



Junit *On IntelliJ IDEA*





IntelliJ IDEA

- ④ Every aspect of IntelliJ IDEA is specifically designed to maximize developer productivity.
- ④ Together, powerful static code analysis and ergonomic design make development not only productive but also an enjoyable experience.



- 内置JUnit4&5
 - 内置测试覆盖度工具
 - 测试类生成插件
 - 导出测试报告及测试覆盖率报告
-



Junit on IntelliJ

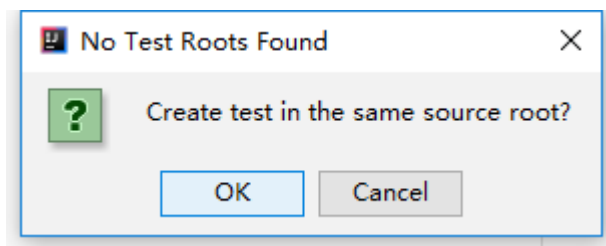
The screenshot shows the IntelliJ IDEA IDE interface. The top toolbar includes icons for running, debugging, and other IDE functions. The left sidebar displays the project structure for 'Hello', with the 'src' folder selected. The main editor window shows the 'Hello.java' file with the following code:

```
1 public class Hello {  
2     public int abs(int n) {  
3         return n >= 0 ? n : (-n);  
4     }  
5 }  
6
```



创建Test类

- 右键被测类.java文件，选择“Go To” - “Test”，打开测试类生成界面。弹出提示

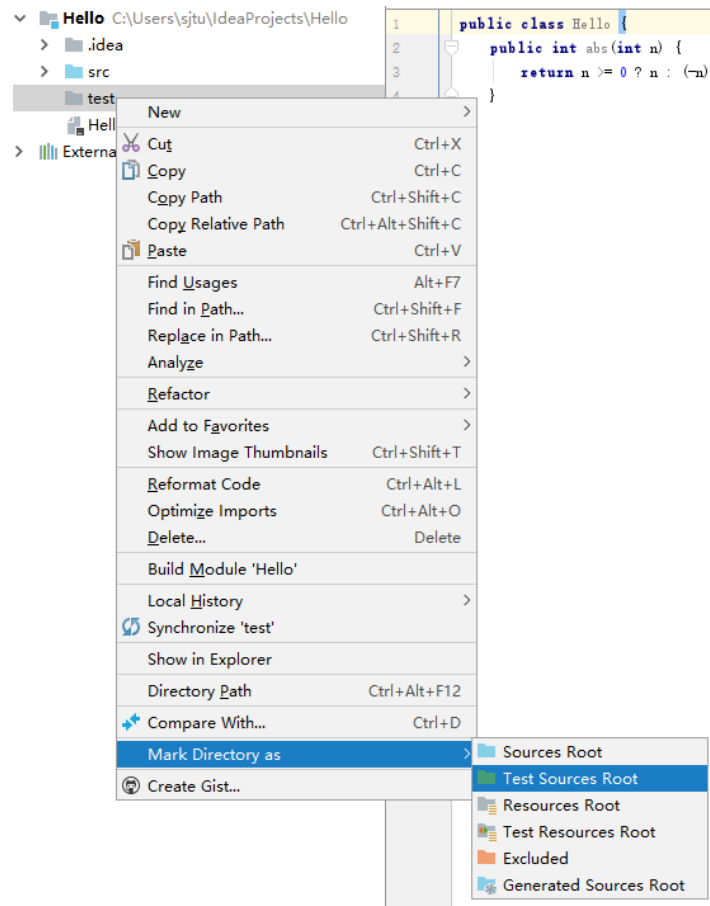


- 为了保持良好的代码结构，一般将测试类与主程序放在不同的根目录下



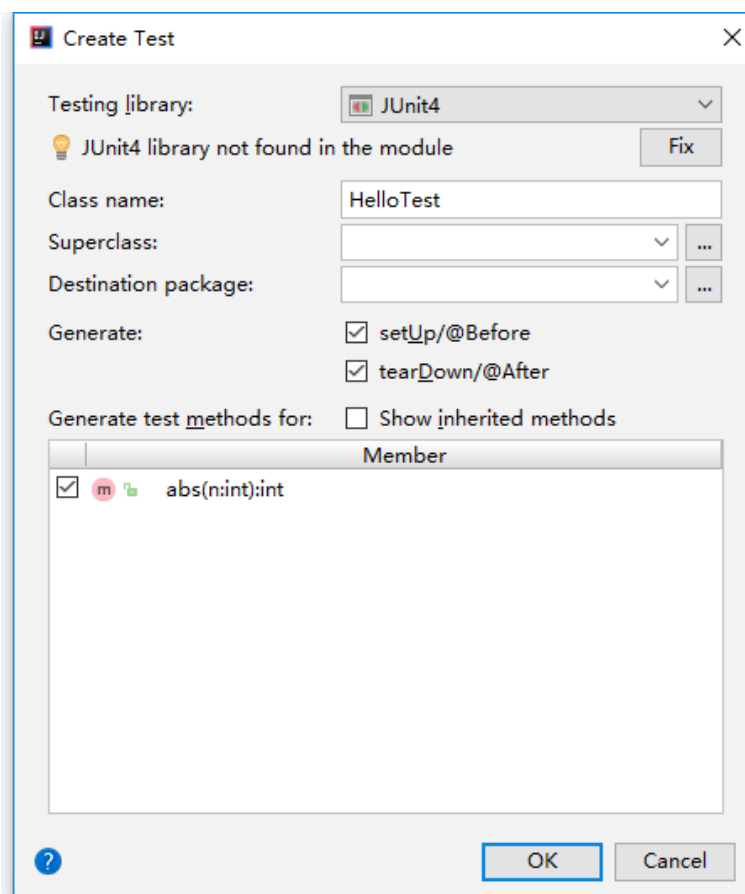
创建Test类

创建test文件夹，并将其设置为Test的根目录



创建Test类

进入创建测试类面板，注意“JUnit4 library not found in the module”的提示，点击“Fix”为工程加入JUnit包





创建Test类

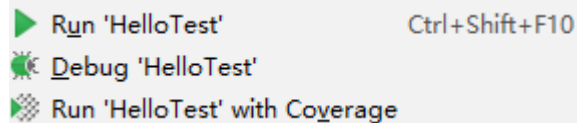
```
1 import org.junit.After;
2 import org.junit.Before;
3 import org.junit.Test;
4
5 import static org.junit.Assert.*;
6
7 public class HelloTest {
8
9     @Before
10    public void setUp() throws Exception {
11    }
12
13    @After
14    public void tearDown() throws Exception {
15    }
16
17    @Test
18    public void abs() {
19    }
20 }
```




开始测试



右键test类或在test类文件中点击  图标，进行测试

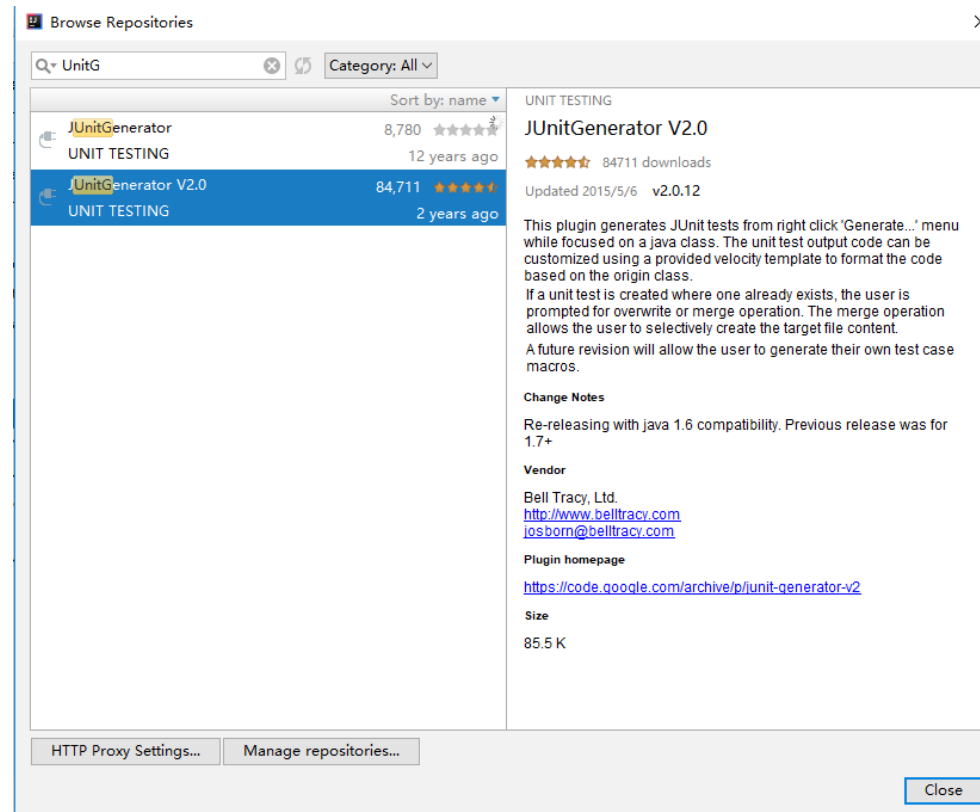


Coverage HelloTest			
↑ 100% classes, 100% lines covered in 'all classes in scope'			
Element	Class, %	Method, %	Line, %
com			
java			
javafx			
javax			
jdk			
META-INF			
netscape			
oracle			
org			
sun			
Hello	100% (1/1)	100% (1/1)	100% (2/2)



使用插件生成测试代码

- 打开File->Settings->Plugins->Browse Repositories，搜索JUnitGenerator并安装（可能要挂代理）。



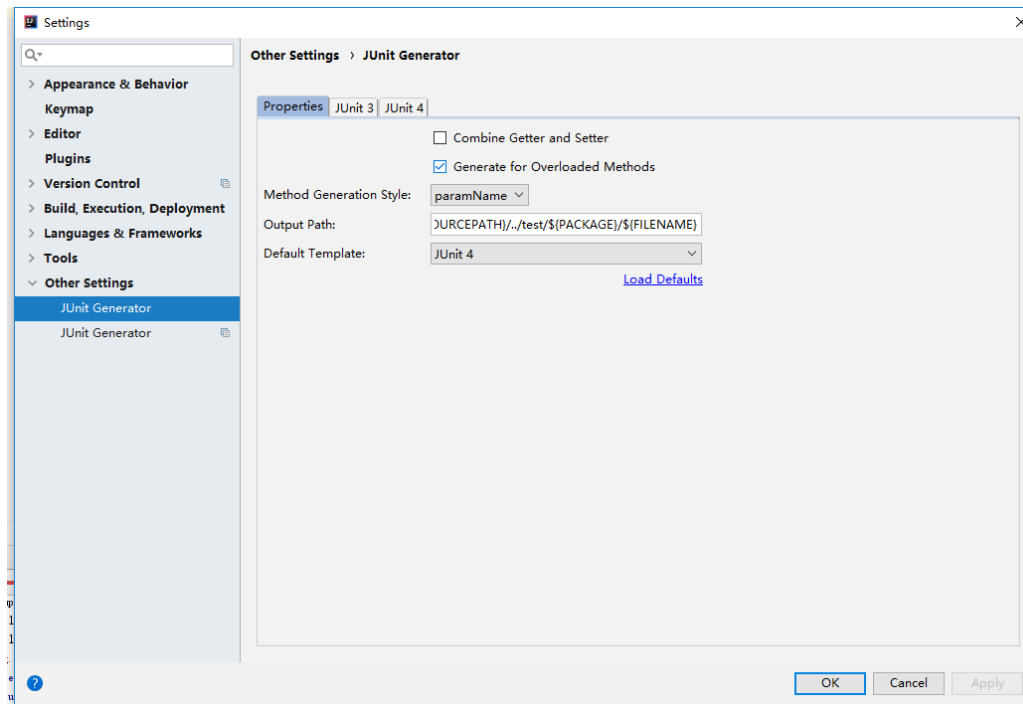


配置插件

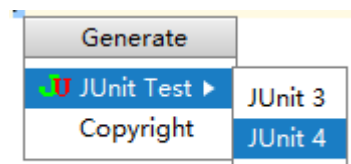


在File->Settings->Other Settings里找到JUnit Generator，修改里面的设置以适应工程：

- Output Path：自动生成测试代码的输出路径
- JUnit 4：生成测试文件的模板



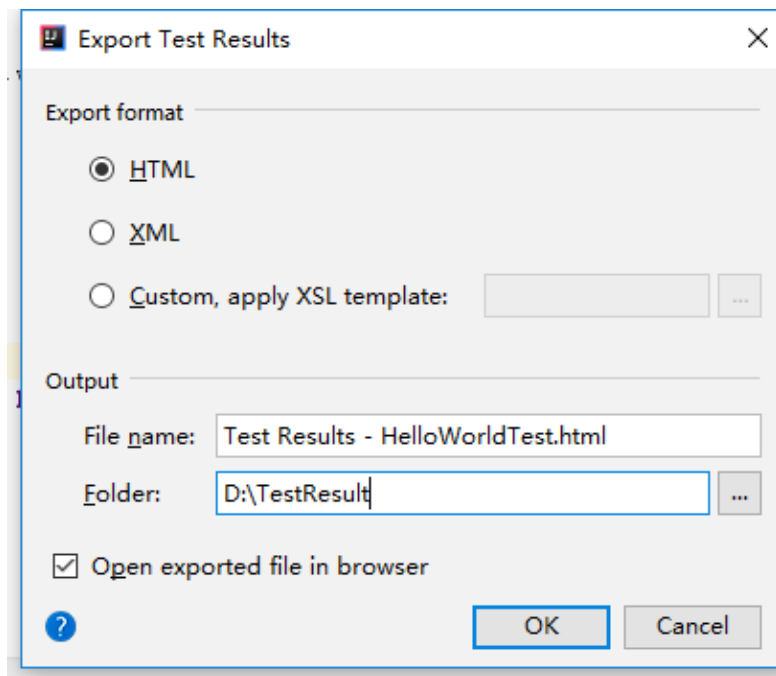
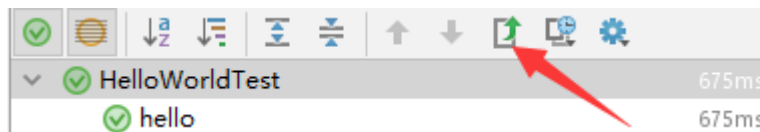
在被测代码文件里右键选择Generate->JUnit Test->JUnit 3/4。或者使用快捷键Alt+Insert。



该插件会为测试类下的每个方法生成测试函数，可以使用IntelliJ自带的方式选择只为部分函数生成测试函数

导出测试报告

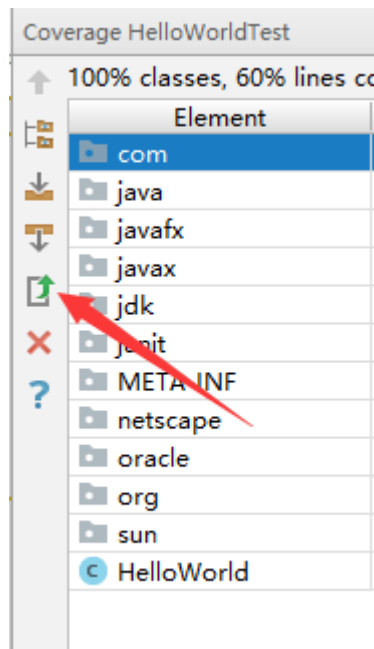
在测试控制窗口界面点击Export Test Results，选择导出格式和路径。





导出测试覆盖率报告

在测试覆盖率窗口点击Generate Coverage Report，选择导出路径。



[all classes] [<empty package name>]

Coverage Summary for Class: HelloWorld (<empty package name>)

Class	Method, %	Line, %
HelloWorld	66.7% (2/ 3)	60% (3/ 5)
HelloWorld\$\$EnhancerByMockitoWithCGLIB\$\$78b37baf		
HelloWorld\$\$EnhancerByMockitoWithCGLIB\$\$78b37baf\$\$FastClassByMockitoWithCGLIB\$\$3553d01e		
HelloWorld\$\$FastClassByMockitoWithCGLIB\$\$1a2fa200		
total	66.7% (2/ 3)	60% (3/ 5)

```
1 public class HelloWorld {
2     public int count = 0;
3     public String hello() {
4         return this.getStr();
5     }
6
7     public String getStr() {
8         count++;
9         return "Hello world!";
10    }
11 }
```