

Maven

1.Maven入门

1.1 什么是maven

1.2 maven安装

 1.2.1 解压Maven核心程序

 1.2.2 配置环境变量

 1.2.3 指定本地仓库

 1.2.4 配置阿里云提供的镜像仓库

1.3 Maven工程

 1.3.1 新建maven项目

 1.3.2 编写java代码

 1.3.3 导入junit依赖

 1.3.4 编写测试代码

1.4 maven命令

1.5 Maven依赖

 1.5.1 依赖流程

 1.5.2 新建HelloFriend项目

 1.5.3 导入依赖

 1.5.4 测试

 1.5.5 中文乱码

 1.5.6 maven远程仓库

 1.5.7 寻找坐标

1.6 Maven工作流程

1.7 依赖范围

1.8 Maven创建web工程

 1.8.1 新建maven web项目

 1.8.2 补全目录

 1.8.3 删除pom.xml中额外的配置

 1.8.4 web项目打包

1.8.5 骨架的web.xml约束有问题

1.9 maven的继承

1.新建子项目

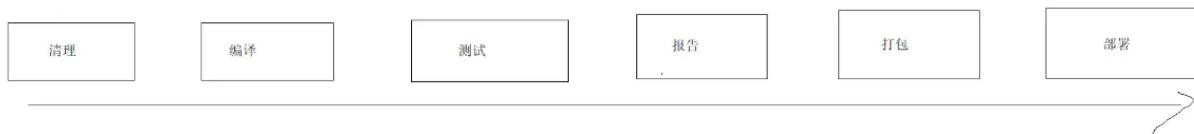
1.2 新建maven父工程

1.3 子工程继承父工程

1.Maven入门

1.1 什么是maven

Maven是Apache软件基金会组织维护的一款跨平台的项目管理工具，主要用于基于java平台的项目构建，依赖管理。



- 依赖管理: maven对项目的第三方构件（jar包）进行统一管理。向工程中加入jar包不要手工从其它地方拷贝，通过maven定义jar包的坐标，自动从maven仓库中去下载到工程中。
- 项目构建: maven提供一套对项目生命周期管理的标准，开发人员、和测试人员统一使用maven进行项目构建。项目生命周期管理：编译、测试、打包、部署、运行。
- maven对工程分模块构建，提高开发效率。

1.2 maven安装

1.2.1 解压Maven核心程序

核心程序压缩包：apache-maven-3.3.9-bin.zip，解压到**非中文、没有空格**的目录。例如：

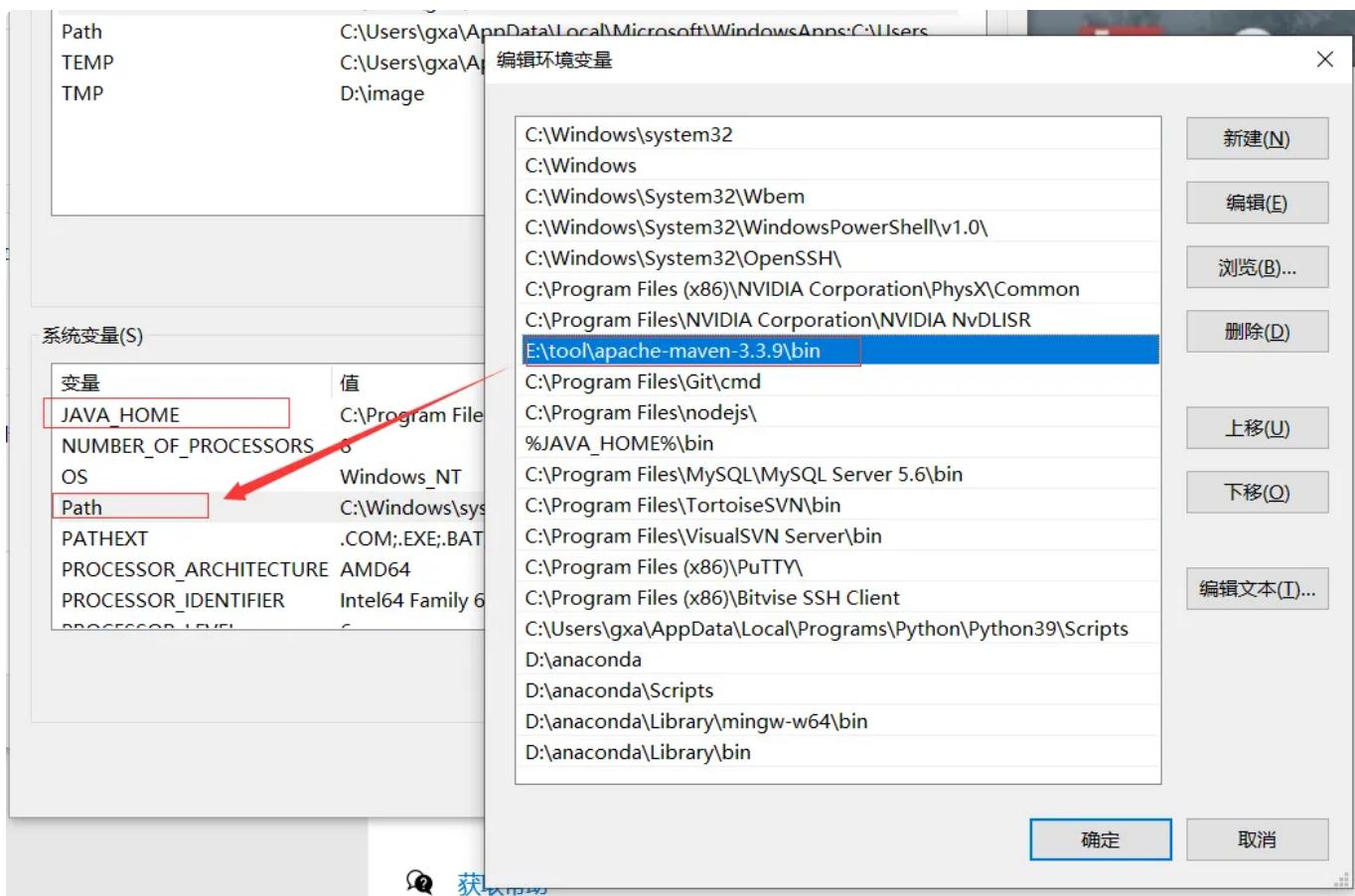
在解压目录中，我们需要着重关注Maven的核心配置文件：conf/settings.xml

bin	核心脚本文件	2021/1/20 8:59	文件夹
boot		2021/1/20 8:59	文件夹
conf	配置文件	2021/1/20 10:31	文件夹
lib		2021/1/20 8:59	文件夹
LICENSE		2015/11/10 11:44	文件 19 KB
NOTICE		2015/11/10 11:44	文件 1 KB
README.txt		2015/11/10 11:38	文本文档 3 KB

在解压目录中，我们需要着重关注Maven的核心配置文件：conf/settings.xml

名称	修改日期	类型	大小
logging	2021/1/20 8:59	文件夹	
settings.xml	2020/11/17 13:44	XML 文档	12 KB
toolchains.xml	2015/11/10 11:38	XML 文档	4 KB

1.2.2 配置环境变量



```
C:\Users\gxa>mvn -version
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
Maven home: E:\tool\apache-maven-3.3.9\bin\..
Java version: 1.8.0_181, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_181\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 10", version: "10.0", arch: "amd64", family: "dos"
```

1.2.3 指定本地仓库

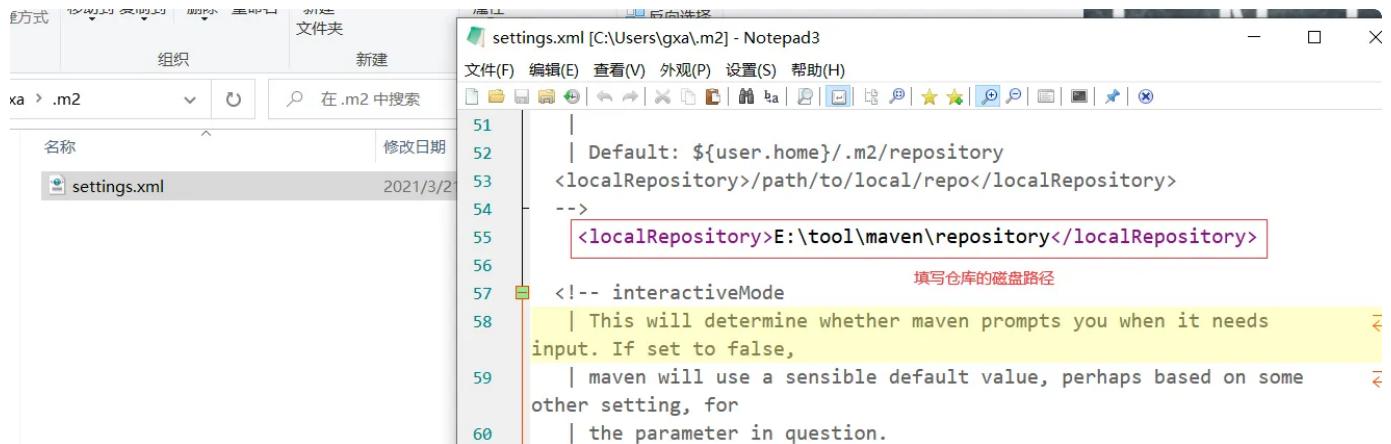
本地仓库默认值：用户家目录/.m2/repository。由于本地仓库的默认位置是在用户的家目录下，而家目录往往是在C盘，也就是系统盘。将来Maven仓库中jar包越来越多，仓库体积越来越大，可能会拖慢C盘运行速度，影响系统性能。所以建议将Maven的本地仓库放在其他盘符下。配置方式如下：

默认：

> 此电脑 > Windows-SSD (C:) > 用户 > gxa > .m2

名称	修改日期	类型	大小
repository	存放jar包	2022/6/13 15:30	文件夹

修改默认的路径:



先拷贝参考到指定的目录,如果不想配置就直接copy到c判断的.m2文件夹下面,但是如果磁盘空间有限,拷贝其它盘。

```
1  <!-- localRepository
2  | The path to the local repository maven will use to store artifacts.
3  |
4  | Default: ${user.home}/.m2/repository
5  <localRepository>/path/to/local/repo</localRepository>
6  -->
7  <localRepository>D:\maven-rep</localRepository>
```

本地仓库这个目录,我们手动创建一个空的目录即可。

记住:一定要把localRepository标签**从注释中拿出来**。

注意:本地仓库本身也需要使用一个**非中文、没有空格**的目录。

1.2.4 配置阿里云提供的镜像仓库

Maven下载jar包默认访问境外的中央仓库,速度很慢。改成阿里云提供的镜像仓库,访问国内网站,可以让Maven下载jar包的时候速度更快。配置的方式是:将下面mirror标签整体复制到mirrors标签的内部。

```
1
2 ▼ <mirrors>
3 ▼   <mirror>
4     <id>nexus-aliyun</id>
5     <mirrorOf>central</mirrorOf>
6     <name>Nexus aliyun</name>
7     <url>http://maven.aliyun.com/nexus/content/groups/public</url>
8   </mirror>
9 </mirrors>
```

1.3 Maven工程

maven的约定:

src/main/java 存放java代码

src/main/resources 存放配置文件

src/test/java 存放测试的java代码

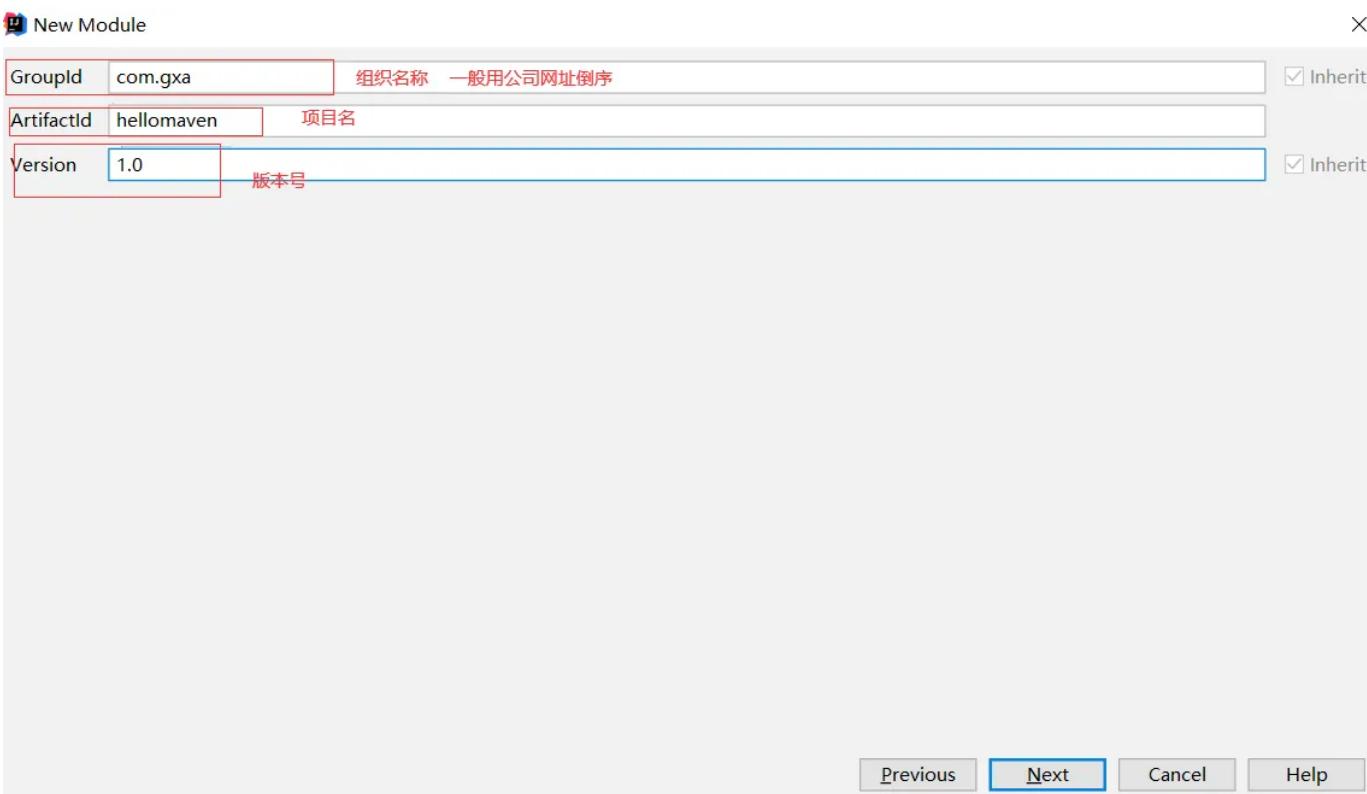
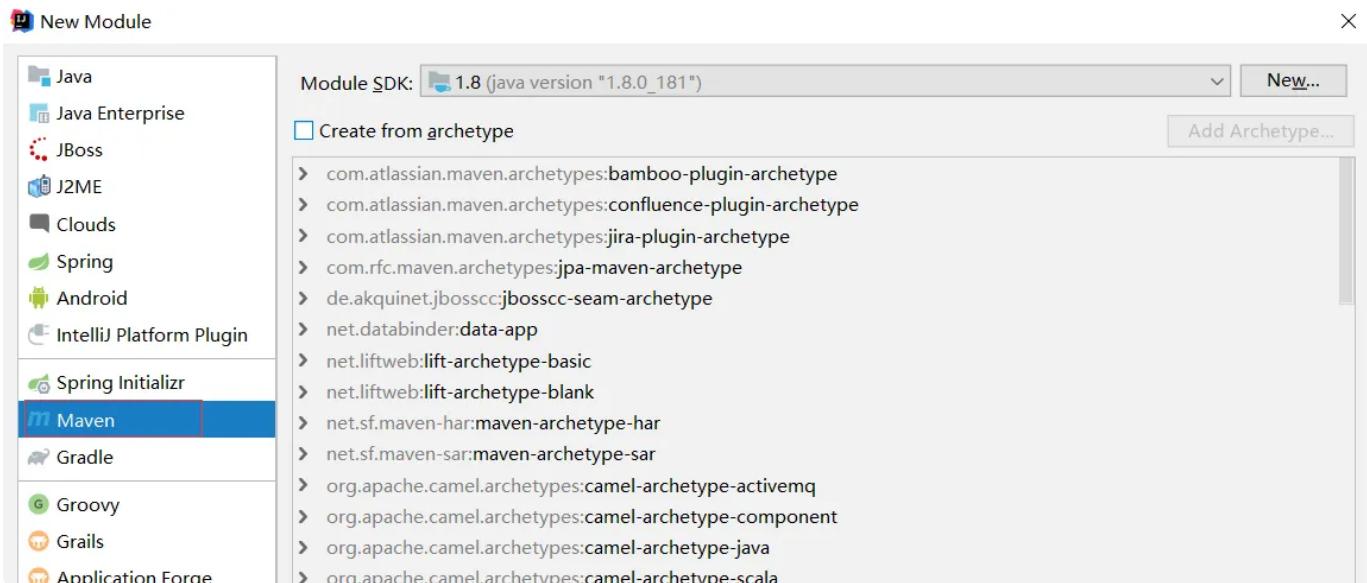
src/test/resources 存放测试的配置文件

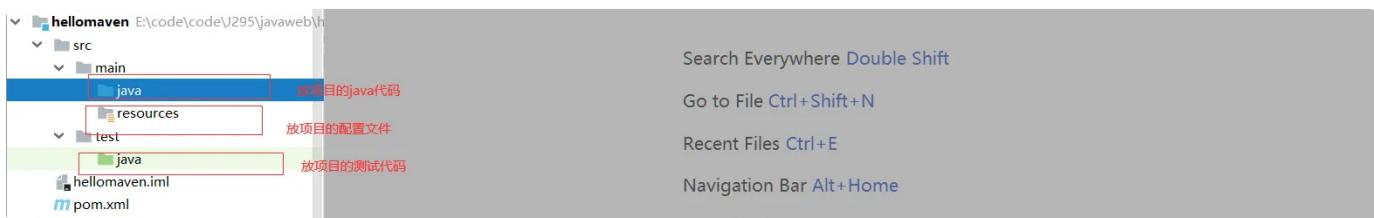
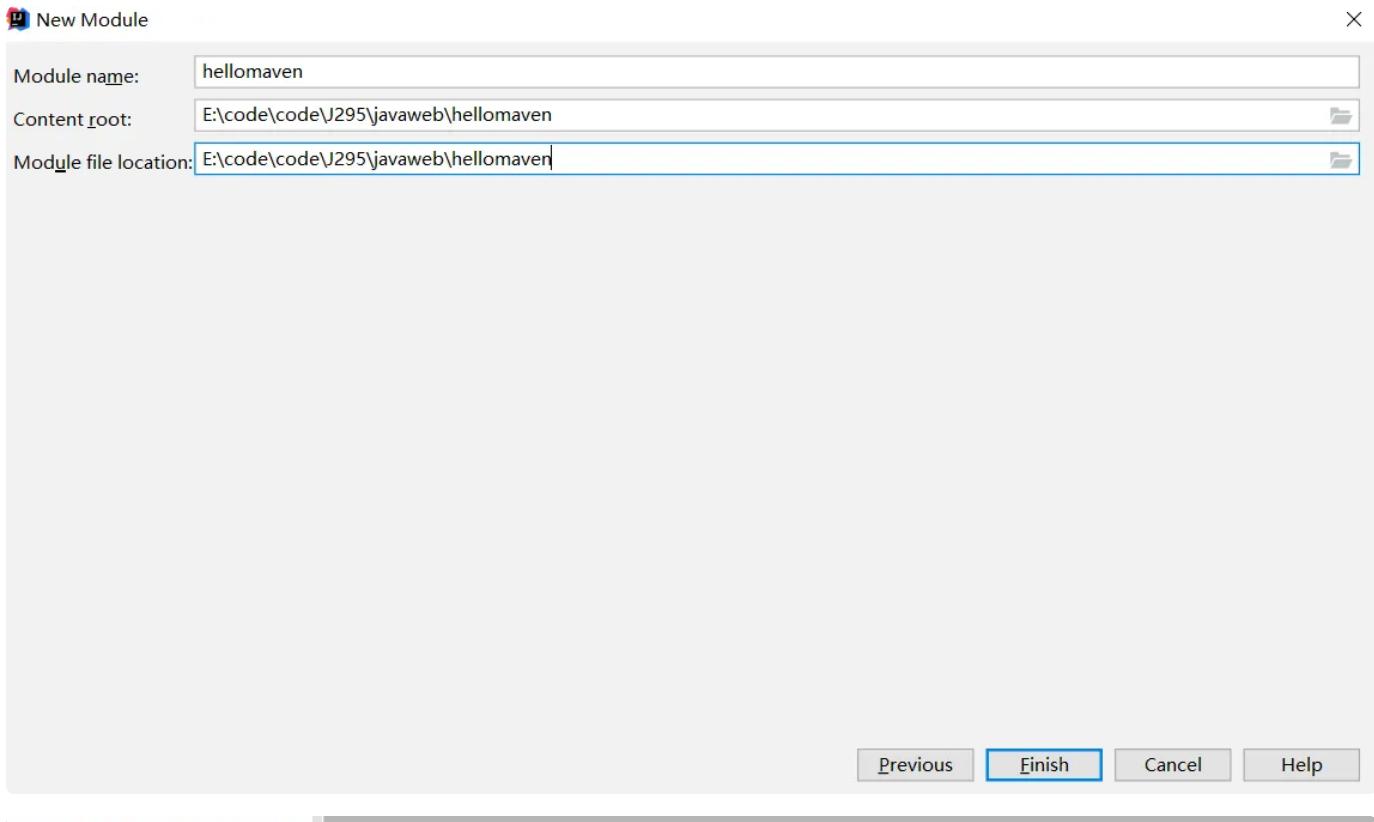
target 项目打包的输出文件

pom.xml maven项目的核心配置文件

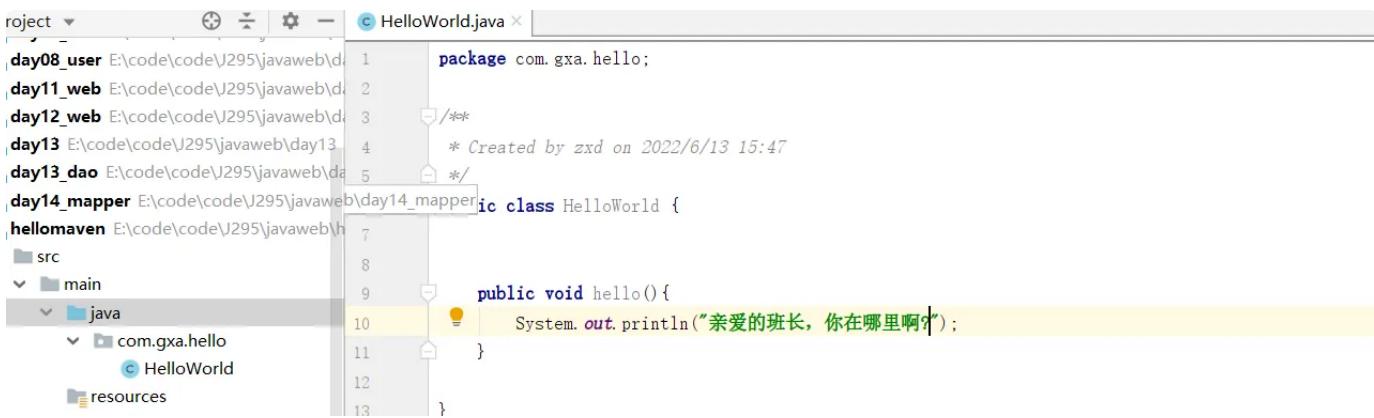
1.3.1 新建maven项目

Java项目 web项目





1.3.2 编写java代码



```
project ▾ + - ⚙ - c HelloWorld.java ×
day08_user E:\code\code\J295\javaweb\d
day11_web E:\code\code\J295\javaweb\d
day12_web E:\code\code\J295\javaweb\d
day13 E:\code\code\J295\javaweb\day13
day13_dao E:\code\code\J295\javaweb\da
day14_mapper E:\code\code\J295\javaweb\day14_mapperic class HelloWorld {
hellomaven E:\code\code\J295\javaweb\h
  src
    main
      java
        com.gxa.hello
          c HelloWorld
      resources
    test
      java
        pom.xml
```

```
1 package com.gxa.hello;
2
3 /**
4  * Created by zxd on 2022/6/13 15:47
5 */
6
7 public class HelloWorld {
8
9     public void hello(){
10         System.out.println("亲爱的班长，你在哪里啊？");
11     }
12 }
13 }
```

1.3.3 导入junit依赖

一个maven项目有一个 pom.xml文件 这个文件就是这个maven项目的配置文件

```
▼ pom.xml Java

1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>com.gxa</groupId>
8      <artifactId>hellomaven</artifactId>
9      <version>1.0</version>
10
11
12      <dependencies>
13
14          <!--导入测试的jar包-->
15          <dependency>
16              <groupId>junit</groupId>
17              <artifactId>junit</artifactId>
18              <version>4.12</version>
19              <scope>test</scope>
20          </dependency>
21
22      </dependencies>
23
24
25  </project>
```

1.3.4 编写测试代码

```

1  /**
2   * Created by zxd on 2022/6/13 15:48
3   */
4  public class TestHello {
5
6
7      @Test
8      public void testHello(){
9
10         HelloWorld helloWorld=new HelloWorld();
11         helloWorld.hello();
12
13     }
14
15 }
16

```



1.4 maven命令

1、清理操作

mvn clean

效果：删除target目录

```
E:\code\code\J295\javaweb\hellomaven>mvn clean
```

2、编译操作

主程序编译: mvn compile

```
en > target      在 target 中
名称
  classes
  generated-sources
  maven-status
修@ C:\Users\gxa\.m2\settings.xml, line 314, column 11
[WARNING] Unrecognised tag: 'mirror' (position: START_TAG seen ...</a
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building hellomaven 1.0
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ hell
[WARNING] Using platform encoding (GBK actually) to copy filtered resou
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ hell
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK
[INFO] Compiling 1 source file to E:\code\code\J295\javaweb\hellomave
[WARNING] /E:/code/code/J295/javaweb/hellomaven/src/main/java/com/gxa
符
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.993 s
[INFO] Finished at: 2022-06-13T15:58:19+08:00
[INFO] Final Memory: 15M/297M
[INFO] -----
```

E:\code\code\J295\javaweb\hellomaven>mvn compile

测试程序编译: mvn test-compile

```
test-classes      在 test-classes 中
名称
  TestHello.class
修@ C:\Users\gxa\.m2\settings.xml, line 314, column 11
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building hellomaven 1.0
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ hell
[WARNING] Using platform encoding (GBK actually) to copy filtered resou
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile)
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResourc
[WARNING] Using platform encoding (GBK actually) to copy filtered resou
[INFO] skip non existing resourceDirectory E:\code\code\J295\javaweb\he
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile)
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding GBK
[INFO] Compiling 1 source file to E:\code\code\J295\javaweb\he
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.015 s
[INFO] Finished at: 2022-06-13T15:59:40+08:00
[INFO] Final Memory: 15M/298M
[INFO] -----
```

E:\code\code\J295\javaweb\hellomaven>mvn test-compile

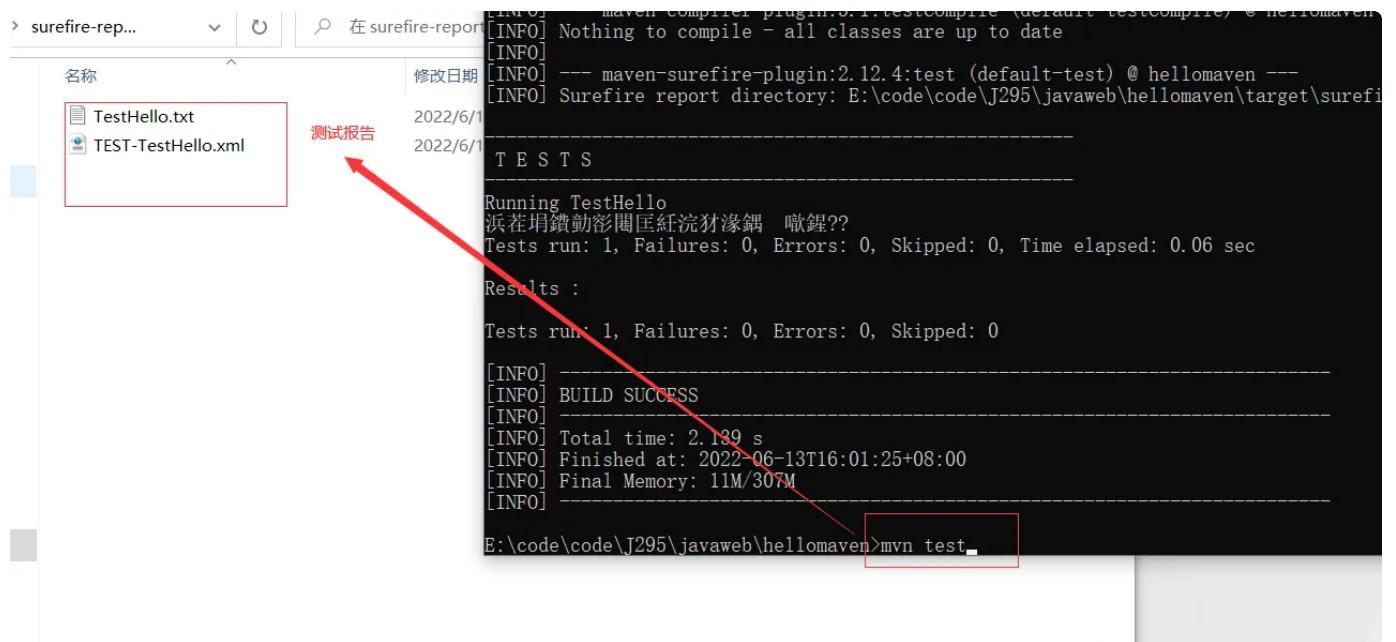
主体程序编译结果存放的目录：target/classes

测试程序编译结果存放的目录：target/test-classes

3、测试操作

mvn test

测试的报告会存放在target/surefire-reports目录下



```
[INFO]           maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ helломавен
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ helломавен ---
[INFO] Surefire report directory: E:\code\code\J295\javaweb\хелломавен\target\surefire-reports

T E S T S

Running TestHello
浜茬堦饋動彆闇匡絃浣犖祫鐸???
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.06 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 2.189 s
[INFO] Finished at: 2022-06-13T16:01:25+08:00
[INFO] Final Memory: 11M/307M
[INFO]

E:\code\code\J295\javaweb\хелломавен>mvn test
```

```
Test set: TestHello
-----
Tests run: 2, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.055 sec <<< FAILURE!
testHello2(TestHello)  Time elapsed: 0.002 sec  <<< ERROR!
java.lang.ArithmaticException: / by zero
    at TestHello.testHello2(TestHello.java:22)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.junit.runners.model.FrameworkMethod$1.runReflectiveCall(FrameworkMethod.java:50)
    at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12)
    at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:47)
    at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:17)
    at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)
    at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)
    at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)
    at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)
    at org.junit.runners.ParentRunner$1.schedule(ParentRunner.java:71)
    at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
    at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)
    at org.junit.runners.ParentRunner$2.evaluate(ParentRunner.java:268)
```

mvn 在运行后面的命令的时候正常情况下会自动运行前面的命令 但是有的时候会抽风

手动 maven clean一下

4、打包操作

mvn package

打包的结果会存放在target目录下



名称	修改日期	类型	大小
classes	2022/6/13 16:20	文件夹	
generated-sources	2022/6/13 16:20	文件夹	
generated-test-sources	2022/6/13 16:20	文件夹	
maven-archiver	2022/6/13 16:20	文件夹	
maven-status	2022/6/13 16:20	文件夹	
surefire-reports	2022/6/13 16:20	文件夹	
test-classes	2022/6/13 16:20	文件夹	
hellomaven-1.0.jar	2022/6/13 16:20	Executable Jar File	3 KB

5、安装操作

mvn install

The screenshot shows a terminal window with the title "ellomaven > 1.0". The command "mvn install" was run, resulting in the following output:

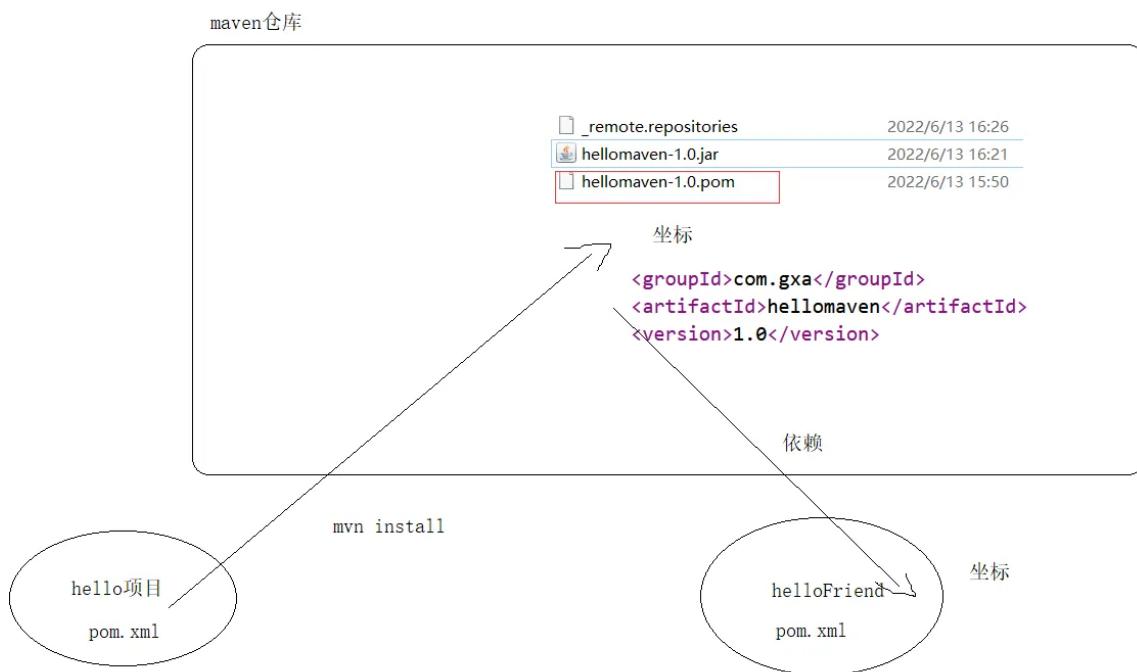
```
Running TestHello
浜菴堀鑽勤巣閻匡紓浣犖濛鏞 嘴鍾??
浜菴堀鑽勤巣閻匡紓浣犖濛鏞 嘴鍾??
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.057 sec
Results :
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ hellomaven ---
[INFO] Building jar: E:\code\code\J295\javaweb\hellomaven\target\hellomav
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ hellomave
[INFO] Installing E:\code\code\J295\javaweb\hellomaven\target\hellomaven-
lomaven\1.1\hellomaven-1.1.jar
[INFO] Installing E:\code\code\J295\javaweb\hellomaven\pom.xml to E:\too
aven-1.1.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.985 s
[INFO] Finished at: 2022-06-13T16:28:15+08:00
[INFO] Final Memory: 12M/307M
[INFO]

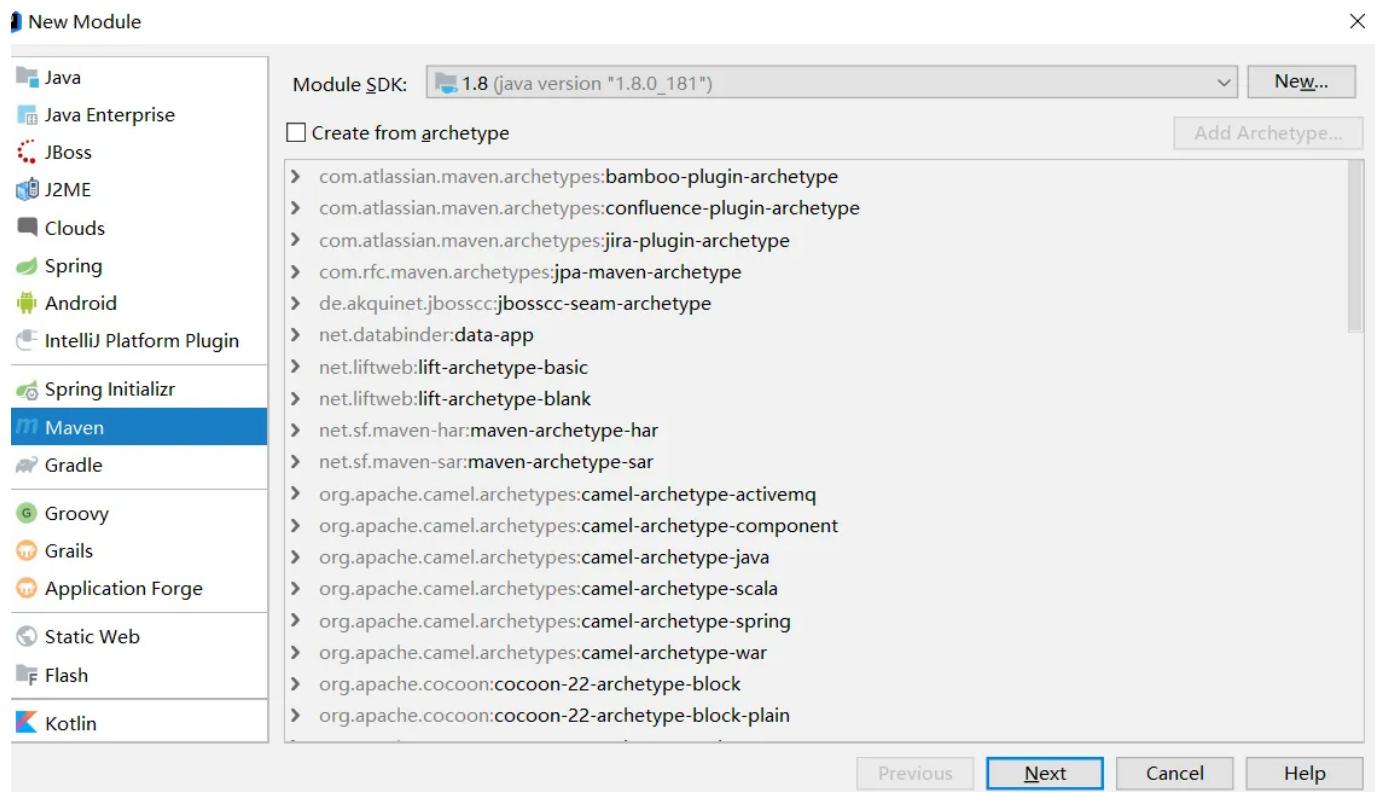
E:\code\code\J295\javaweb\hellomaven>mvn install
```

1.5 Maven依赖

1.5.1 依赖流程



1.5.2 新建HelloFriend项目



New Module

Add as module to <none>

Parent <none>

GroupId com.gxa Inherit

ArtifactId hellofriend

Version 1.0 Inherit

New Module

Module name: hellofriend

Content root: E:\code\code\J295\javaweb\hellofriend

Module file location: E:\code\code\J295\javaweb\hellofriend

1.5.3 导入依赖

```
    pom.xml                                     Java |  
1  <?xml version="1.0" encoding="UTF-8"?>  
2  <project xmlns="http://maven.apache.org/POM/4.0.0"  
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
4      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://mave  
n.apache.org/xsd/maven-4.0.0.xsd">  
5      <modelVersion>4.0.0</modelVersion>  
6  
7      <groupId>com.gxa</groupId>  
8      <artifactId>hellofriend</artifactId>  
9      <version>1.0</version>  
10  
11  
12      <!--导入依赖-->  
13      <dependencies>  
14  
15          <dependency>  
16              <groupId>com.gxa</groupId>  
17              <artifactId>hellomaven</artifactId>  
18              <version>1.0</version>  
19          </dependency>  
20  
21          <!--导入测试的jar包-->  
22          <dependency>  
23              <groupId>junit</groupId>  
24              <artifactId>junit</artifactId>  
25              <version>4.12</version>  
26              <scope>test</scope>  
27          </dependency>  
28  
29  
30  
31      </dependencies>  
32  
33  
34  </project>
```

1.5.4 测试

The screenshot shows an IDE interface with two main panes. The left pane displays a Java file named `TestHello.java` with the following code:

```
3  * /ok
4  * Created by zxd on 2022/6/13 15:48
5  */
6
7  public class TestHello {
8
9
10 }
11
12
13
14
15
16
17
18  @Test
19  public void testHello(){
20
21      HelloWorld helloWorld=new HelloWorld();
22      helloWorld.hello();
23
24  }
25
26 }
```

The right pane shows the Maven dependency tree for the project `hellomaven`:

```
> Profiles
  > hellomaven
    > Lifecycle
    > Plugins
    > Dependencies
      > com.gxa:hellomaven:1.0
        > junit:junit4.12 (test)
    > hellomaven
```

A red arrow points from the `junit:junit4.12 (test)` dependency in the Maven tree to the `@Test` annotations in the `TestHello.java` code.

1.5.5 中文乱码

maven工程默认在打包的时候会出现中文乱码

在hellomaven项目配置打包的插件，指定用UTF-8编码进行打包

pom.xml

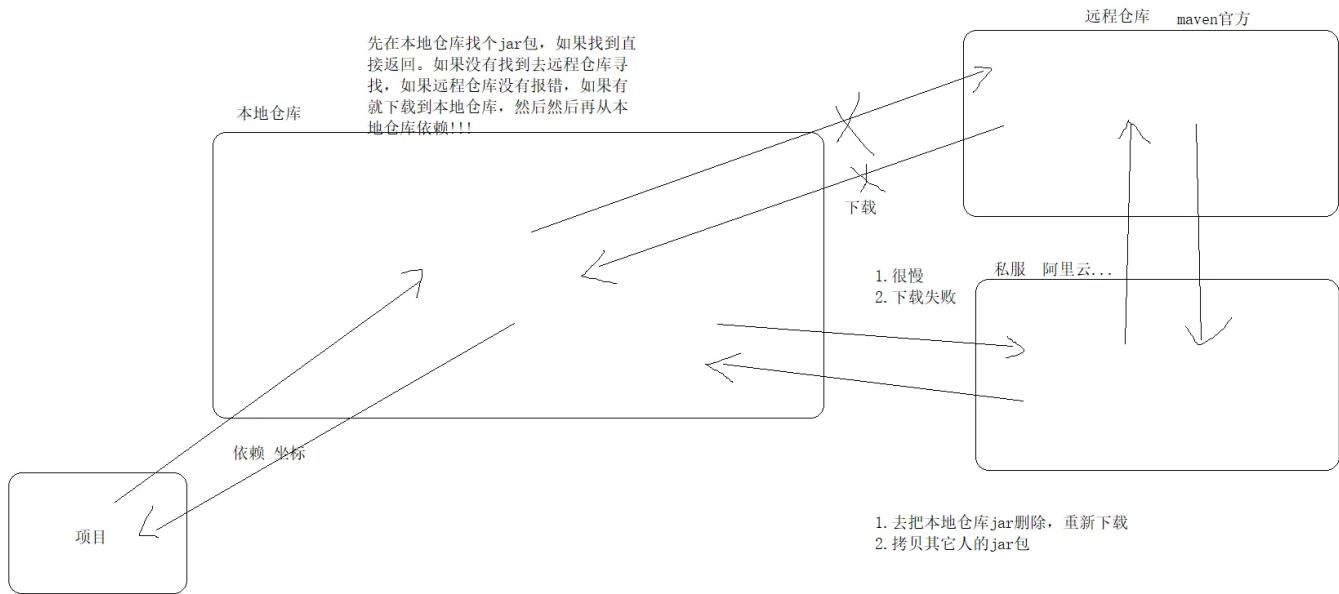
Java

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>com.gxa</groupId>
8      <artifactId>hellomaven</artifactId>
9      <version>1.0</version>
10
11     <!--配置maven项目的编码-->
12     <properties>
13         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
14     </properties>
15
16
17     <dependencies>
18
19         <!--导入测试的jar包-->
20         <dependency>
21             <groupId>junit</groupId>
22             <artifactId>junit</artifactId>
23             <version>4.12</version>
24             <scope>test</scope>
25         </dependency>
26
27     </dependencies>
28
29     <!--编译插件-->
30     <build>
31         <plugins>
32             <plugin>
33                 <groupId>org.apache.maven.plugins</groupId>
34                 <artifactId>maven-compiler-plugin</artifactId>
35                 <version>3.1</version>
36                 <configuration>
37                     <target>1.8</target>
38                     <source>1.8</source>
39                     <encoding>${project.build.sourceEncoding}</encoding>
40                 </configuration>
41             </plugin>
42         </plugins>
43     </build>
44
```

45

</project>

1.5.6 maven远程仓库



1.5.7 寻找坐标

<https://search.maven.org/>

druid

Group ID	Artifact ID	Latest Version	Updated	OSS Index	Download
com.alibaba	druid	1.2.11	(99+)	13-Jun-2022	
io.github.linceln	druid	1.1.11	(2)	29-Apr-2022	
org.apache.druid	druid	0.22.1	(20)	11-Dec-2021	
io.shulie	druid	1.2.5.7	(8)	04-Nov-2021	
org.azolla.com.alibaba	druid7_preview_05_Azolla02	(2)	25-Jul-2021	
com.weicoder	druid	3.3.8-jdk8	(4)	07-May-2020	
com.pig4cloud	druid	0.1.21	(1)	03-Nov-2019	
io.druid	druid	0.12.3	(21)	09-Oct-2018	
com.n3twork.druid	druid	0.6.105-hadoop2.4.0	(1)	29-May-2014	
org.apache.druid	druid-benchmarks	0.22.1	(20)	11-Dec-2021	

javaweb [E:\code\code\J295\javaweb] - hellotriend - IntelliJ IDEA

Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

hellotriend > pom.xml

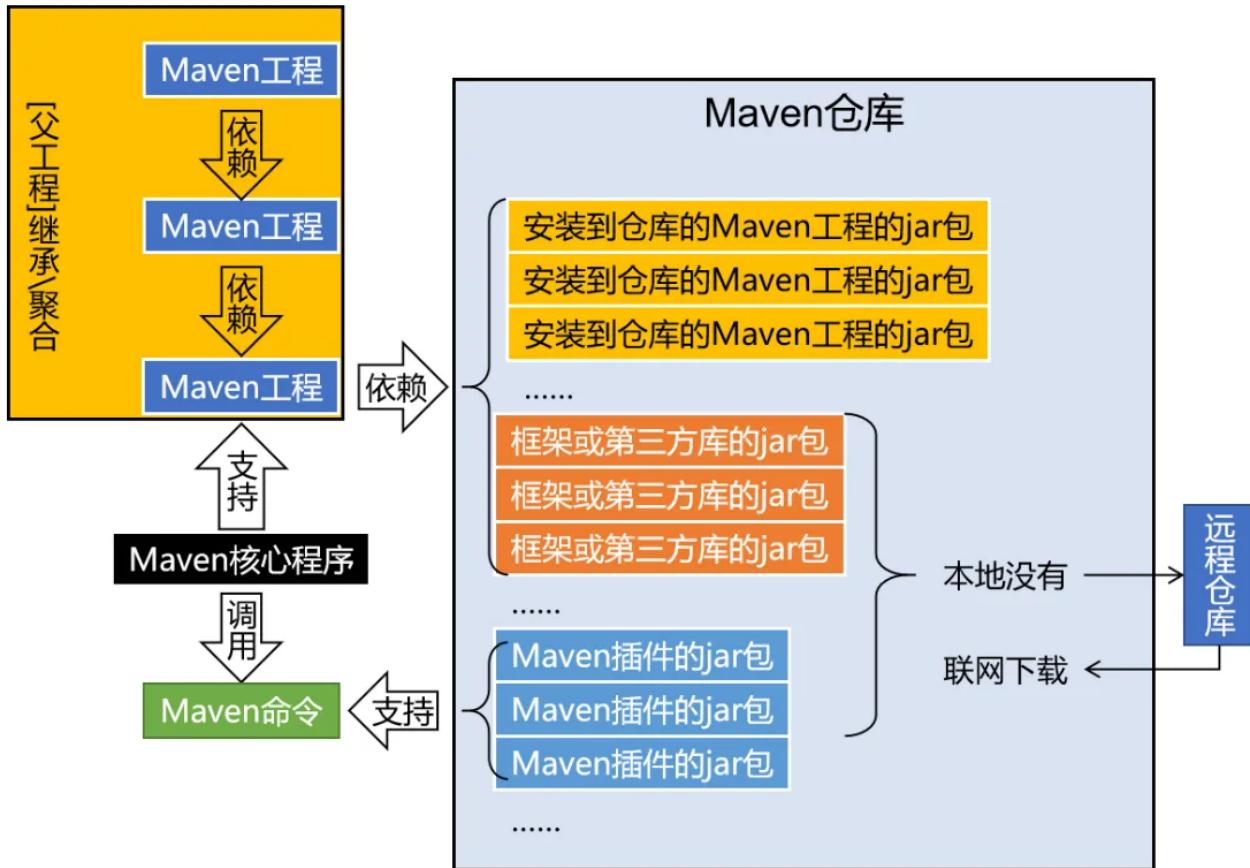
```

Project  TestHello.java  hellomaven  hellotriend
crud E:\code\code\J295\javaweb\crud
day04_web E:\code\code\J295\javaweb\da
day05_web E:\code\code\J295\javaweb\da
day06_web E:\code\code\J295\javaweb\da
day07_web E:\code\code\J295\javaweb\da
day08_user E:\code\code\J295\javaweb\da
day11_web E:\code\code\J295\javaweb\da
day12_web E:\code\code\J295\javaweb\da
day13 E:\code\code\J295\javaweb\day13
day13_dao E:\code\code\J295\javaweb\da
day14_mapper E:\code\code\J295\javaweb\da
hellotriend E:\code\code\J295\javaweb\hel
  src
    main
      java
      resources
    test
      java
        TestHello
  target
  hellotriend.iml
  pom.xml
hellomaven E:\code\code\J295\javaweb\he
  src

16   <groupId>com.gxa</groupId>
17   <artifactId>hellomaven</artifactId>
18   <version>1.0</version>
19   </dependency>
20
21   <!--导入测试的jar包-->
22   <dependency>
23     <groupId>junit</groupId>
24     <artifactId>junit</artifactId>
25     <version>4.12</version>
26     <scope>test</scope>
27   </dependency>
28
29   <dependency>
30     <groupId>com.alibaba</groupId>
31     <artifactId>druid</artifactId>
32     <version>1.2.10</version>
33   </dependency>
34
35   </dependencies>
36
37
38

```

1.6 Maven工作流程



仓库名称	作用
本地仓库	相当于缓存，工程第一次会从远程仓库（互联网）去下载jar包，将jar包存在本地仓库（在程序员的电脑上）。第二次不需要从远程仓库去下载。先从本地仓库找，如果找不到才会去远程仓库找。
中央仓库	就是远程仓库，仓库中jar由专业团队（maven团队）统一维护。中央仓库的地址： http://repo1.maven.org/maven2/

远程仓库(私服)	私服是一种特殊的远程仓库，它是架设在局域网内的仓库服务，私服代理广域网上的远程仓库，供局域网内的Maven用户使用。当Maven需要下载构件的时候。它从私服请求，假设私服上不存在该构件，则从外部的远程仓库下载，缓存在私服上之后，再为Maven的下载请求提供服务。
----------	---

1.7 依赖范围

依赖范围	对于编译 classpath 有效	对于测试 classpath 有效	对于运行时 classpath 有效	例子
compile	Y	Y	Y	spring-core
test	-	Y	-	Junit
provided	Y	Y	-	servlet-api
runtime	-	Y	Y	JDBC驱动

① compile 和 test 对比

	main目录 (空间)	test目录 (空间)	开发过程 (时间)	部署到服务器 (时间)
compile	有效	有效	有效	有效
test	无效	有效	有效	无效

②compile和provided对比

	main目录 (空间)	test目录 (空间)	开发过程 (时间)	部署到服务器 (时间)
compile	有效	有效	有效	有效
provided	有效	有效	有效	无效

③结论

1.compile：通常使用的第三方框架的jar包这样在项目实际运行时真正要用到的jar包都是以compile范围进行依赖的。比如SSM框架所需jar包。默认不配置就是compile

2.test：测试过程中使用的jar包，以test范围依赖进来。比如junit。

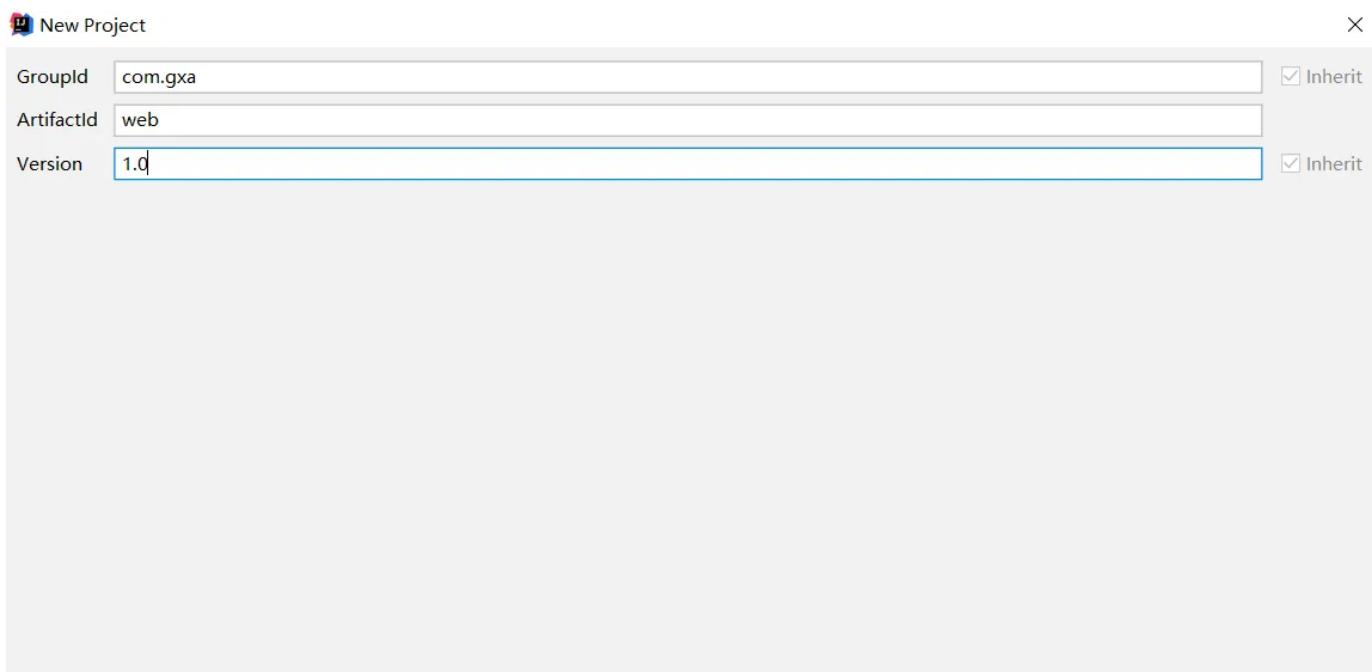
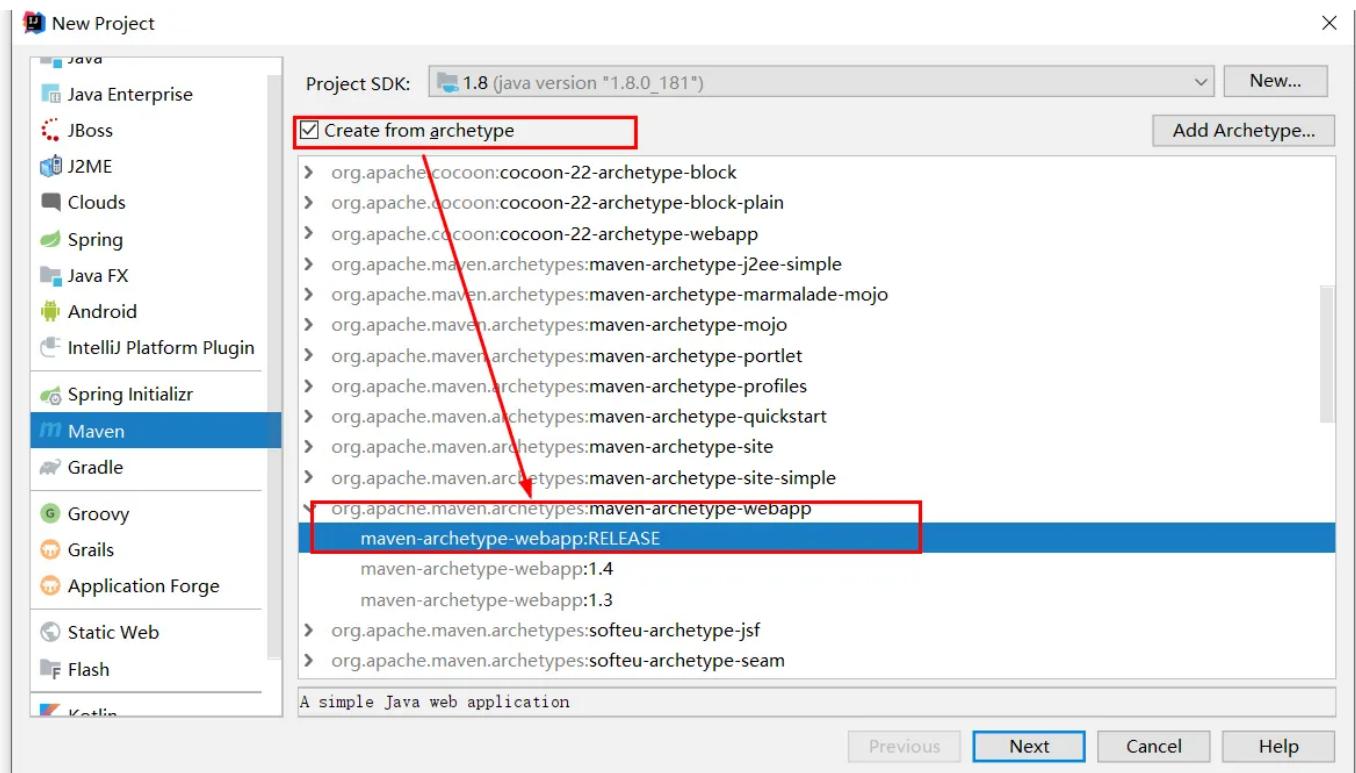
3.provided：在开发过程中需要用到的“服务器上的jar包”通常以provided范围依赖进来。比如servlet-api、jsp-api。而这个范围的jar包之所以不参与部署、不放进war包，就是避免和服务器上已有的同类jar包产生冲突，同时减轻服务器的负担。说白了就是：“服务器上已经有了，你就别带啦！”

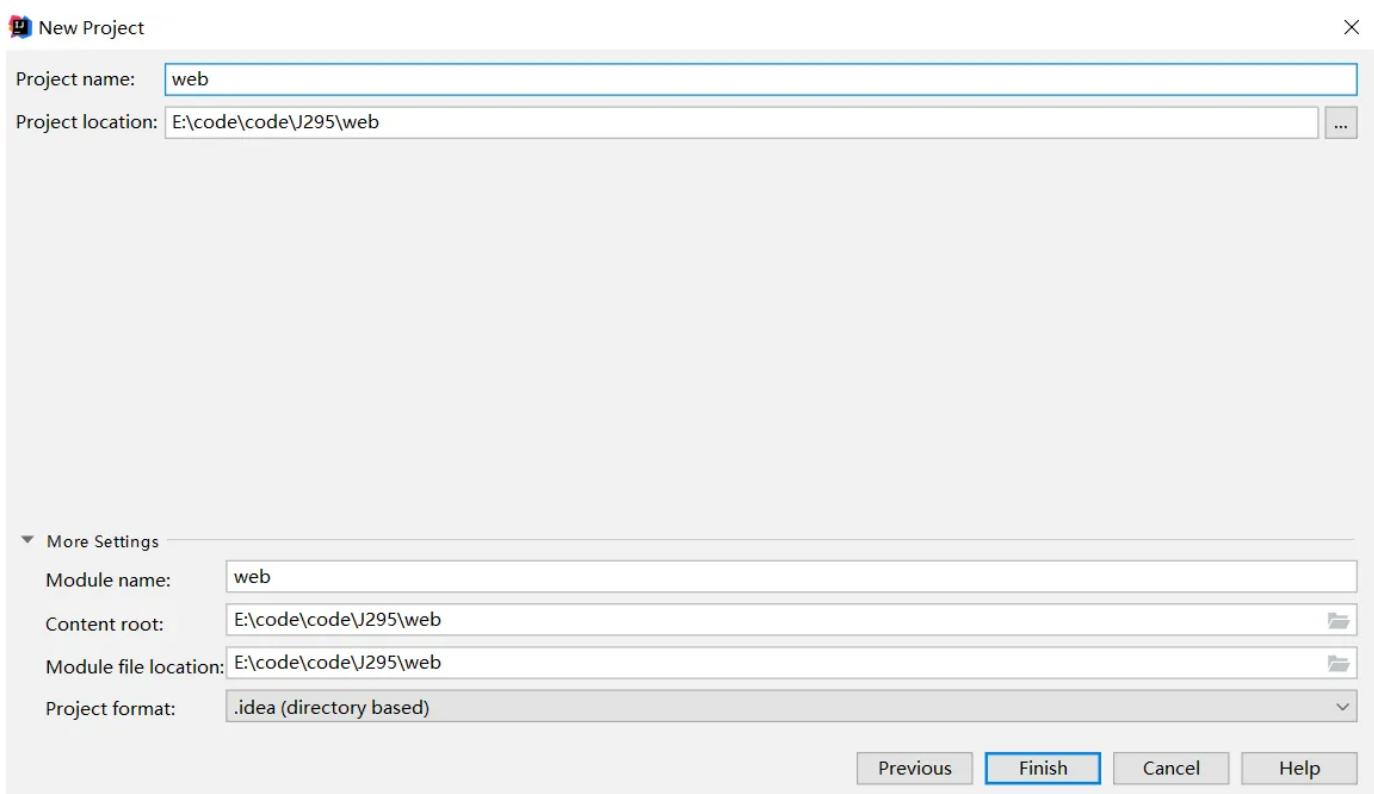
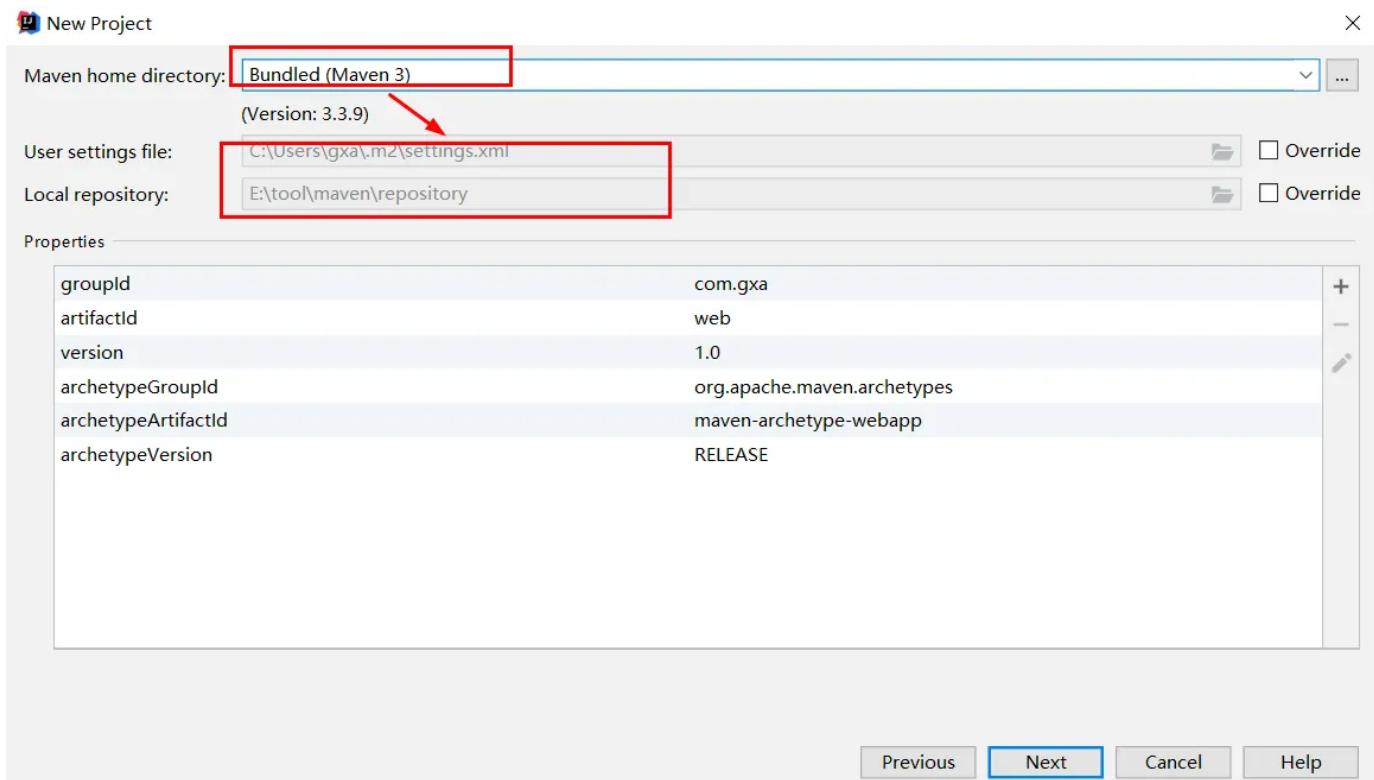
1.8 Maven创建web工程

一个web项目打包 war包，可以直接放在web服务器的webapp下面，web服务器自动解压部署

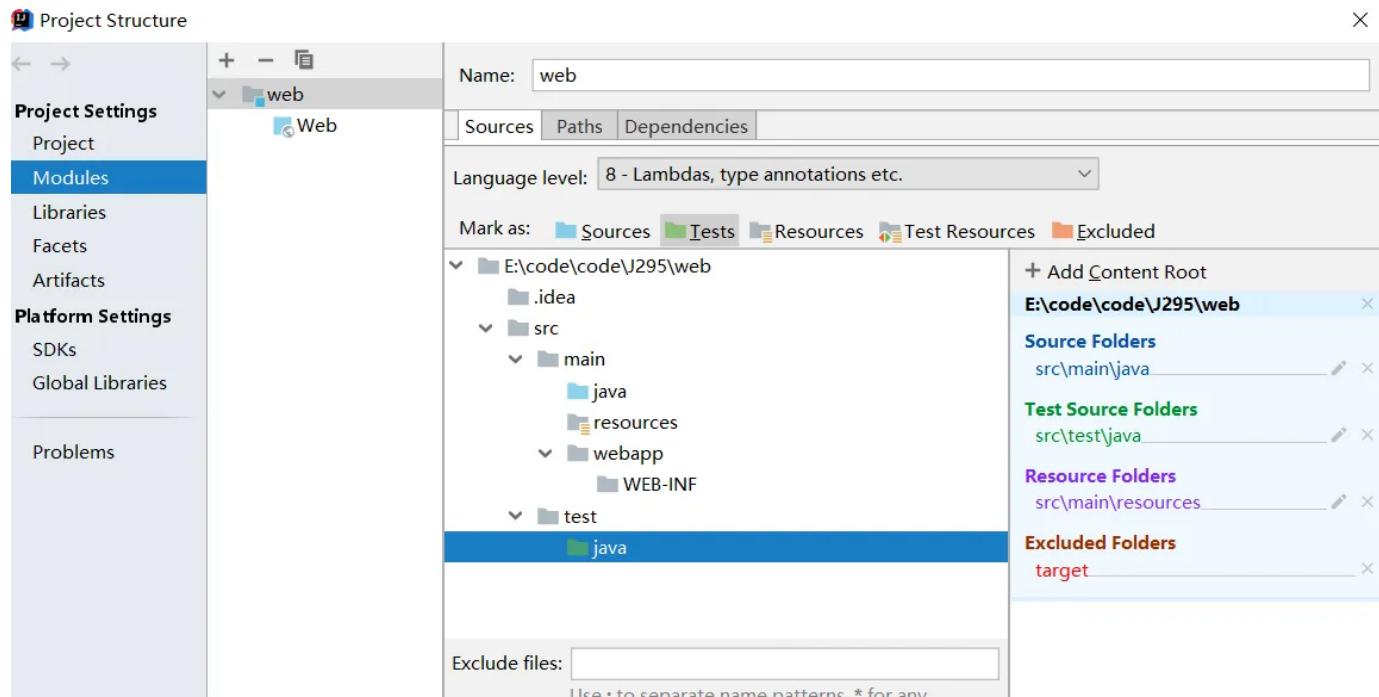
骨架：maven官方给我提供搭建项目的模板，从maven的远程仓库下载到本地仓库

1.8.1 新建maven web项目





1.8.2 补全目录



1.8.3 删掉pom.xml中额外的配置

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.gxa</groupId>
8   <artifactId>web</artifactId>
9   <version>1.0</version>
10  <!--web项目 部署到tomcat里面-->
11  <packaging>war</packaging>
12
13
14 </project>
15
```

1.8.4 web项目打包

The screenshot shows a Java IDE interface with the following components:

- Project Explorer:** On the left, it displays a tree view of the project structure:
 - maven-archiver
 - maven-status
 - web-1.0
 - web-1.0.war** (highlighted with a red box)
 - pom.xml
 - web.iml
- Maven Project View:** A central panel showing the Maven pom.xml file content:

```
<groupId>com.gxa</groupId>
<artifactId>web</artifactId>
<version>1.0</version>
<packaging>war</packaging>
```

Line numbers 7 through 14 are visible.
- Terminal:** At the bottom, the terminal window shows the command and its execution:

```
[INFO] -----
[INFO] Total time: 1.469 s
[INFO] Finished at: 2022-06-14T09:42:01+08:00
[INFO] Final Memory: 16M/211M
[INFO] -----
```

The command `E:\code\code\J295\web>mvn package` is shown at the bottom of the terminal, with a red box highlighting the path and the command itself.

1.8.5 骨架的web.xml约束有问题

The screenshot shows an IDE interface with a project tree on the left and a code editor on the right. The code editor displays the contents of a `web.xml` file. A red box highlights the first few lines of the XML, which define the document type as a Sun Microsystems DTD from 2003. A tooltip or status bar at the top right indicates that the version is too low and that `web.xml` typically does not use DTD constraints.

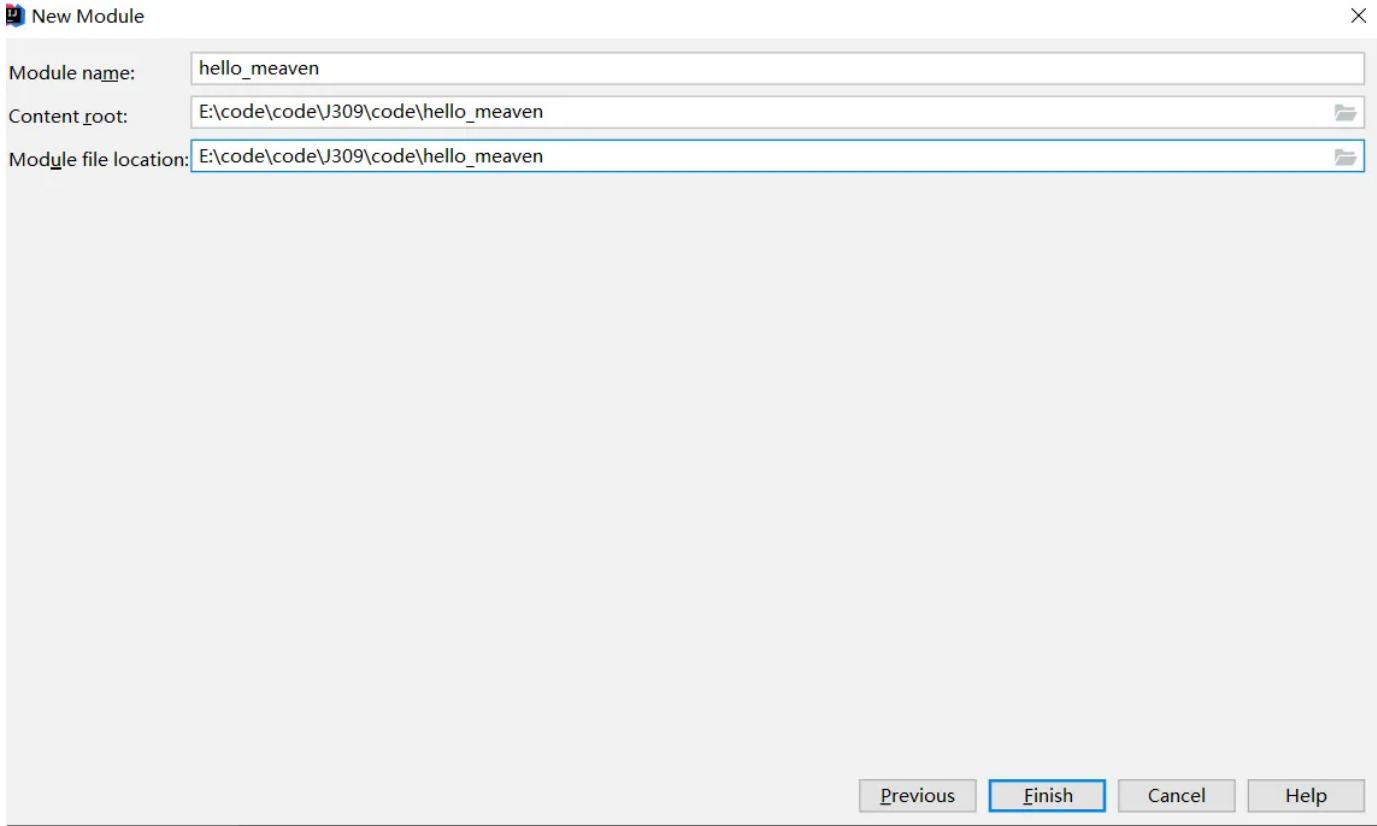
```
<!DOCTYPE web-app PUBLIC  
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
"http://java.sun.com/dtd/web-app_2_3.dtd" >  
  
<web-app>  
    <display-name>Archetype Created Web Application</display-name>  
</web-app>
```

This screenshot shows the same IDE environment with a modern `web.xml` file. The XML declaration uses the `version` and `encoding` attributes, and includes namespace declarations for Java EE and XML Schema. The XML schema location is also specified.

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"  
         version="4.0">  
  
</web-app>
```

1.9 maven的继承

1.新建子项目



pom.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>com.gxa</groupId>
8     <artifactId>hello_meaven</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11    <dependencies>
12
13      <dependency>
14        <groupId>com.alibaba</groupId>
15        <artifactId>druid</artifactId>
16        <version>1.2.14</version>
17      </dependency>
18
19      <dependency>
20        <groupId>org.mybatis</groupId>
21        <artifactId>mybatis</artifactId>
22        <version>3.5.11</version>
23      </dependency>
24
25    </dependencies>
26
27
28  </project>

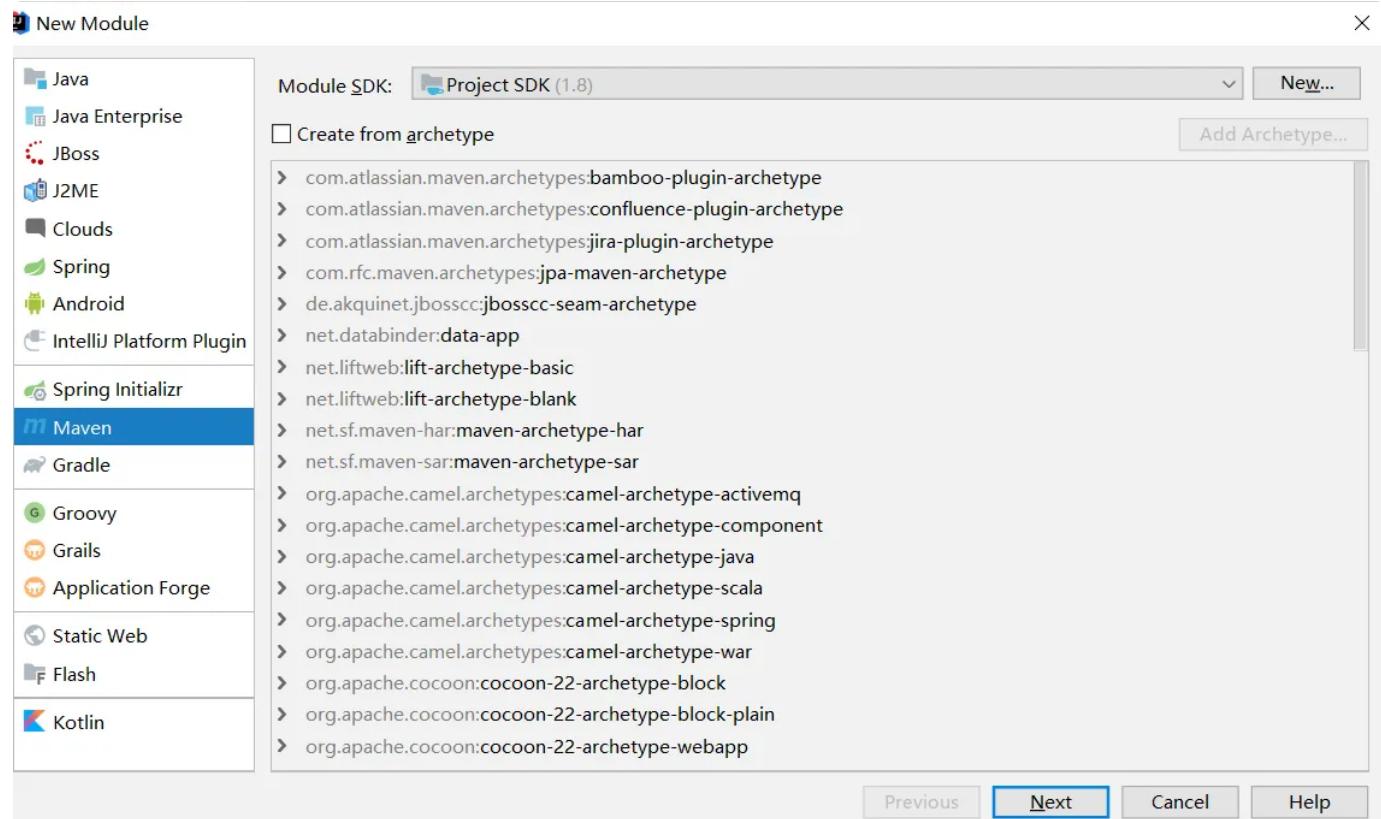
```

思考?

我们有很多的项目，使用到了相同的maven配置。

而是把maven的配置如果能放到一个工程，其它工程都去继承过来，很完美了

1.2 新建maven父工程



New Module

Add as module to <none>

Parent <none> ...
GroupId com.gxa ...
ArtifactId parent Inherit
Version 1.0-SNAPSHOT Inherit

Previous Next Cancel Help

New Module

Module name: parent

Content root: E:\code\code\J309\code\parent

Module file location: E:\code\code\J309\code\parent

Previous Finish Cancel Help

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7      <groupId>com.gxa</groupId>
8      <artifactId>parent</artifactId>
9      <version>1.0-SNAPSHOT</version>
10     <!--打包后只有pom的内容-->
11     <packaging>pom</packaging>
12
13     <dependencies>
14
15         <dependency>
16             <groupId>com.alibaba</groupId>
17             <artifactId>druid</artifactId>
18             <version>1.2.14</version>
19         </dependency>
20
21         <dependency>
22             <groupId>org.mybatis</groupId>
23             <artifactId>mybatis</artifactId>
24             <version>3.5.11</version>
25         </dependency>
26
27     </dependencies>
28
29
30 </project>
```

The screenshot shows a Java development environment with a project tree on the left and a code editor on the right. The project tree includes modules like day01, day02, day03, day04_mybatis, day05_mapper, day07, day08, day08_web, day10, day11, and hello_meaven. A module named 'parent' is expanded, showing its .iml and pom.xml files. The 'External Libraries' section lists JRE 1.8, Maven, and Tomcat. The code editor displays the pom.xml file for the 'hello_meaven' module, which defines the project's XML structure, group ID, artifact ID, version, model version, dependencies on com.alibaba:druid and org.mybatis:mybatis, and packaging type pom.

```
<?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.gxa</groupId>
    <artifactId>parent</artifactId>
    <version>1.0-SNAPSHOT</version>
    <!--打包后只有pom的内容-->
    <packaging>pom</packaging>

    <dependencies>
        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>druid</artifactId>
            <version>1.2.14</version>
        </dependency>

        <dependency>
            <groupId>org.mybatis</groupId>
            <artifactId>mybatis</artifactId>
            <version>3.5.11</version>
        </dependency>
    </dependencies>

```

1.3 子工程继承父工程

pom.xml

XML |

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6
7
8      <parent>
9          <groupId>com.gxa</groupId>
10         <artifactId>parent</artifactId>
11         <version>1.0-SNAPSHOT</version>
12     </parent>
13
14
15     <groupId>com.gxa</groupId>
16     <artifactId>hello_meaven</artifactId>
17     <version>1.0-SNAPSHOT</version>
18
19     <dependencies>
20         <dependency>
21             <groupId>commons-dbcp</groupId>
22             <artifactId>commons-dbcp</artifactId>
23             <version>20030825.184428</version>
24         </dependency>
25     </dependencies>
26
27
28 </project>
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

parent

```

<parent>
    <groupId>com.gxa</groupId>
    <artifactId>parent</artifactId>
    <version>1.0-SNAPSHOT</version>
</parent>

<groupId>com.gxa</groupId>
<artifactId>hello_meaven</artifactId>
<version>1.0-SNAPSHOT</version>

<dependencies>
    <dependency>
        <groupId>commons-dbcp</groupId>
        <artifactId>commons-dbcp</artifactId>
        <version>20030825.184428</version>
    </dependency>
</dependencies>

</project>

```

Maven

- Profiles
- hello_meaven
 - Lifecycle
 - Plugins
 - Dependencies
 - commons-dbcp:commons-dbcp:20030825.184428
 - com.alibaba:druid:1.2.14
 - org.mybatis:mybatis:3.5.11
- parent
 - Lifecycle
 - Plugins
 - Dependencies
 - com.alibaba:druid:1.2.14
 - org.mybatis:mybatis:3.5.11

project

Text Dependency Analyzer