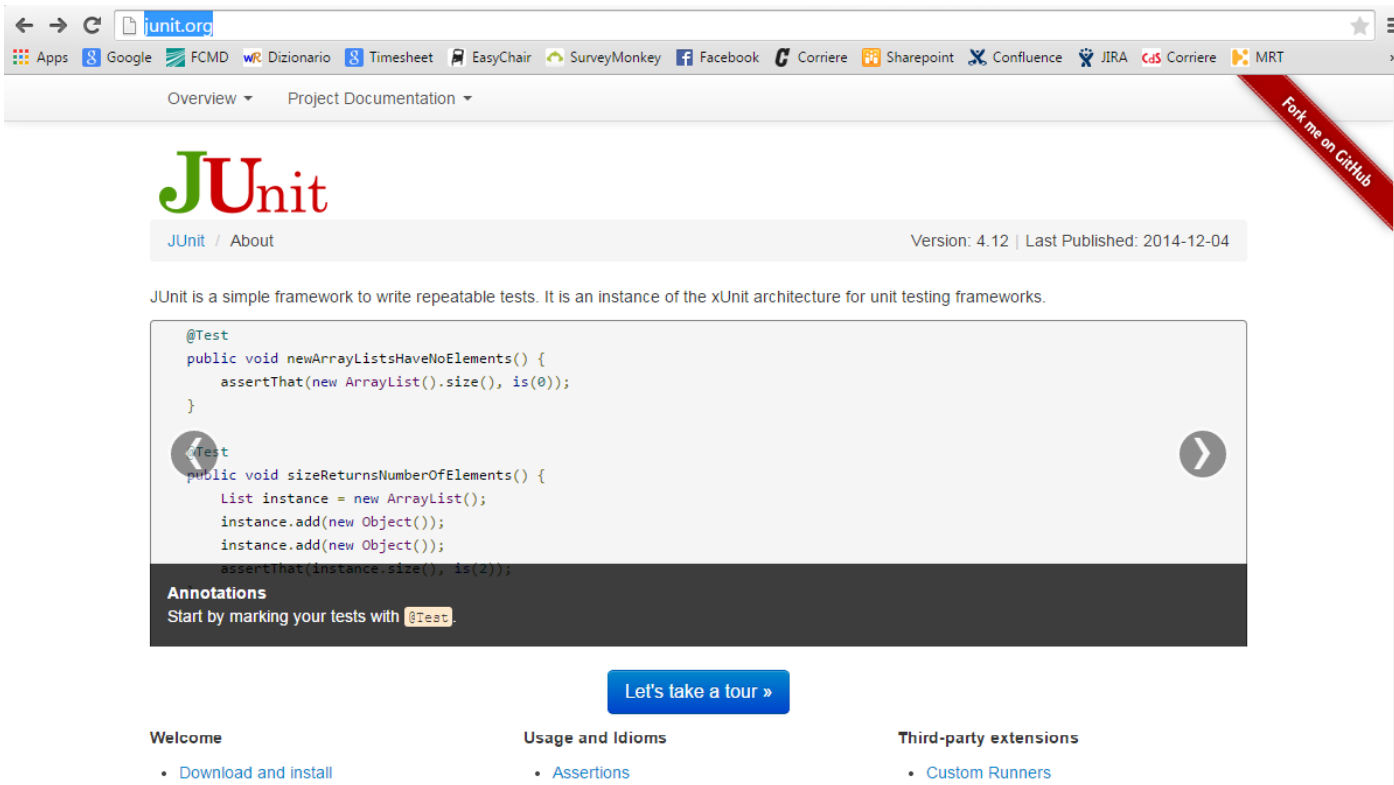
- Write tests in a separate package (e.g., test)
- Write at least one test class for each logic class.
  - Do not write one test class for more than one logic class (it does not apply to type of testing different than unit testing)
- Use multiple test methods for the same method under testing.
- Use one “assert” per each test method.
- Name the test class as TestX where X is the class you are testing (e.g., TestCalculations).
- Name the test method testYZ where Y is the method being tested and Z is a specific case. (e.g., testDivisionByZeros for testing the method Division in case the divisor is zero).

# How to install Junit (if required)

# INSTALLING JUNIT IN ECLIPSE

Installation steps:

1) Download Junit form JUnit.org



The screenshot shows the JUnit.org website in a web browser. The browser's address bar displays "junit.org". The website features the JUnit logo, navigation links for "Overview" and "Project Documentation", and a red banner that says "Fork me on GitHub". Below the logo, it states "JUnit / About" and "Version: 4.12 | Last Published: 2014-12-04". A description of JUnit as a simple framework for writing repeatable tests is provided. Two code snippets are shown: one for a test that checks if a new ArrayList has no elements, and another for a test that checks if the size of an ArrayList is 2. Below the code, there are "Annotations" explaining that tests should be marked with the @Test annotation. At the bottom, there is a blue button that says "Let's take a tour »" and three sections: "Welcome" with a link to "Download and install", "Usage and Idioms" with a link to "Assertions", and "Third-party extensions" with a link to "Custom Runners".

JUnit

JUnit / About Version: 4.12 | Last Published: 2014-12-04

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());
    assertThat(instance.size(), is(2));
}
```

**Annotations**  
Start by marking your tests with `@Test`

[Let's take a tour »](#)

**Welcome**

- [Download and install](#)

**Usage and Idioms**

- [Assertions](#)

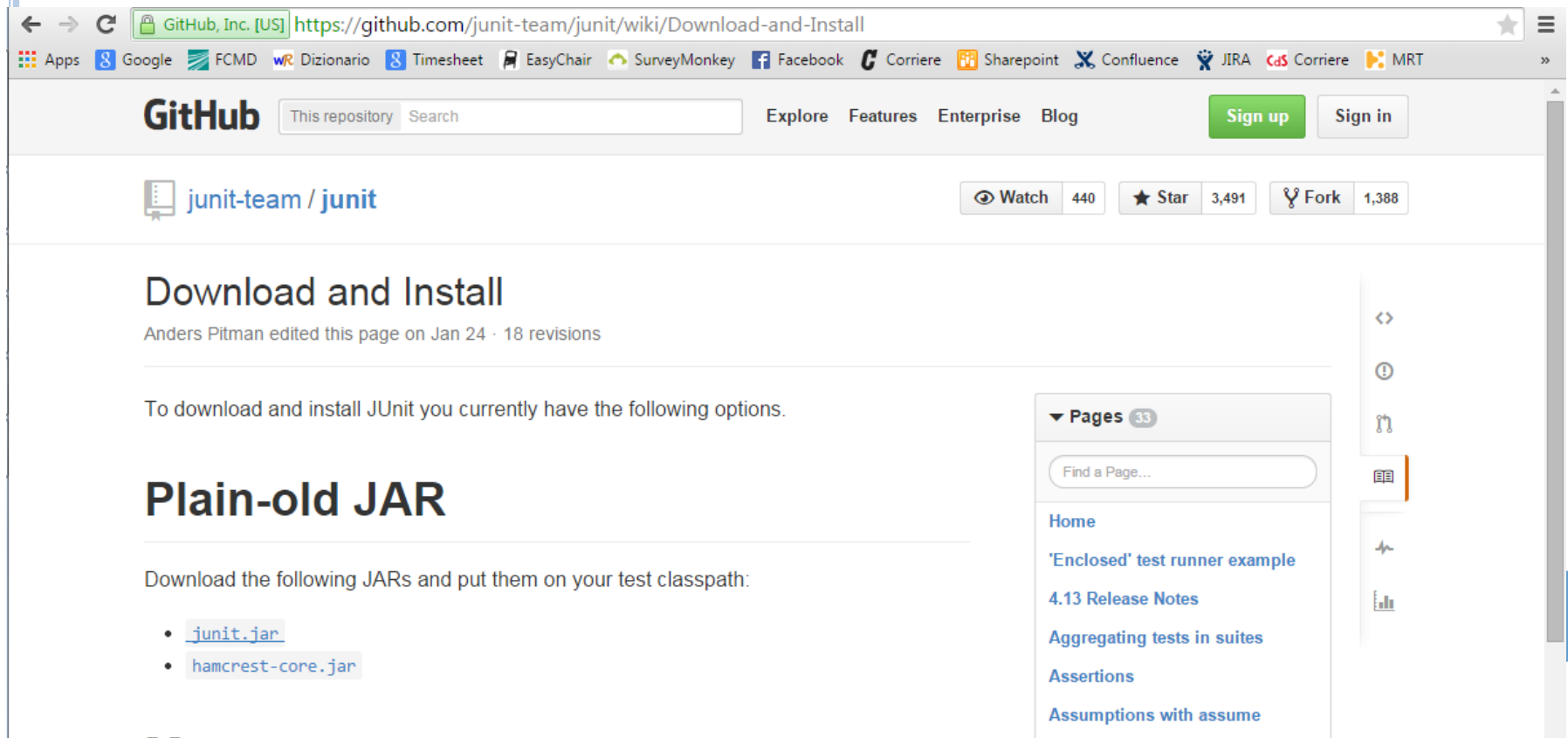
**Third-party extensions**

- [Custom Runners](#)

# INSTALLING JUNIT IN ECLIPSE

Installation steps:

2 ) Select Junit.jar



The screenshot shows the GitHub repository page for `junit-team/junit`. The browser address bar displays the URL `https://github.com/junit-team/junit/wiki/Download-and-Install`. The GitHub header includes the logo, a search bar, and navigation links like 'Explore', 'Features', 'Enterprise', and 'Blog'. The repository name 'junit-team / junit' is shown with statistics: 440 watches, 3,491 stars, and 1,388 forks. The main heading is 'Download and Install', with a note that Anders Pitman edited the page on Jan 24 with 18 revisions. The text states: 'To download and install JUnit you currently have the following options.' Below this, the section 'Plain-old JAR' is highlighted, with instructions to 'Download the following JARs and put them on your test classpath:'. A list of JARs is provided: `junit.jar` and `hamcrest-core.jar`. A right-hand sidebar contains a 'Pages' section with 33 items, including 'Home', 'Enclosed test runner example', '4.13 Release Notes', 'Aggregating tests in suites', 'Assertions', and 'Assumptions with assume'.

← → ↻ GitHub, Inc. [US] `https://github.com/junit-team/junit/wiki/Download-and-Install` ★ ☰

Apps Google FCMD Dizionario Timesheet EasyChair SurveyMonkey Facebook Corriere Sharepoint Confluence JIRA CdS Corriere MRT »

**GitHub** This repository Search Explore Features Enterprise Blog Sign up Sign in

junit-team / junit Watch 440 Star 3,491 Fork 1,388

## Download and Install

Anders Pitman edited this page on Jan 24 · 18 revisions

To download and install JUnit you currently have the following options.

### Plain-old JAR

Download the following JARs and put them on your test classpath:

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

--

Pages 33

Find a Page...

Home

'Enclosed' test runner example

4.13 Release Notes

Aggregating tests in suites

Assertions

Assumptions with assume

# INSTALLING JUNIT IN ECLIPSE

Installation steps:

3 ) Save the file “junit-4.12.jar” in “C:\JUnit”

4) Set the environment variable JUNIT\_HOME to C:\JUnit

4) Open Eclipse

5) Right click on a (new) project, Properties, Java Build Path, Libraries, Add External Jar, select “junit-4.12.jar” from “C:\JUnit”

