

# Unit Testing with junit

Add this maven library to your project: `junit:junit:4.12`

Then you can create classes with public methods marked with `@Test`. Here is a complete example:

```
import org.junit.Assert;
import org.junit.Test;

public class StringConcatTest {
    @Test
    public void testConcatStrings()
    {
        Assert.assertEquals(
            "3,5,8,9,10",
            "3," + 5 + "," + 8 + "," + 9 + "," + 10
        );
    }
}
```

When you do `assertEquals`, the first parameter is the **expected** value, and the second parameter is the **actual** value. If they are equal, the test passes. If not, the test reports a failure and shows both values.

Here's a further example of testing our `SaltedHashedPassword` class. Instead of `assertEquals`, this one primarily relies on `assertTrue` and `assertFalse`.

```
package auth;

import org.junit.Assert;
import org.junit.Test;

import java.security.NoSuchAlgorithmException;

public class SaltedHashedPasswordTest {

    @Test
    public void testEmptyPassword() throws NoSuchAlgorithmException {
        SaltedHashedPassword pwd =
            SaltedHashedPassword.generate("");
        Assert.assertFalse(pwd.check("secret"));
        Assert.assertTrue(pwd.check(""));
    }
}
```

```
@Test
public void testActualPassword() throws NoSuchAlgorithmException {
    SaltedHashedPassword pwd =
        SaltedHashedPassword.generate("secr3t!");
    Assert.assertFalse(pwd.check("secret!"));
    Assert.assertTrue(pwd.check("secr3t!"));
}

@Test
public void testStringRoundTrip() throws NoSuchAlgorithmException {
    SaltedHashedPassword pwd1 =
        SaltedHashedPassword.generate("foo");
    String s = pwd1.toString();
    SaltedHashedPassword pwd2 = new SaltedHashedPassword(s);
    Assert.assertEquals(pwd1.toString(), pwd2.toString());
    Assert.assertTrue(pwd2.check("foo"));
}
}
```

To run all the tests in your project, go to **Run » Edit configurations...**, select the plus sign in the upper left, and then **JUnit**. Set the **Name** to “All Tests”, change **Test kind** to “All in directory”, and then select your src directory. Press **OK** and running **All Tests** should show results from any methods marked `@Test` throughout your project.

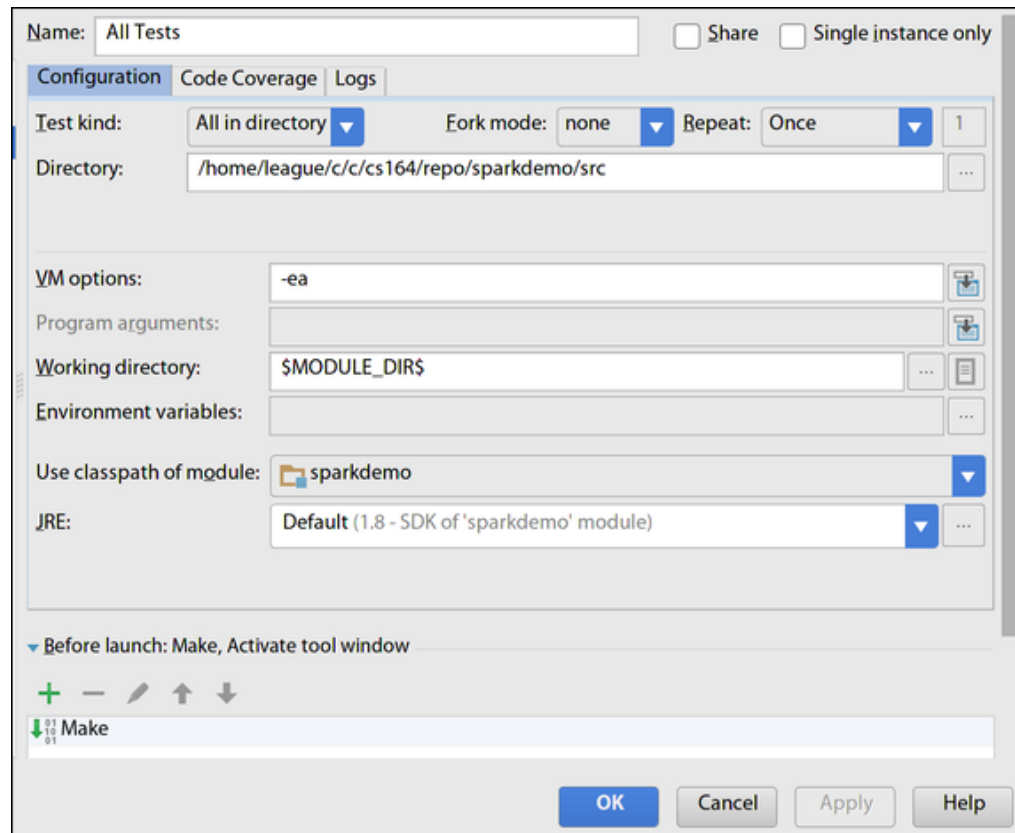


Figure 1: Run all tests in your project directory