1、在eclipse中创建一个maven项目

new -> maven -> maven project -> maven archetype quickstart

2、设置一下项目的坐标

groupId：com.zhss.maven

artifactId：maven-helloworld

version：1.0-SNAPSHOT

package：com.zhss.maven

在pom.xml中将junit的依赖从3.8.1版本修改成4.12版本

在eclipse开发maven项目的话，每次你保存pom.xml，eclipse就会让maven自动去解析pom.xml，然后下载你最新声明的依赖，立即从远程仓库下载到本地仓库

但是如果纯手工通过archetype去创建maven工程的话，是需要在mvn test package打包的时候，才会去检查pom.xml，才会去下载依赖

在eclipse，window -> preferences -> 搜索font -> color and font -> basic -> text font，设置字体为courier new

3、编写代码

src/main/java

package com.zhss.maven;

public class HelloWorld {

public String sayHello(String name) {

return "hello, " + name;

}

public static void main(String[] args) {

System.out.println("hello world......");

}

}

4、编写测试代码

package com.zhss.maven;

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class HelloWorldTest {

@Test

public void testSayHello() {

HelloWorld helloWorld = new HelloWorld();

String result = helloWorld.sayHello("leo");

assertEquals("hello, leo", result);

}

}

5、用eclipse执行maven命令

右击eclipse工程 -> run configurations -> maven build -> new ->

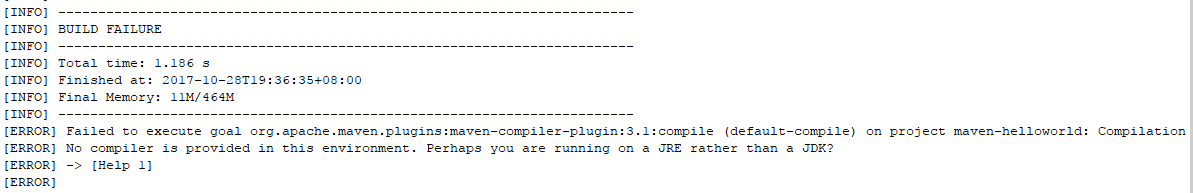
maven-helloworld

选择工程目录中的maven项目

clean package

run运行这个maven命令

很多同学对我讲课过程中遇到的问题是怎么排查的很感兴趣



JRE是用来运行的，JDK是用来编译的

windows -> preferences - java - installed jres -> 默认用的时jre目录 -> 修改为 jdk目录

右击工程，build path -> configure build path -> 将用的jdk修改为刚才设置的那个jdk就可以了

853769620