## 开发工具

推荐：IntelliJ IDEA



## JDK版本

IDE运行需要安装jdk1.8

当前项目开发基于1.7+

## 数据库

公司产品一般选mysql

除非客户指定oracle，一般客户项目也采用mysql

## 系统

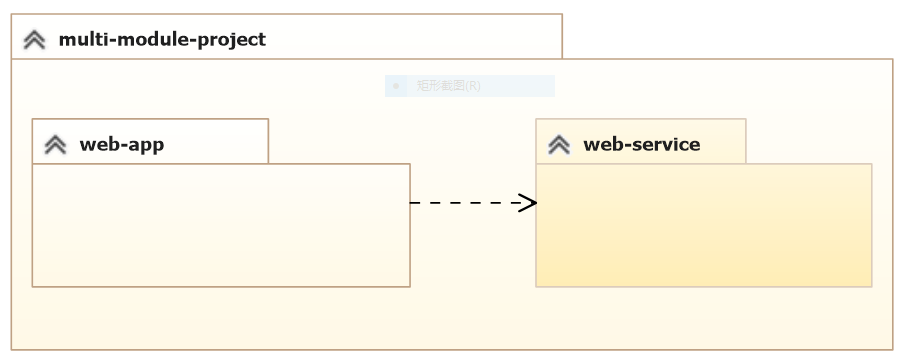
Widows，linux，mac系统等

## 5.流程

### 5.1 网上资料

本文通过一个例子来介绍利用maven来构建一个多模块的jave项目。开发工具：intellij idea。

**项目结构**

  
  
multi-module-project是主工程，里面包含两个模块（Module）：

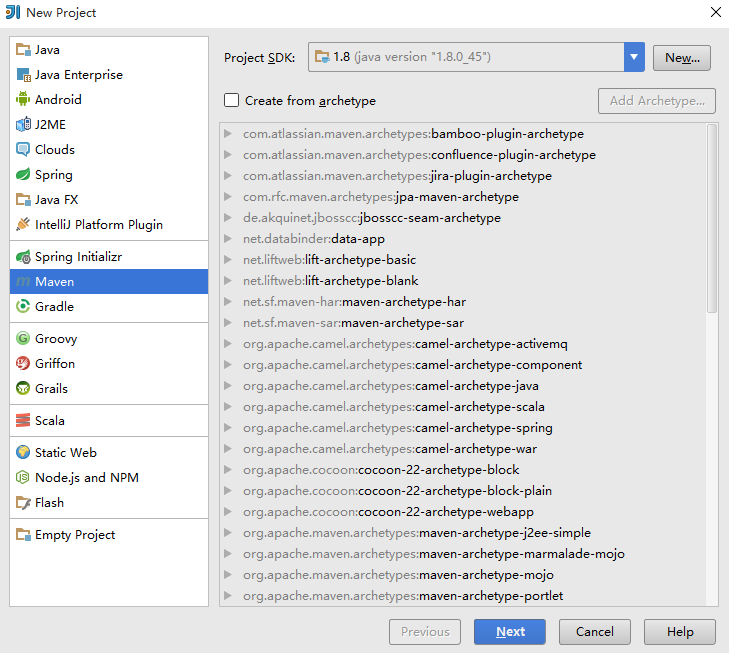
web-app是应用层，用于界面展示，依赖于web-service参的服务。

web-service层是服务层，用于给app层提供服务。

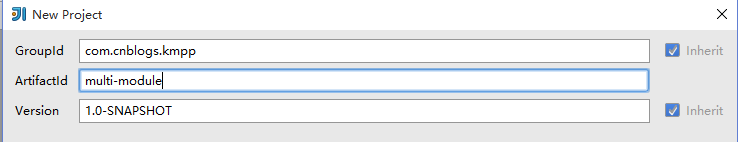
**构建项目**

**Parent Project**

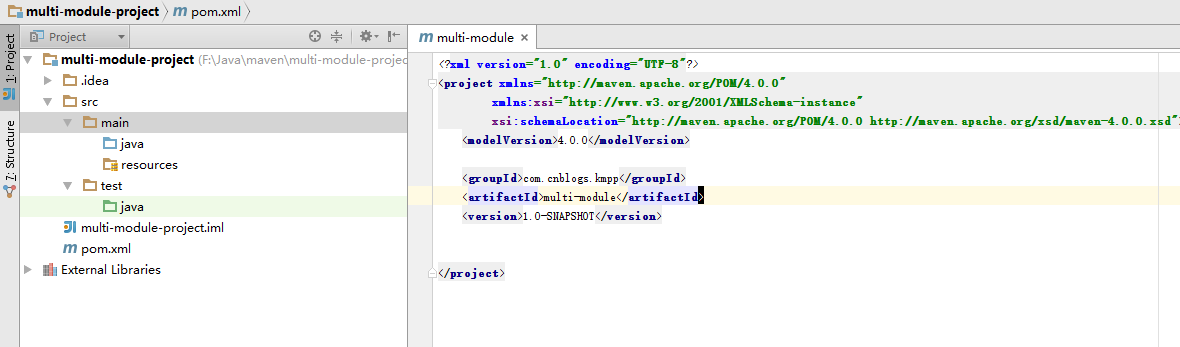
新建一个空白标准maven project（不要选择Create from archetype选项）



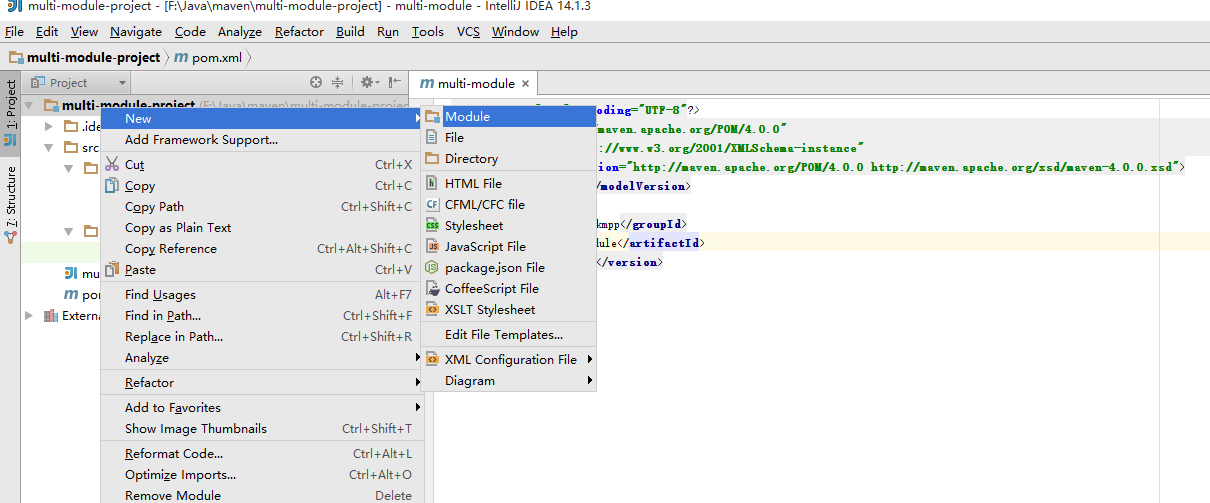
填写项目坐标



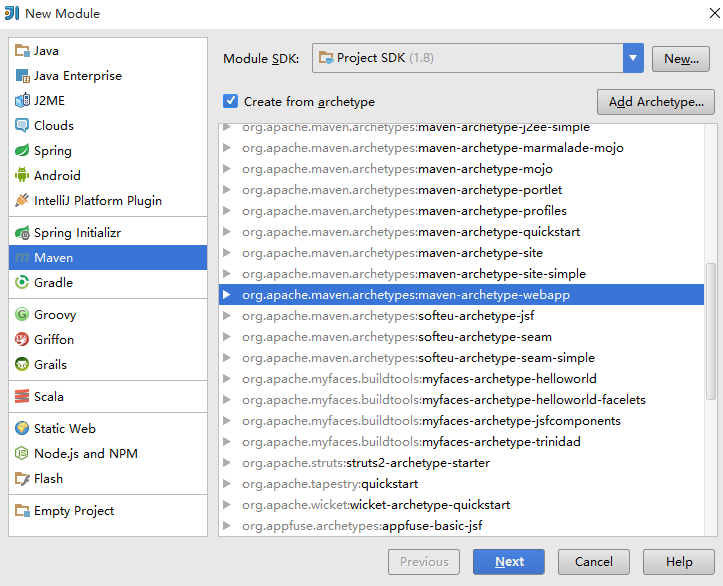
得到一个标准的maven项目，因为该项目是作为一个Parent project存在的，可以直接删除src文件夹。



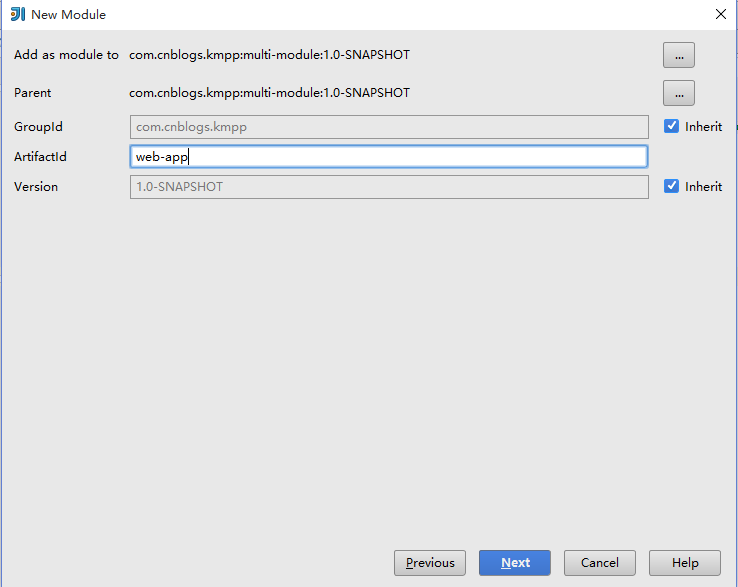
**增加web-app模块（Module）**



选择从archetype创建（选择webapp选项）

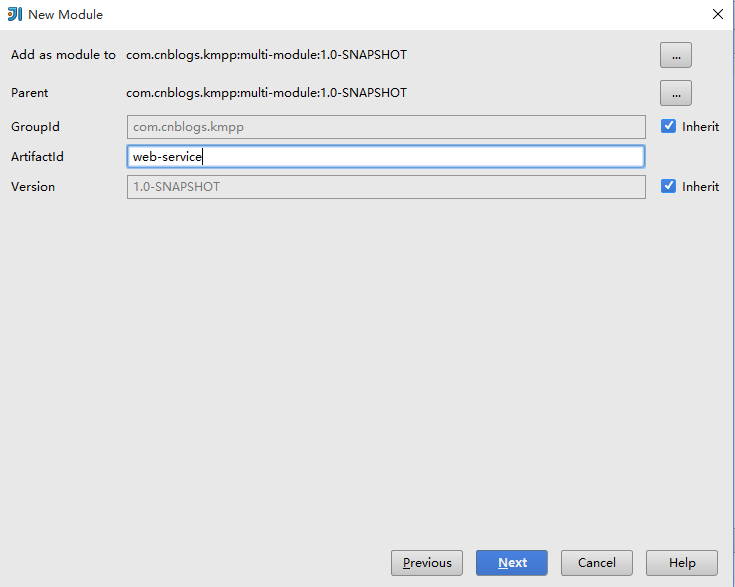


groupId和version继承自Parent project，这里只需要填写artifactId即可。

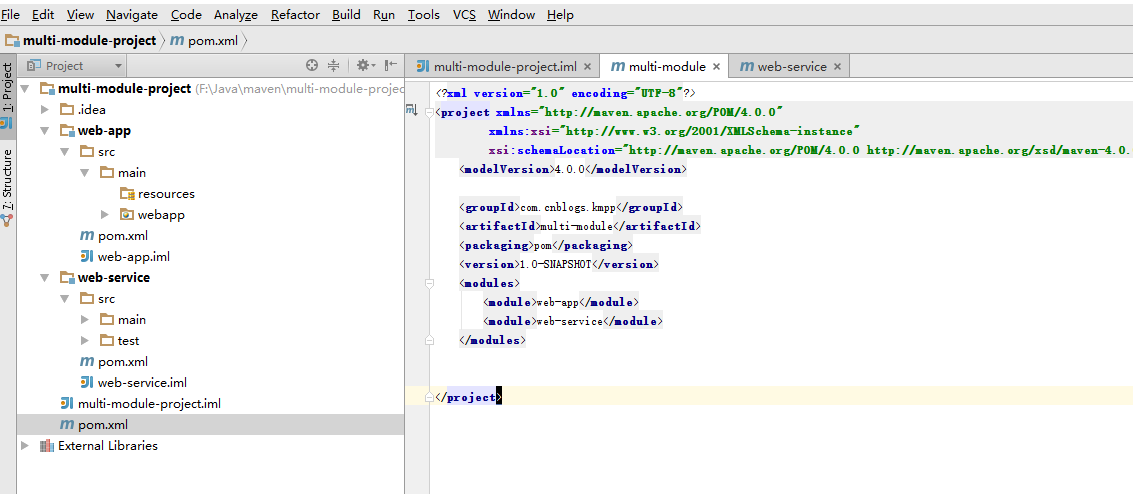


**增加web-service模块**

用同样的方法创建web-service模块（不过该模块是一个空白maven标准项目，不要从archetype创建）



**得最终项目结构**



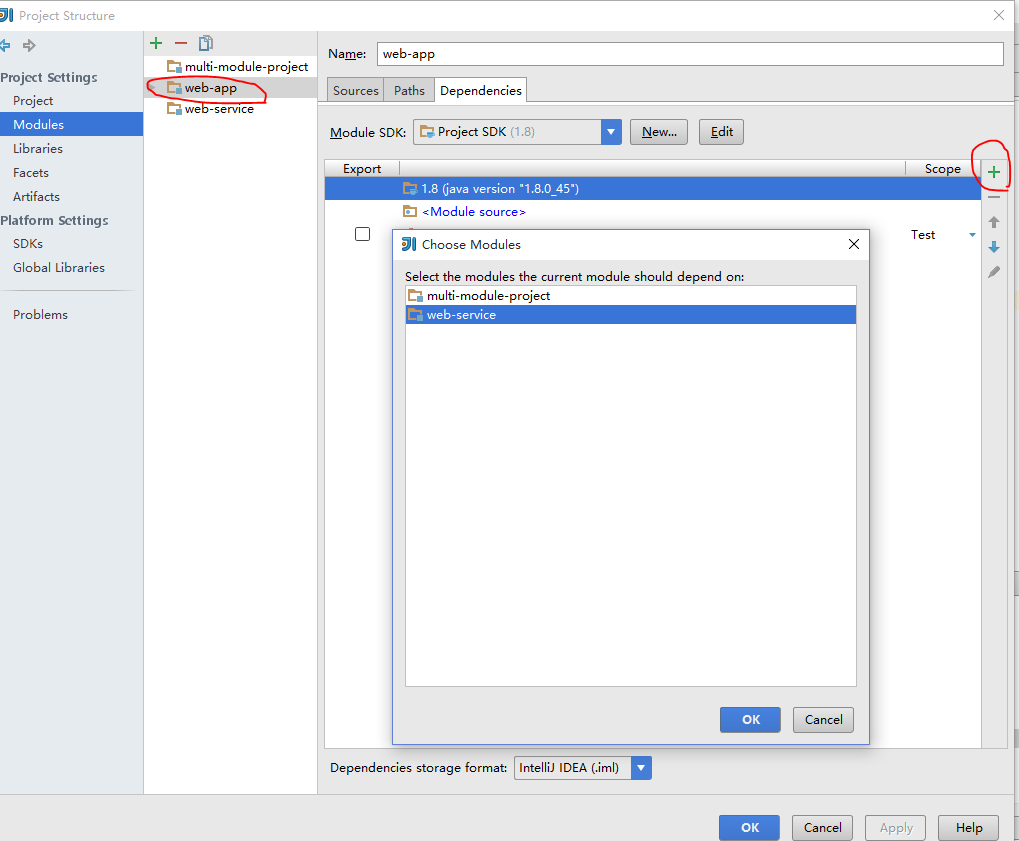
**关键几点**

Parent project和各个Module拥有独立pom文件，他们之间的关系后续会专门写文章介绍。

Parent project用于组织不同的Module，不实现逻辑

Module集成Parent project的groupId和version，Module只需要指定自己的artifactId即可。

**添加项目依赖**



上面的操作是添加web-app对web-service模块的依赖，完成上述操作后web-Service中public的类已经在web-app模块中可见了。但是这个时候在app模块使用了service模块中的类，通过maven进行编译（compile）的时候还是会报错XX找不到（XX为service模块的类），要解决这个问题需要在app的pom中增加对service的依赖：

<dependency>

<groupId>com.cnblogs.kmpp</groupId>

<artifactId>web-service</artifactId>

<version>1.0-SNAPSHOT</version>

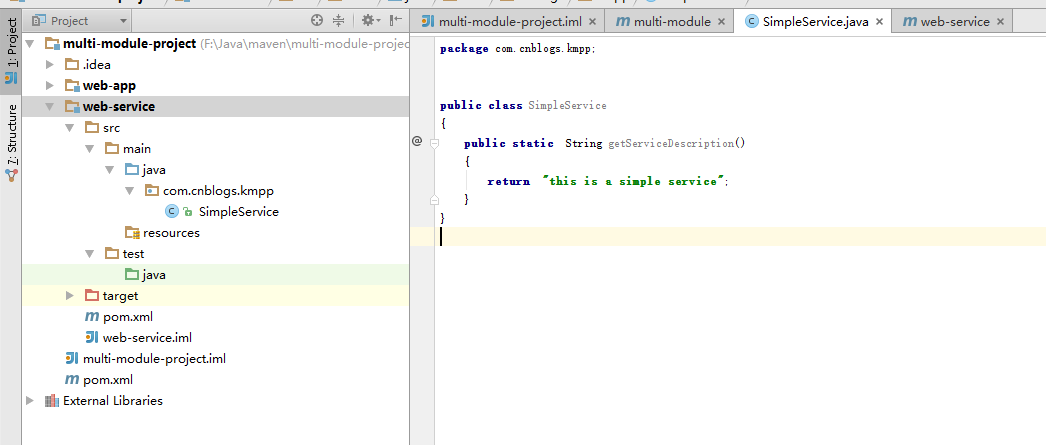
</dependency>

以上，项目依赖的添加已经完成

**开始编程**

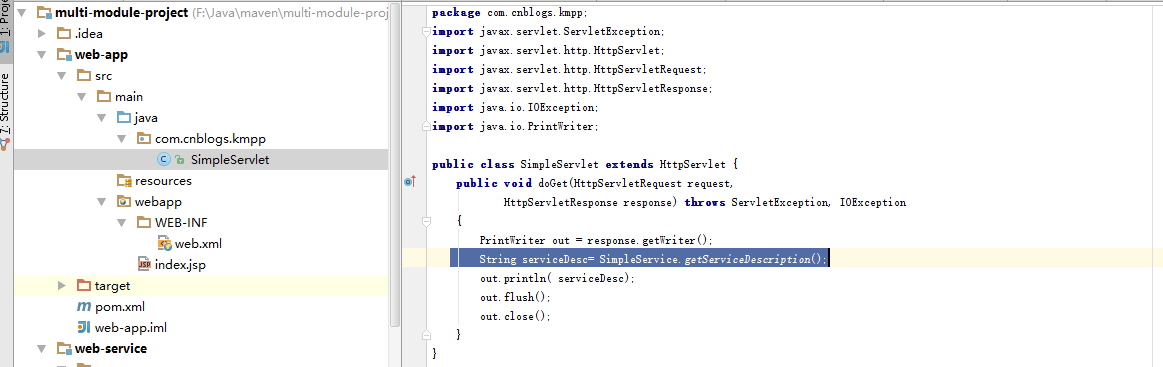
**web-service模块编程**

在web-service模块中增加一个Service类（SimpleService.java)



**web-app模块编程**

 在web-app模块增加一个Servlet，并且调用web-service模块的SimpleService类的getServiceDescription方法。



**配置Servlet**

[IMG_268](http://www.cnblogs.com/Brake/p/javascript:void(0);" \o "复制代码)

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Archetype Created Web Application</display-name>

<servlet>

<servlet-name>Simple</servlet-name>

<servlet-class>com.cnblogs.kmpp.SimpleServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Simple</servlet-name>

<url-pattern>/simple-servlet</url-pattern>

</servlet-mapping></web-app>

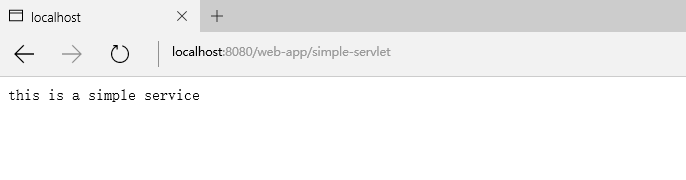
[IMG_269](http://www.cnblogs.com/Brake/p/javascript:void(0);" \o "复制代码)

**运行**

在web-app的pom中增加j2ee的依赖，以及jetty插件的依赖。运行jetty。

在浏览器中输入：http://localhost:8080/web-app/simple-servlet

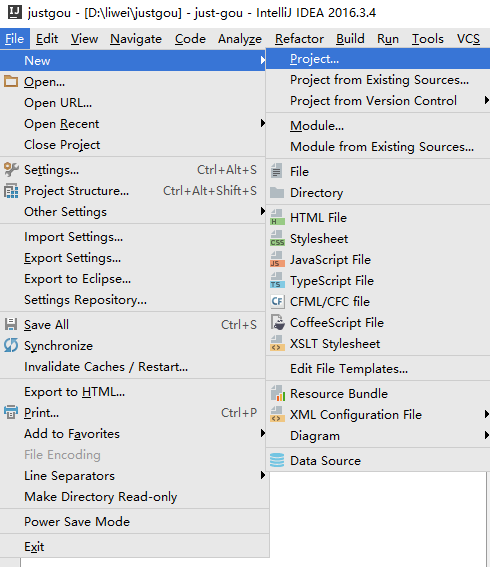
得到运行结果：

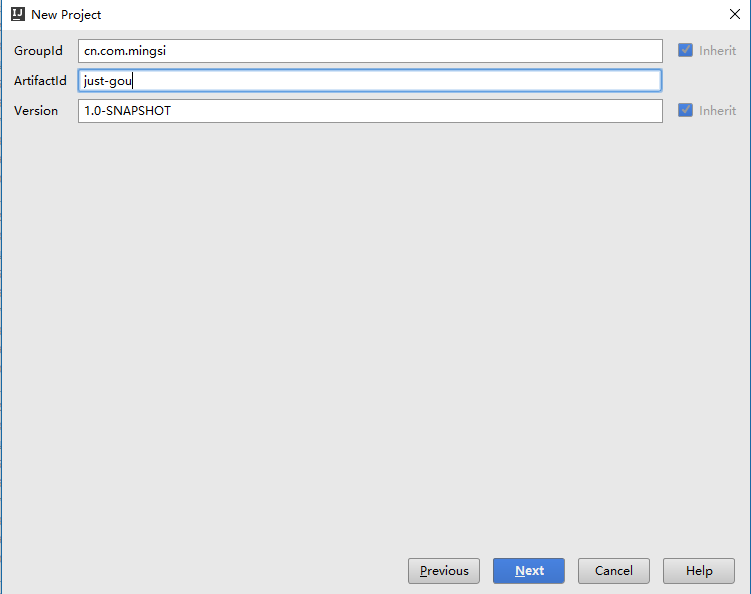


### 5.2 搭建步骤

1. 创建 parent project

名称取项目或产品名称 如justgou





groupId一般取值：cn.com.公司简称

ArtifactId 一般取值：项目名称

指定项目位置：Project location

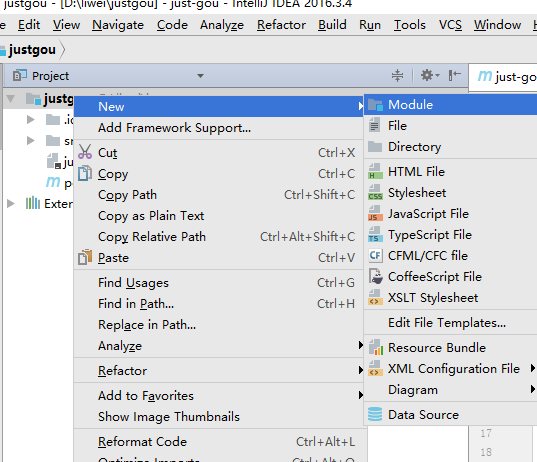
其余步骤默认即可

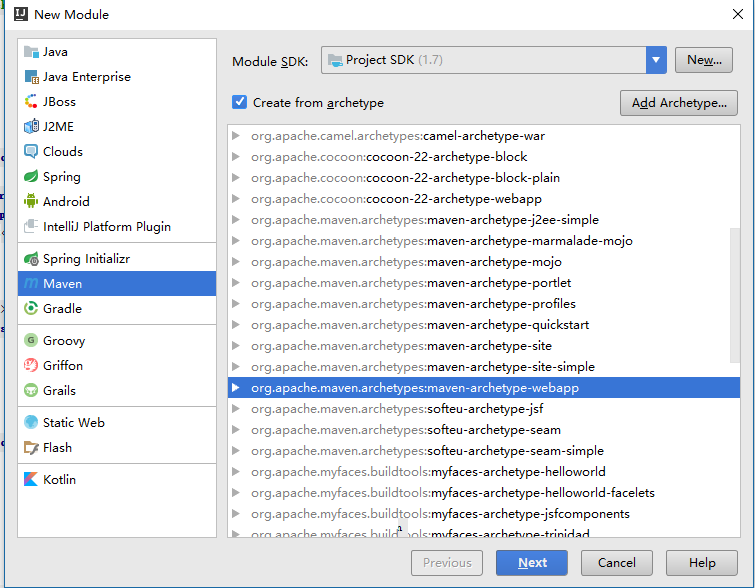
Parent project 搭建完成

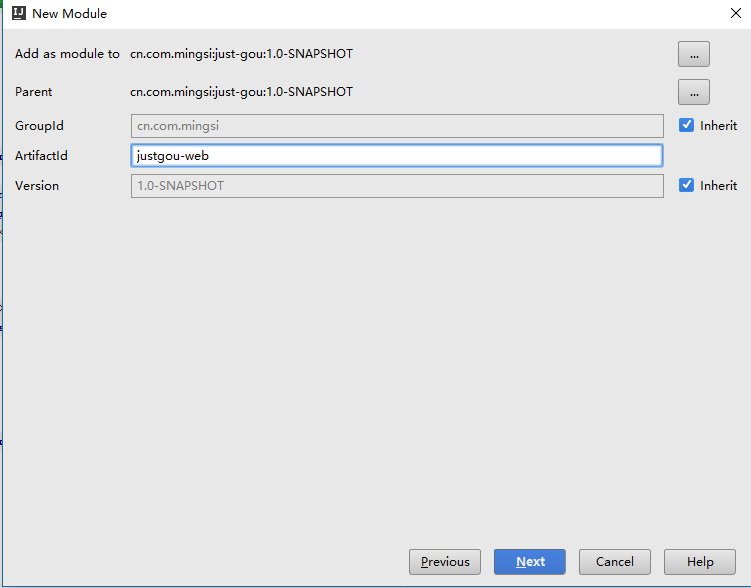
修改pom.xml 参照

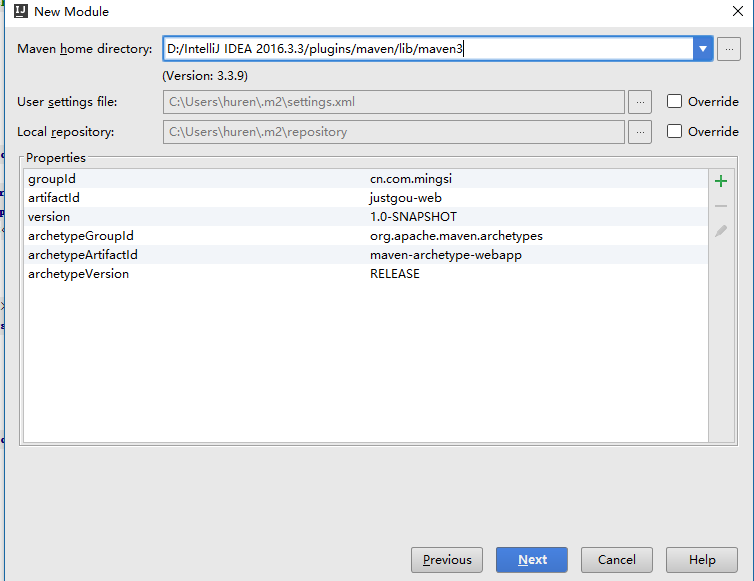


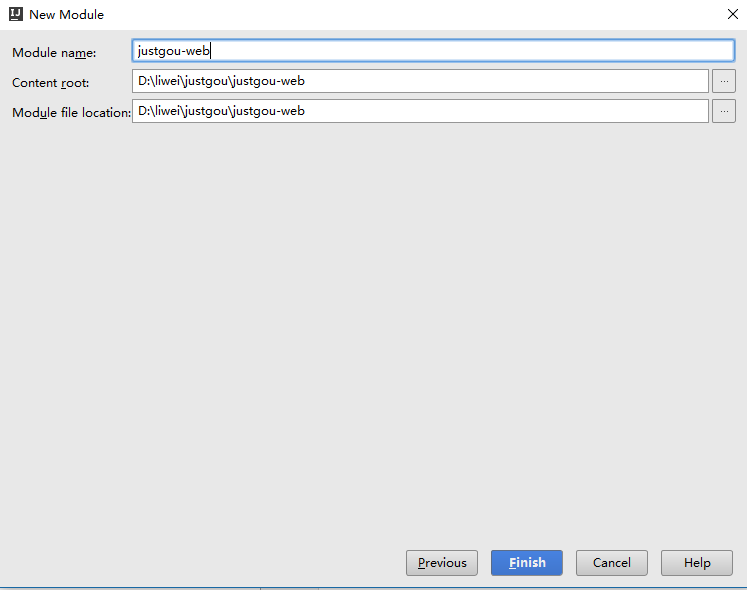
1. web应用层项目





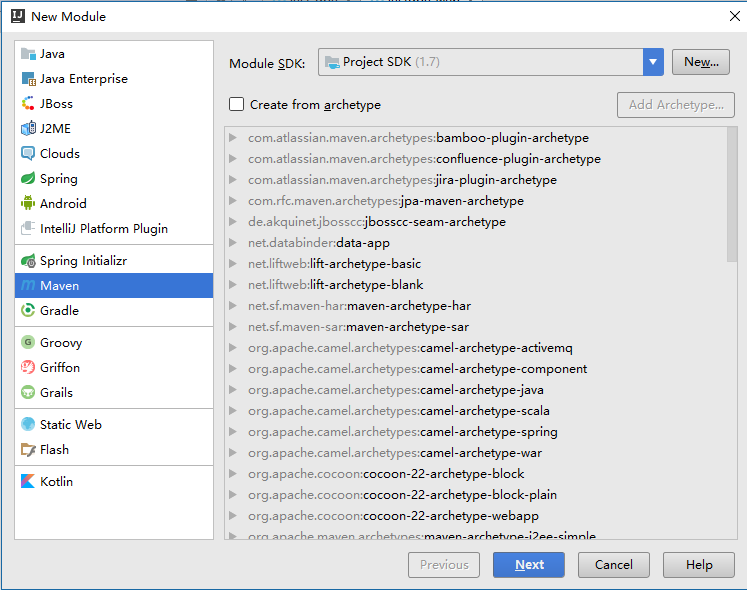






完成

1. 通用模块（注意处于框架搭建方便的目的，建议此模块不要添加任何新代码）



1. 通用模块扩展
2. 业务接口模块
3. 业务实现模块
4. 其他模块

report-table

Location

install mybatis-generator:generate -DconfigurationFile=generatorConfig-report.xml

Install

clean install