搭建

heima-leadnews

后端工程

VERSION - 1.0.0

|  |  |  |
| --- | --- | --- |
| 文档版本说明 | | |
| 文件状态  [ ] 草稿  [ √ ] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本 |  |
| 作者 | miukoo |
| 完成日期 | 2019-05-31 |

目录

[目录 2](#_Toc10476257)

[1. 概述 4](#_Toc10476258)

[2. 创建父项目[heima-leadnews] 4](#_Toc10476259)

[1.1 创建工程 5](#_Toc10476260)

[1.2 基本设置 6](#_Toc10476261)

[1.2.1 设置本地仓库[可选] 6](#_Toc10476262)

[1.2.2 设置项目编码 7](#_Toc10476263)

[1.2.3 删除多余目录 7](#_Toc10476264)

[1.2.4 pom.xml配置 7](#_Toc10476265)

[3. 创建common子项目[heima-leadnews-common] 11](#_Toc10476266)

[2.1 创建工程 12](#_Toc10476268)

[2.2 创建结构 15](#_Toc10476269)

[4. 创建utils子项目[heima-leadnews-utils] 16](#_Toc10476270)

[3.1 创建工程 16](#_Toc10476272)

[3.2 创建结构 19](#_Toc10476273)

[5. 创建model子项目[heima-leadnews-model] 20](#_Toc10476274)

[4.1 创建工程 20](#_Toc10476276)

[4.2 创建结构 24](#_Toc10476277)

[6. 创建apis子项目[heima-leadnews-apis] 24](#_Toc10476278)

[5.1 创建工程 25](#_Toc10476280)

[5.2 创建结构 28](#_Toc10476281)

[7. 创建测试子项目[heima-leadnews-tests] 29](#_Toc10476282)

[6.1 创建工程 29](#_Toc10476284)

[6.2 创建结构 37](#_Toc10476285)

[6.3 添加配置/代码 37](#_Toc10476286)

[6.3.1 添加POM依赖 37](#_Toc10476287)

[6.3.2 添加多环境配置 38](#_Toc10476288)

[6.3.3 添加配置文件 38](#_Toc10476289)

[6.3.4 添加代码 39](#_Toc10476290)

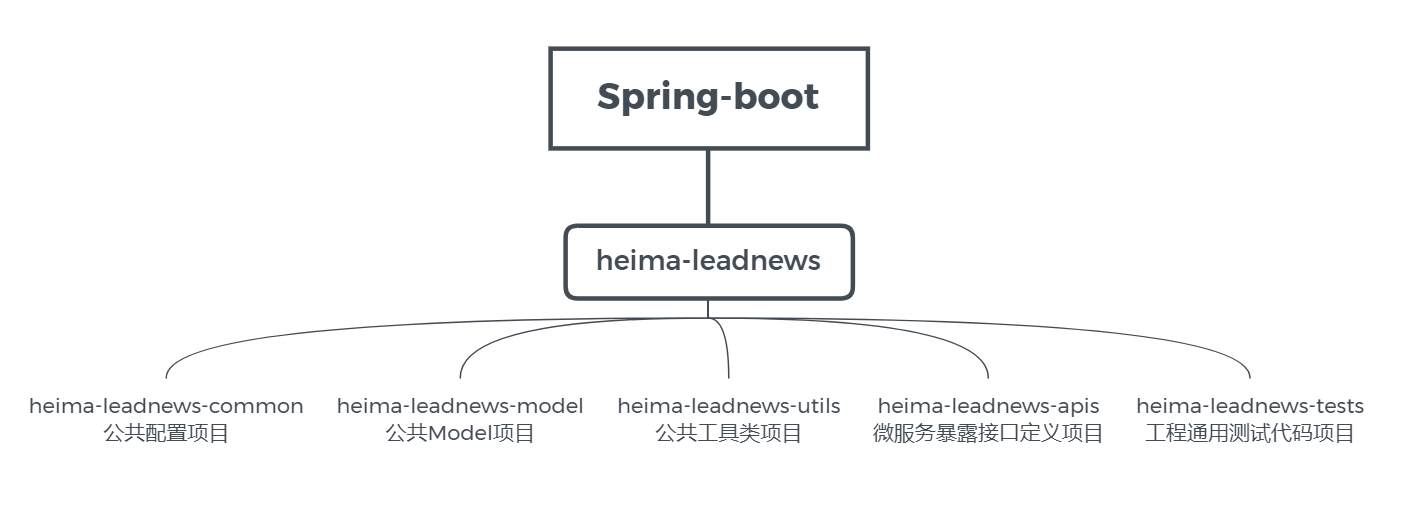
[6.3.5 项目运行 41](#_Toc10476291)

# 概述

工程基于Spring-boot 2.1.5.RELEASE 版本构建，工程父项目为heima-leadnews，并通过继承方式集成Spring-boot。

父项目下分5个公共子项目：

* heima-leadnews-common : 是整个工程的配置核心，包括所有集成三方框架的配置定义，比如redis、kafka等。除此之外还包括项目每个模块及整个项目的常量定义;
* heima-leadnews-model ：项目中用到的Dto、Pojo、Mapper、Enums定义工程;
* heima-leadnews-utils : 工程公用工具类项目，包含加密/解密、Date、JSON等工具类;
* heima-leadnew-apis : 整个项目微服务暴露的接口的定义项目，按每个模块进行子包拆分;
* heima-leadnew-tests : 用于存放工程中通用的测试用例；



# 环境

项目依赖环境（需提前安装好）：

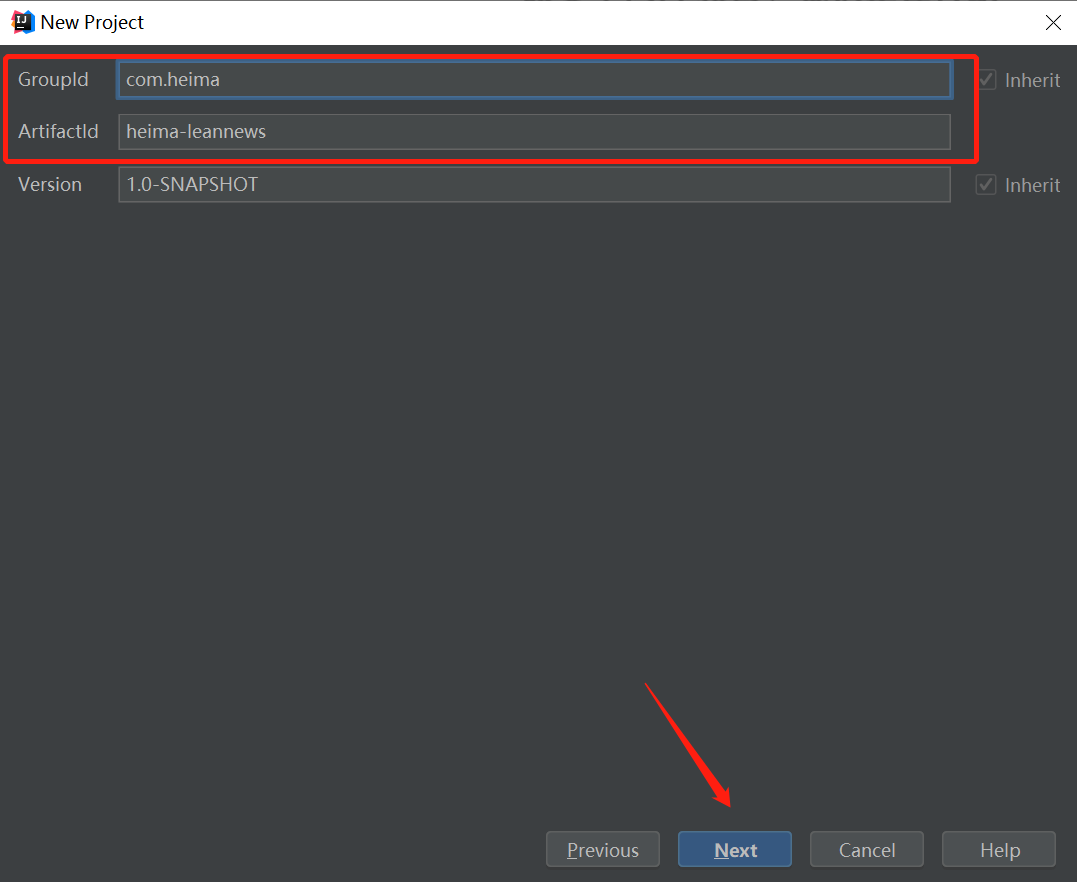
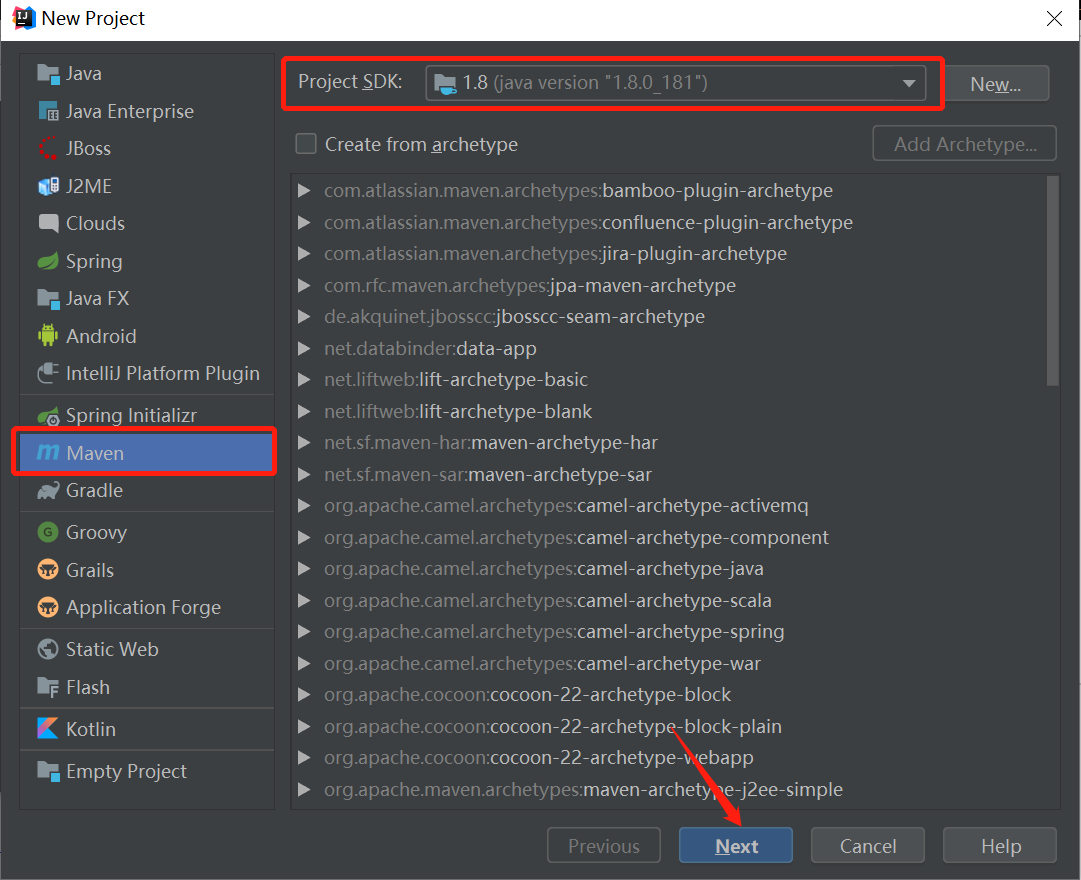
* JDK1.8
* Intellij Idea
* Tomcat 8.5
* Git

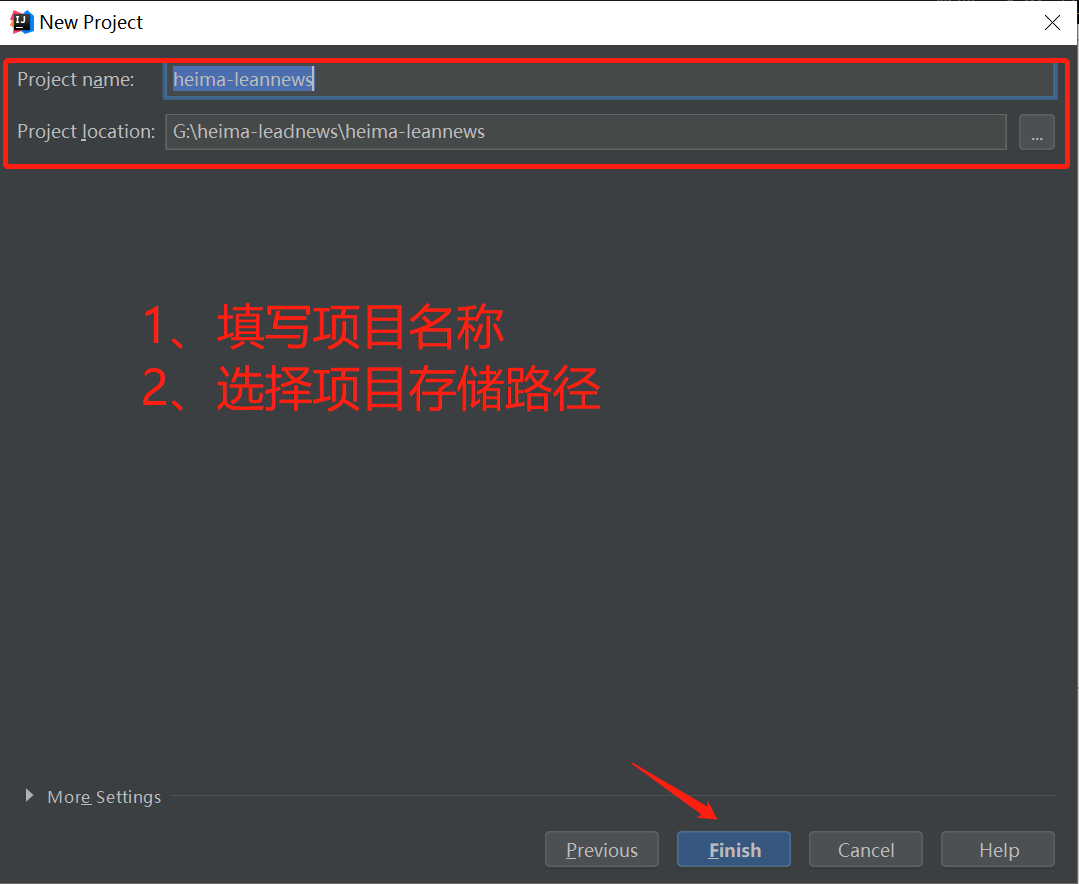
# 创建父项目[heima-leadnews]

父工程，用于管理工程中的依赖。



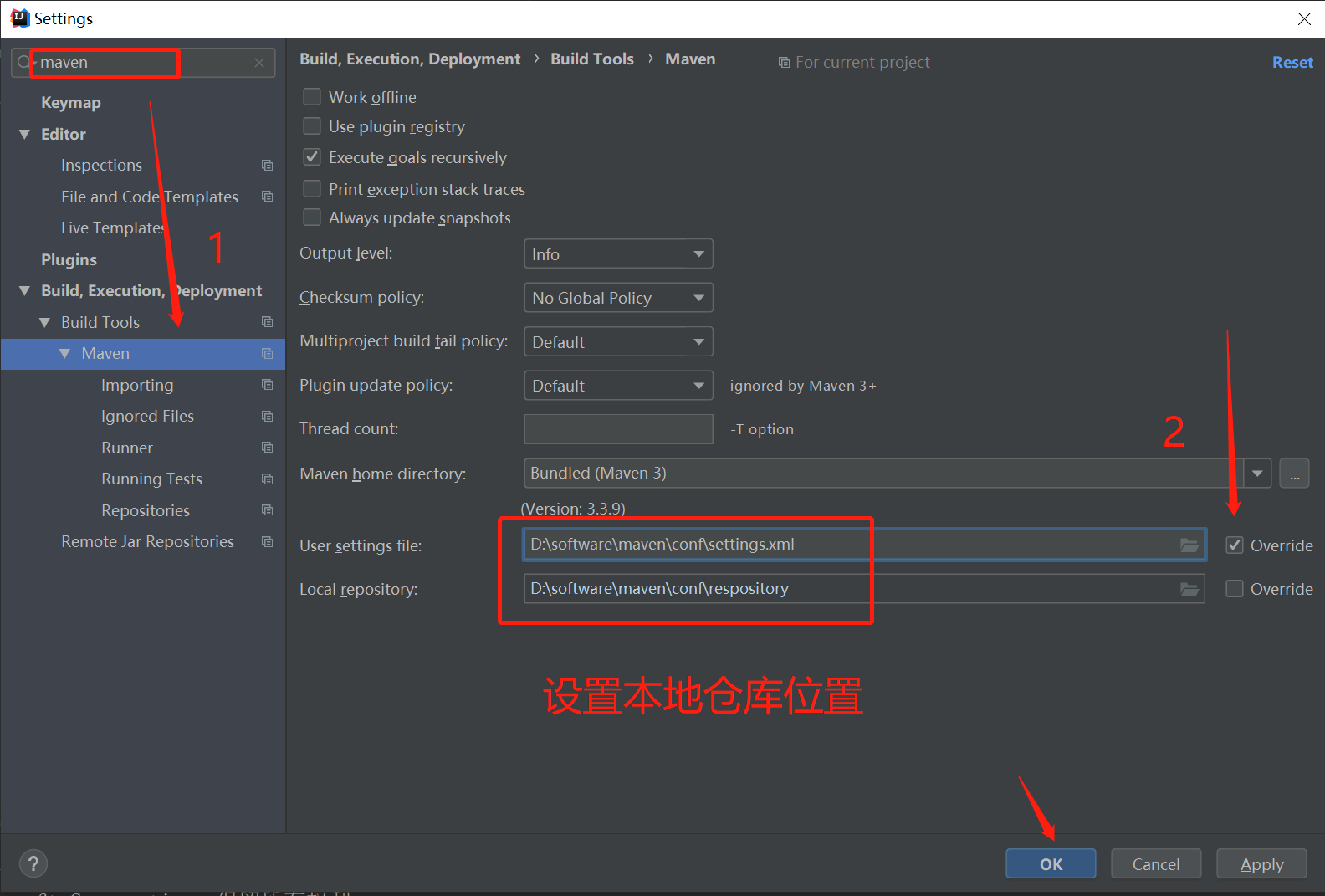
## 创建工程



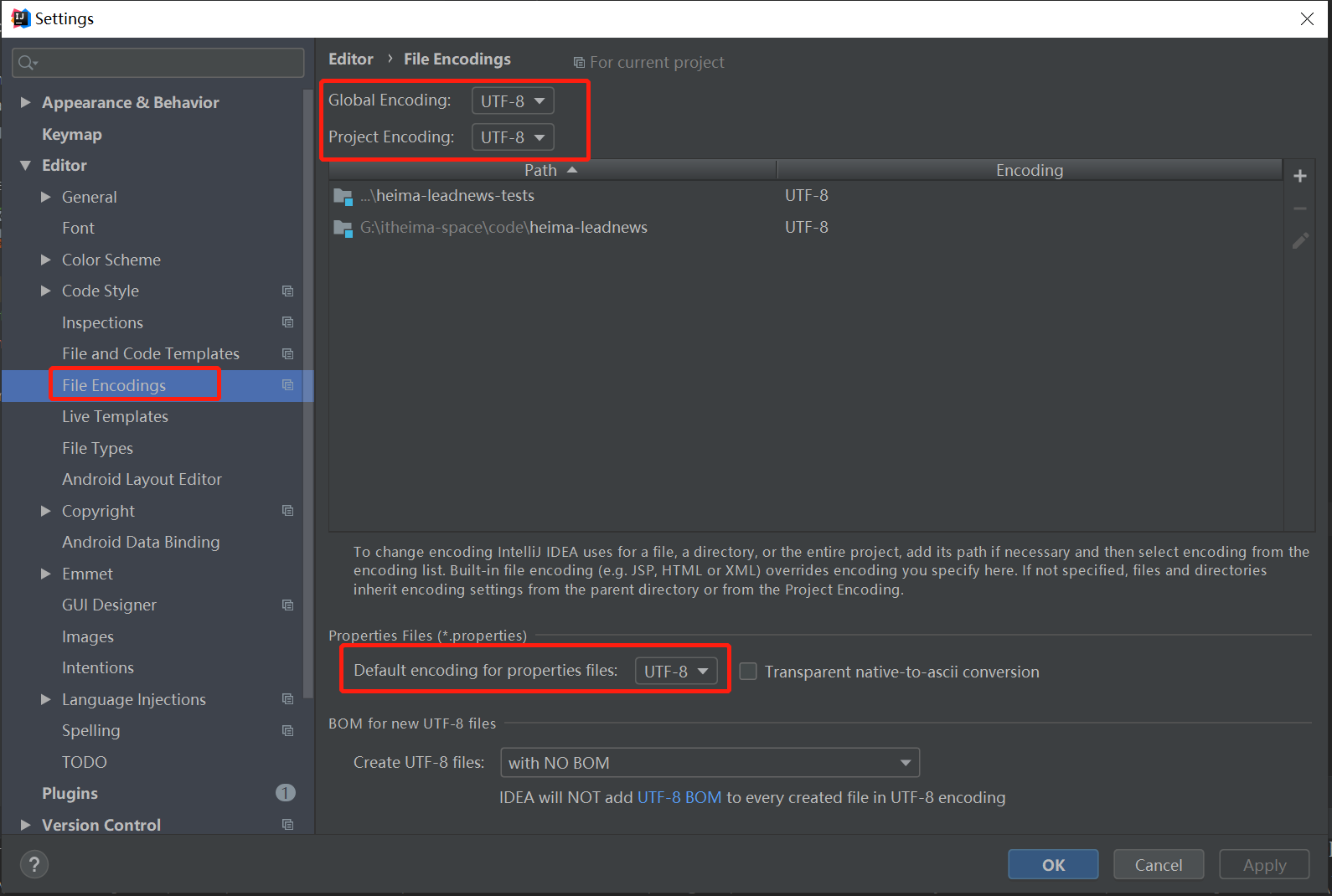


## 基本设置

### 设置本地仓库[可选]



### 设置项目编码



### 删除多余目录

* 删除src源码文件夹

### pom.xml配置

#### 更packaging类型

1. **<groupId>**com.heima**</groupId>**
2. **<artifactId>**heima-leannews**</artifactId>**
3. **<version>**1.0-SNAPSHOT**</version>**
4. **<packaging>**pom**</packaging>**

#### 集成Spring boot

1. <!-- 继承Spring boot工程 -->
2. **<parent>**
3. **<groupId>**org.springframework.boot**</groupId>**
4. **<artifactId>**spring-boot-starter-parent**</artifactId>**
5. **<version>**2.1.5.RELEASE**</version>**
6. **</parent>**

#### 设置编码&JDK

1. <!-- 项目源码及编译输出的编码 -->
2. **<project.build.sourceEncoding>**UTF-8**</project.build.sourceEncoding>**
3. **<project.reporting.outputEncoding>**UTF-8**</project.reporting.outputEncoding>**
4. <!-- 项目编译JDK版本 -->
5. **<maven.compiler.source>**1.8**</maven.compiler.source>**
6. **<maven.compiler.target>**1.8**</maven.compiler.target>**

#### 多环境配置

项目分三个环境：

* dev : 开发环境
* test : 测试环境
* prod: 生产环境

1. **<profiles>**
2. **<profile>**
3. **<id>**dev**</id>**
4. **<build>**
5. **<filters>**
6. **<filter>**maven\_dev.properties**</filter>**
7. **</filters>**
8. **</build>**
9. **</profile>**
10. **<profile>**
11. **<id>**test**</id>**
12. **<build>**
13. **<filters>**
14. **<filter>**maven\_test.properties**</filter>**
15. **</filters>**
16. **</build>**
17. **</profile>**
18. **<profile>**
19. **<id>**prod**</id>**
20. **<build>**
21. **<filters>**
22. **<filter>**maven\_prod.properties**</filter>**
23. **</filters>**
24. **</build>**
25. **</profile>**
26. **</profiles>**

#### Jar依赖导入

##### properties版本配置

1. <!-- 依赖包版本管理 -->
2. **<commons.io.version>**2.6**</commons.io.version>**
3. **<commons.codec.version>**1.12**</commons.codec.version>**

##### dependencies导入

1. **<dependencies>**
2. <!--日志处理包-->
3. **<dependency>**
4. **<groupId>**org.springframework.boot**</groupId>**
5. **<artifactId>**spring-boot-starter-log4j2**</artifactId>**
6. **</dependency>**
7. **</dependencies>**

##### dependencyManagement导入

1. **<dependencyManagement>**
2. **<dependencies>**
3. <!--内部依赖工程-->
4. **<dependency>**
5. **<artifactId>**heima-leadnews-apis**</artifactId>**
6. **<groupId>**com.heima**</groupId>**
7. **<version>**${project.version}**</version>**
8. **</dependency>**
9. **<dependency>**
10. **<artifactId>**heima-leadnews-common**</artifactId>**
11. **<groupId>**com.heima**</groupId>**
12. **<version>**${project.version}**</version>**
13. **</dependency>**
14. **<dependency>**
15. **<artifactId>**heima-leadnews-model**</artifactId>**
16. **<groupId>**com.heima**</groupId>**
17. **<version>**${project.version}**</version>**
18. **</dependency>**
19. **<dependency>**
20. **<artifactId>**heima-leadnews-utils**</artifactId>**
21. **<groupId>**com.heima**</groupId>**
22. **<version>**${project.version}**</version>**
23. **</dependency>**
24. <!--IO流处理包-->
25. **<dependency>**
26. **<groupId>**commons-io**</groupId>**
27. **<artifactId>**commons-io**</artifactId>**
28. **<version>**${commons.io.version}**</version>**
29. **</dependency>**
30. <!--加密处理包-->
31. **<dependency>**
32. **<groupId>**commons-codec**</groupId>**
33. **<artifactId>**commons-codec**</artifactId>**
34. **<version>**${commons.codec.version}**</version>**
35. **</dependency>**
36. **</dependencies>**
37. **</dependencyManagement>**

#### Build配置

##### resource配置

1. **<resources>**
2. **<resource>**
3. **<directory>**src/main/resources**</directory>**
4. **<includes>**
5. **<include>**\*\*/\*.\***</include>**
6. **</includes>**
7. **<filtering>**true**</filtering>**
8. **</resource>**
9. **</resources>**
10. **<testResources>**
11. **<testResource>**
12. **<directory>**src/test/resources**</directory>**
13. **<filtering>**true**</filtering>**
14. **</testResource>**
15. **</testResources>**

##### plugins配置

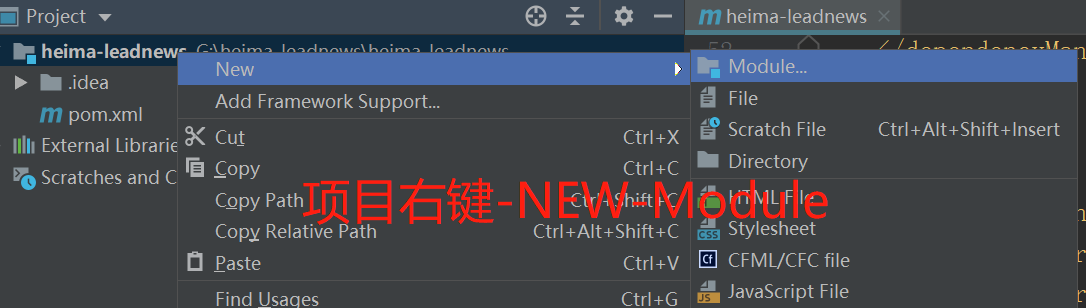
1. **<plugins>**
2. **<plugin>**
3. **<artifactId>**maven-resources-plugin**</artifactId>**
4. **<configuration>**
5. **<useDefaultDelimiters>**true**</useDefaultDelimiters>**
6. **</configuration>**
7. **</plugin>**
8. **<plugin>**
9. **<groupId>**pl.project13.maven**</groupId>**
10. **<artifactId>**git-commit-id-plugin**</artifactId>**
11. **<executions>**
12. **<execution>**
13. **<goals>**
14. **<goal>**revision**</goal>**
15. **</goals>**
16. **</execution>**
17. **</executions>**
18. **<configuration>**
19. **<dotGitDirectory>**../../.git**</dotGitDirectory>**
20. **<dateFormat>**yyyy-MM-dd HH:mm:ss**</dateFormat>**
21. **<useNativeGit>**true**</useNativeGit>**
22. **<abbrevLength>**8**</abbrevLength>**
23. **<generateGitPropertiesFile>**false**</generateGitPropertiesFile>**
24. **</configuration>**
25. **</plugin>**
26. **</plugins>**

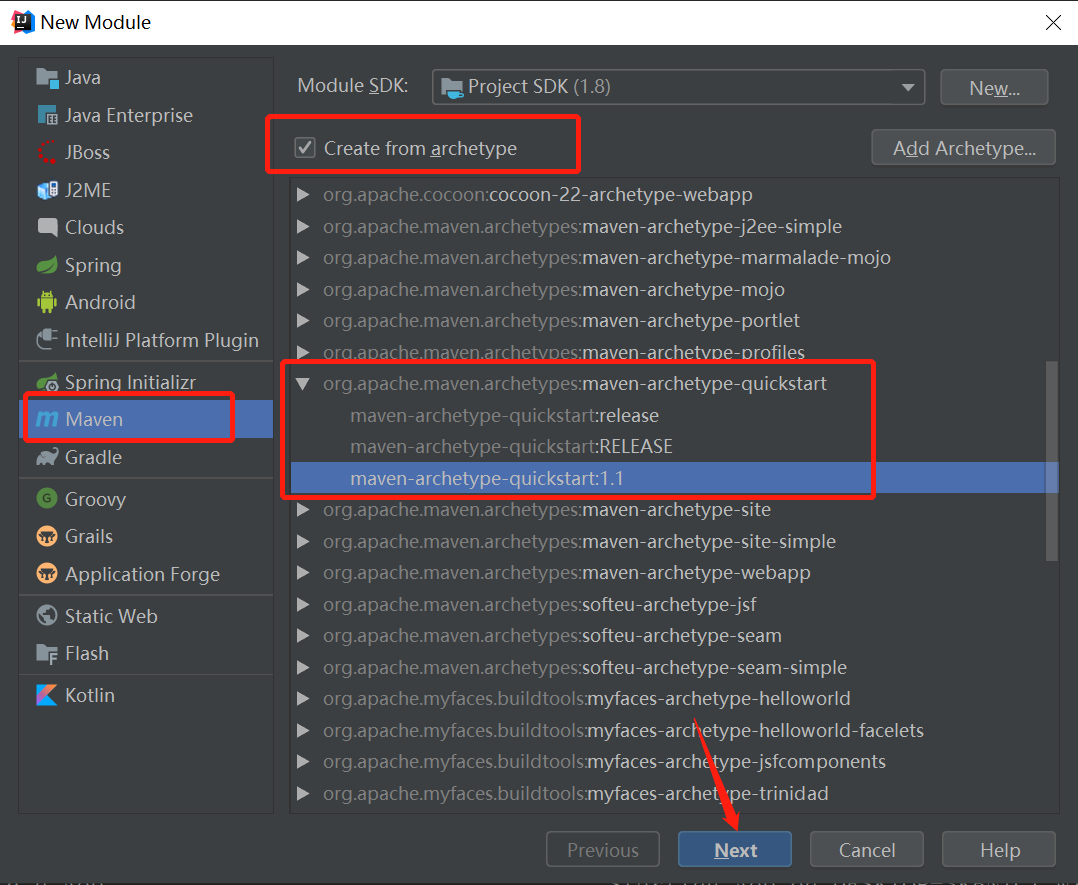
# 创建common子项目[heima-leadnews-common]

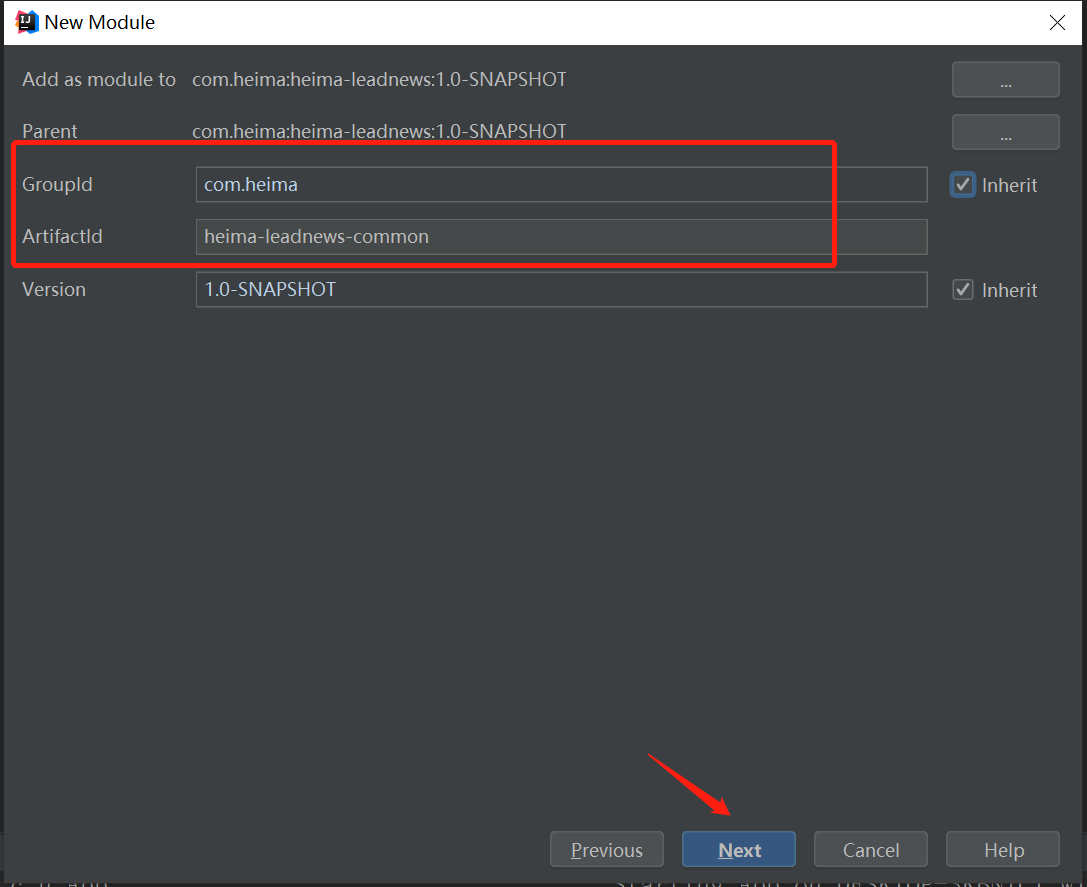
Conmmon项目是整个工程的配置核心，包括所有集成三方框架的配置定义，比如redis、kafka等。除此之外还包括项目每个模块及整个项目的常量定义。

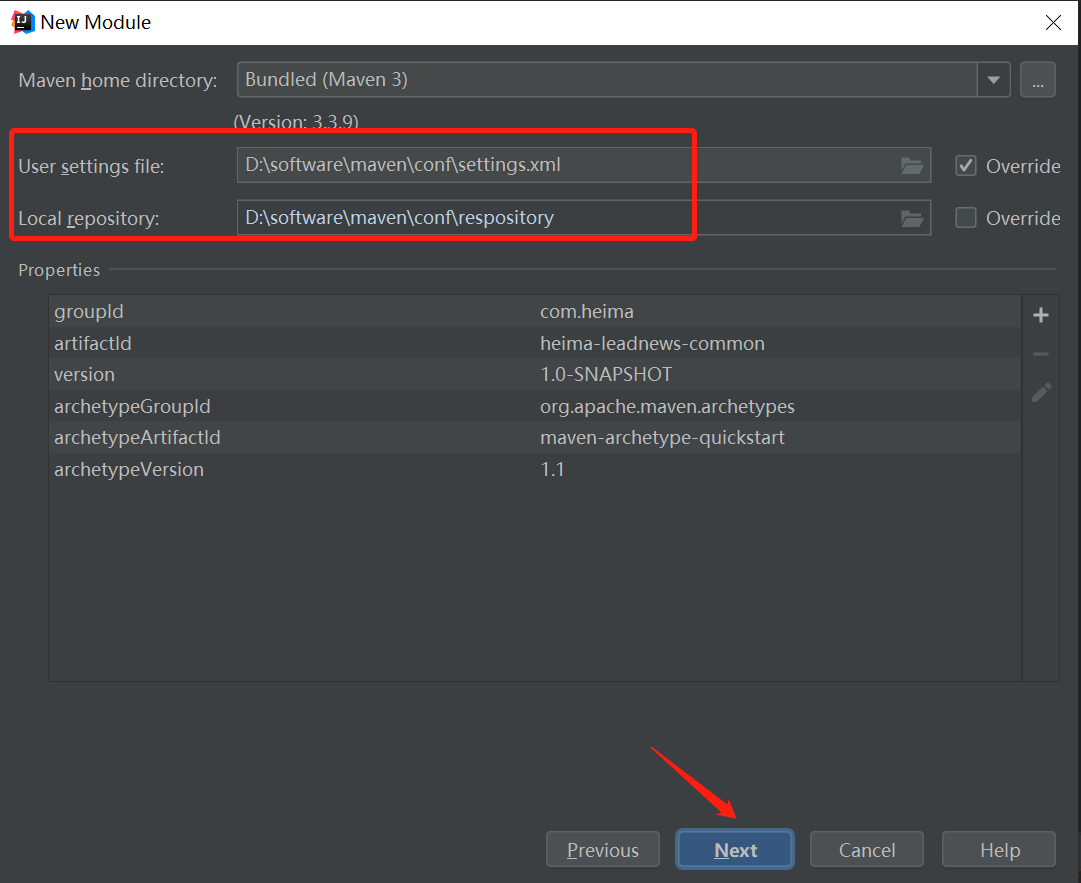


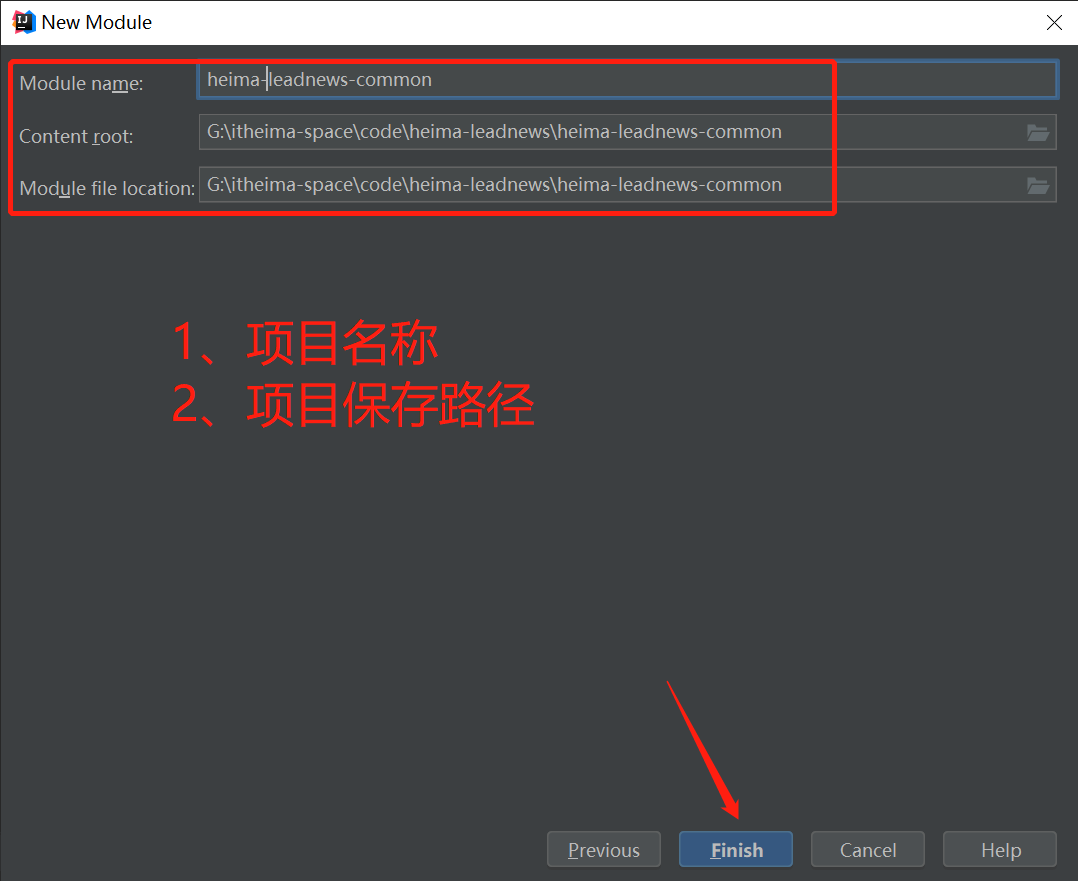
## 创建工程



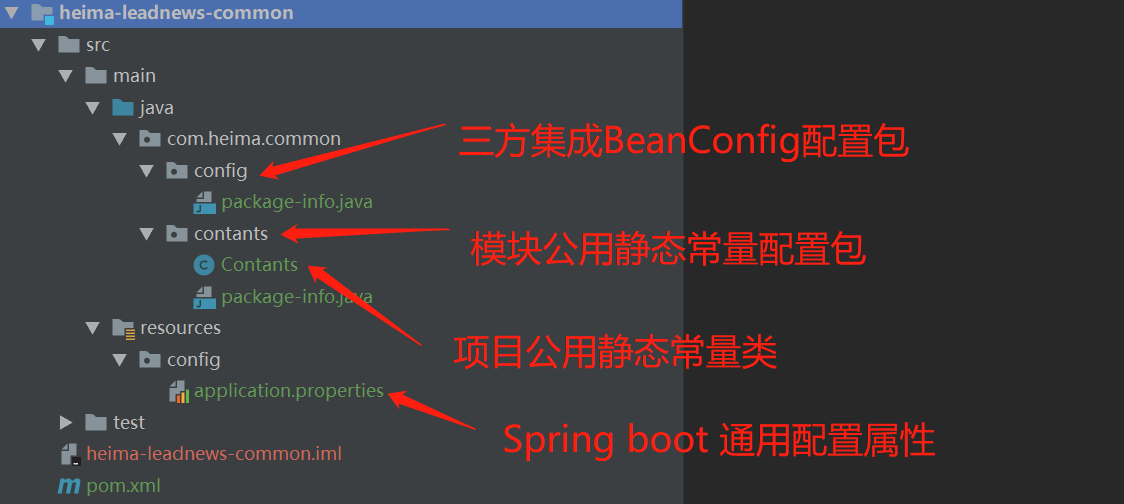








## 创建结构

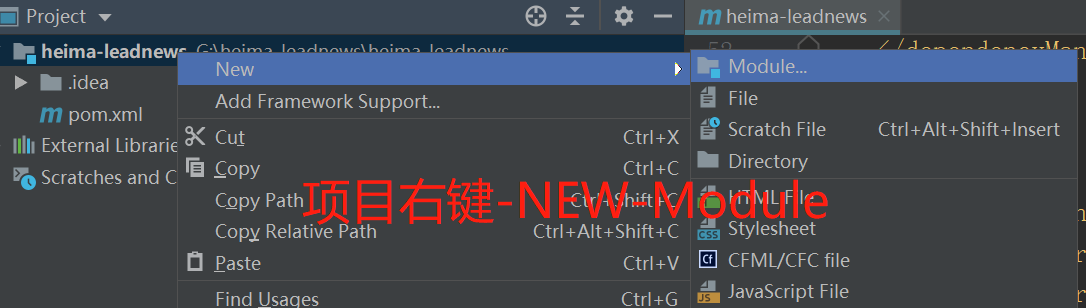


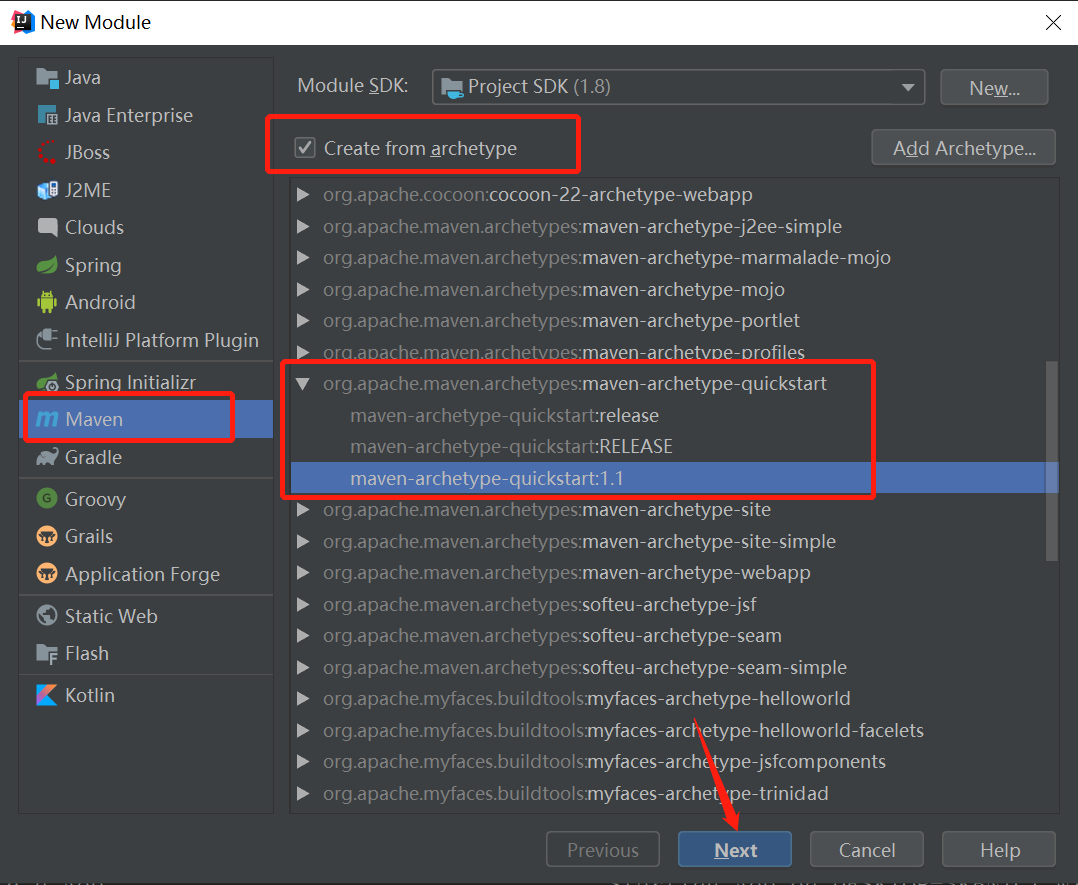
# 创建utils子项目[heima-leadnews-utils]

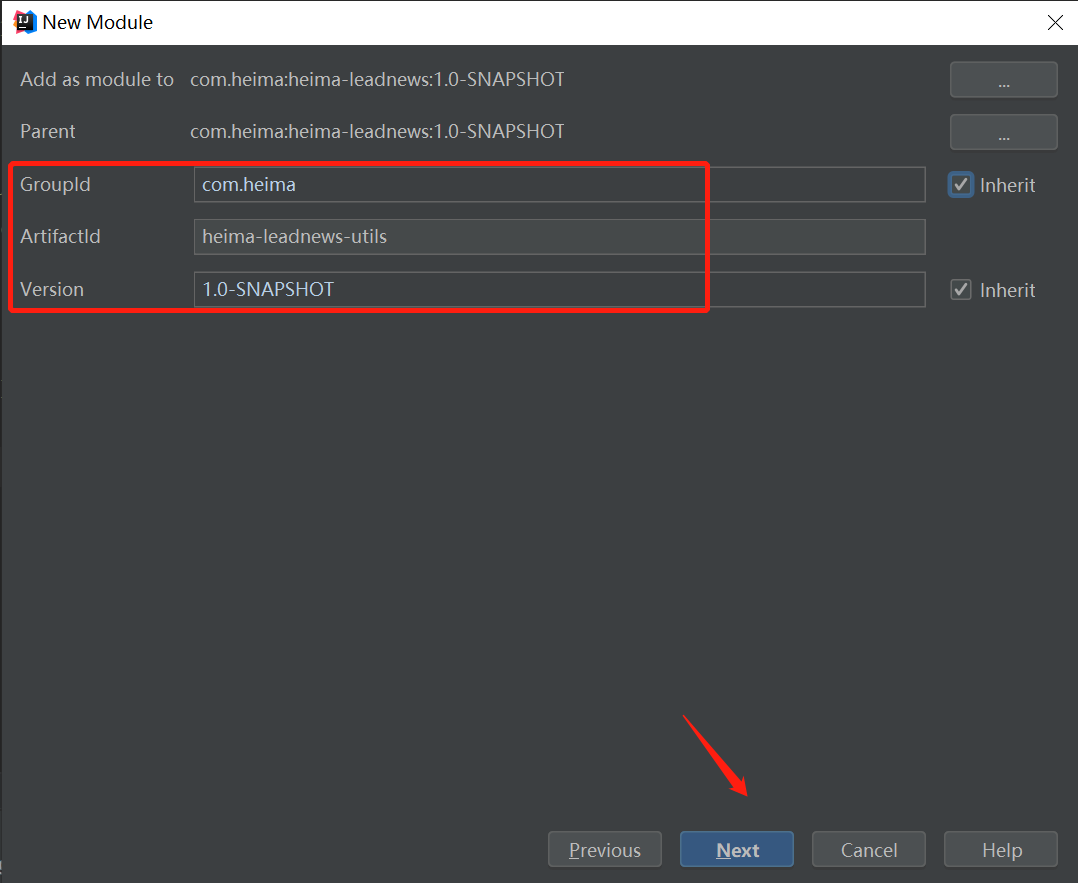
Utils是工程公用工具类项目，包含加密/解密、Date、JSON等工具类。

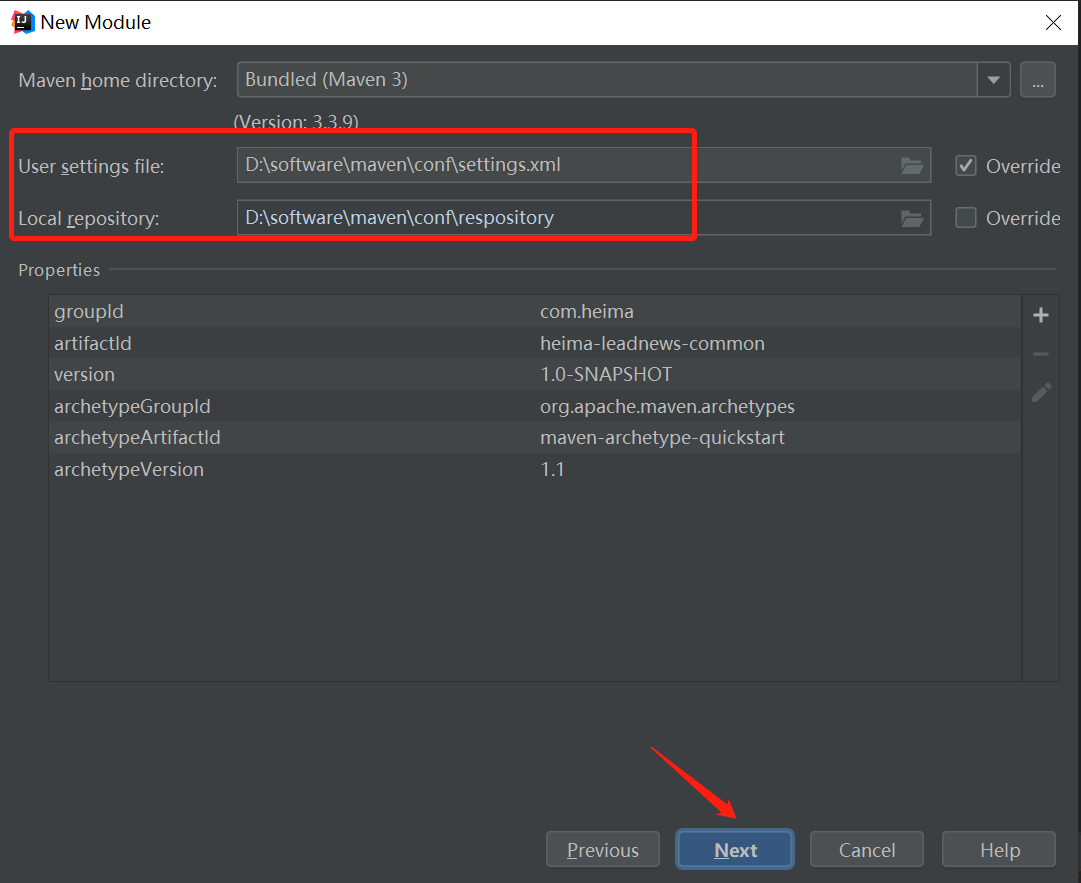


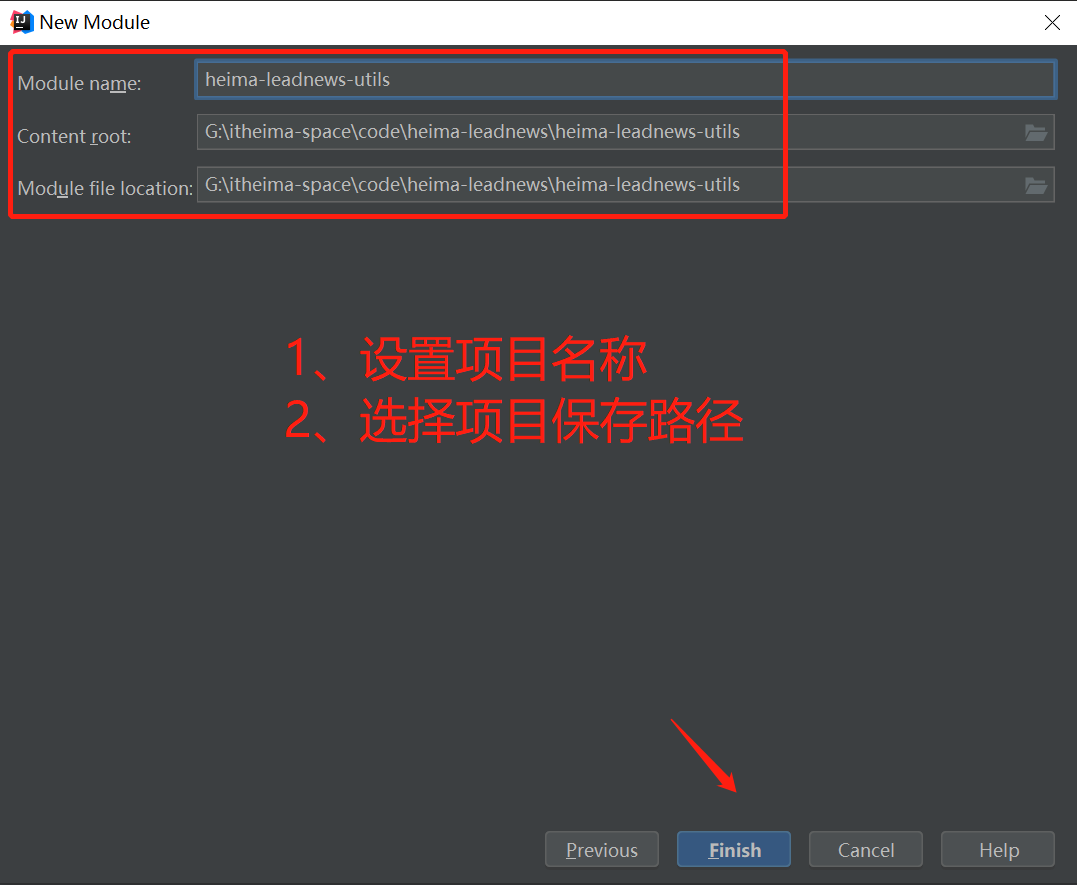
## 创建工程



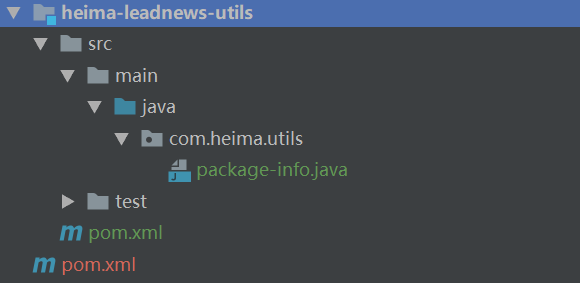








## 创建结构



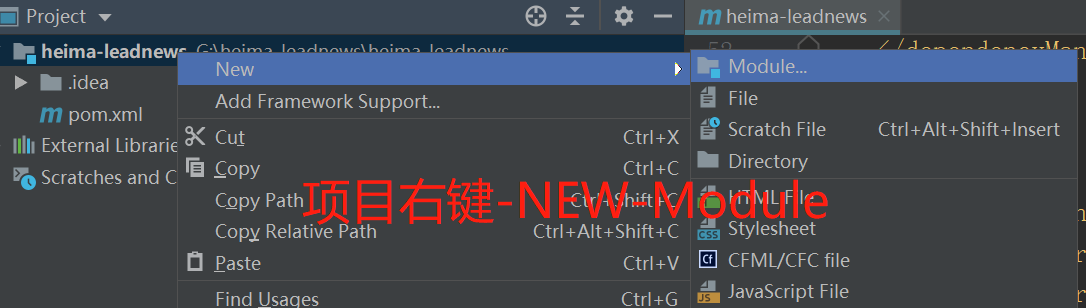
* utils下面存放项目所有模块公用的工具类，比如加密解密、日期、JSON处理等。

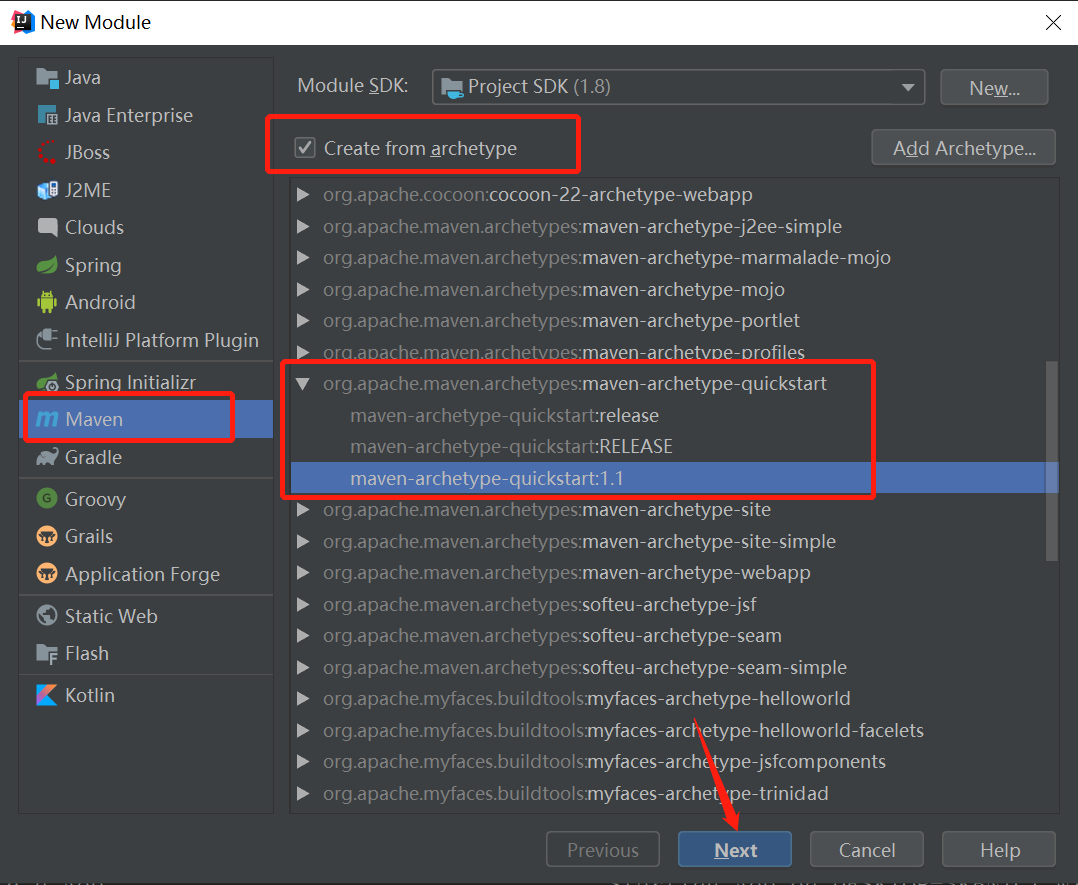
# 创建model子项目[heima-leadnews-model]

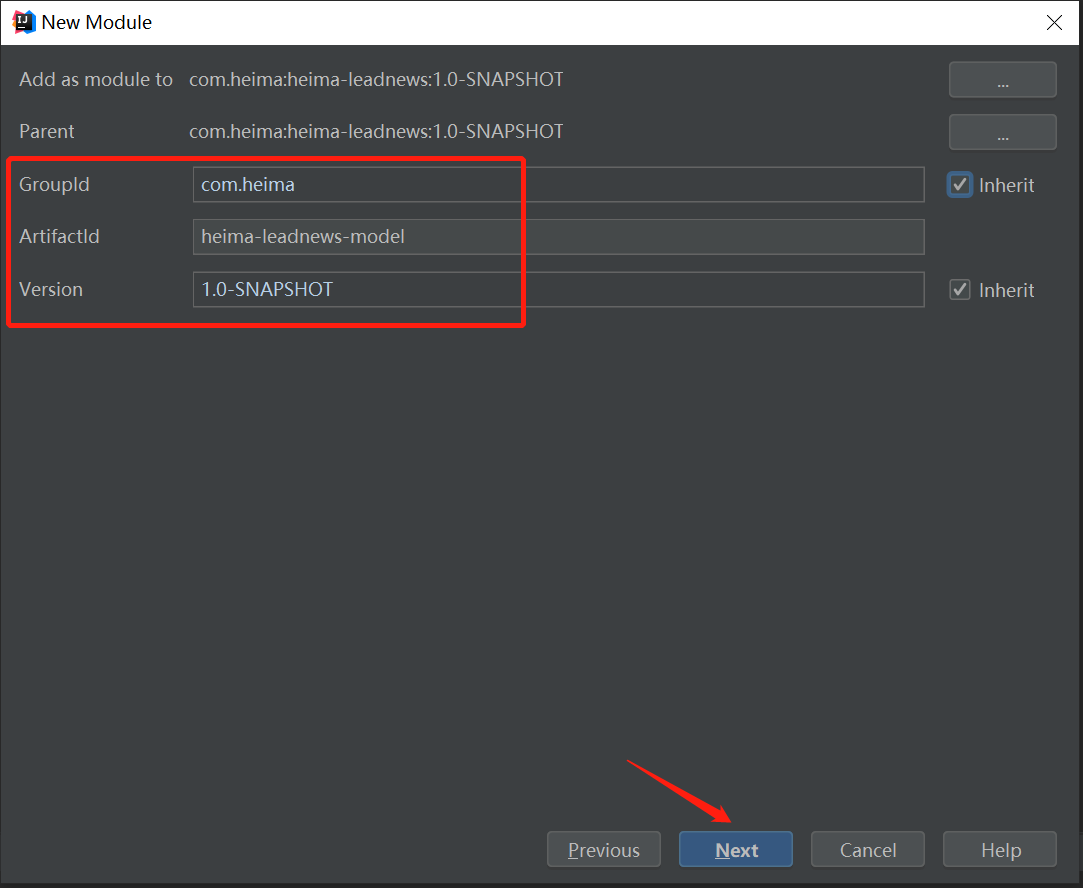
Model是项目中用到的Dto、Pojo、Mapper、Enums定义工程。

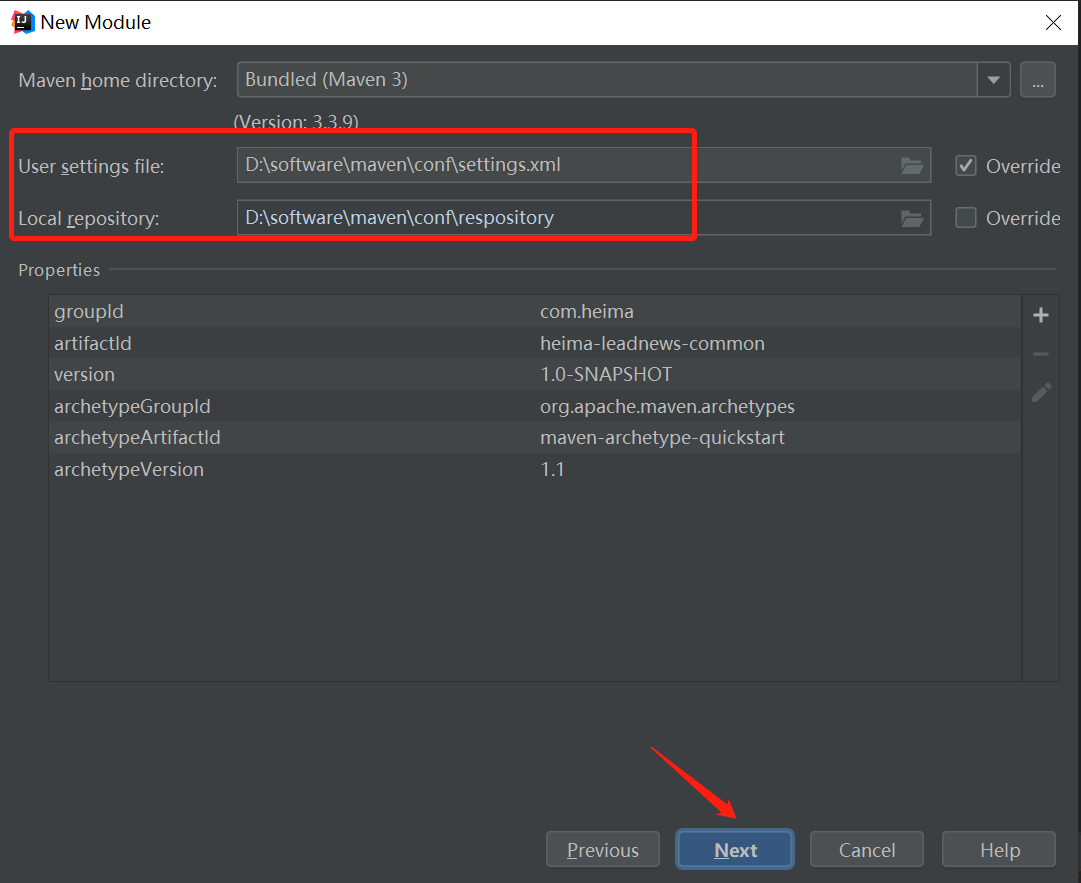


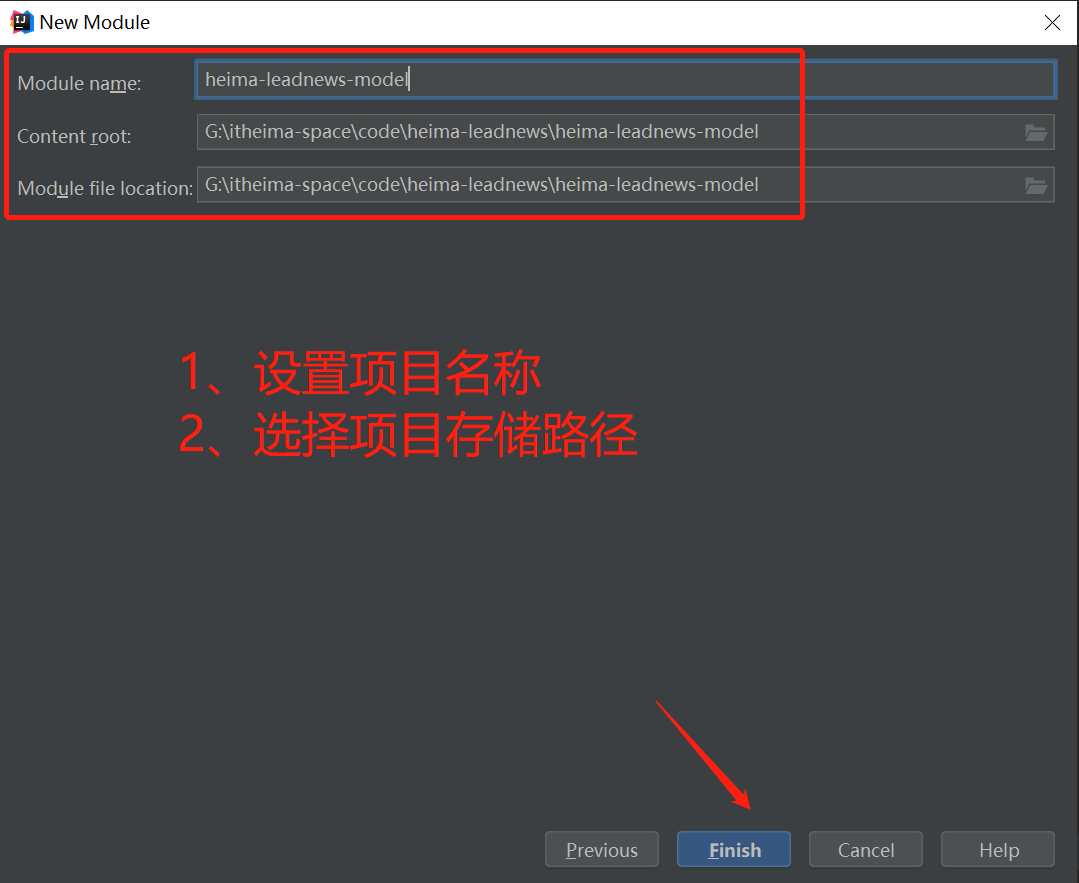
## 创建工程



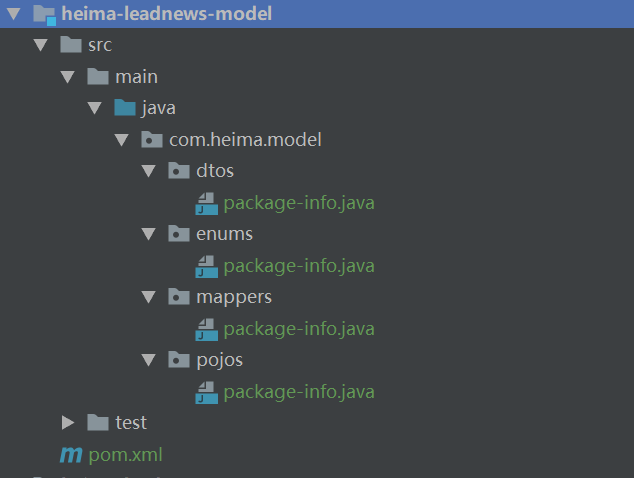








## 创建结构



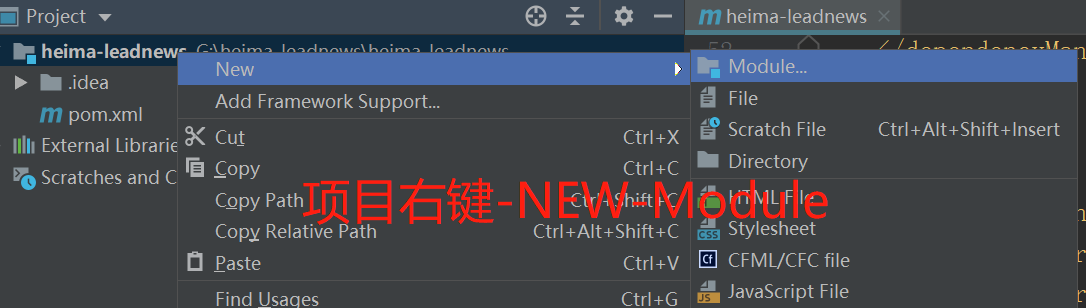
* dtos ：用于存放数据对象的转换，常用于Controller层
* enums：用于定义模块的枚举类，例如一些code的枚举
* mappers: 用于定义Mybatis 代理Mapper类
* pojos：用于定义数据库表对应的JAVA简单对象

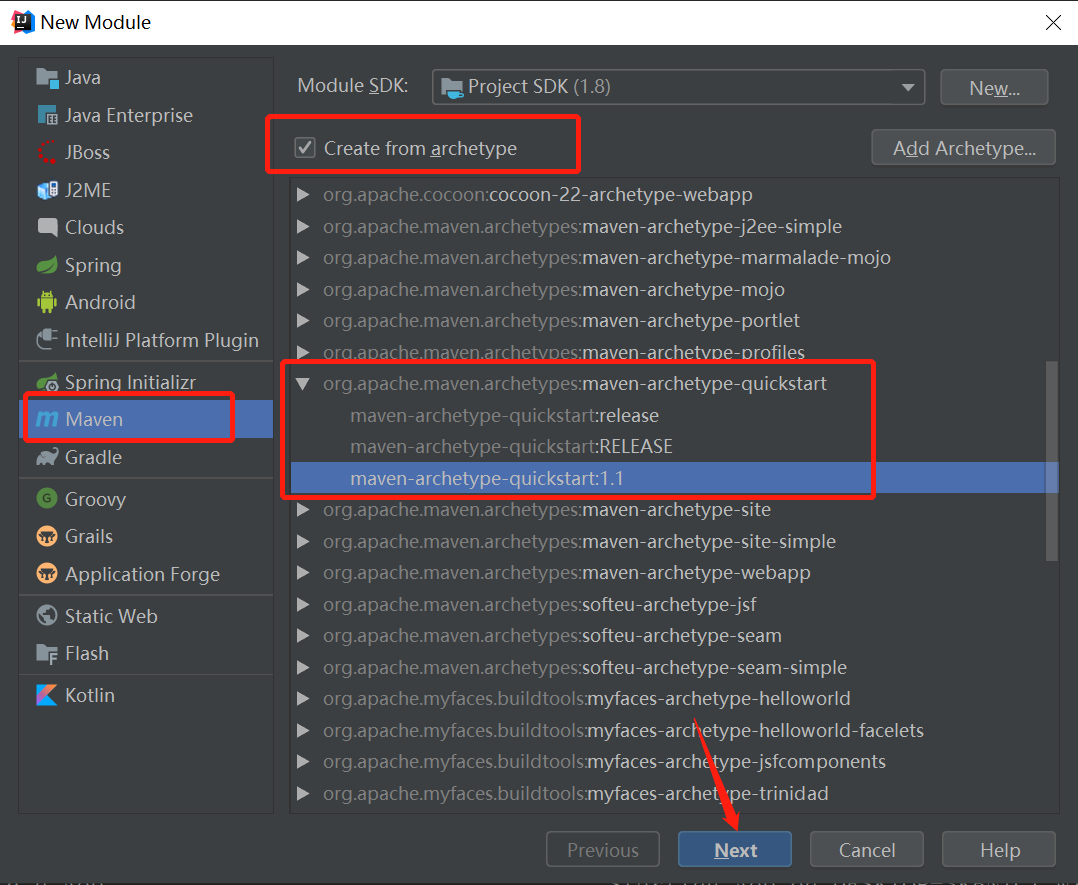
# 创建apis子项目[heima-leadnews-apis]

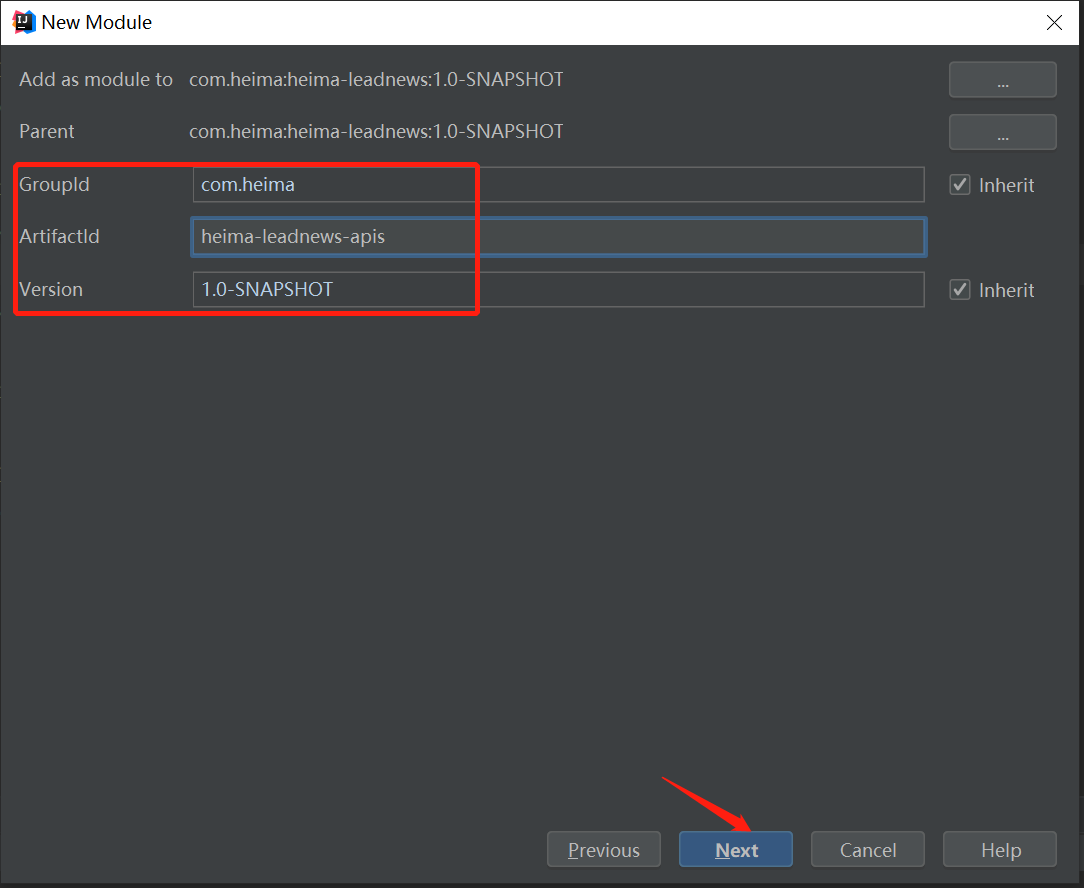
Apis : 是整个项目微服务暴露的接口的定义工程，按每个模块进行子包拆分。

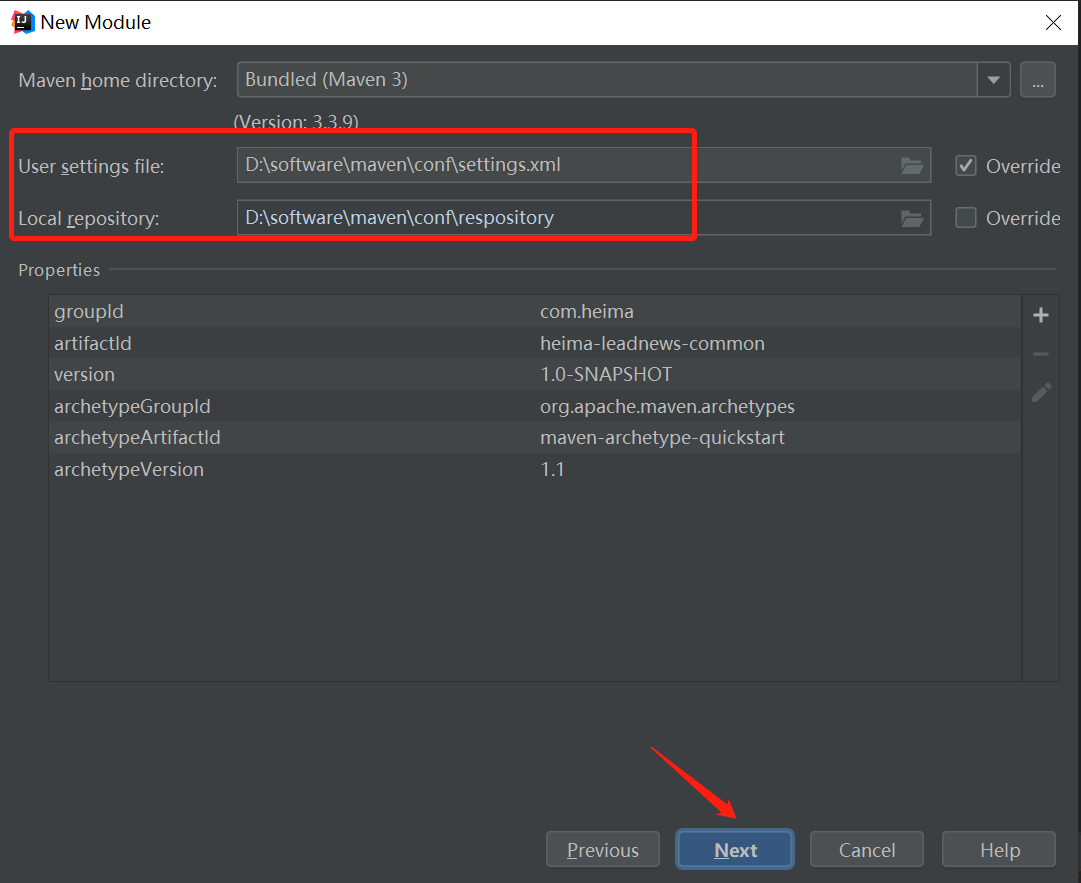


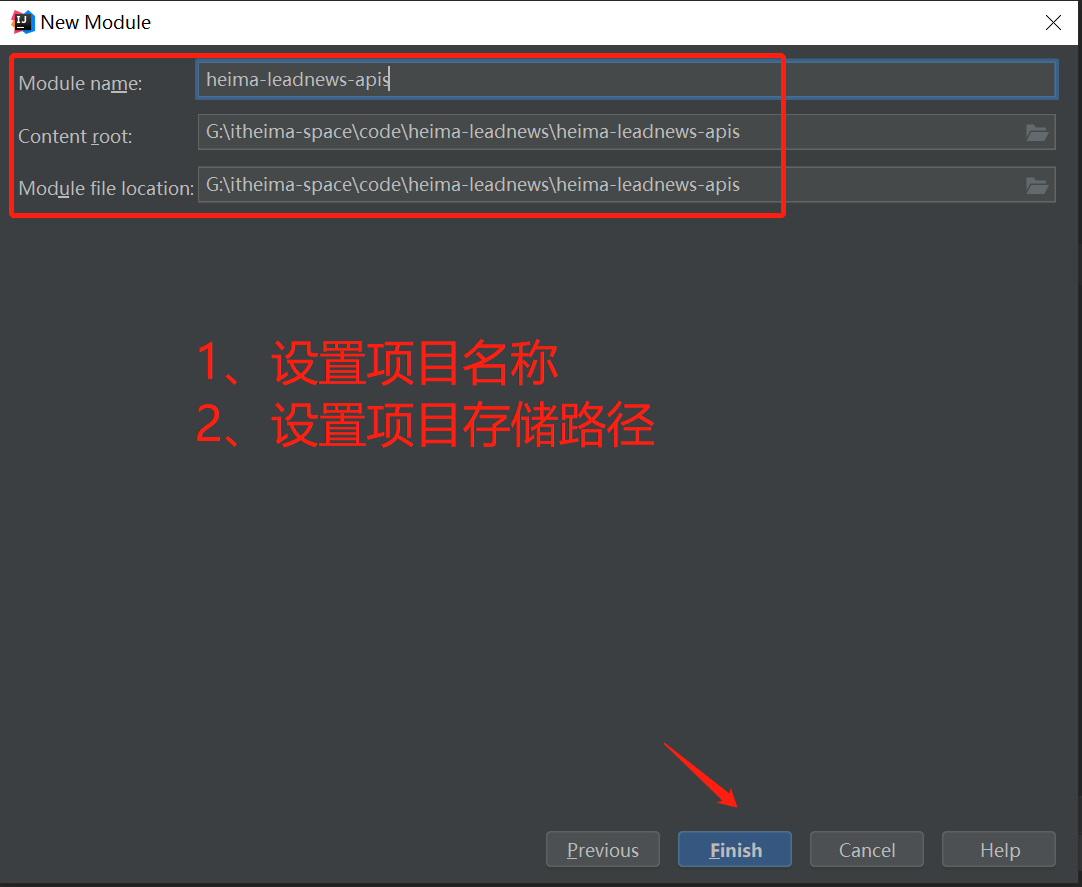
## 创建工程



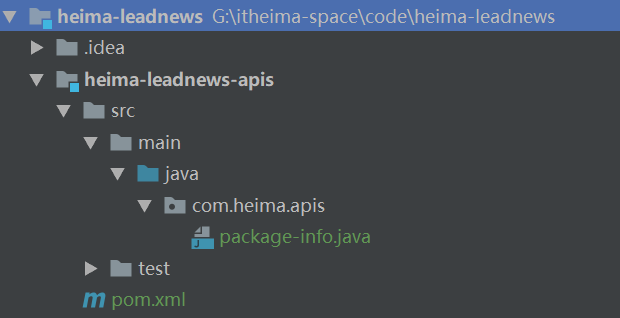








## 创建结构



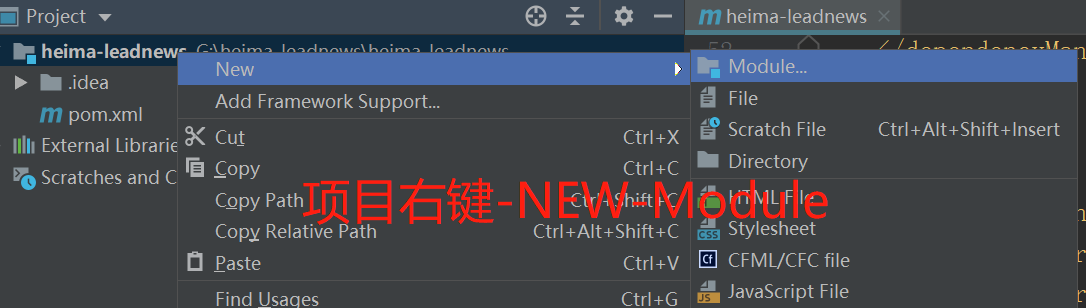
* apis：按照每个模块进行定义子包，存放微服务暴露的接口

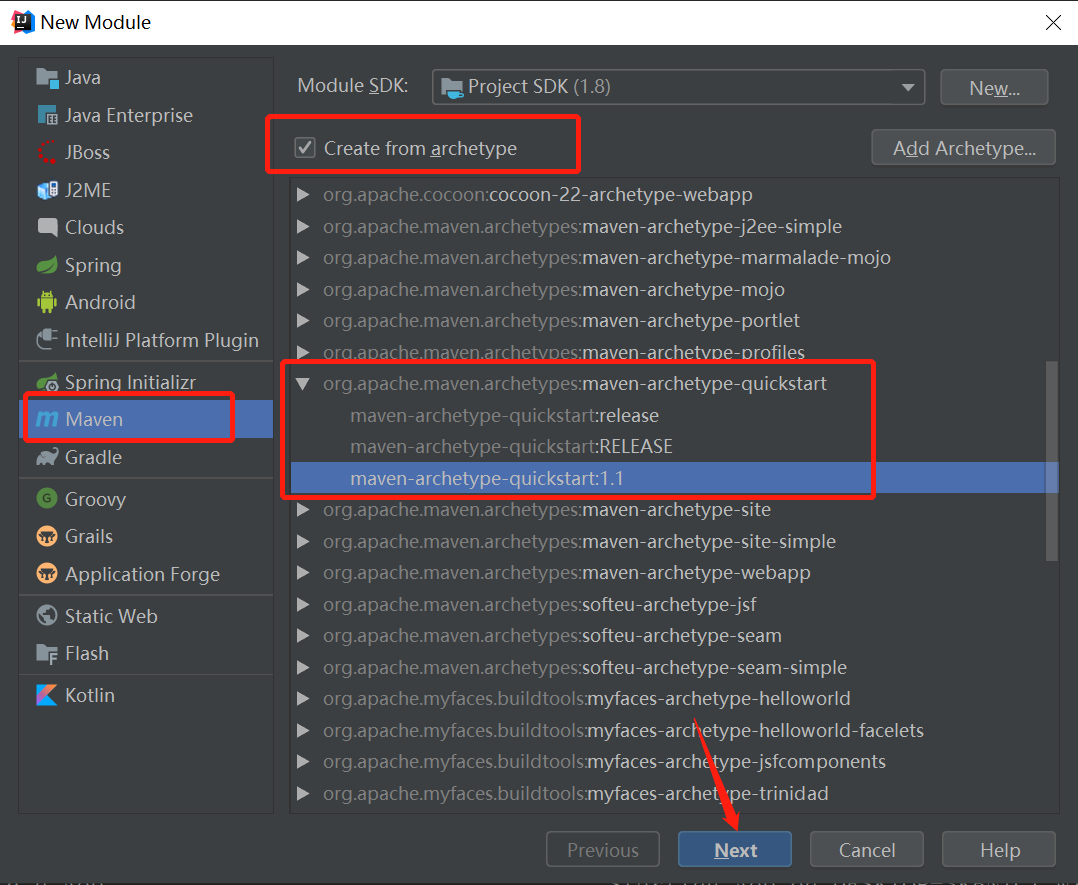
# 创建测试子项目[heima-leadnews-tests]

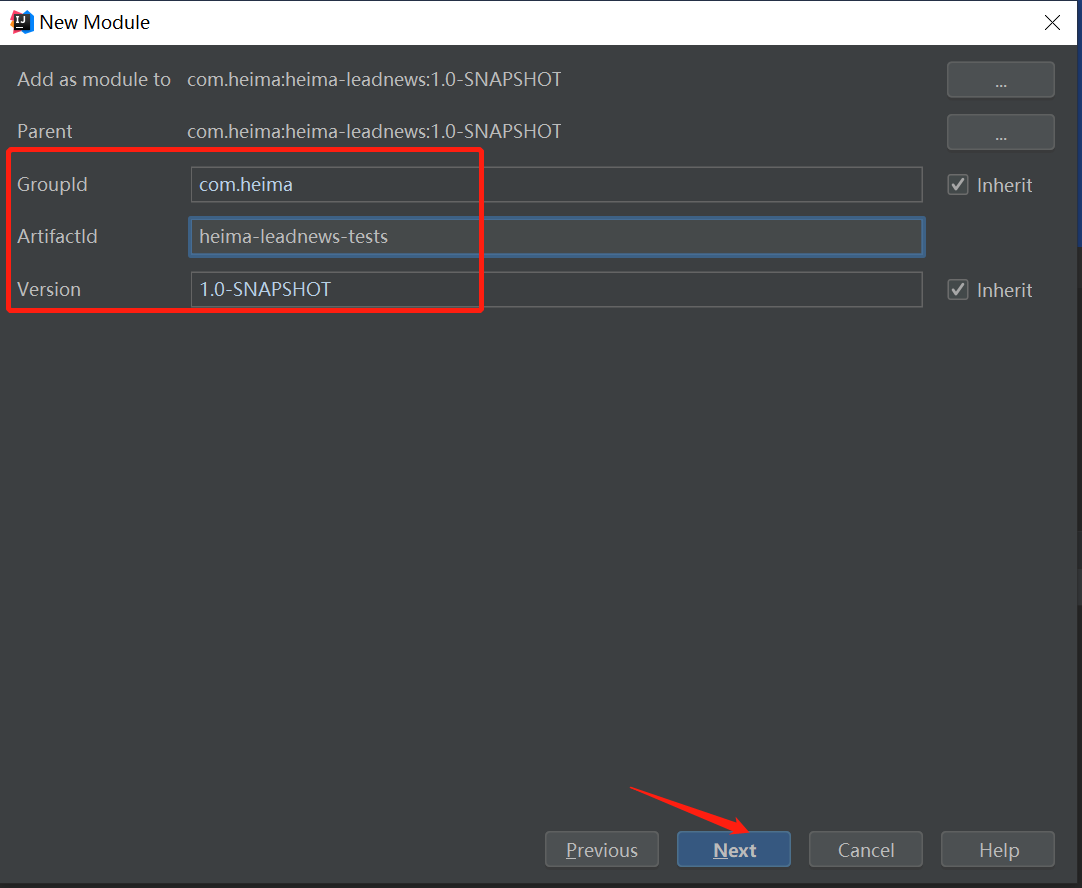
Tests项目用于存放工程中通用的测试用例。

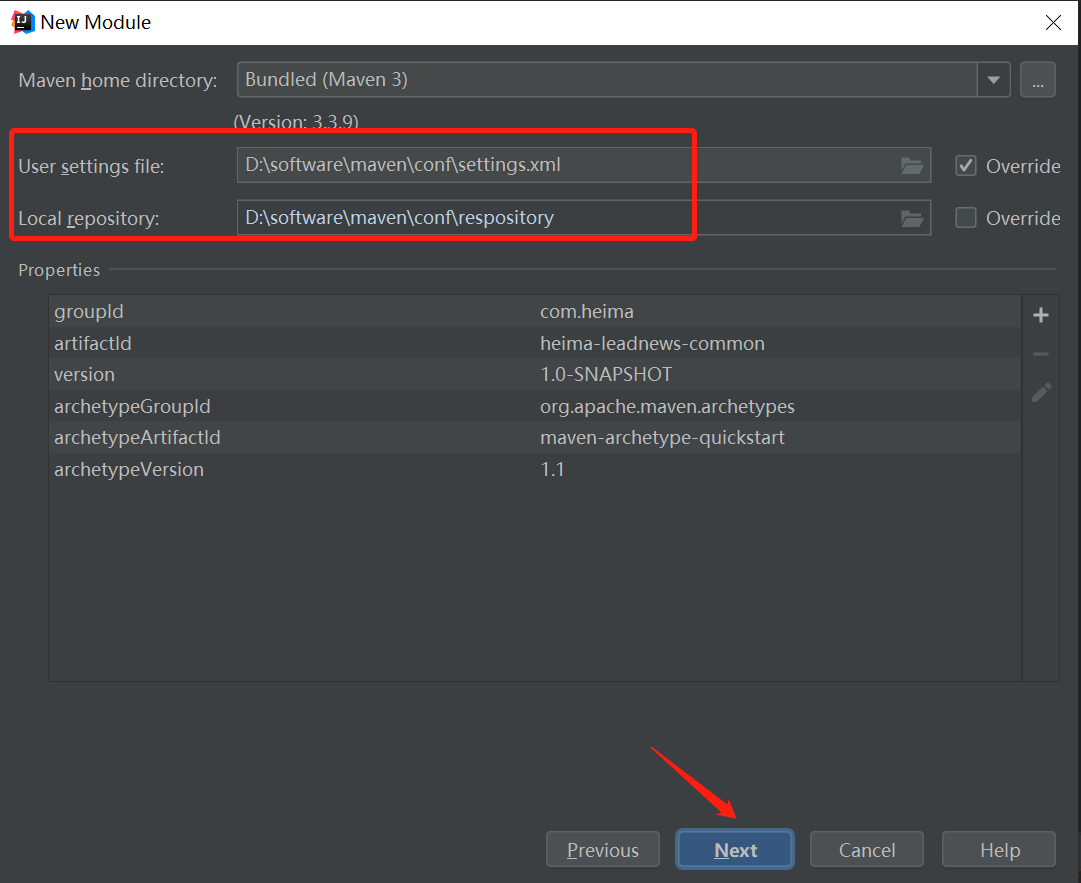


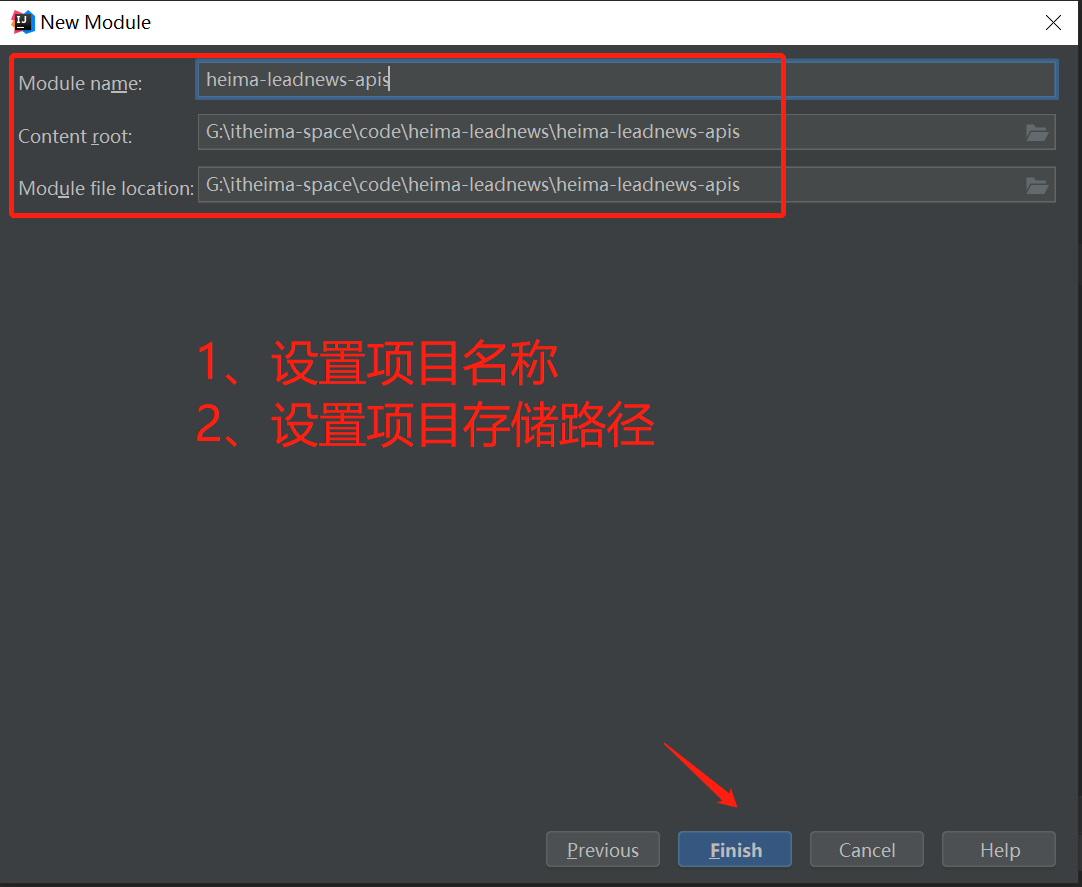
## 创建工程

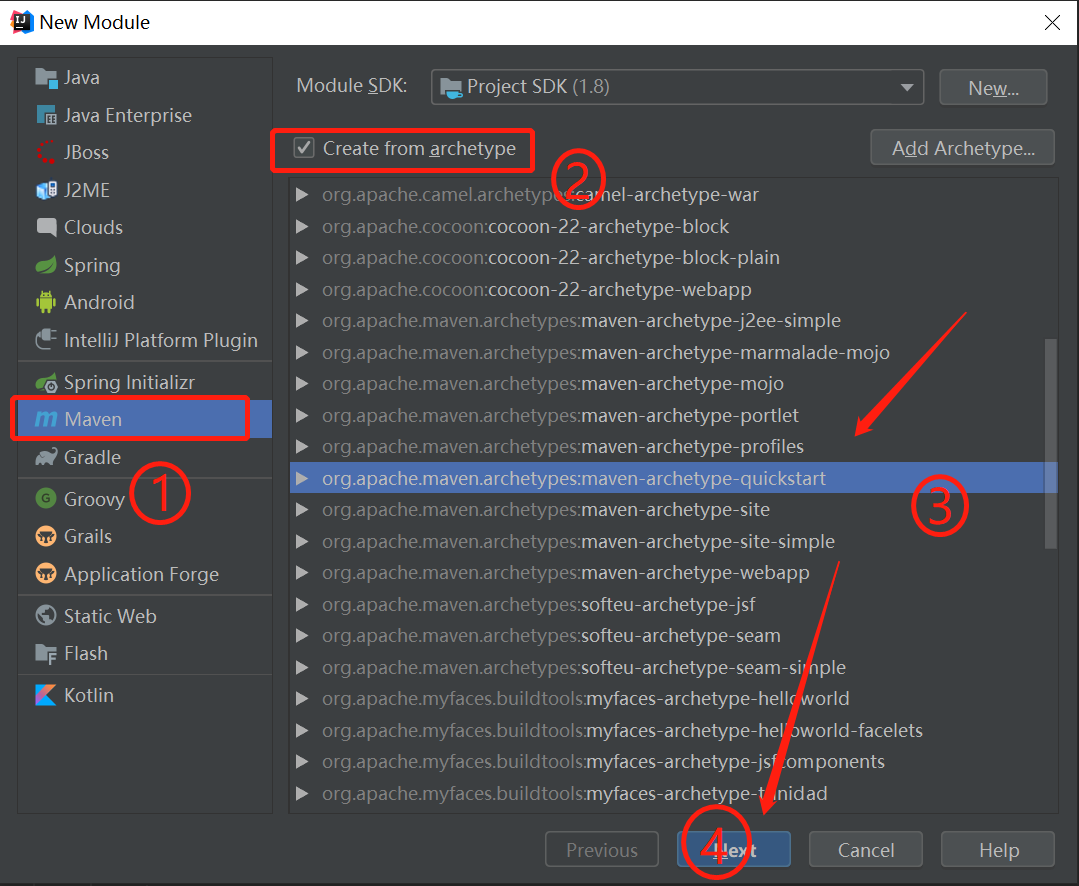
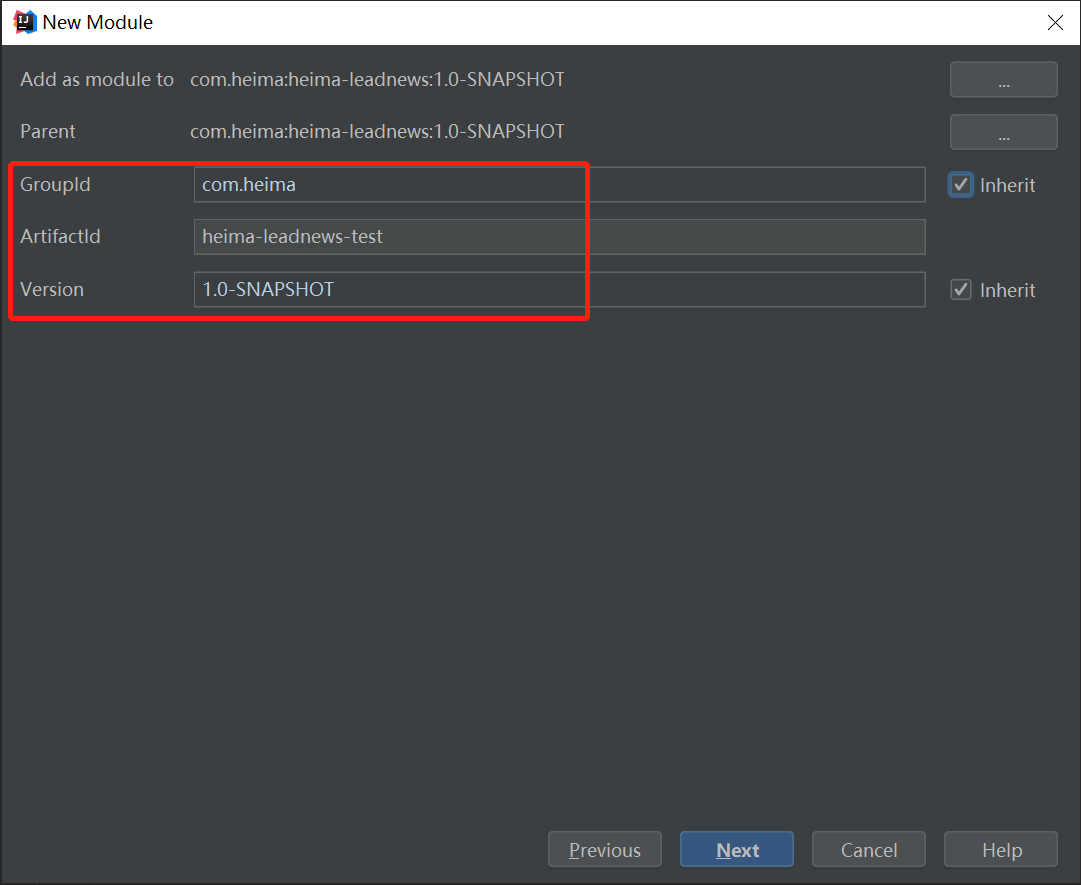


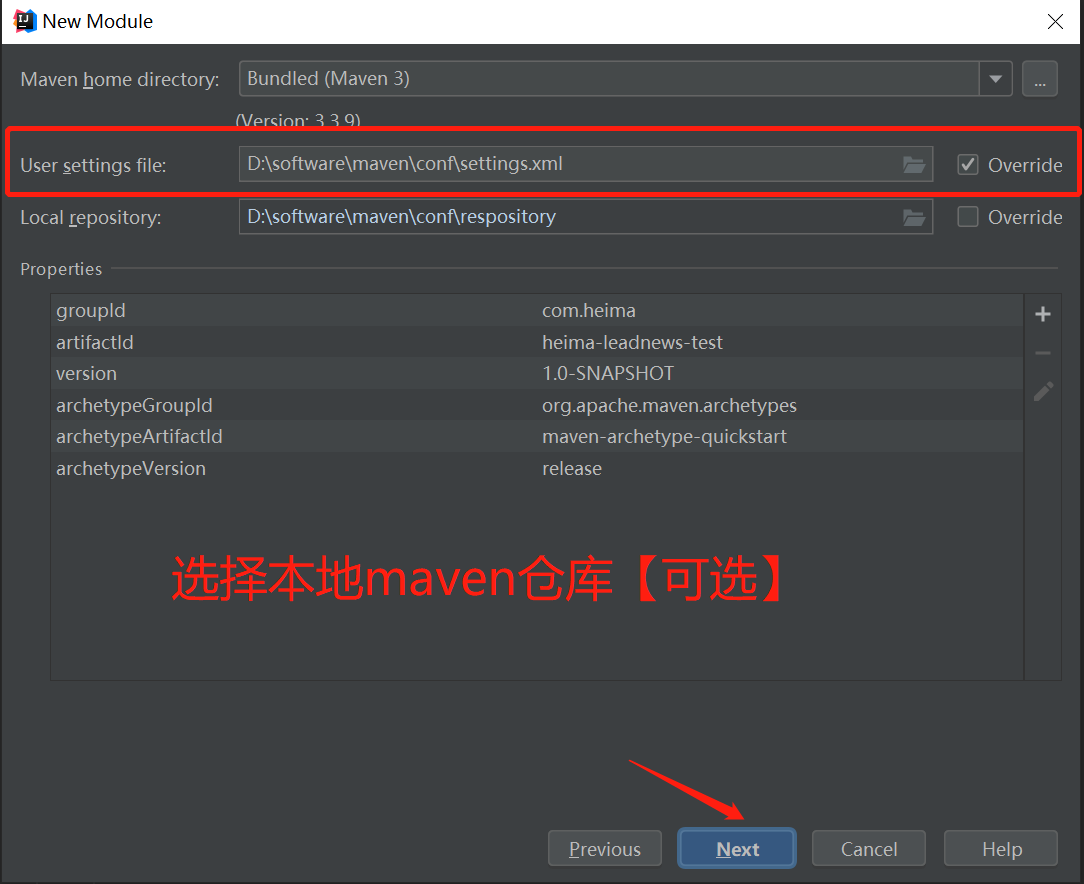


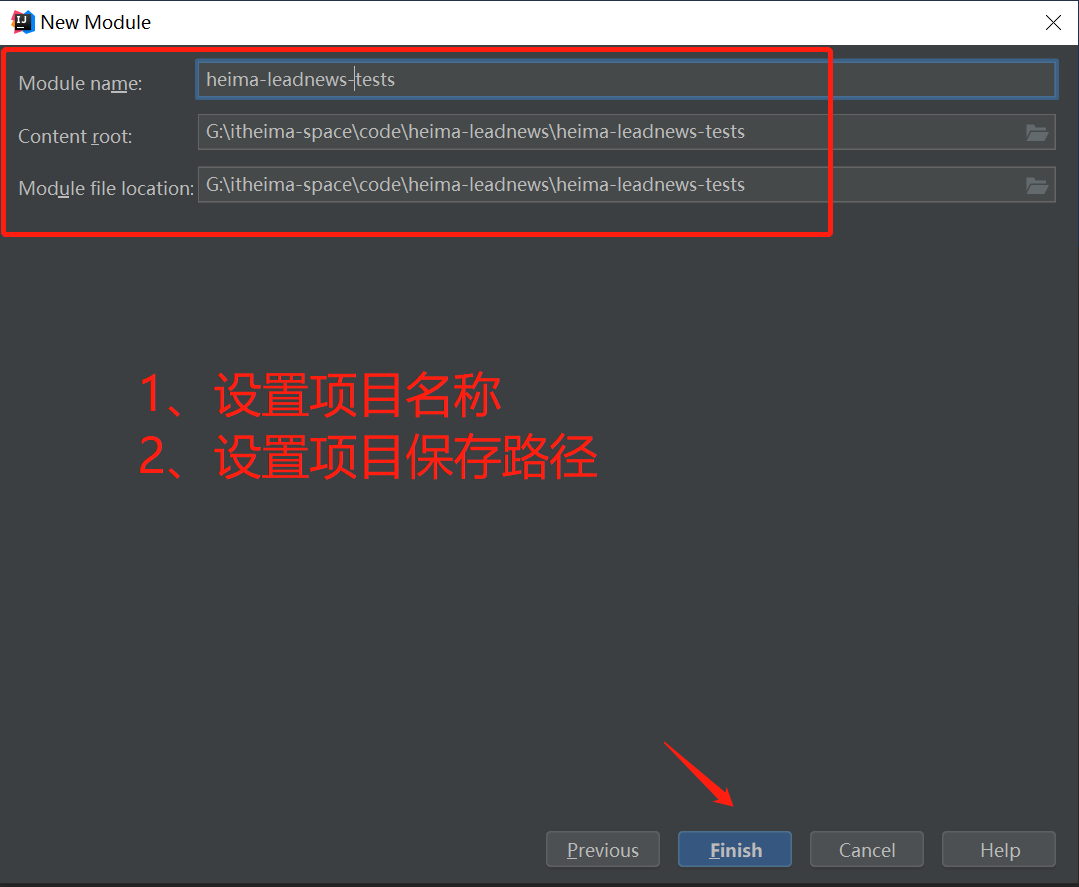










## 创建结构



## 添加配置/代码

### 添加POM依赖

1. **<dependencies>**
2. <!--内部依赖-->
3. **<dependency>**
4. **<artifactId>**heima-leadnews-common**</artifactId>**
5. **<groupId>**com.heima**</groupId>**
6. **</dependency>**
7. <!-- Spring boot starter -->
8. **<dependency>**
9. **<groupId>**org.springframework.boot**</groupId>**
10. **<artifactId>**spring-boot-starter-web**</artifactId>**
11. <!-- 排除默认的logback日志，使用log4j-->
12. **<exclusions>**
13. **<exclusion>**
14. **<groupId>**org.springframework.boot**</groupId>**
15. **<artifactId>**spring-boot-starter-logging**</artifactId>**
16. **</exclusion>**
17. **<exclusion>**
18. **<groupId>**org.slf4j**</groupId>**
19. **<artifactId>**slf4j-log4j12**</artifactId>**
20. **</exclusion>**
21. **</exclusions>**
22. **</dependency>**
23. **</dependencies>**

### 添加多环境配置

#### maven\_dev.properties

1. log.level=ALL
2. log.pattern=%d{DEFAULT}^|%sn^|%level^|%t^|%c^|%M^|%msg%n

#### maven\_test.properties

1. log.level=DEBUG
2. log.pattern=%d{DEFAULT}^|%sn^|%level^|%t^|%c^|%M^|%msg%n

#### maven\_dev.properties

1. log.level=INFO
2. log.pattern=%d{DEFAULT}^|%sn^|%level^|%t^|%c^|%M^|%msg%n

### 添加配置文件

#### appliaction.properties

1. log.level = ${log.level}
2. log.pattern = ${log.pattern}

#### log4j2.xml

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<configuration>**
3. **<properties>**
4. **<property** name="CONSOLE\_PATTERN"**>**${log.pattern}**</property>**
5. **<property** name="FILE\_NAME"**>**${project.build.finalName}**</property>**
6. **</properties>**
7. <!--先定义所有的appender-->
8. **<appenders>**
9. <!--这个输出控制台的配置-->
10. **<console** name="Console" target="SYSTEM\_OUT"**>**
11. <!--输出日志的格式-->
12. **<PatternLayout** pattern="${CONSOLE\_PATTERN}"**/>**
13. **</console>**
14. <!-- 这个会打印出所有的info及以下级别的信息，每次大小超过size，则这size大小的日志会自动存入按年份-月份建立的文件夹下面并进行压缩，作为存档-->
15. **<RollingFile** name="RollingFileInfo" fileName="${sys:user.home}/logs/${FILE\_NAME}.log"
16. filePattern="${sys:user.home}/logs/$${date:yyyy-MM}/${FILE\_NAME}-%d{yyyy-MM-dd}-%i.log"**>**
17. **<PatternLayout** pattern="${CONSOLE\_PATTERN}"**/>**
18. **<Policies>**
19. **<TimeBasedTriggeringPolicy/>**
20. **<SizeBasedTriggeringPolicy** size="100 MB"**/>**
21. **</Policies>**
22. **</RollingFile>**
23. **</appenders>**
24. **<loggers>**
25. <!--过滤掉spring和mybatis的一些无用的DEBUG信息-->
26. **<logger** name="org.springframework" level="INFO"**></logger>**
27. **<logger** name="org.mybatis" level="INFO"**></logger>**
28. **<logger** name="org.hibernate" level="INFO"**></logger>**
29. **<root** level="${log.level}"**>**
30. **<appender-ref** ref="Console"**/>**
31. **<appender-ref** ref="RollingFileInfo"**/>**
32. **</root>**
33. **</loggers>**
34. **</configuration>**

### 添加代码

#### Contants.java

在heima-leadnews-common工程中编写Contants.java类。

1. **package** com.heima.common.contants;
3. /\*\*
4. \* 项目工程通用常量定义类；
5. \* 如是模块通用，请定义在自模块常量类中；
6. \*/
7. **public** **class** Contants {
9. // 项目名称
10. **public** **static** String APP\_NAME = "APP\_NAME";
12. }

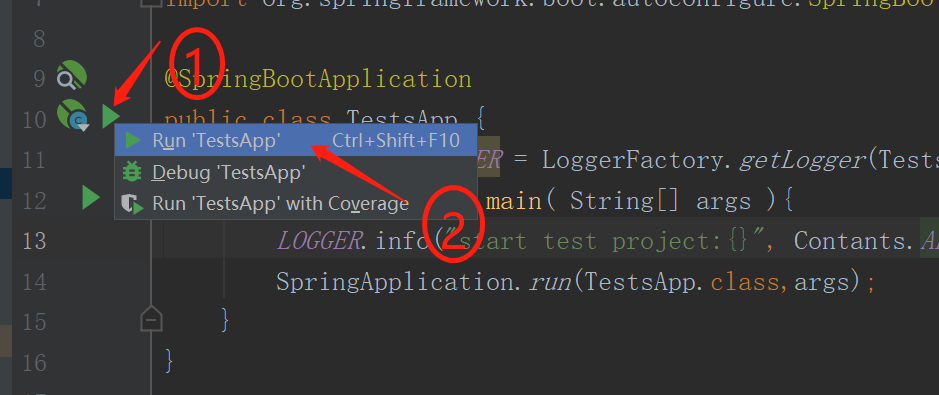
#### TestApp.java

在heima-leadnews-tests项目中编写TestApp.java类。

1. **package** com.heima.tests;
3. **import** com.heima.common.contants.Contants;
4. **import** org.slf4j.Logger;
5. **import** org.slf4j.LoggerFactory;
6. **import** org.springframework.boot.SpringApplication;
7. **import** org.springframework.boot.autoconfigure.SpringBootApplication;
9. @SpringBootApplication
10. **public** **class** TestsApp {
11. **static** Logger LOGGER = LoggerFactory.getLogger(TestsApp.**class**);
12. **public** **static** **void** main( String[] args ){
13. LOGGER.info("start test project:{}", Contants.APP\_NAME);
14. SpringApplication.run(TestsApp.**class**,args);
15. }
16. }

### 项目运行

#### 启动项目



#### 结果验证



