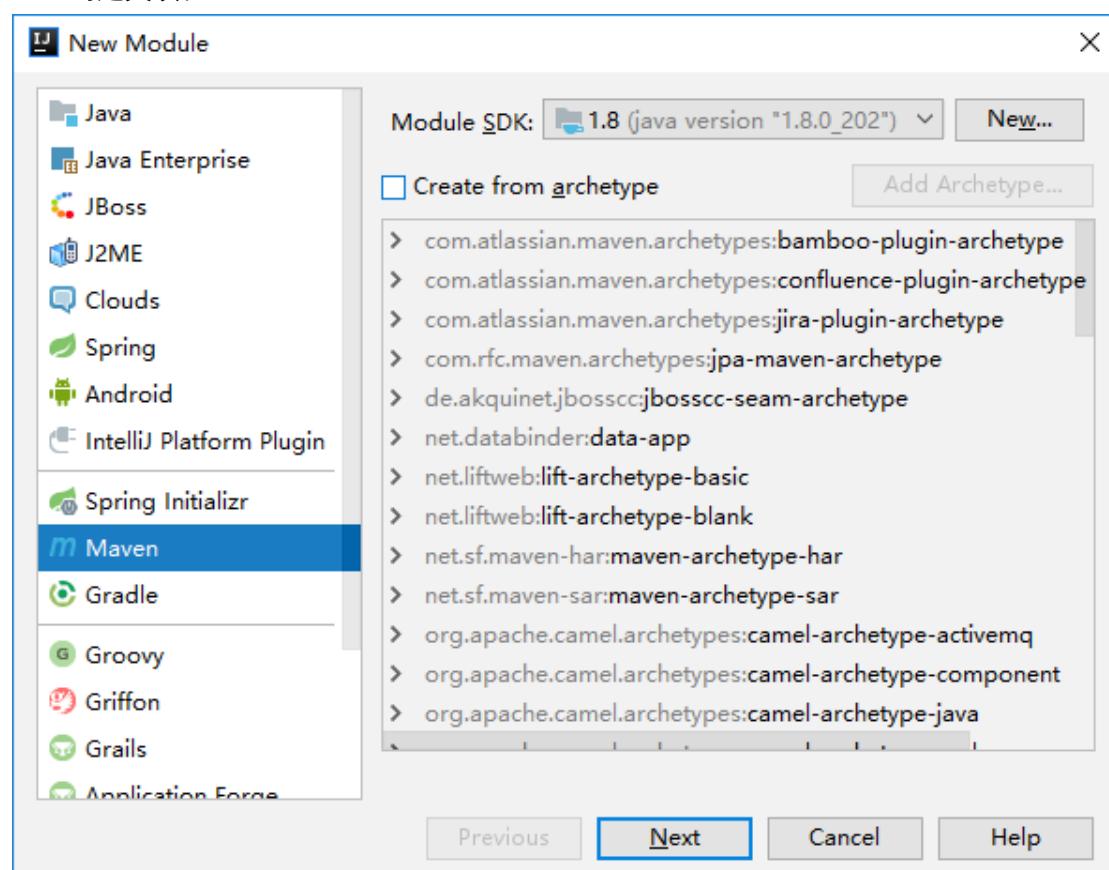


# 后台管理系统——项目框架搭建

## 1、ego-parent 父模块

### 1.1、创建 pom 项目

创建父项目



New Project

GroupId	com.ego	<input checked="" type="checkbox"/> Inherit
ArtifactId	ego-parent	
Version	1.0-SNAPSHOT	<input checked="" type="checkbox"/> Inherit

Previous Next Cancel Help

New Project

Project name:	ego-parent	
Project location:	D:\IdeaProjects\Dylan\ego-parent	...

More Settings

Previous Finish Cancel Help

## 1.2、编辑 pom.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.ego</groupId>
  <artifactId>ego-parent</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <modules>
    <module>ego-common</module>
    <module>ego-manager</module>
  </modules>

  <packaging>pom</packaging>

  <!-- 集中定义依赖组件版本号 -->
  <properties>
    <!-- junit 依赖 -->
    <junit.version>4.12</junit.version>
    <!-- spring 依赖 -->
    <spring-version>4.3.4.RELEASE</spring-version>
    <!-- hibernate 数据验证框架依赖 -->
    <hibernate-validator.version>5.3.3.Final</hibernate-validator.version>
    <!-- spring-mybatis 依赖 -->
    <mybatis-spring.version>1.3.0</mybatis-spring.version>
    <!-- mybatis 依赖 -->
    <mybatis.version>3.4.1</mybatis.version>
    <!-- servlet api 依赖 -->
    <servlet.version>3.0.1</servlet.version>
    <!-- jsp 依赖 -->
    <jstl.version>1.2</jstl.version>
    <standard.version>1.1.2</standard.version>
    <!-- slf4j 依赖 -->
    <slf4j-log4j12.version>1.7.22</slf4j-log4j12.version>
    <!-- json 相关依赖 -->
    <jackson.version>2.8.5</jackson.version>
    <!-- 数据库连接池相关依赖 -->
    <druid.version>1.0.27</druid.version>
```

```
<!-- mysql 依赖 -->
<mysql.version>5.1.8</mysql.version>
<!-- 分页工具依赖 -->
<pagehelper.version>4.1.6</pagehelper.version>
<!-- 文件上传依赖 -->
<commons-fileupload.version>1.3.2</commons-fileupload.version>
<!-- redis 依赖2.9.1 版本为了兼容springdata-redis -->
<jedis.version>2.9.1</jedis.version>
<!-- spring data redis 依赖使用1+最新版本因为2+需要spring5+版本 -->
<springdata-redis.version>1.8.18.RELEASE</springdata-redis.version>
<!-- solr 依赖 -->
<solrj.version>7.6.0</solrj.version>
<!-- jdk 依赖 -->
<jdk.version>1.8</jdk.version>
<!-- Apache 其他依赖 -->
<maven-compiler-plugin.version>3.5.1</maven-compiler-plugin.version>
<poi.version>3.15</poi.version>
<commons-net.version>3.3</commons-net.version>
<commons-logging.version>1.2</commons-logging.version>
</properties>
<!-- 只定义依赖的版本，并不实际依赖 -->
<dependencyManagement>
    <dependencies>
        <!-- servlet api -->
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>${servlet.version}</version>
            <!-- provided 作用域表示：只在编译和测试的时候 运行时，如果容器已经提供了该jar 包，直接调用容器jar 包即可。 -->
            <scope>provided</scope>
        </dependency>
        <!-- jsp 相关 -->
        <dependency>
            <groupId>jstl</groupId>
            <artifactId>jstl</artifactId>
            <version>${jstl.version}</version>
        </dependency>
        <!-- Spring -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>${spring-version}</version>
        </dependency>
    </dependencies>
</dependencyManagement>
```

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aspects</artifactId>
    <version>${spring-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${spring-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>${spring-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${spring-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${spring-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-orm</artifactId>
    <version>${spring-version}</version>
</dependency>
<!-- 使用hibernate 的数据验证框架 -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>${hibernate-validator.version}</version>
</dependency>
<!-- Jackson Json 处理工具包 -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>${jackson.version}</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
```

```
<artifactId>jackson-annotations</artifactId>
<version>${jackson.version}</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>${jackson.version}</version>
</dependency>
<dependency>
    <groupId>taglibs</groupId>
    <artifactId>standard</artifactId>
    <version>${standard.version}</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>${mybatis-spring.version}</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>${mybatis.version}</version>
</dependency>
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper</artifactId>
    <version>${pagehelper.version}</version>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
</dependency>
<!-- 日志处理 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>${slf4j-log4j12.version}</version>
</dependency>
<!-- MySQL -->
<dependency>
    <groupId>mysql</groupId>
```

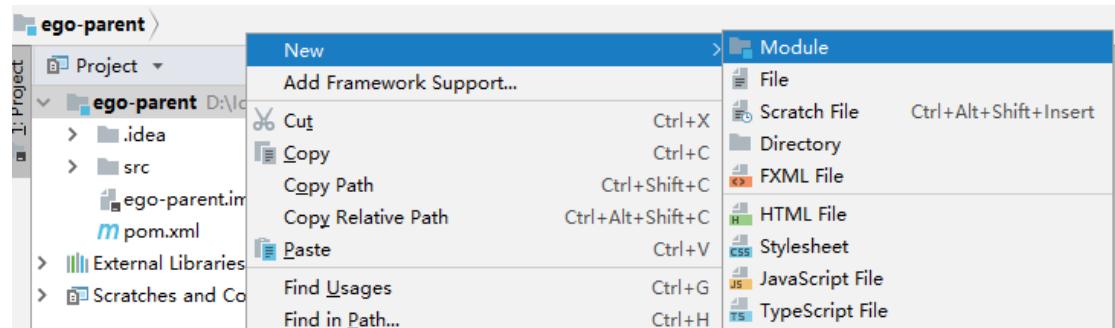
```
<artifactId>mysql-connector-java</artifactId>
<version>${mysql.version}</version>
</dependency>
<!-- 连接池 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
    <version>${druid.version}</version>
</dependency>
<!-- 文件上传组件 -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>${commons-fileupload.version}</version>
</dependency>
<!-- Redis 客户端 -->
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>${jedis.version}</version>
</dependency>
<!-- spring data redis 依赖 -->
<dependency>
    <groupId>org.springframework.data</groupId>
    <artifactId>spring-data-redis</artifactId>
    <version>${springdata-redis.version}</version>
</dependency>
<!-- solr 客户端 -->
<dependency>
    <groupId>org.apache.solr</groupId>
    <artifactId>solr-solrj</artifactId>
    <version>${solrj.version}</version>
</dependency>
<!-- Apache 其他组件 -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>${poi.version}</version>
</dependency>
<dependency>
    <groupId>commons-net</groupId>
    <artifactId>commons-net</artifactId>
    <version>${commons-net.version}</version>
</dependency>
```

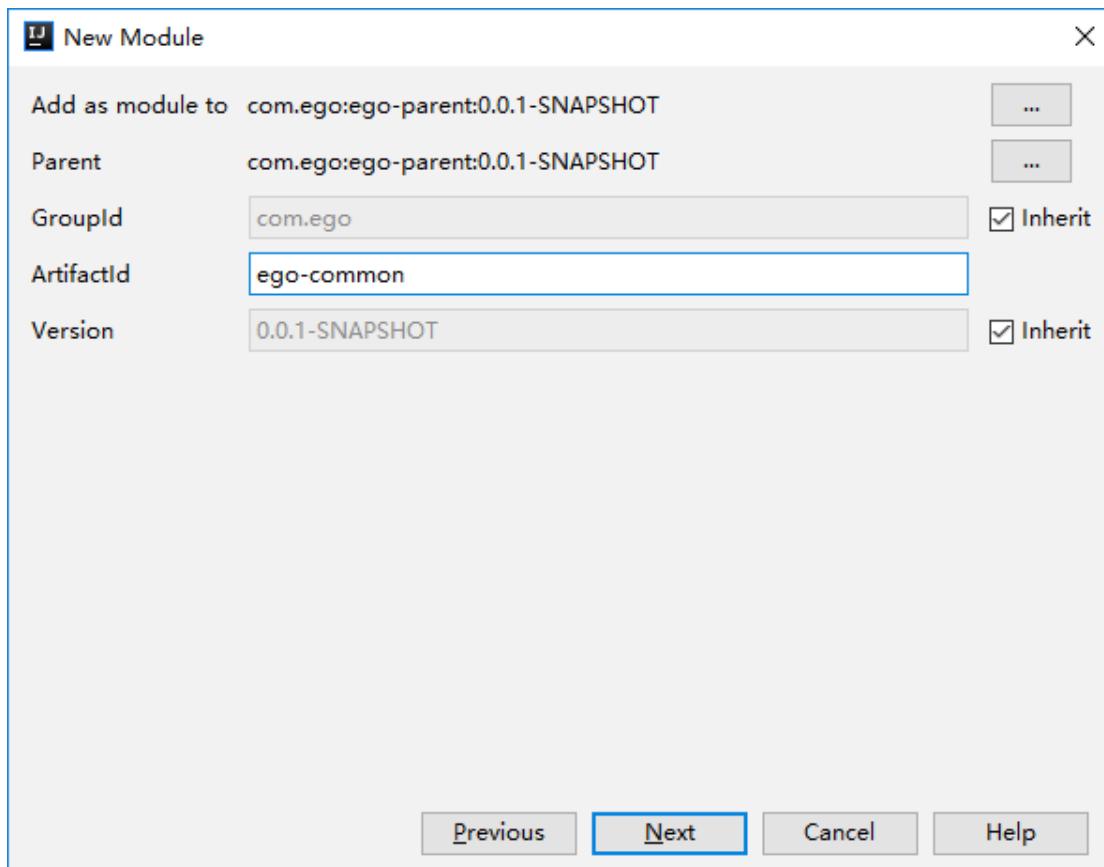
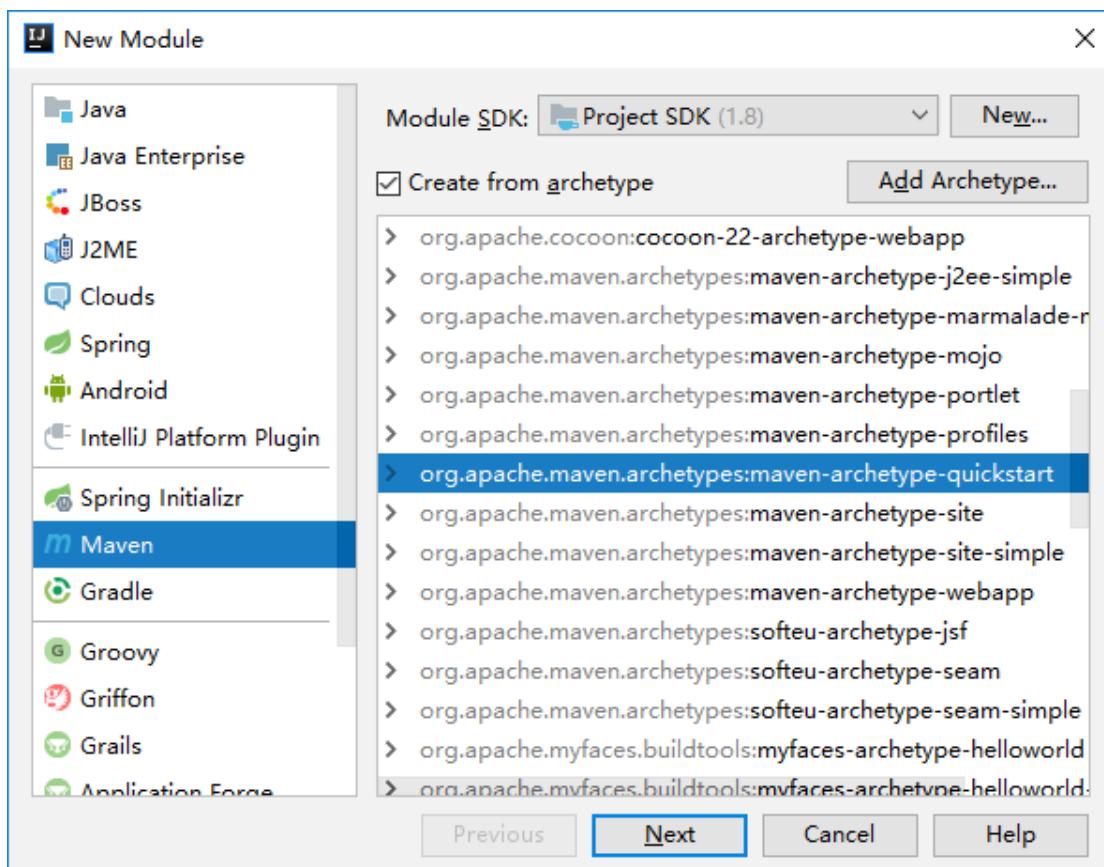
```
<dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
    <version>${commons-logging.version}</version>
</dependency>
</dependencies>
</dependencyManagement>
</project>
```

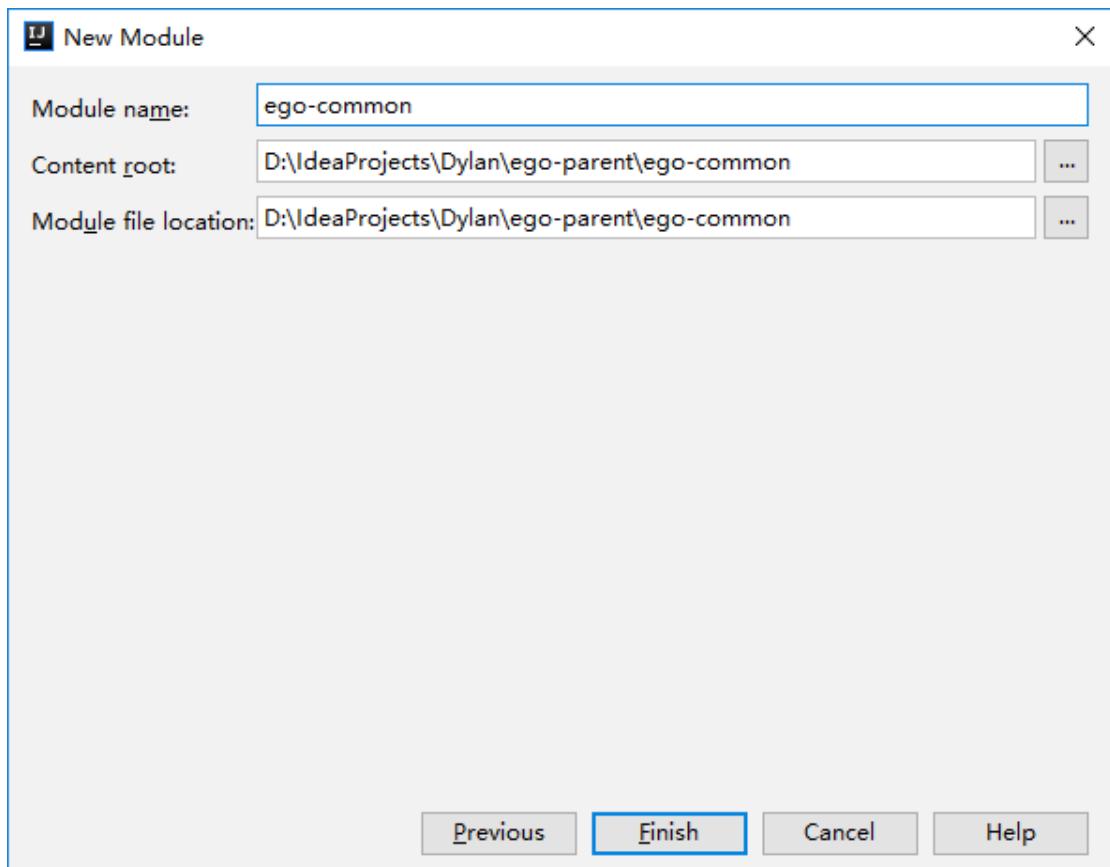
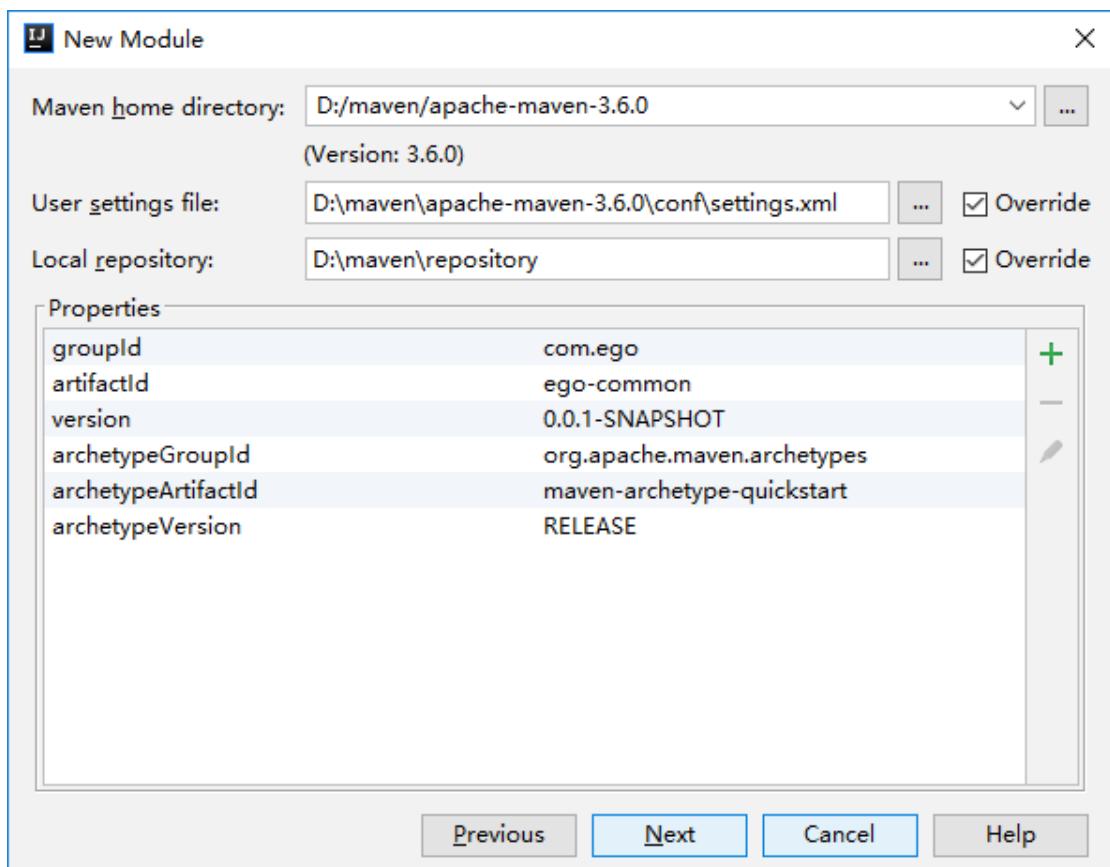
## 2、ego-common 子模块

用于添加公共的工具类、枚举类、拦截器等。

### 2.1、创建 jar 项目







## 2.2、编辑 pom.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <parent>
    <artifactId>ego-parent</artifactId>
    <groupId>com.ego</groupId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>ego-common</artifactId>

  <name>ego-common</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- junit 单元测试 -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-test</artifactId>
    </dependency>
    <!-- 分页工具 -->
    <dependency>
      <groupId>com.github.pagehelper</groupId>
      <artifactId>pagehelper</artifactId>
    </dependency>
    <dependency>
```

```

<groupId>commons-net</groupId>
<artifactId>commons-net</artifactId>
</dependency>
<!-- 添加mybatis 依赖 -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
</dependency>
<!-- Redis 客户端 -->
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
</dependency>
<!-- servlet api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <scope>provided</scope>
</dependency>
</dependencies>
</project>

```

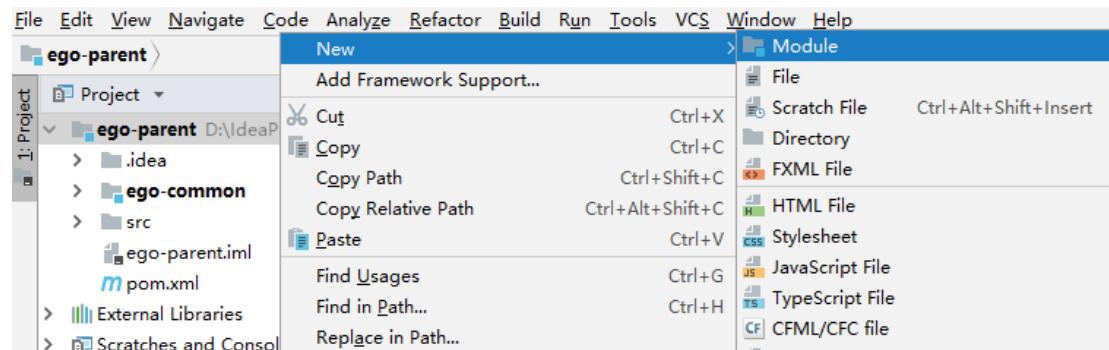
### 3、ego-manager 父模块

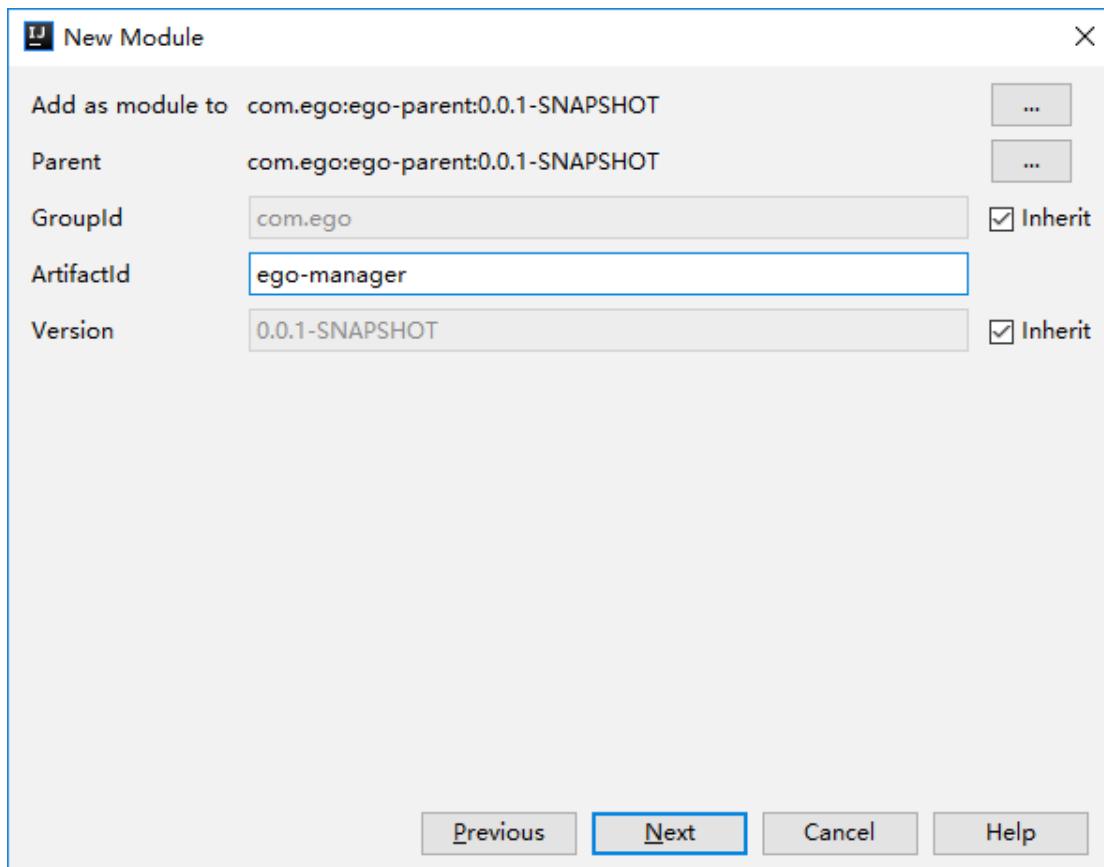
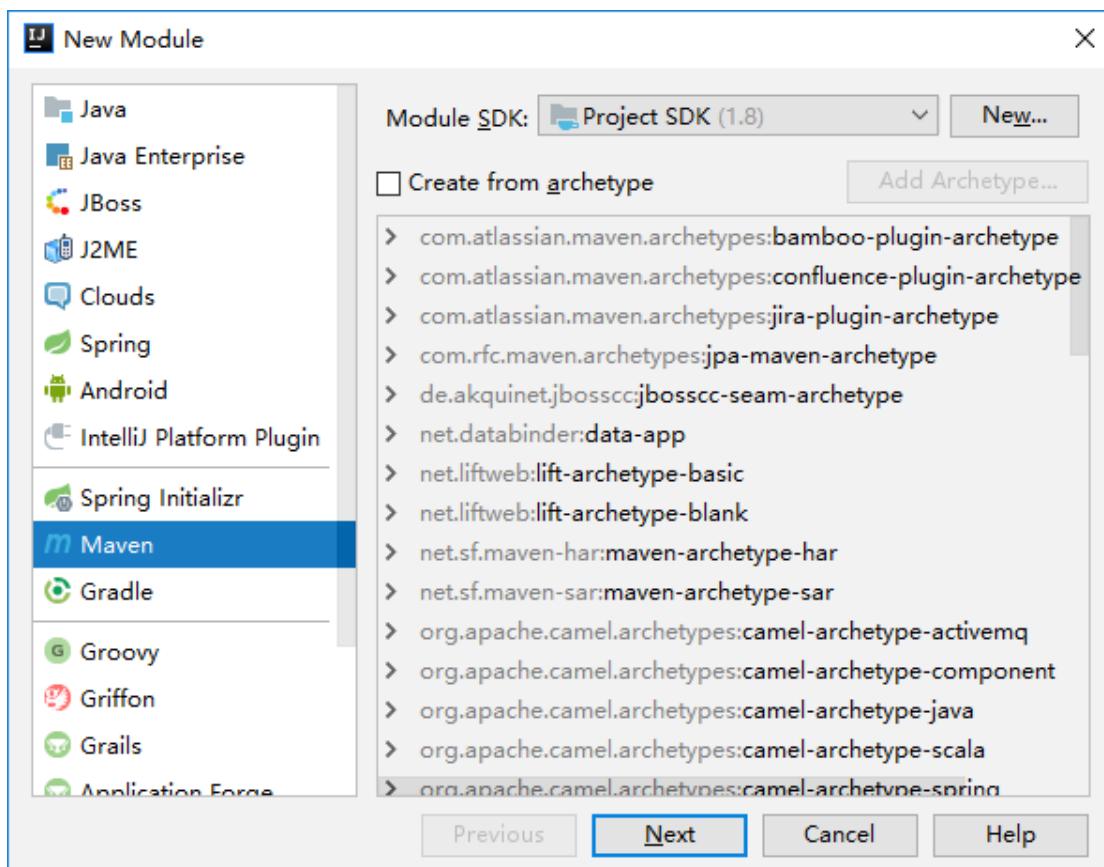
后台管理系统（采用分布式）

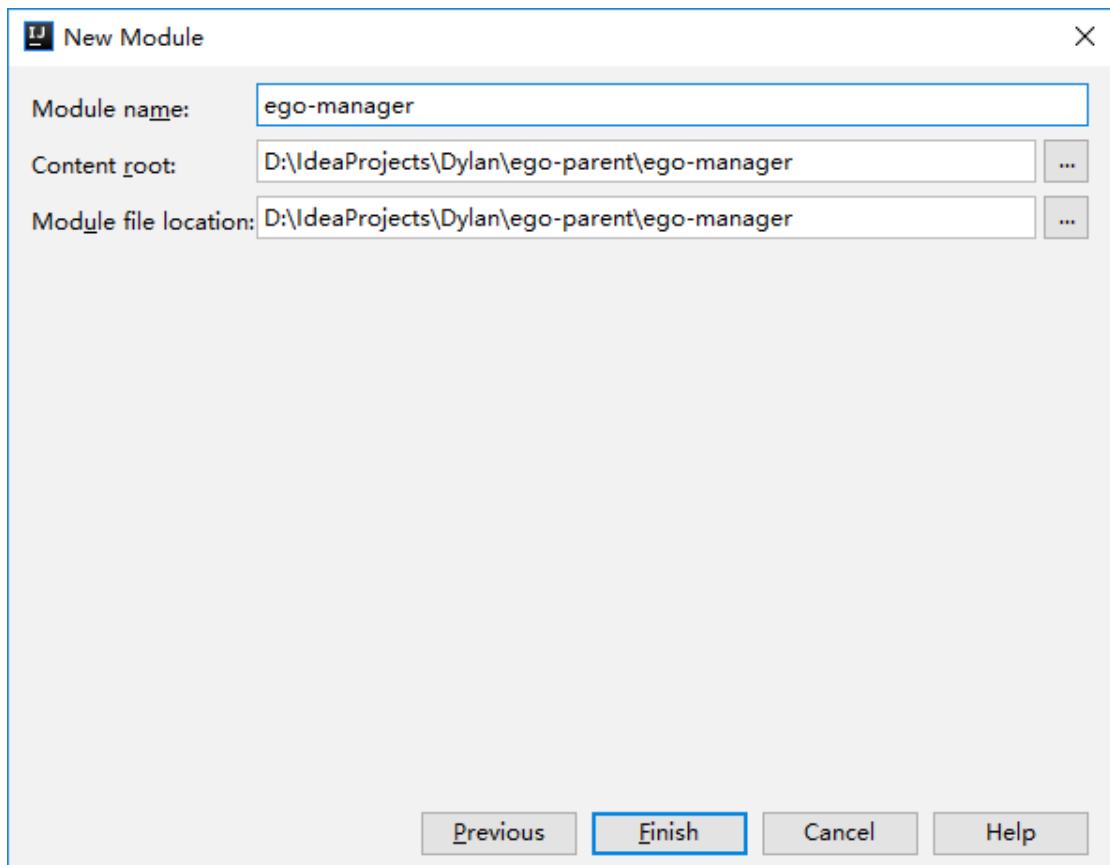
Maven 提供了聚合类型的项目，其本质就是一个分布式架构。

Maven 分布式架构-聚合类型，需要一个 parent 项目，manager 就是 parent 项目。

#### 3.1、创建 ego-manager 项目，pom 父模块







编辑 ego-manager 的 pom.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <parent>
        <artifactId>ego-parent</artifactId>
        <groupId>com.ego</groupId>
        <version>0.0.1-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>ego-manager</artifactId>

    <packaging>pom</packaging>

    <modules>
        <module>ego-manager-pojo</module>
        <module>ego-manager-mapper</module>
        <module>ego-manager-service</module>
        <module>ego-manager-web</module>
    </modules>

```

```

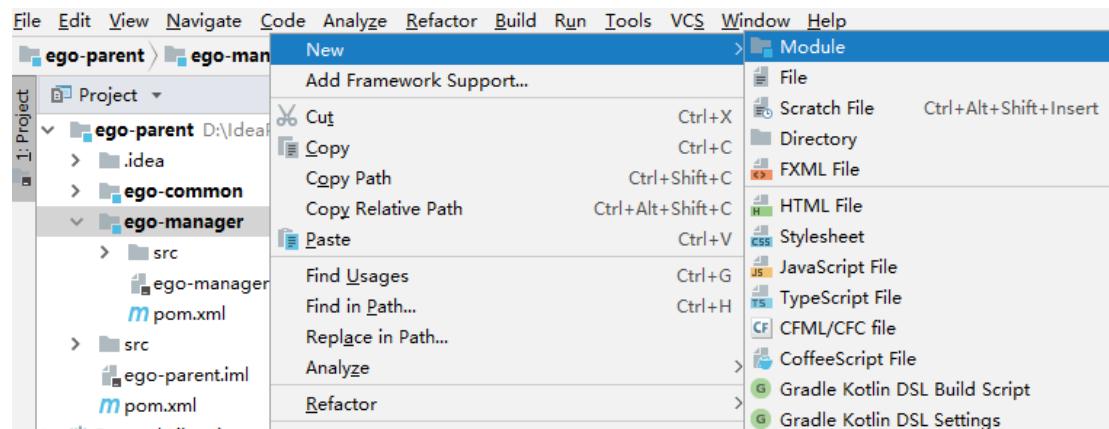
<dependencies>
    <!-- ego-common 依赖 -->
    <dependency>
        <groupId>com.ego</groupId>
        <artifactId>ego-common</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
</dependencies>

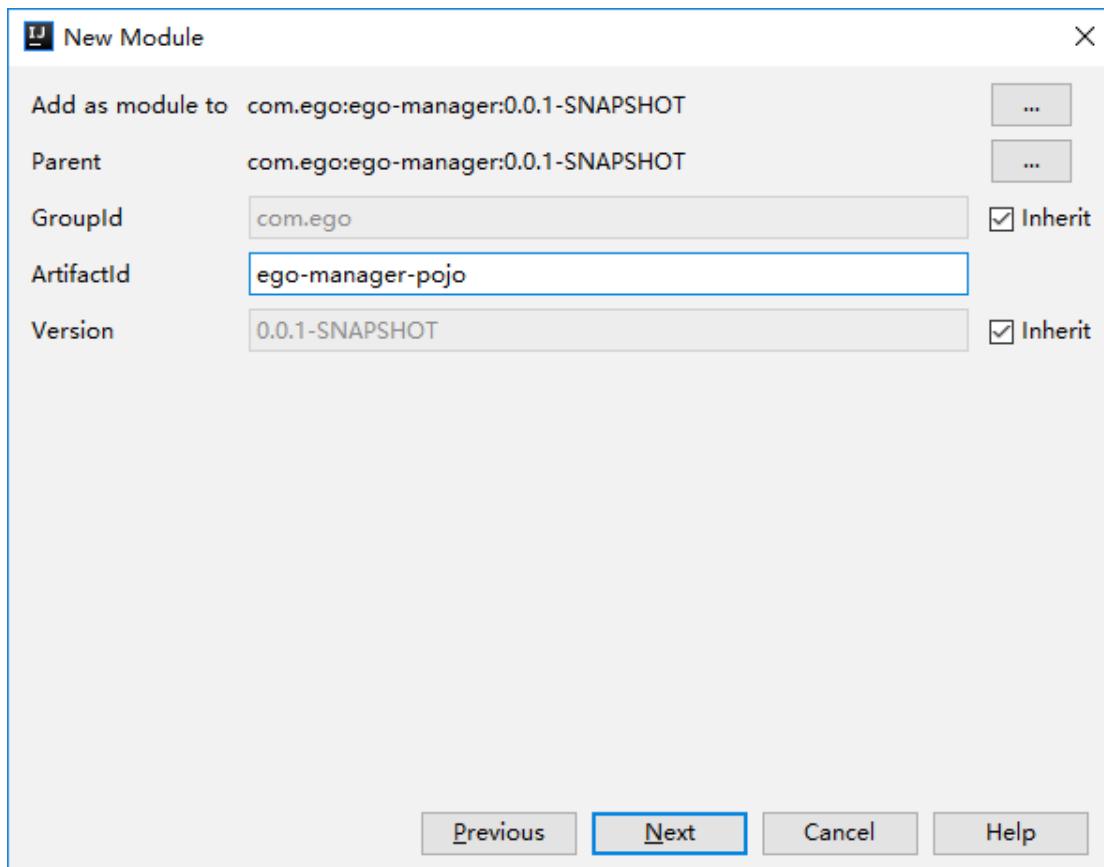
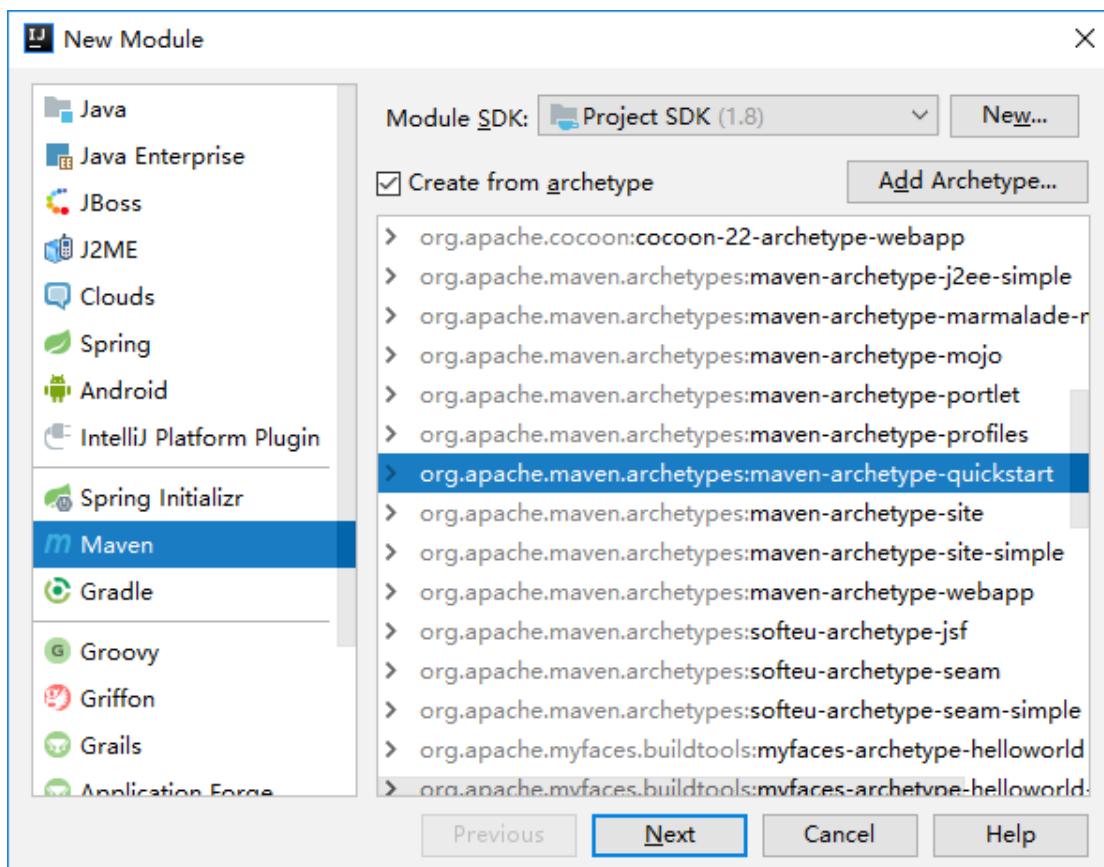
<!-- build 标签 常用于添加插件及编译配置 -->
<build>
    <!-- 读取配置文件 -->
    <resources>
        <resource>
            <directory>src/main/resources</directory>
        </resource>
        <resource>
            <directory>src/main/java</directory>
            <includes>
                <include>**/*.xml</include>
                <include>**/*.properties</include>
                <include>**/*.tld</include>
            </includes>
            <filtering>false</filtering>
        </resource>
    </resources>
</build>
</project>

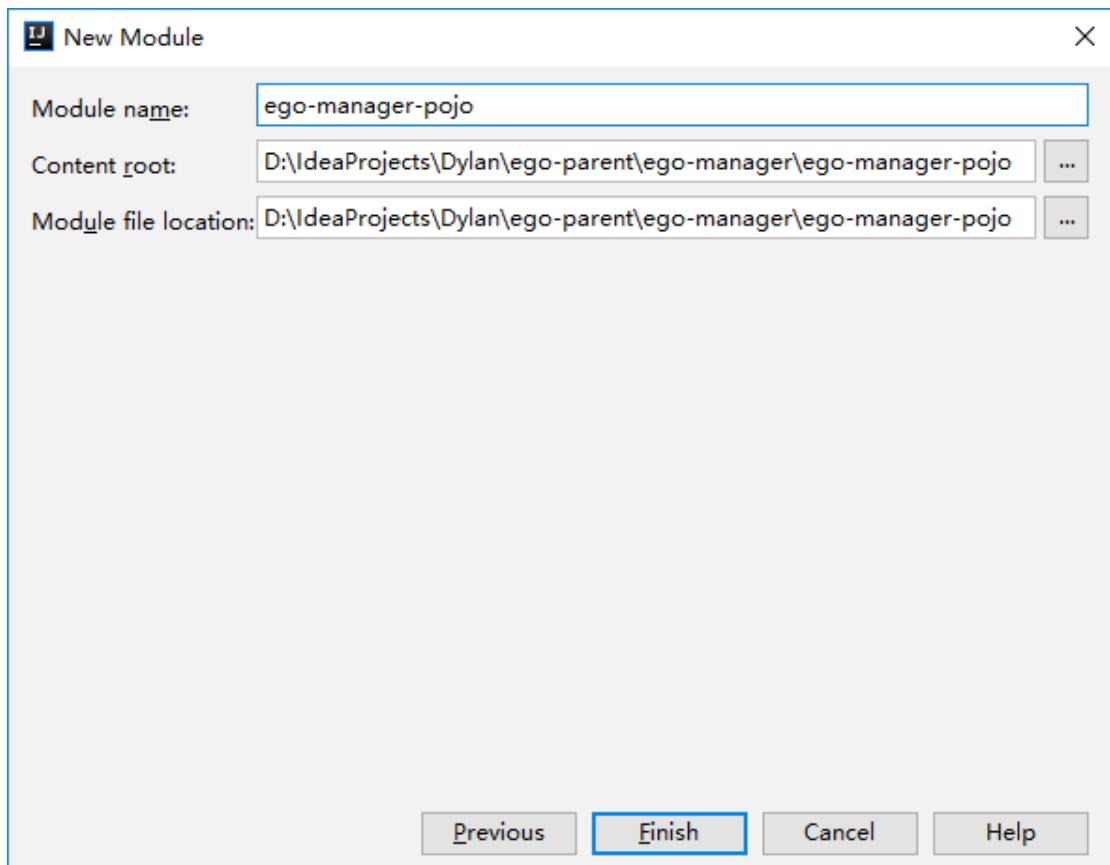
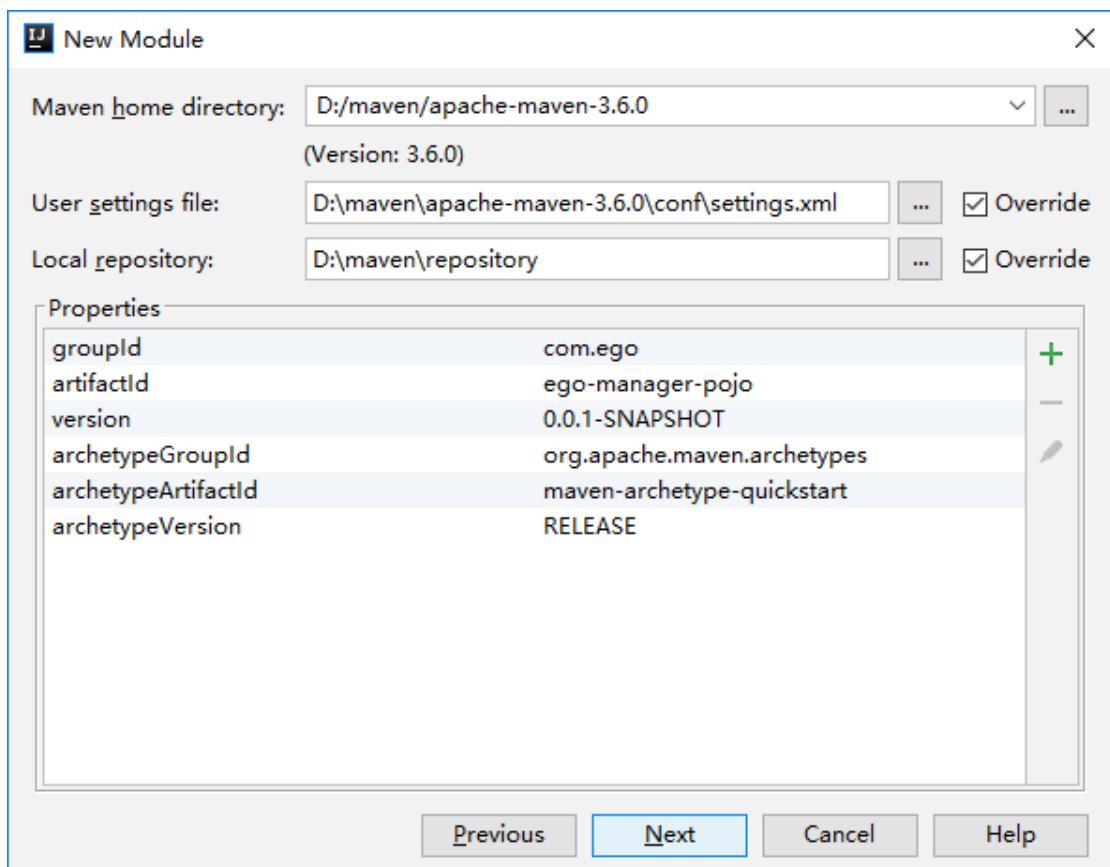
```

### 3.2、创建 ego-manager-pojo 项目，jar 子模块

在 ego-manager 项目上右键 new







编辑 ego-manager-pojo 的 pom.xml 配置文件

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <parent>
    <artifactId>ego-manager</artifactId>
    <groupId>com.ego</groupId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>ego-manager-pojo</artifactId>

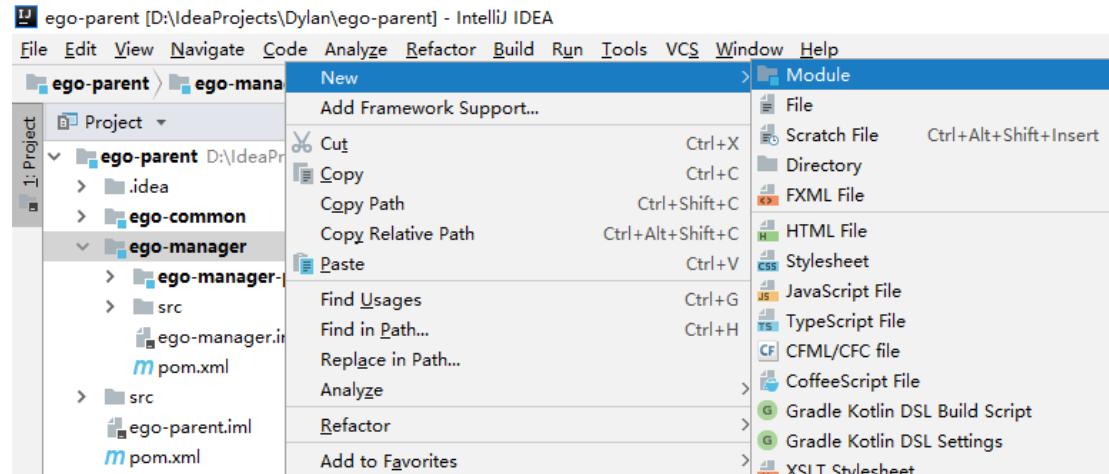
  <name>ego-manager-pojo</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

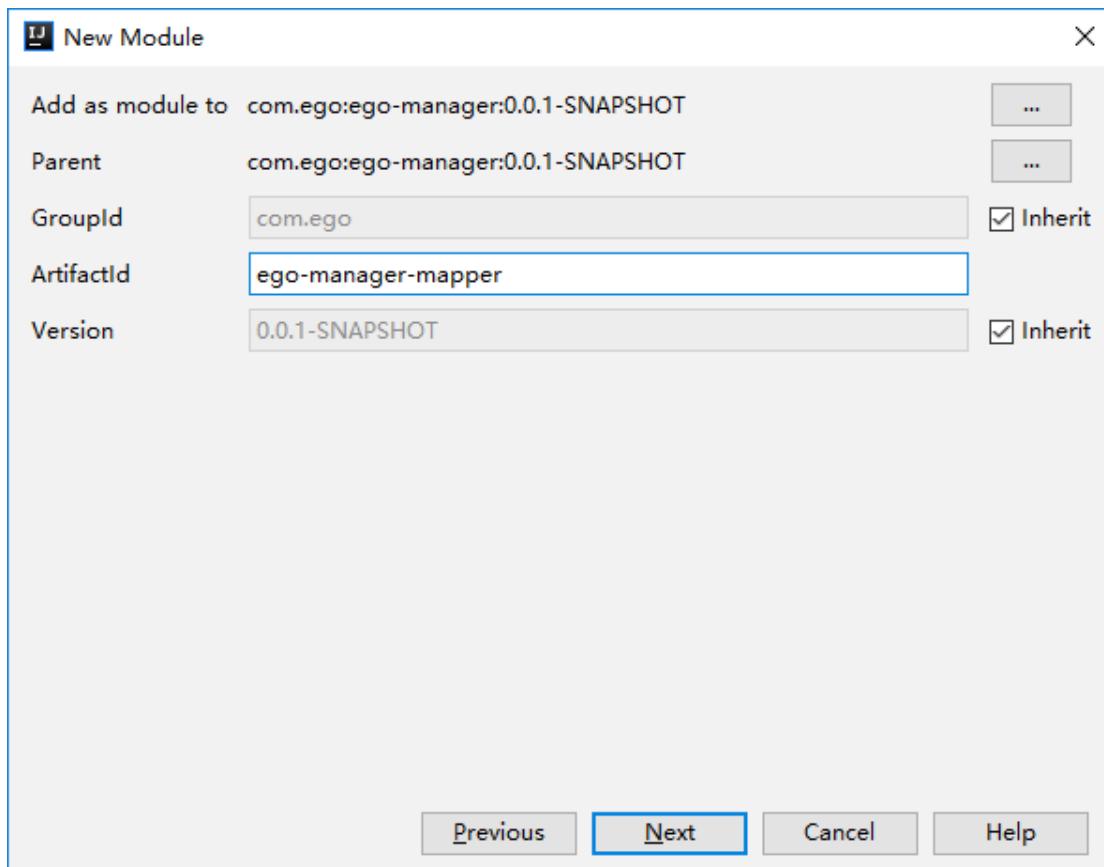
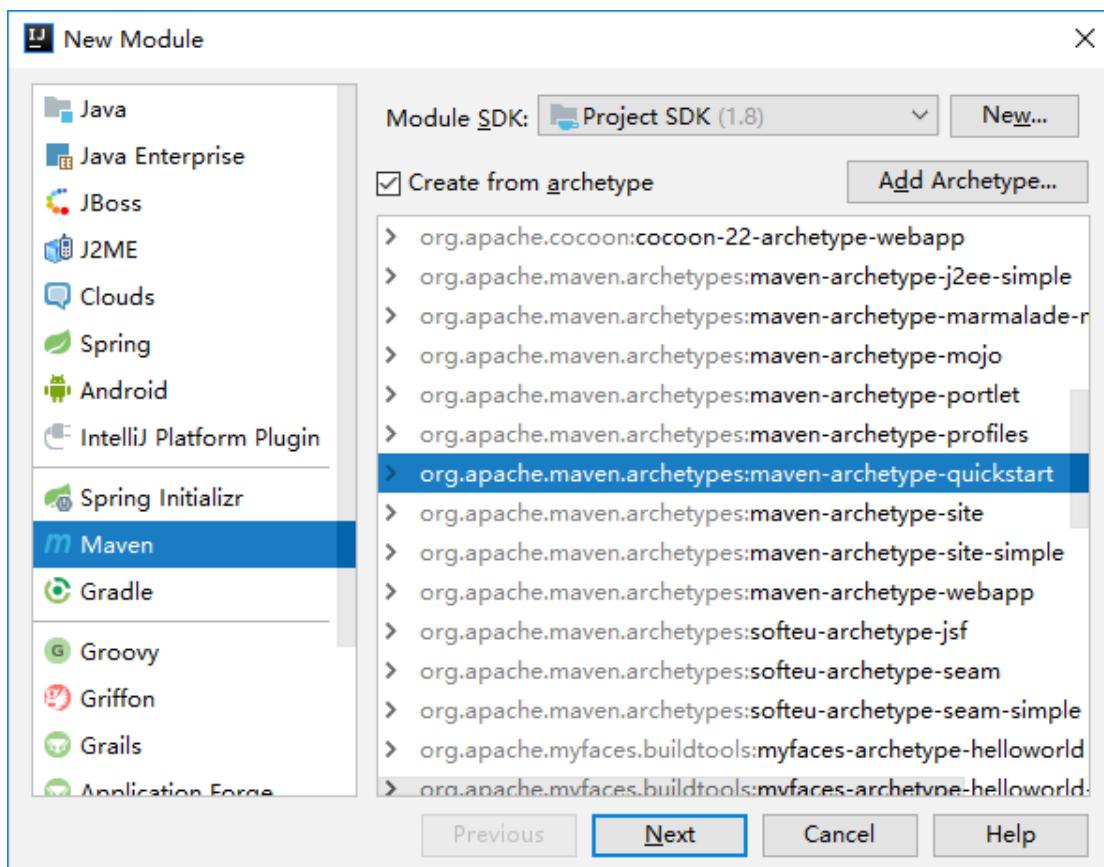
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
</project>

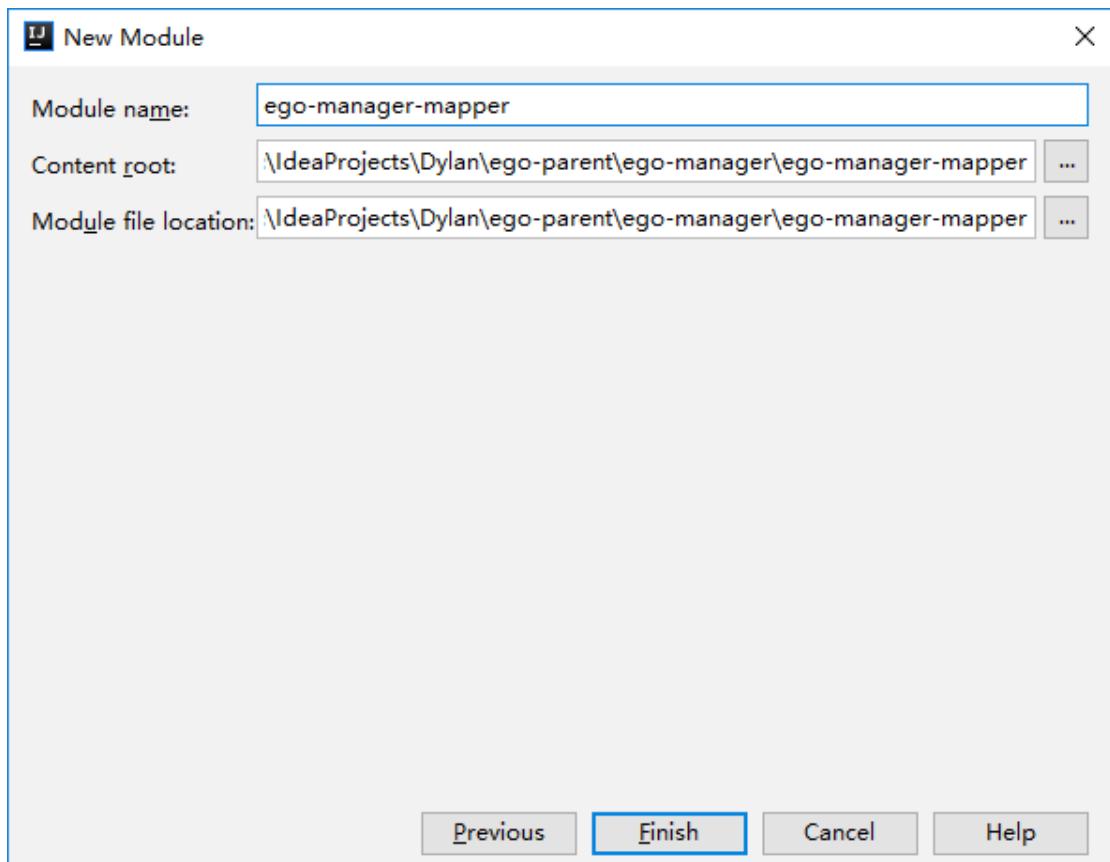
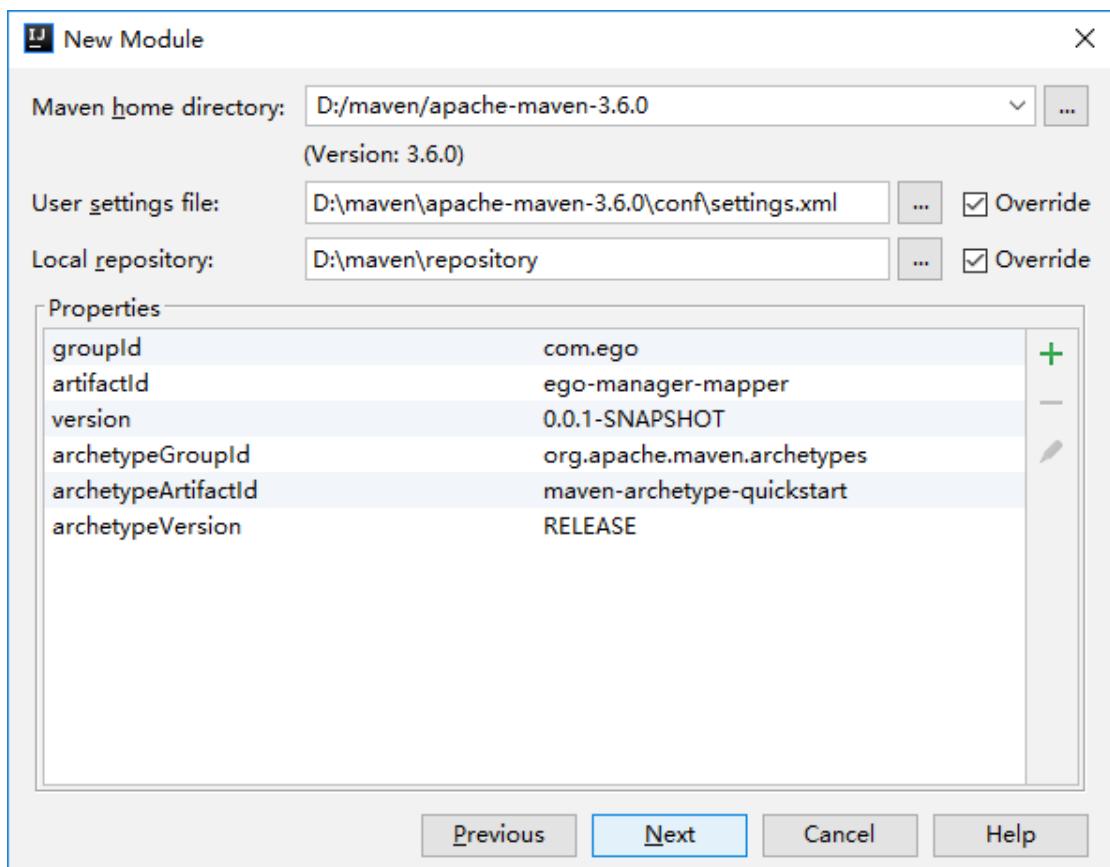
```

### 3.3、创建 ego-manager-mapper 项目，jar 子模块

在 ego-manager 项目上右键 new







编辑 ego-manager-mapper 的 pom.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <parent>
    <artifactId>ego-manager</artifactId>
    <groupId>com.ego</groupId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>ego-manager-mapper</artifactId>

  <name>ego-manager-mapper</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

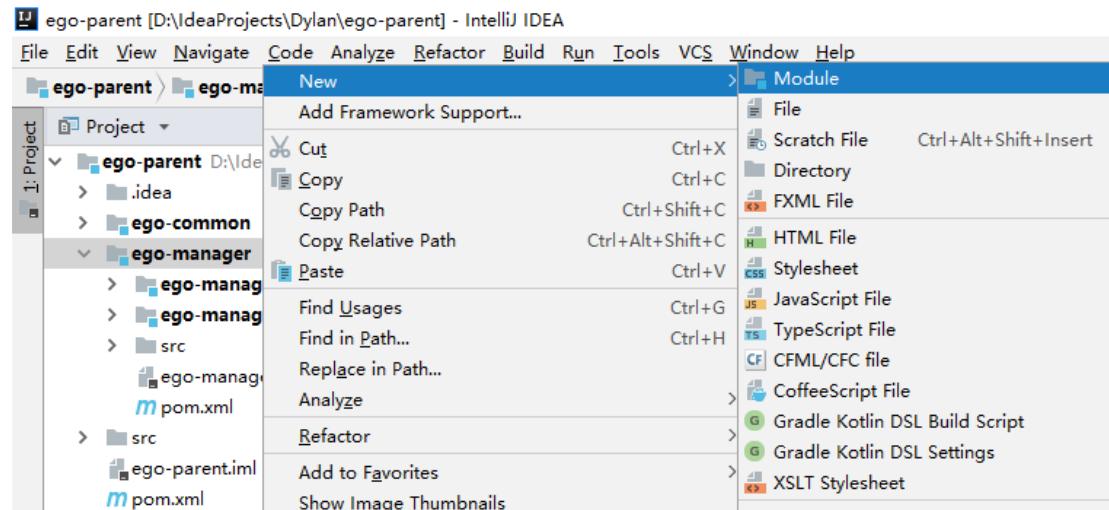
  <dependencies>
    <!-- ego-manager-pojo 依赖 -->
    <dependency>
      <groupId>com.ego</groupId>
      <artifactId>ego-manager-pojo</artifactId>
      <version>0.0.1-SNAPSHOT</version>
    </dependency>
    <!-- mybatis 依赖 -->
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
    </dependency>
    <dependency>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis-spring</artifactId>
    </dependency>
    <!-- mysql 依赖 -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
    </dependency>
  </dependencies>

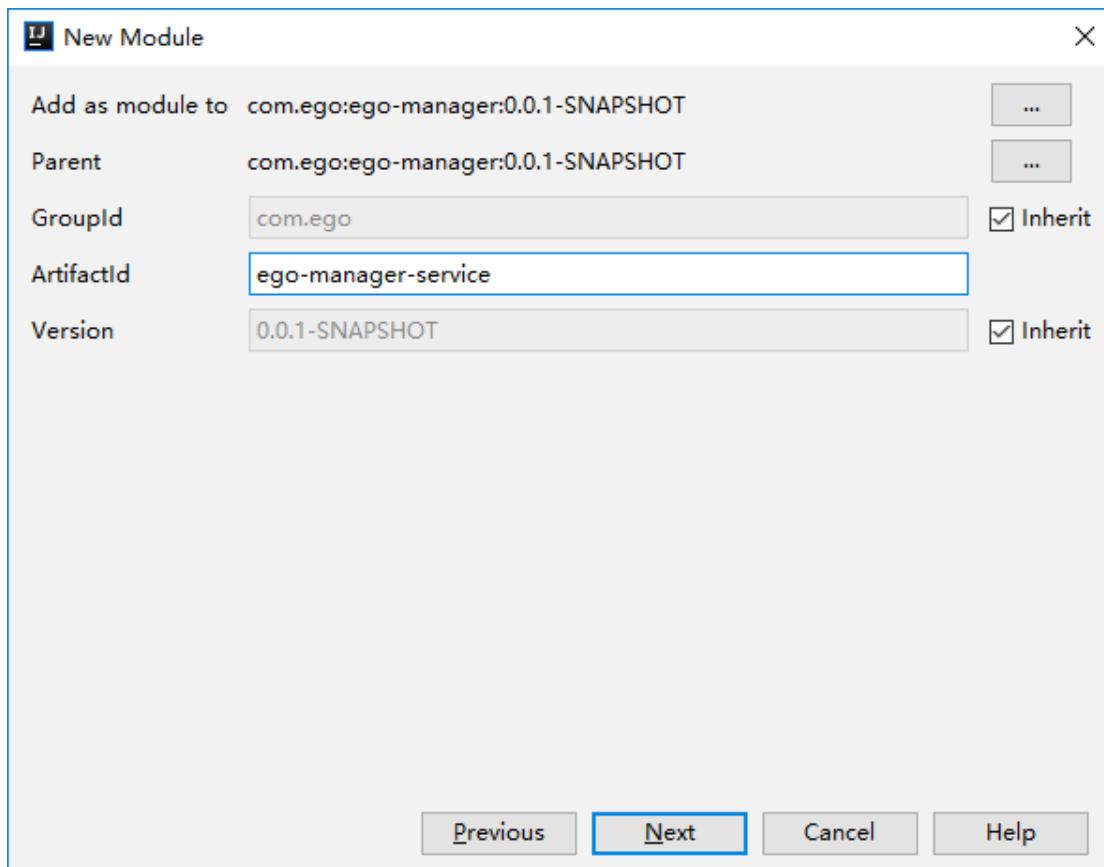
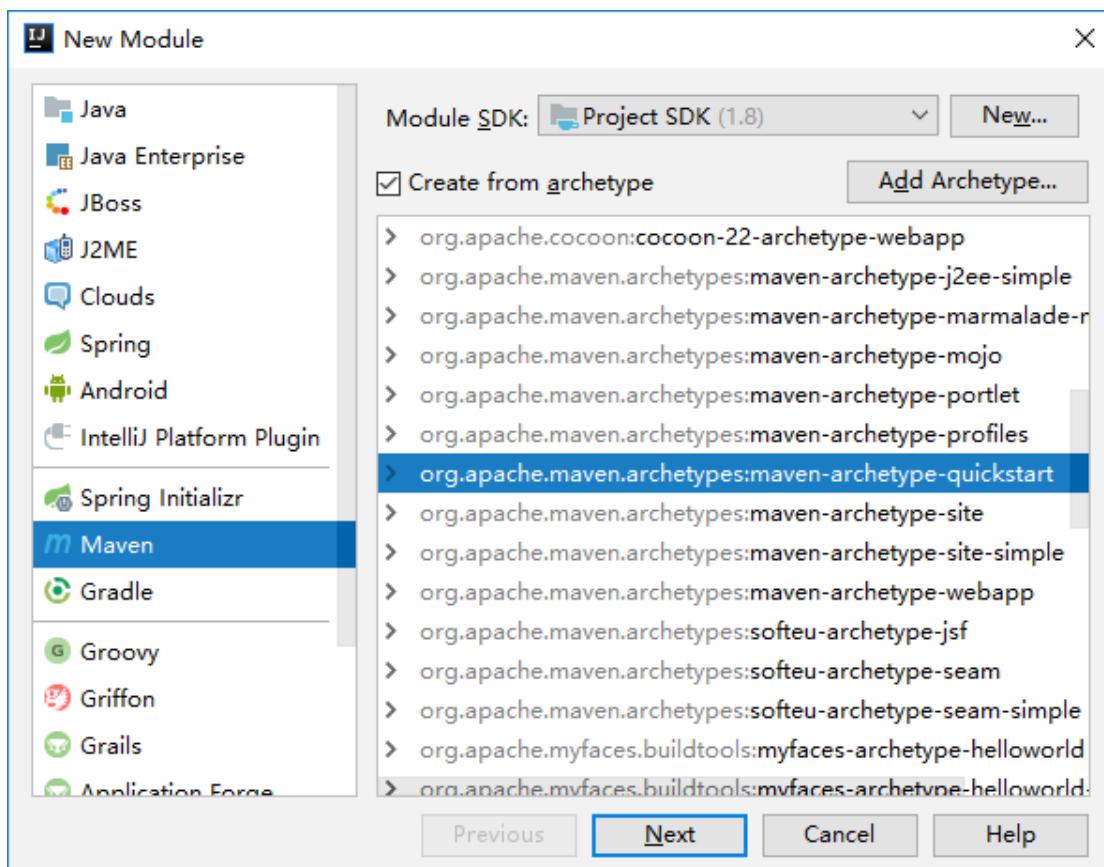
```

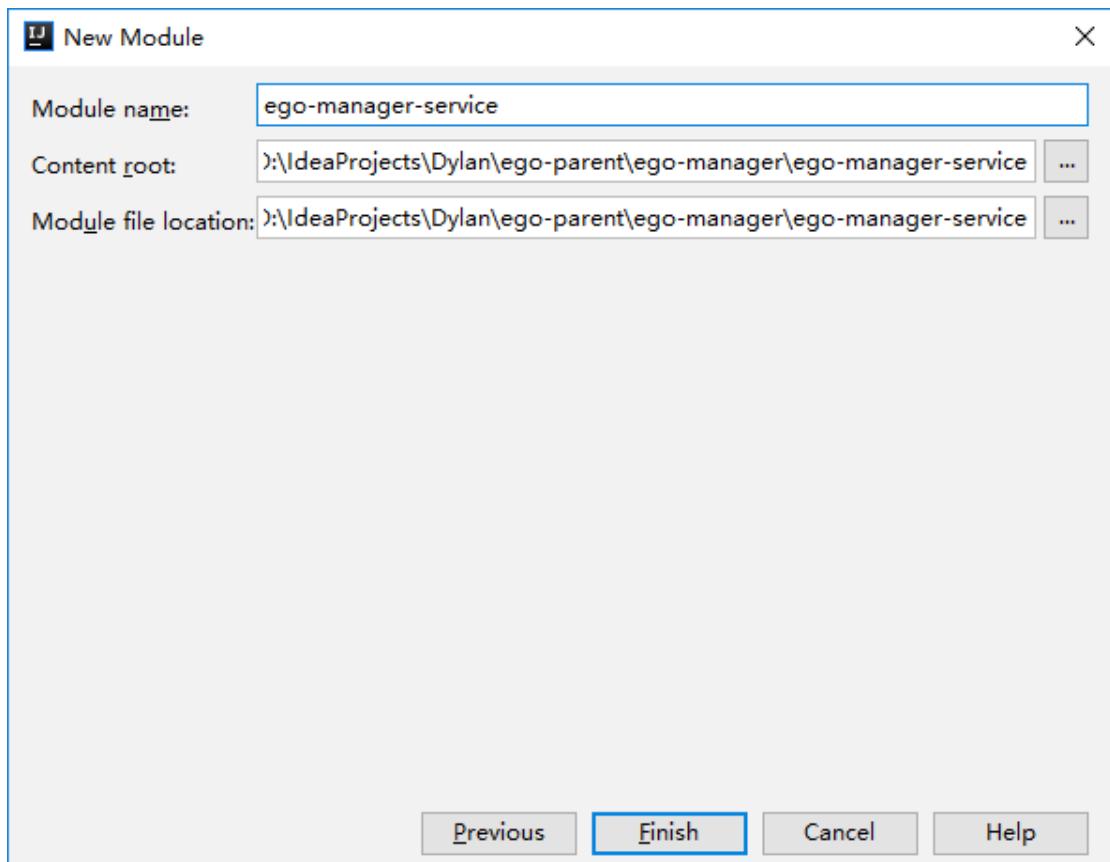
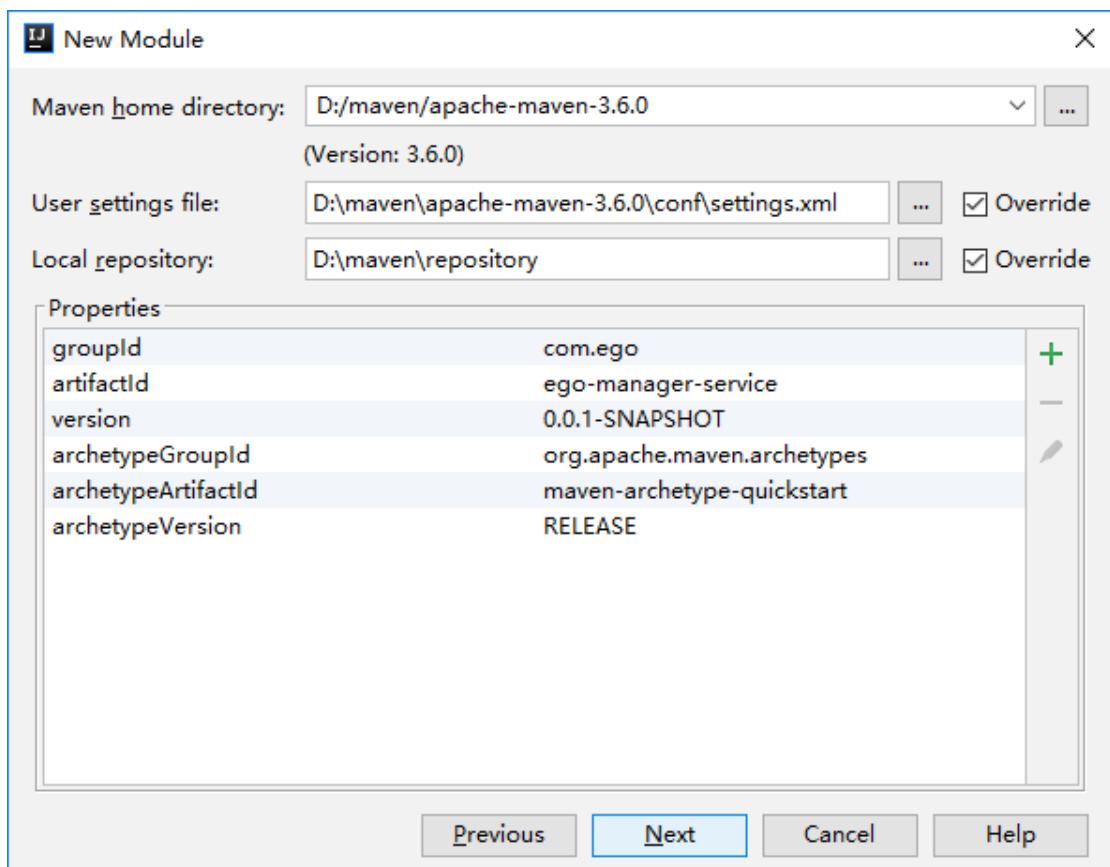
```
</dependency>
<!-- druid 连接池 -->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
</dependency>
<!-- 分页工具 -->
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper</artifactId>
</dependency>
</dependencies>
</project>
```

### 3.4、创建 ego-manager-service 项目，jar 子模块

在 ego-manager 项目上右键 new







编辑 ego-manager-service 的 pom.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <parent>
    <artifactId>ego-manager</artifactId>
    <groupId>com.ego</groupId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>ego-manager-service</artifactId>

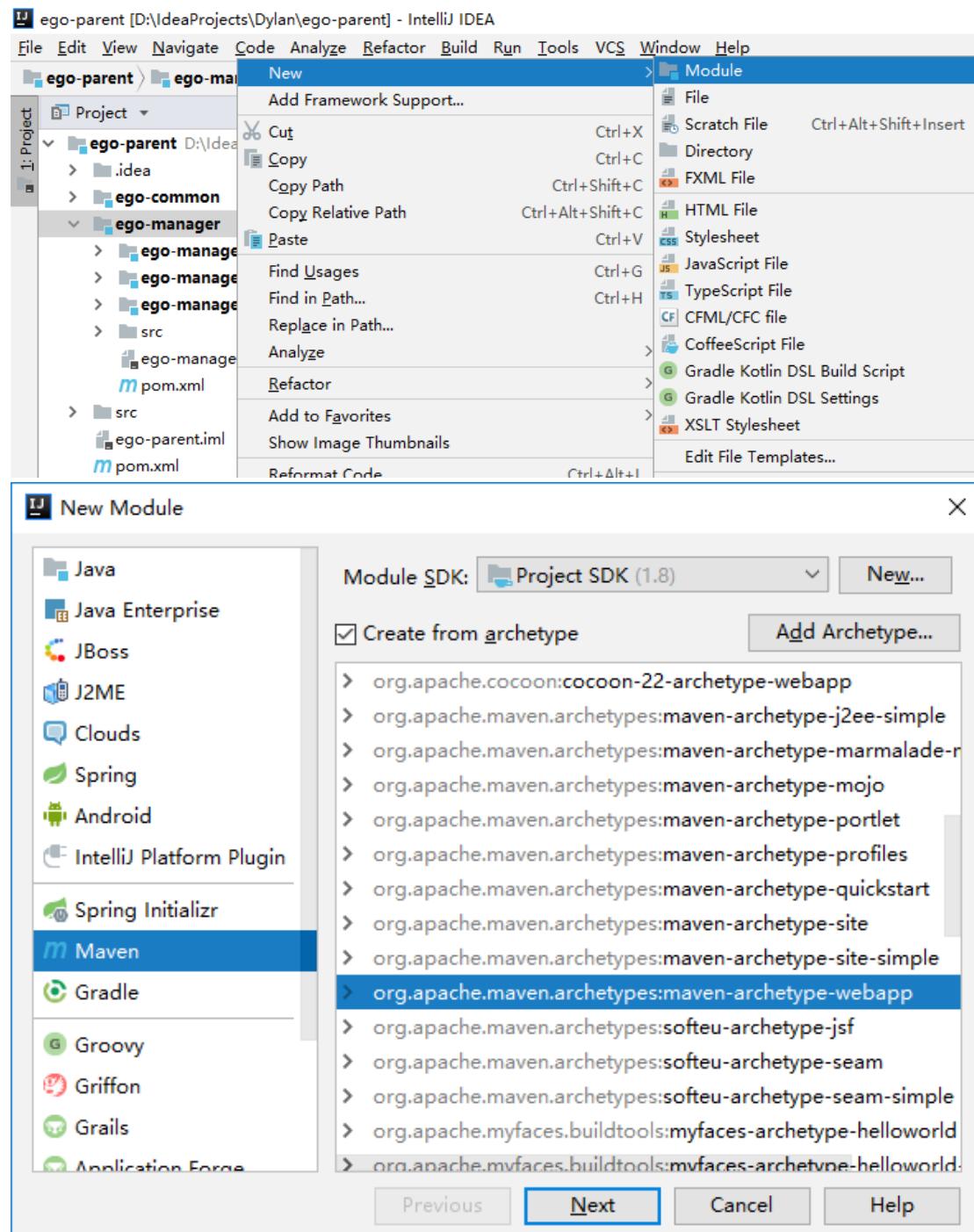
  <name>ego-manager-service</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <!-- ego-manager-mapper 依赖 -->
    <dependency>
      <groupId>com.ego</groupId>
      <artifactId>ego-manager-mapper</artifactId>
      <version>0.0.1-SNAPSHOT</version>
    </dependency>
    <!-- spring 依赖 -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
    </dependency>
    <!-- 日志打印 -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
    </dependency>
  </dependencies>
</project>
```

### 3.5、创建 ego-manager-web 项目，war 子模块

在 ego-manager 项目上右键 new



New Module

Add as module to com.ego:ego-manager:0.0.1-SNAPSHOT

Parent com.ego:ego-manager:0.0.1-SNAPSHOT

GroupId com.ego  Inherit

ArtifactId ego-manager-web

Version 0.0.1-SNAPSHOT  Inherit

Previous Next Cancel Help

New Module

Maven home directory: D:/maven/apache-maven-3.6.0  (Version: 3.6.0)

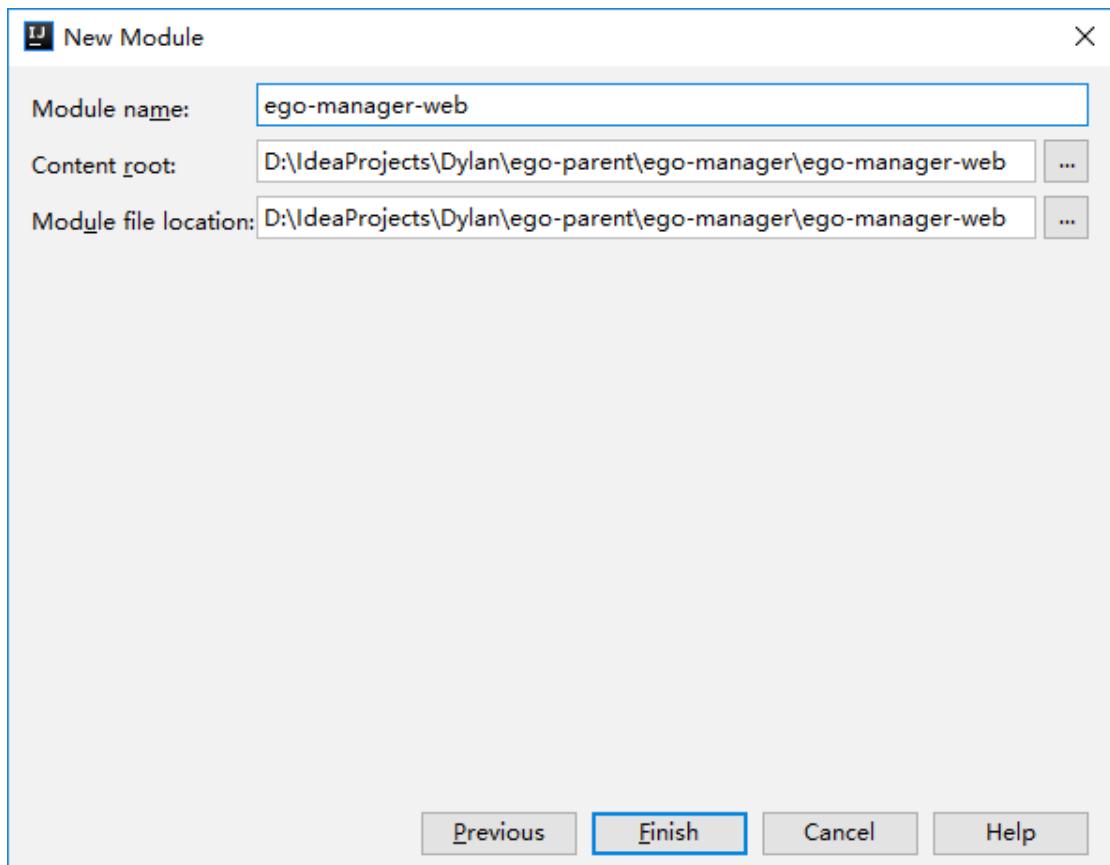
User settings file: D:\maven\apache-maven-3.6.0\conf\settings.xml   Override

Local repository: D:\maven\repository   Override

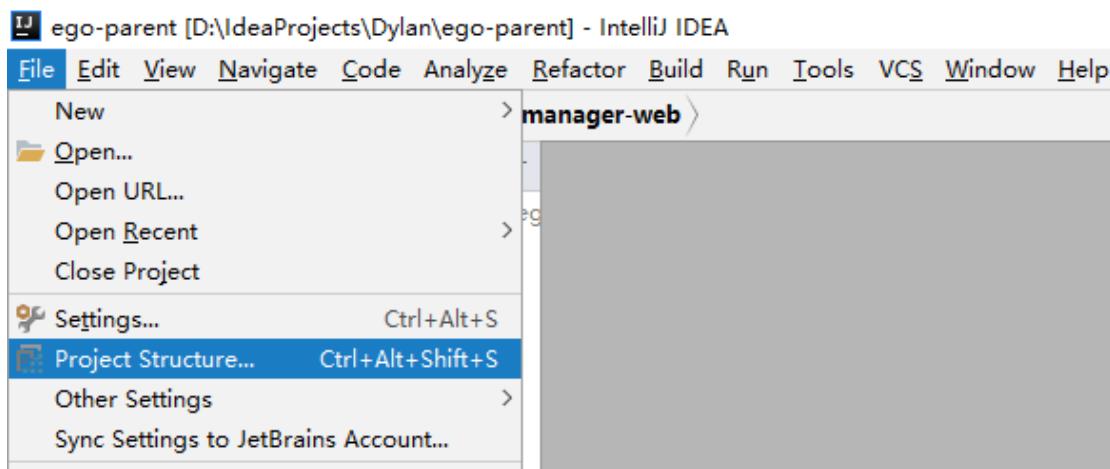
Properties

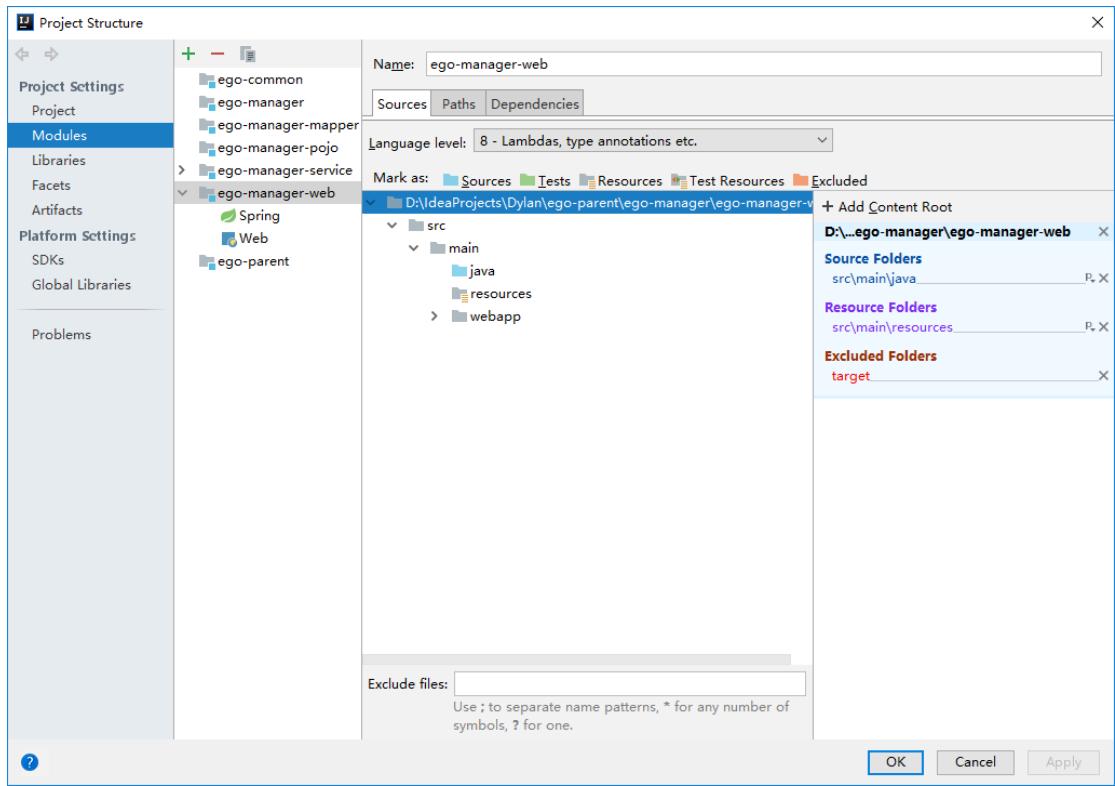
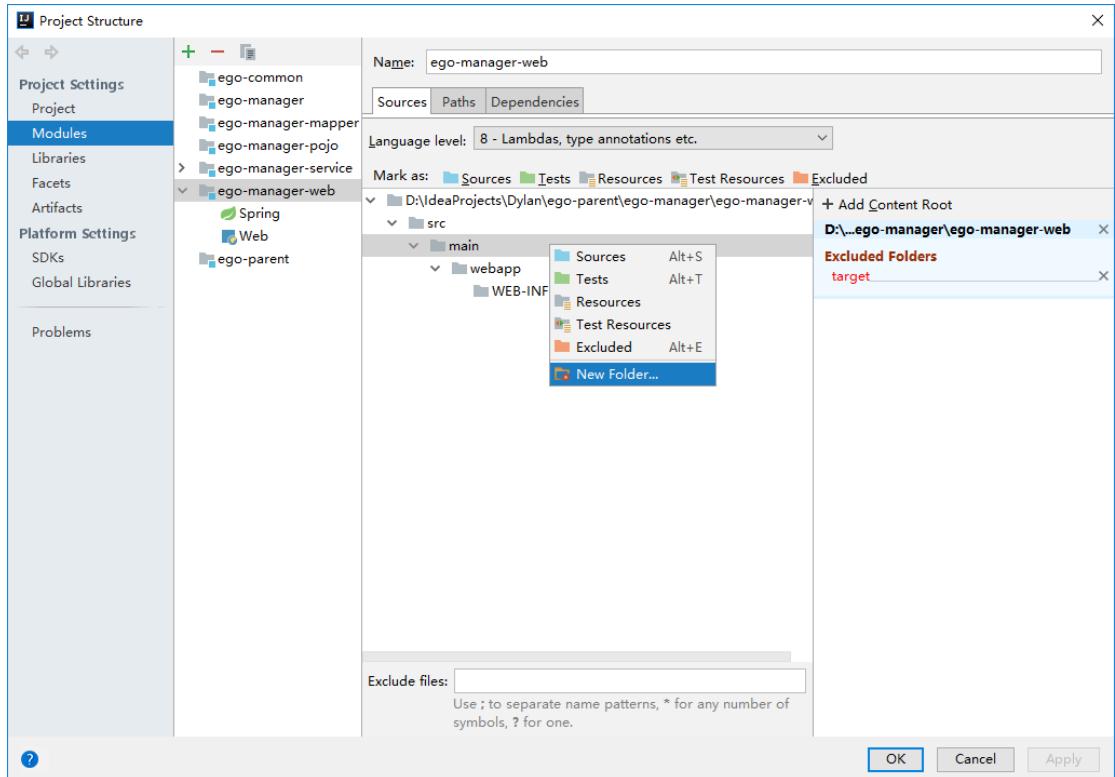
groupId	com.ego	<input type="button" value="+"/>
artifactId	ego-manager-web	<input type="button" value="-"/>
version	0.0.1-SNAPSHOT	<input type="button" value="edit"/>
archetypeGroupId	org.apache.maven.archetypes	
archetypeArtifactId	maven-archetype-webapp	
archetypeVersion	RELEASE	

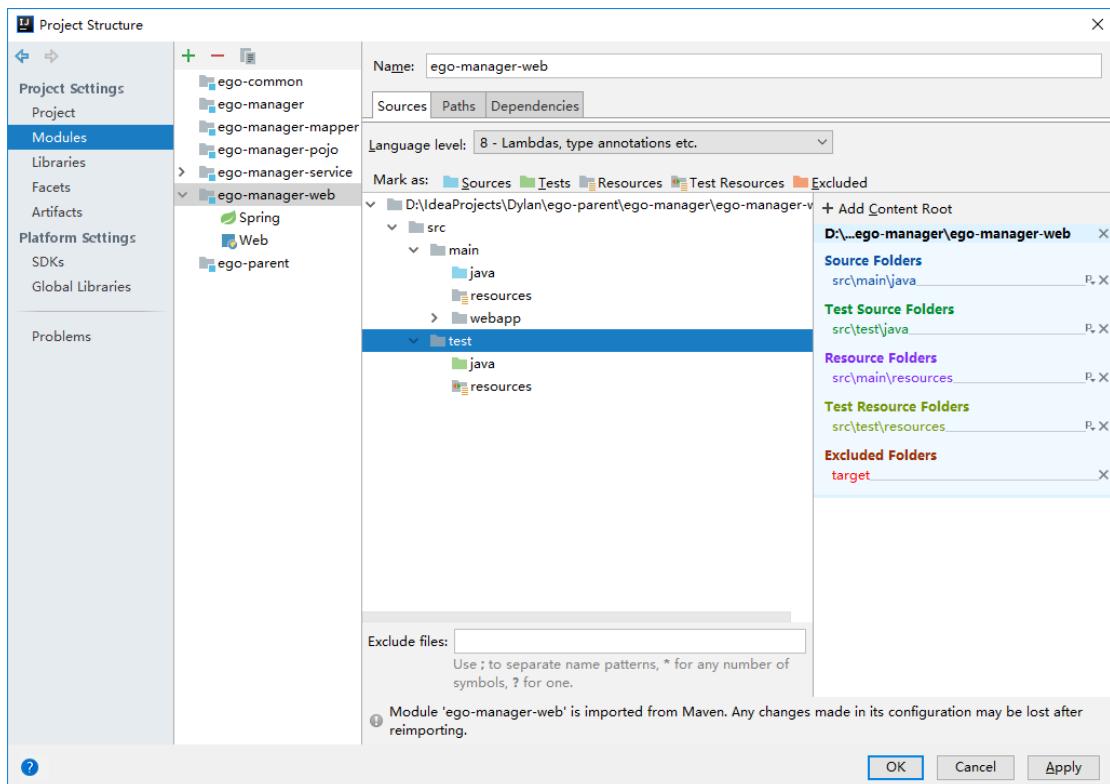
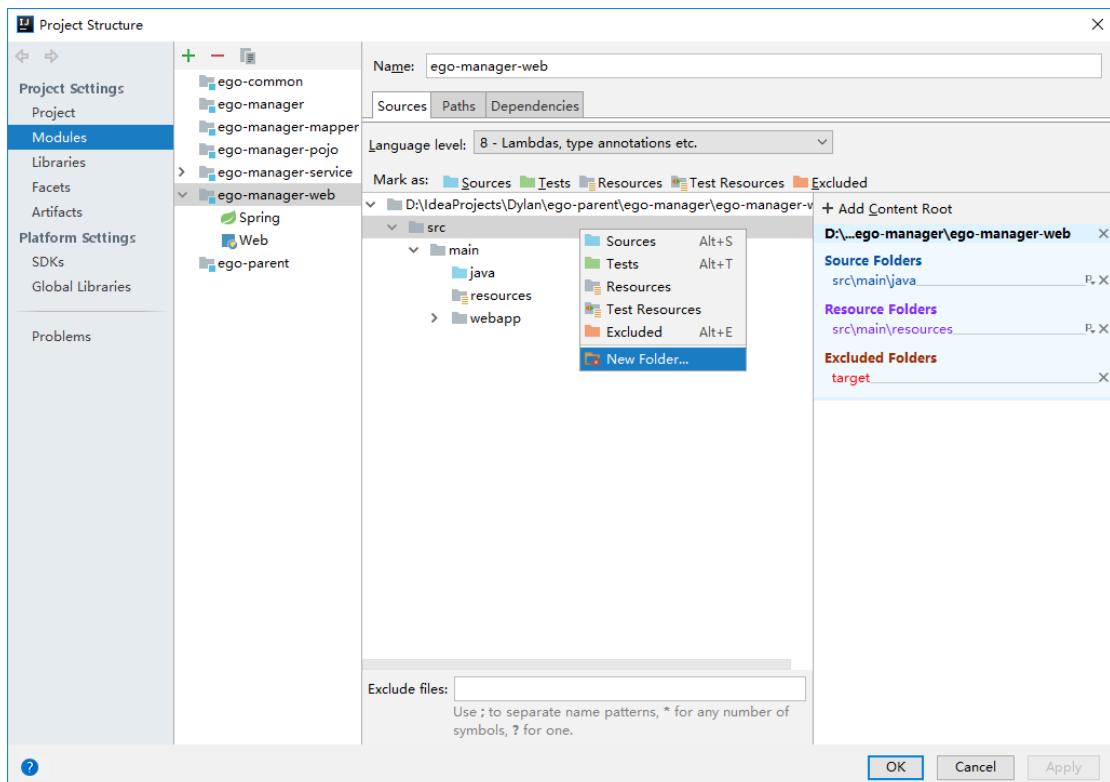
Previous Next Cancel Help



项目调整，为 web 项目添加 java 源文件目录和配置文件目录







编辑 ego-manager-web 的 pom.xml 配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<parent>
    <artifactId>ego-manager</artifactId>
    <groupId>com.ego</groupId>
    <version>0.0.1-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>

<artifactId>ego-manager-web</artifactId>

<packaging>war</packaging>

<name>ego-manager-web Maven Webapp</name>
<!-- FIXME change it to the project's website -->
<url>http://www.example.com</url>

<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>

<dependencies>
    <!-- ego-manager-service 依赖 -->
    <dependency>
        <groupId>com.ego</groupId>
        <artifactId>ego-manager-service</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
    <!-- spring 依赖 -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aspects</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-orm</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
    </dependency>
```

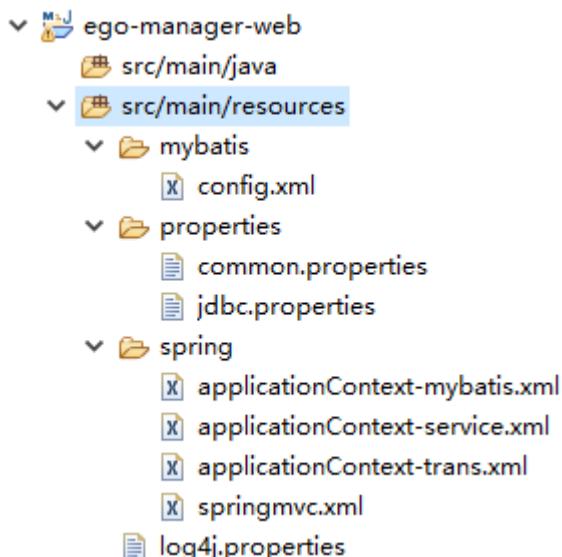
```
</dependency>
<!-- jsp 相关依赖 -->
<dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
</dependency>
<!-- servlet api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <scope>provided</scope>
</dependency>
<!-- Json 处理相关依赖 -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-annotations</artifactId>
</dependency>
<!-- 文件上传组件 -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
</dependency>
</dependencies>

<!-- build 标签 常用于添加插件及编译配置 -->
<build>
    <!-- 编译配置 指定 finalName 用于指定打包的包名 -->
    <finalName>ego-manager-web</finalName>
    <plugins>
        <!-- 添加 maven jetty 插件 -->
        <plugin>
            <groupId>org.mortbay.jetty</groupId>
            <artifactId>maven-jetty-plugin</artifactId>
            <version>6.1.26</version>
            <configuration>
                <scanIntervalSeconds>10</scanIntervalSeconds>
```

```
<!-- 项目访问路径 -->
<contextPath>/ego-manager-web</contextPath>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

## 4、其他配置文件

springmvc.xml  
applicationContext-mybatis.xml  
applicationContext-service.xml  
applicationContext-trans.xml  
jdbc.properties  
log4j.properties  
...



web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd"
  version="2.5">
  <display-name>ego-manager-web</display-name>

  <!-- spring后端配置 begin -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
```

```

<param-value>classpath:spring/applicationContext-*.xml</param-value>
</context-param>
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<!-- spring后端配置 end -->

<!-- springmvc后端配置 begin -->
<servlet>
    <servlet-name>springmvc</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:spring/springmvc.xml</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>springmvc</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
<!-- springmvc后端配置 begin -->

<!-- post方式提交乱码解决 begin -->
<filter>
    <filter-name>characterEncodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>utf-8</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>characterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<!-- post方式提交乱码解决 end -->
</web-app>

```

注意：springmvc 后端配置中的<url-pattern>/</url-pattern>填写 /，如果填写 /\* 将会在请求到 controller 之后分发请求到 jsp 时进行再次拦截，就会报 404 找不到页面的错误。

## 5、项目运行测试

Controller

```

package com.ego.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;

/**
 * @see(功能介绍) : 测试项目访问
 * @version(版本号) : 1.0
 * @author(创建人) : Dylan
 * @since : JDK 1.8
 */
@Controller
public class PageController {

    /**
     * 公共页面跳转 restful 风格
     * 比如: 前台传 welcome 就跳转至/WEB-INF/pages/welcome.jsp
     * 比如: 前台传 Login 就跳转至/WEB-INF/pages/Login.jsp
     * ...
     * @param page
     * @return
     */
    @RequestMapping("/{page}")
    public String page(@PathVariable String page) {
        System.out.println("pageController-----"+page);
        return page;
    }

}

```

welcome.jsp

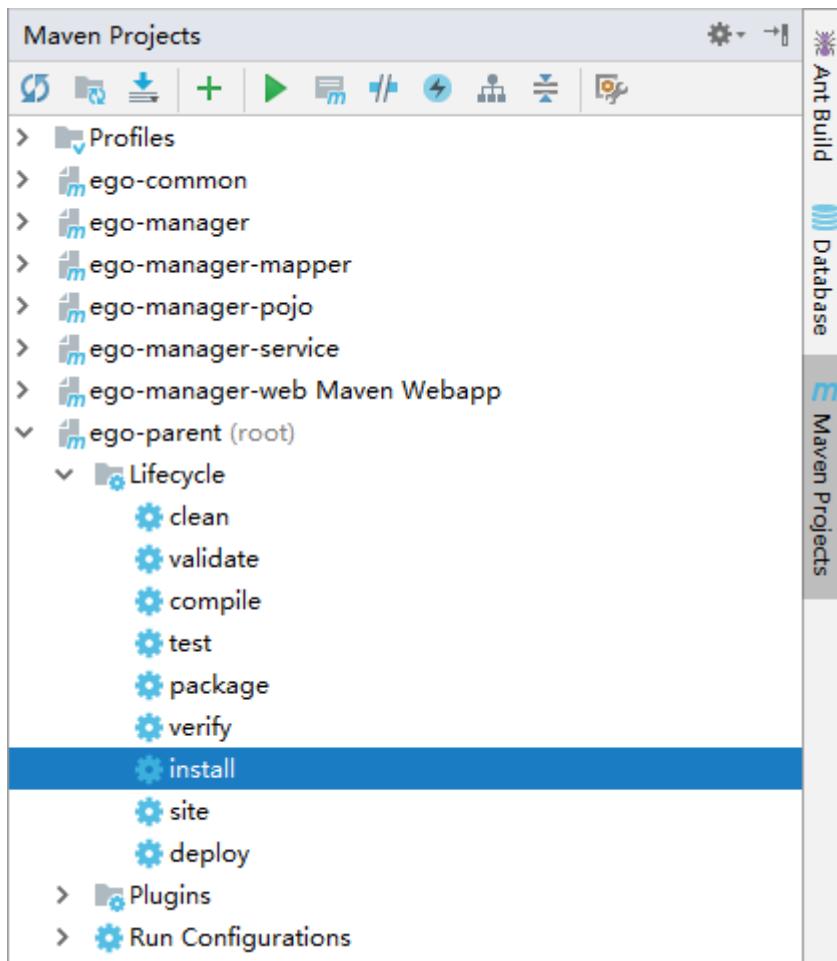
```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="ctx" value="${pageContext.request.contextPath }"></c:set>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ego 商城后台管理系统</title>
</head>
<body>
    <h1>${ctx}欢迎使用 ego 商城后台管理系统</h1>

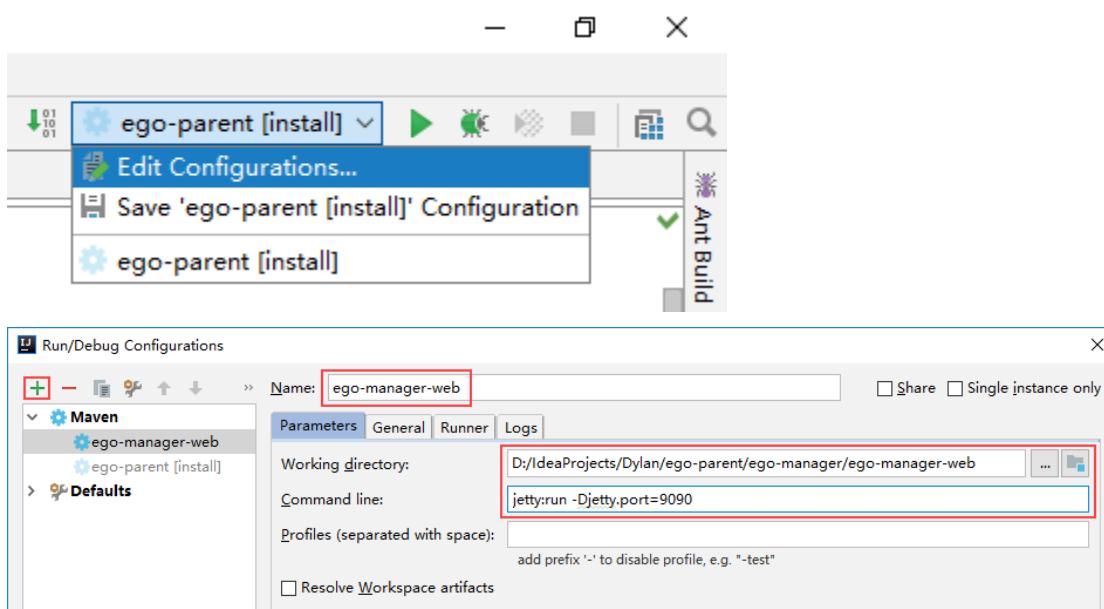
```

```
</body>  
</html>
```

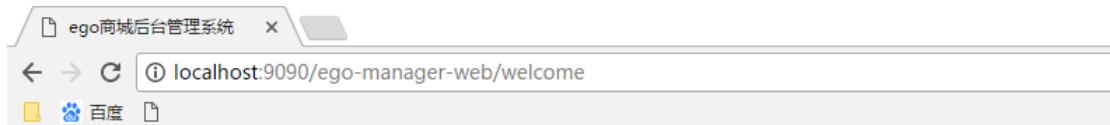
程序运行前先把 parent 项目 install 一下



最后配置 jetty 启动整个项目



测试结果如下：



/ego-manager-web欢迎使用ego商城后台管理系统