**开发环境：**

JDK1.6, Maven 3.0.4, GWT2.1.0

**准备工作：**

1) 安装 JDK 1.6

安装完成后 验证： 开始>运行>cmd，输入 java -v ，确认设置正确

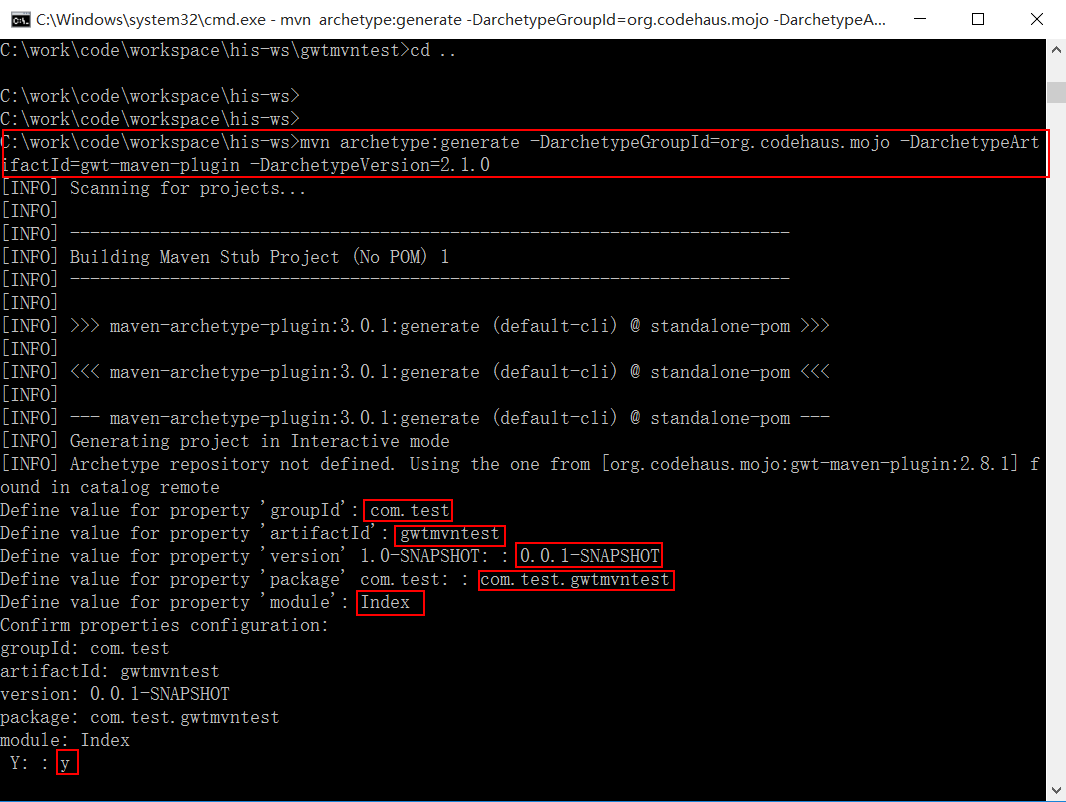
2) 安装 Maven 3.0.4

安装完成后 验证： 开始>运行>cmd，输入 mvn -v ，确认设置正确

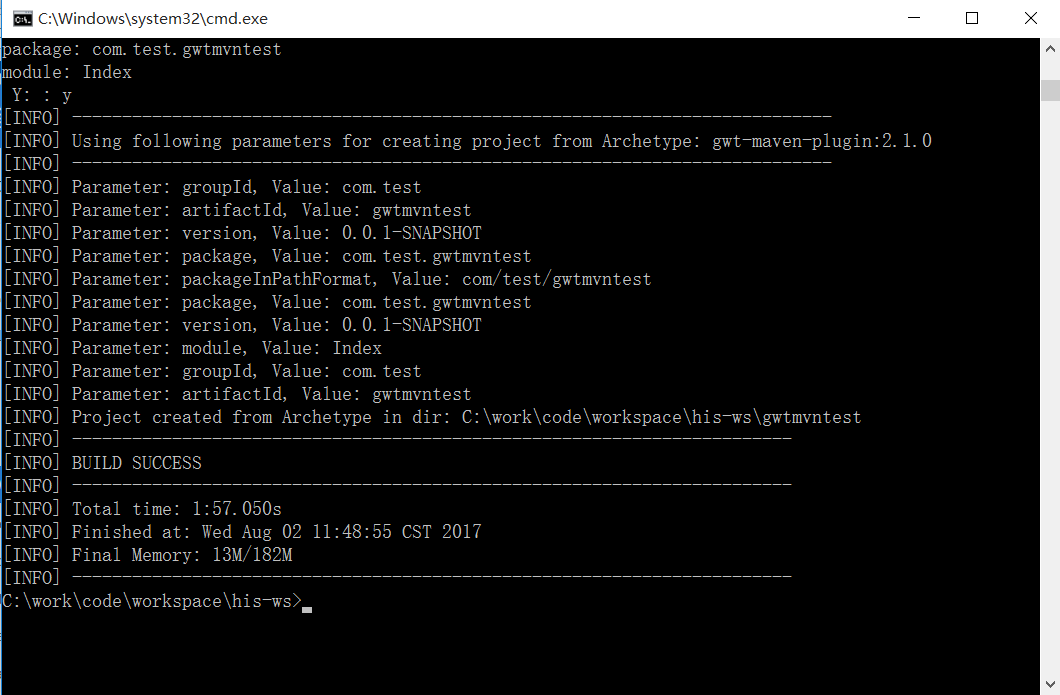
**1、生成 GWT Maven 项目**

开始>运行>cmd，运行如下命令:

mvn archetype:generate -DarchetypeGroupId=org.codehaus.mojo -DarchetypeArtifactId=gwt-maven-plugin -DarchetypeVersion=2.1.0



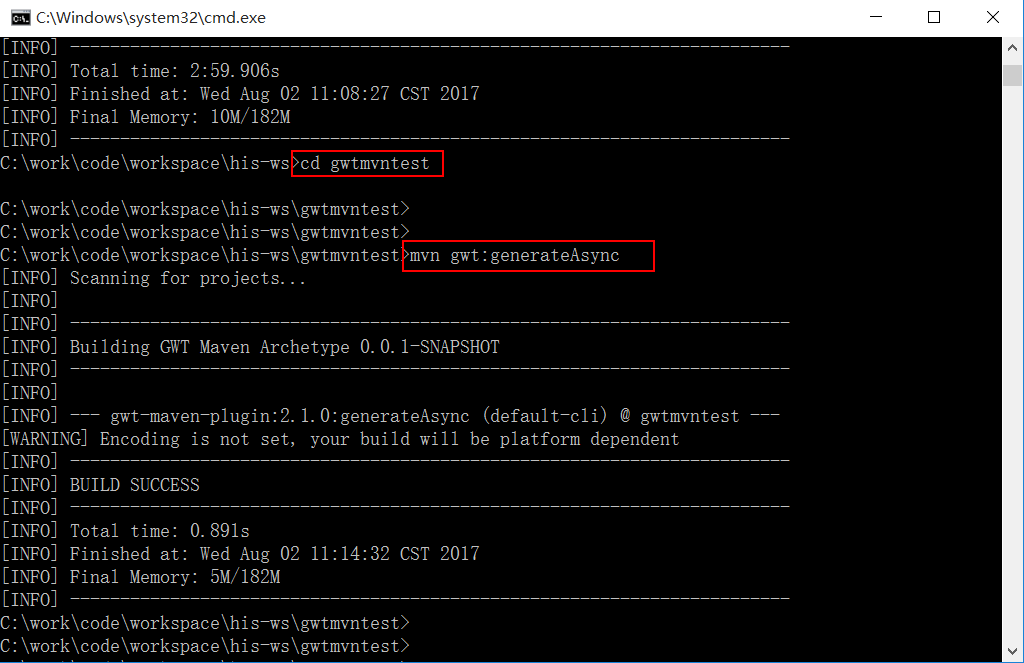
输入上图所示的maven相关信息，然后回车，则创建成功，如下图所示：



**2、自动创建 \*Async 异步接口 + I18N 子类**

1) 创建 Async异步接口

cd 至项目路径, 然后运行 mvn gwt:generateAsync

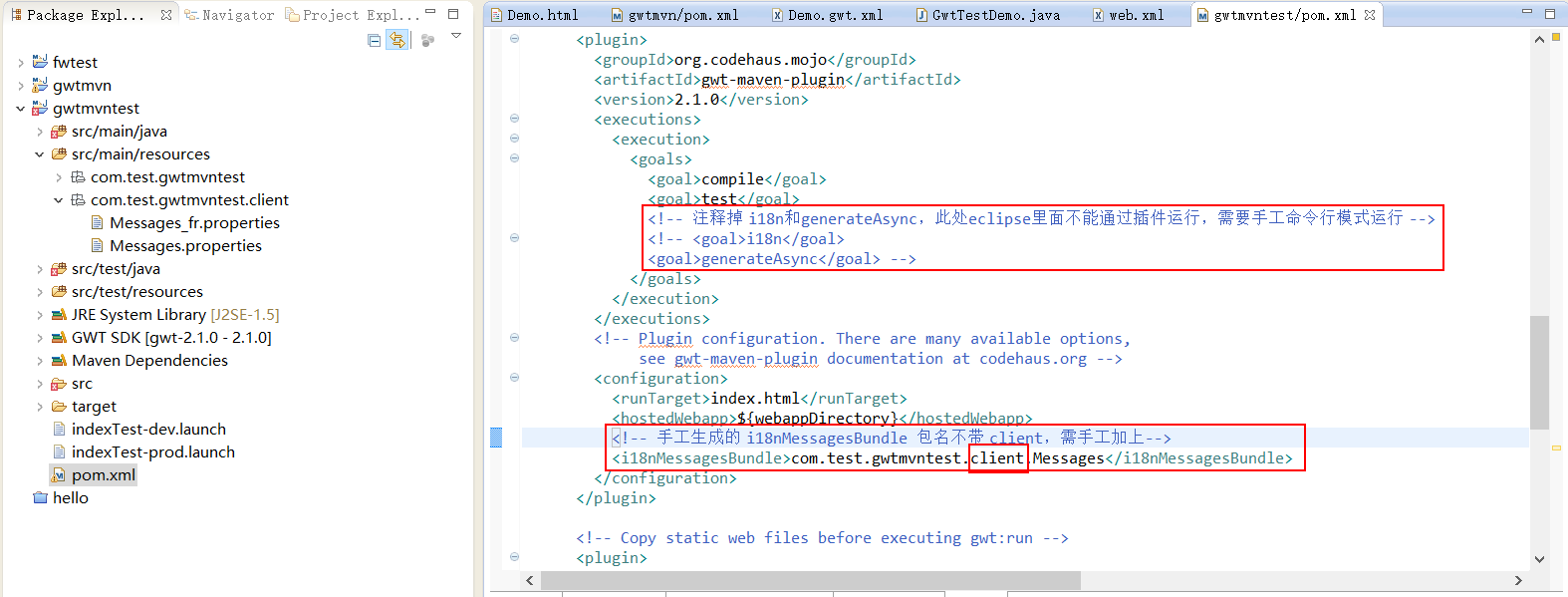


期间会自动下载相关文件到本地 repository，等待即可。

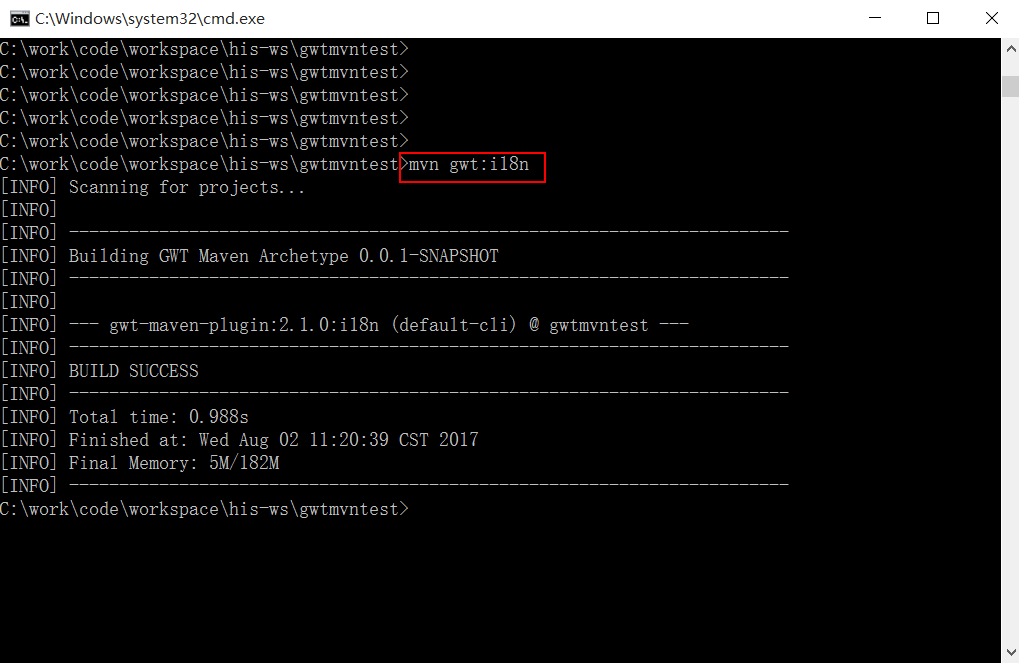
自动在 \target\generated-sources\gwt\com\test\gwtmvntest\client\ 下生成 GreetingServiceAsync，将在 GWT RPC 时调用

2) 创建i18n子类

i> 修改pom.xml文件，主要两个修改点，见下图：



ii> cd 至项目路径, 然后运行 mvn gwt:i18n



期间会自动下载相关文件到本地 repository，等待即可。

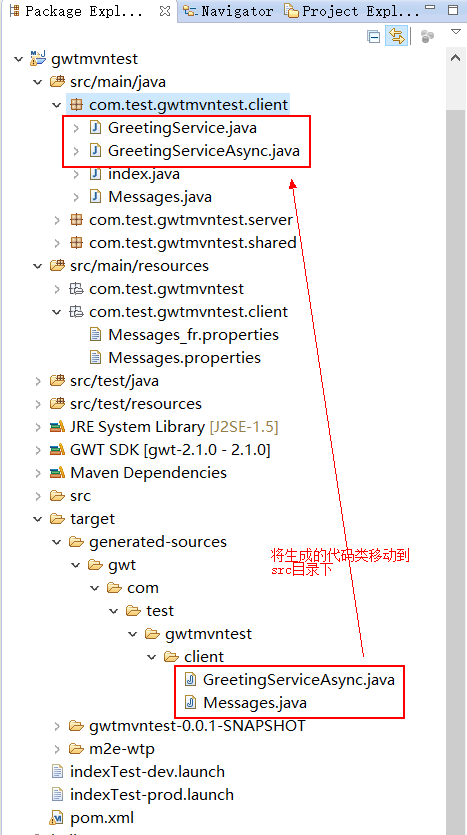
自动在 \target\generated-sources\gwt\com\test\gwtmvntest\client\ 下生成 Messages

静态 I18N key 都在此定义，再由 Messages\_\*.properties 文件配置 value.

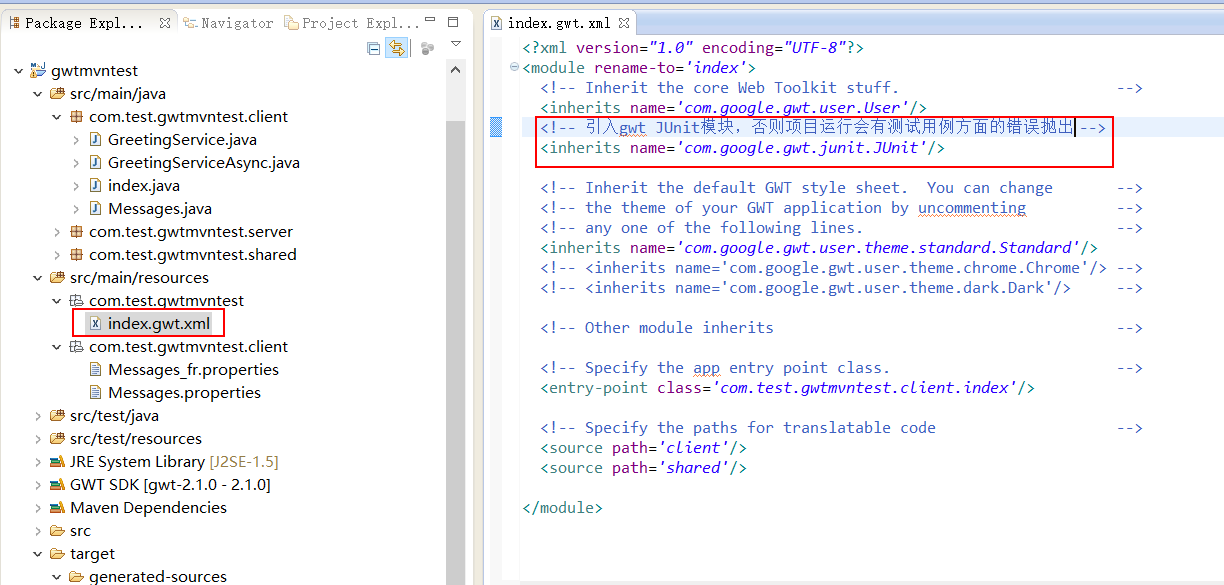
**3、导入项目，并做相关修改**

1) 将项目 gwtmvntest 以maven项目方式导入eclipse

2) 将上面第二步生成的异步接口和i18n子类拷贝到src目录下

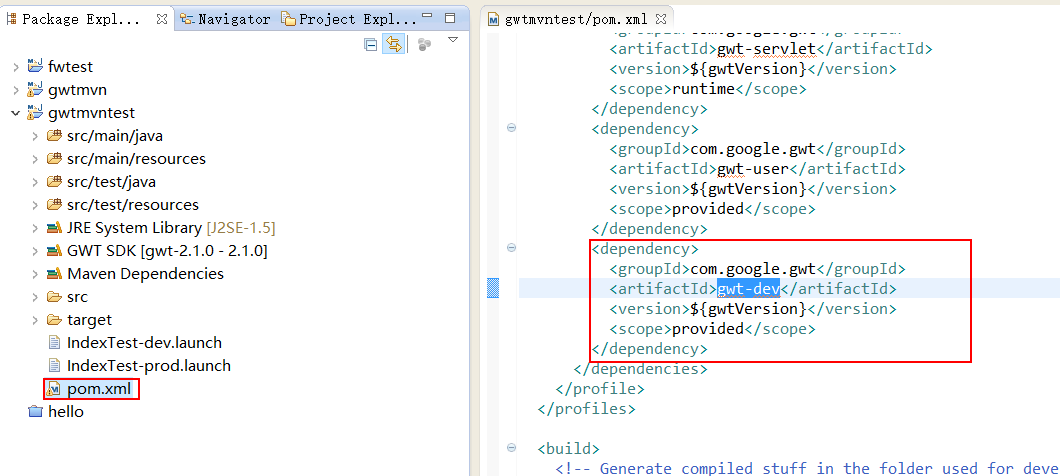


3) 修改 index.gwt.xml文件，引入 gwt JUnit模块

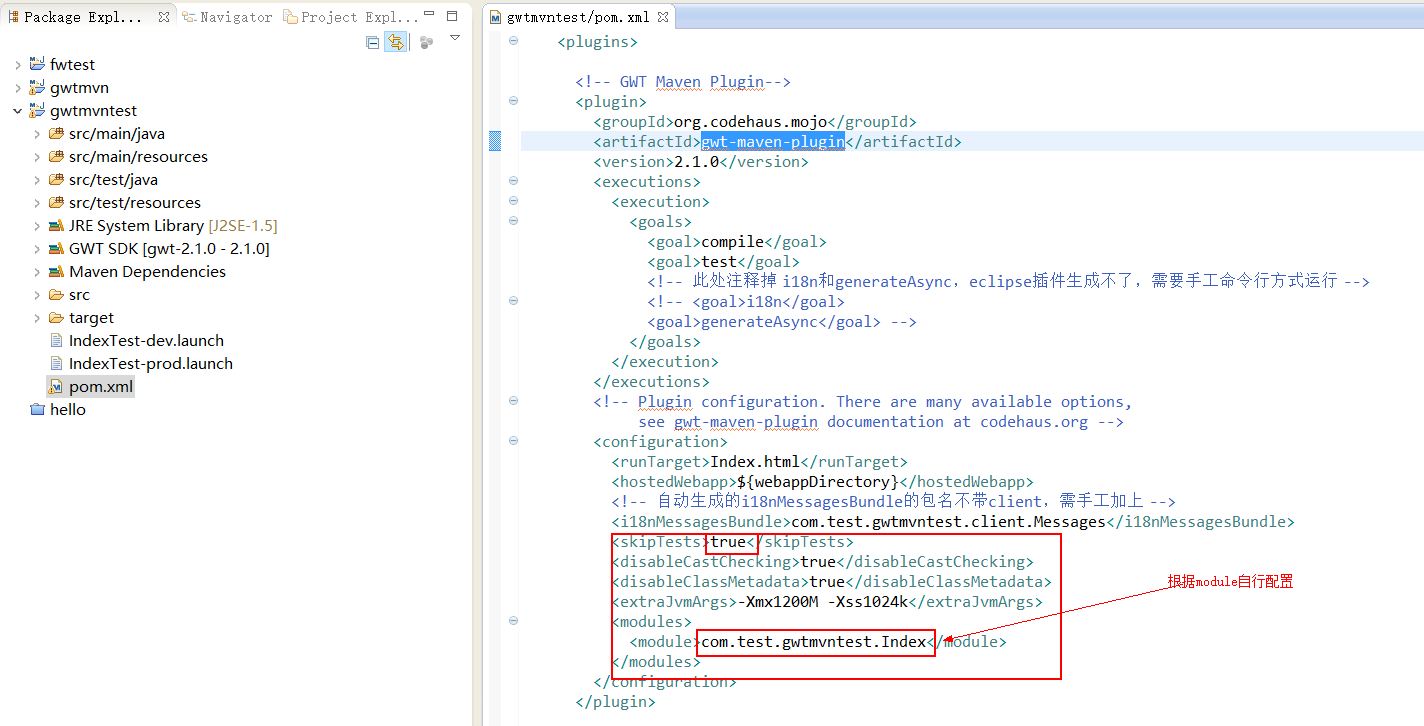


4) 修改pom.xml文件

i> 引入 gwt-dev maven依赖

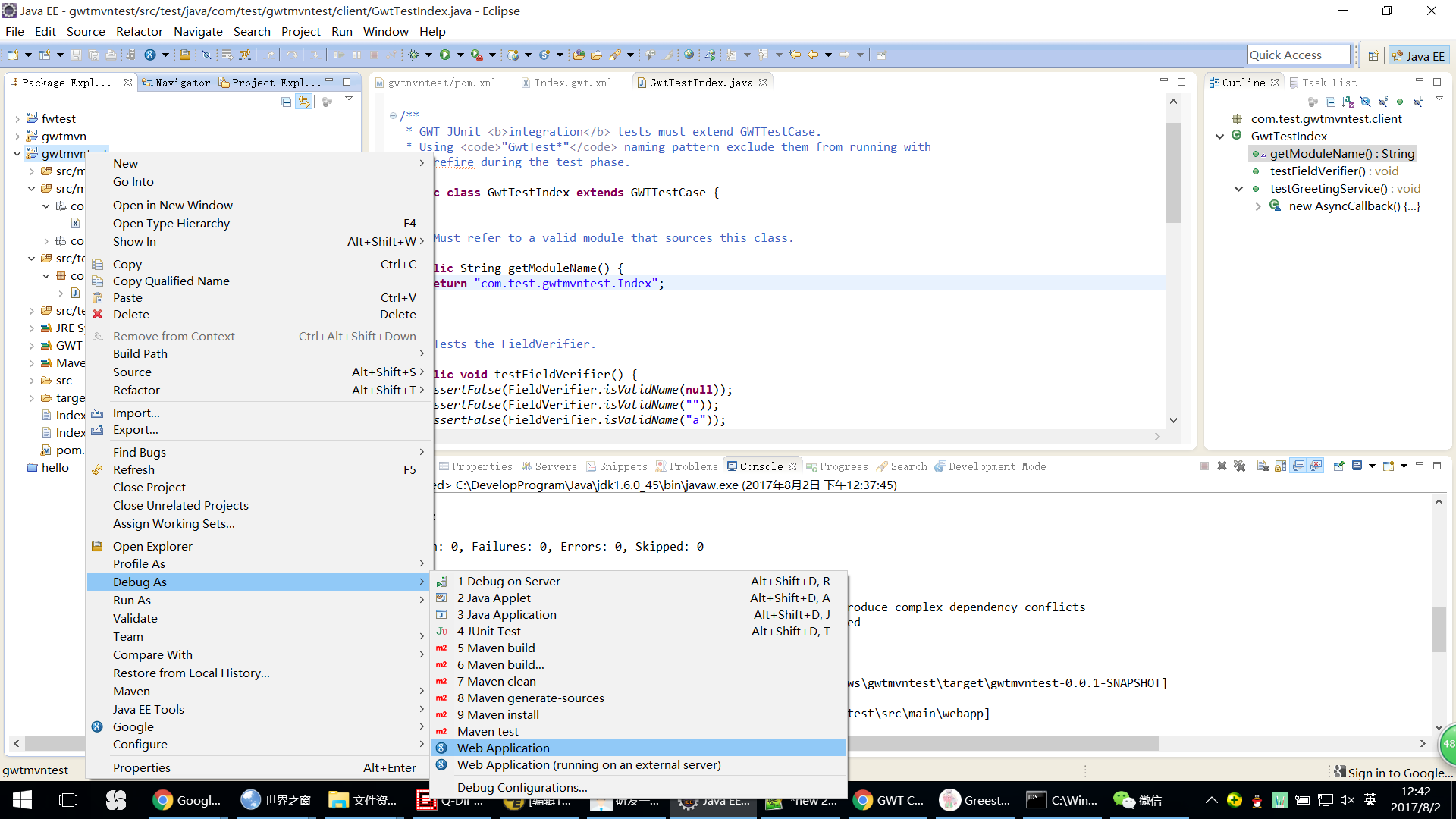


ii> 配置 gwt-maven-plugin



**4、运行项目**

右键选中项目 gwtmvntest，然后选择菜单"Debug As -> Web Application"，点击运行：



成功启动。

<http://127.0.0.1:8888/Index.html?gwt.codesvr=127.0.0.1:9997>然后在浏览器（需安装gwt浏览器插件）输入 进行访问，如下图所示：

附上工程代码：