

Maven

Maven	1
1. 安装	4
1.1. 1.环境: JDK/Eclipse/Maven 3.5.2.zip	4
1.2. 2.环境变量	4
1.2.1. A)新建环境变量MAVEN_HOME, D:\Java\Maven\apache-maven-3.5.4	
B)编辑环境变量Path, 追加%MAVEN_HOME%\bin\;	4
1.2.2. dos 中输入 mvn -v 可以显示 maven信息	4
1.3. 3.配置本地仓库	4
1.3.1. ===== 打开D:\Java\Maven新建maven-	
repository文件夹, 该目录用作maven的本地库 =====	
D:\Java\Maven\apache-maven-3.5.4\conf\settings.xml找到	
<localRepository>/path/to/local/repo</localRepository>	
localRepository节点默认是被注释掉的, 需要把它移到注释之外, 然后将localR	
epository节点的值改为上一步创建的目录 =====	
运行一下DOS命令 mvn help:system ; 如果成功可以发现本地仓库多了很多文件	
4	
1.4. 4.设置maven使用阿里云镜像	4
1.4.1. D:\Java\Maven\apache-maven-3.5.4\conf\settings.xml	
找到<mirror>进行替换 <mirror> <id>UK</id> <name>UK Central</name>	
<url>http://uk.maven.org/maven2</url> <mirrorOf>central</mirrorOf>	
</mirror>	4
1.5. 5.配置Eclipse	5
1.6. 6.添加Maven模板	5
1.6.2. archetype-catalog.xml(选择这个文件)	6
2. Maven Web 搭建	6
2.1. 1.Eclipse新建maven web项目	6
2.2. 2.pom.xml添加servlet-api 依赖	8
2.2.1. <dependency> <groupId>javax.servlet</groupId>	
<artifactId>javax.servlet-api</artifactId> <version>3.1.0</version> <!--	
只在编译和测试时运行 --> <scope>provided</scope> </dependency>	8
2.3. 3.创建maven规定的文件目录(在Navigator下新建文件夹)	8
2.4. 4.把Junit的依赖改为 版本4.12	9
2.5. 5.更新maven项目	9
2.5.1. 右键-maven-update project	9
2.6. 6.转化为Web项目	9
2.6.1. 右键-properties-project facets	9

2.7.	7.修改默认部署配置.....	10
2.7.1.	右键-properties-Deployment Assembly ,移除倒数第三个.....	10
2.8.	8.配置服务器依赖.....	10
2.8.1.	<code><!--Jetty 服务器 --> <!-- <plugin></code> <code><groupId>org.eclipse.jetty</groupId> <artifactId>jetty-maven-plugin</artifactId></code> <code><version>9.4.11.v20180605</version> </plugin> --> <plugin></code> <code><groupId>org.apache.tomcat.maven</groupId> <artifactId>tomcat7-maven-</code> <code>plugin</artifactId> <version>2.2</version> <executions></code> <code><execution> <phase>package</phase> <goals></code> <code><goal>run</goal> </goals> </execution> </executions></code> <code></plugin> </plugins></code>	10
2.9.	9.运行.....	11
2.9.1.	右键-run as -maven //: clean package	11
2.10.	10.浏览器输入	11
2.10.1.	[INFO] Running war on http://localhost:8080/maven2	11
3.	Maven Web的目录结构.....	11
3.2.	Java	
	Resources文件夹下面的代码都是以包的形式展现的，主要是为了开发人员更好的找到修改的代码文件。Java	
	Resources文件夹下的代码和src文件夹下面的代码其实就是一份。	
	这里面写Java代码,在src/main/webapp下写jsp页面,这个目录下的jsp就相当于web Content目录下的JSP	13
3.3.	<code><import resource="classpath:spring-mybatis.xml" /></code> 。这里的这个spring-mybatis.xml文件是怎么找到的呢，其实它生成后，不就是在根目录下的WEB-INF下的classes的路径下嘛，这个就是classpath啦。.....	13
3.4.	src/main/java :放业务包和业务java类 src/main/resources: 放资源文件(这个就是classpath地址) src/main/resources/mapper:放数据库xml src/test/java: 放测试包和测试java类	13
4.	Junit.....	13
4.1.	1.@Test: 测试方法	
	a)(expected=XXException.class)如果程序的异常和XXException.class一样，则测试通过	
	b)(timeout=100)如果程序的执行能在100毫秒之内完成，则测试通过	
	2.@Ignore: 被忽略的测试方法：加上之后，暂时不运行此段代码	
	3.@Before: 每一个测试方法之前运行	
	4.@After: 每一个测试方法之后运行	
	5.@BeforeClass: 方法必须要是静态方法（static	
	声明），所有测试开始之前运行，注意区分before，是所有测试方法	
	6.@AfterClass: 方法必须要是静态方法（static	
	声明），所有测试结束之后运行，注意区分 @After.....	13

4.2.	①测试方法上必须使用@Test进行修饰 ②测试方法必须使用public void进行修饰，不能带任何的参数	
	③新建一个源代码目录来存放我们的测试代码，即将测试代码和项目业务代码分开 ④测试类所在的包名应该和被测试类所在的包名保持一致	
	⑤测试单元中的每个方法必须可以独立测试，测试方法间不能有任何的依赖	
	⑥测试类使用Test作为类名的后缀（不是必须）	
	⑦测试方法使用test作为方法名的前缀（不是必须）	14
4.3.	断言	14
4.3.1.	assertEquals断言:比较实际的值和用户预期的值是否一样	14
5.	常见问题	14
5.1.	1. EL表达式无法解析	14
5.1.1.	Maven的web 版本2.5 默认忽视EL表达 A.在jsp页面添加<%@ page isELIgnored="false" %> B.修改web.xml 成新的 版本 3.0	14
5.2.	2.ojdbc 依赖需要本地仓库	14
5.2.1.	mvn install:install-file -Dfile=D:\Java\ojdbc7.jar -DgroupId=com.oracle -DartifactId=ojdbc7 -Dversion=12.2.0 -Dpackaging=jar -DgeneratePom=true <dependency> <groupId>com.oracle</groupId> <artifactId>ojdbc7</artifactId> <version>12.2.0</version> </dependency>	14
5.3.	3.修改web dynamic module	15

1. 安装

1.1.1.环境: JDK/Eclipse/Maven 3.5.2.zip

1.2.2.环境变量

1.2.1. A)新建环境变量MAVEN_HOME,

D:\Java\Maven\apache-maven-3.5.4

B)编辑环境变量Path, 追加%MAVEN_HOME%\bin\;

1.2.2. dos 中输入 mvn -v 可以显示 maven信息

1.3.3.配置本地仓库

1.3.1. =====

打开D:\Java\Maven新建maven-repository文件夹, 该目录用作maven的本地库

=====

D:\Java\Maven\apache-maven-3.5.4\conf\settings.xml找到

<localRepository>/path/to/local/repo</localRepository>

localRepository节点默认是被注释掉的, 需要把它移到注释之外, 然后将localRepository节点的值改为上一步创建的目录

=====

运行一下DOS命令 mvn help:system ; 如果成功可以发现本地仓库多了很多文件

1.4.4.设置maven使用阿里云镜像

1.4.1. D:\Java\Maven\apache-maven-3.5.4\conf\settings.xml

找到<mirror>进行替换

<mirror>

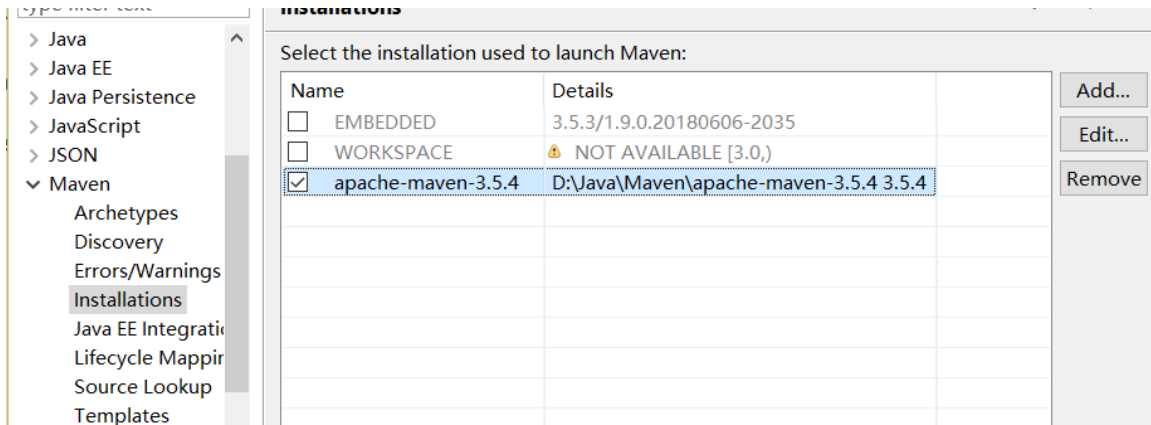
<id>UK</id>

<name>UK Central</name>

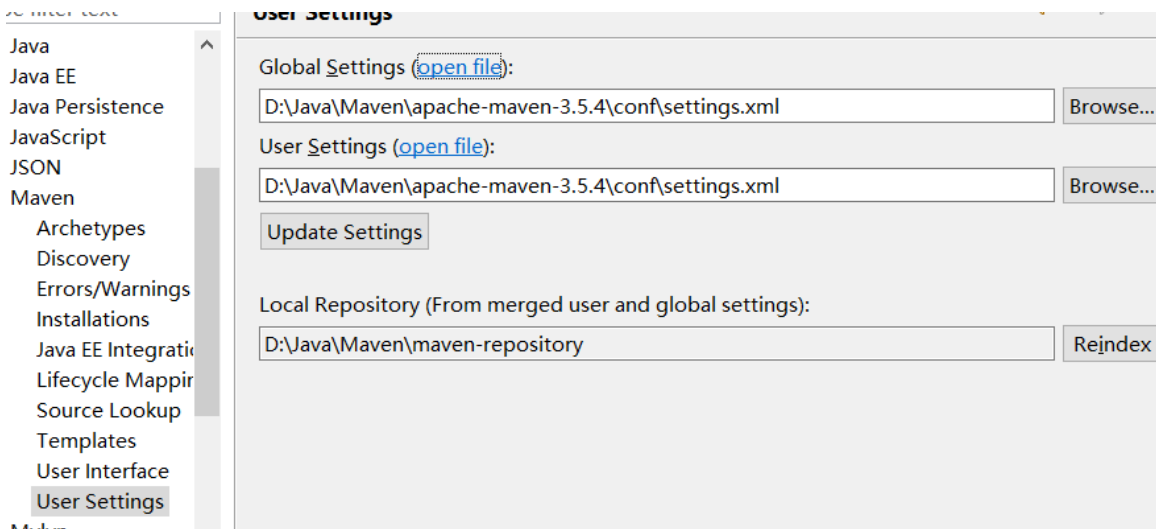
```
<url>http://uk.maven.org/maven2</url>  
<mirrorOf>central</mirrorOf>  
</mirror>
```

1.5.5. 配置Eclipse

1.5.1.

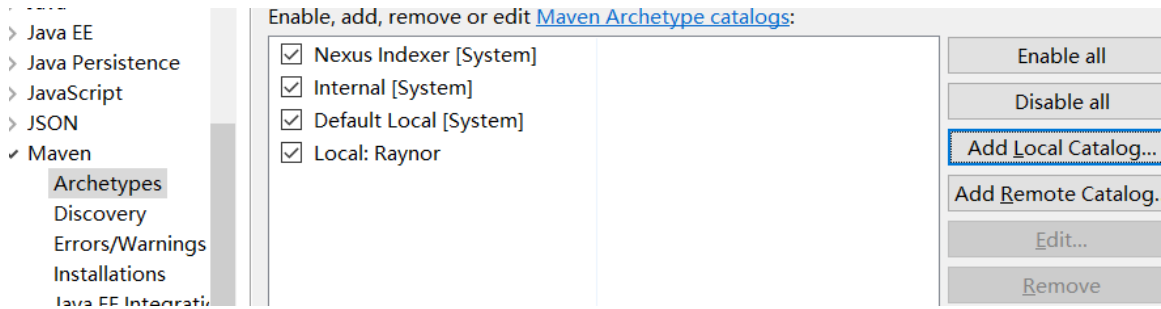


1.5.2.



1.6.6. 添加Maven模板

1.6.1.



1.6.2. archetype-catalog.xml(选择这个文件)

2. Maven Web 搭建

2.1. 1.Eclipse新建maven web项目

2.1.1.

New Maven project

Select project name and location



☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location:

☐ Add project(s) to working set

Working set:

▶ Advanced

2.1.2.

New Maven project

Select an Archetype



Catalog: All Catalogs Configure...

Filter: web ×

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-webapp	1.0

An archetype which contains a sample Maven Webapp project. ^
v

☒ Show the last version of Archetype only ☐ Include snapshot archetypes Add Archetype...

▸ Advanced

? < Back Next > Finish Cancel

2.1.3.

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

Name	Value

Advanced

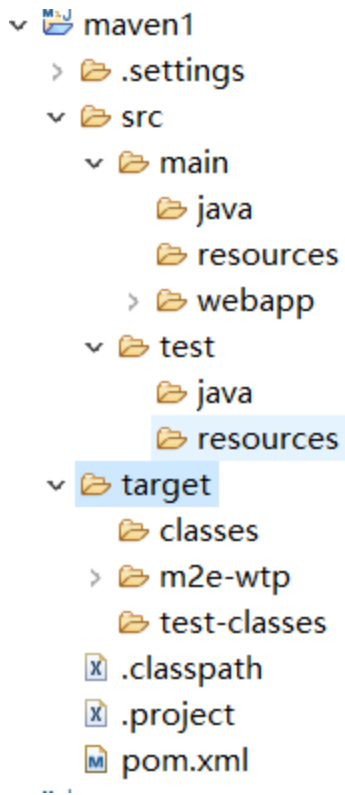
2.2.2.pom.xml添加servlet-api 依赖

2.2.1. <dependency>

```
<groupId>javax.servlet</groupId>
<artifactId>javax.servlet-api</artifactId>
<version>3.1.0</version>
<!-- 只在编译和测试时运行 -->
<scope>provided</scope>
</dependency>
```

2.3.3.创建maven规定的文件目录(在Navigator下新建文件夹)

2.3.1.



2.4.4.把Junit的依赖改为 版本4.12

2.5.5.更新maven项目

2.5.1. 右键-maven-update project

2.6.6.转化为Web项目

2.6.1. 右键-properties-project facets

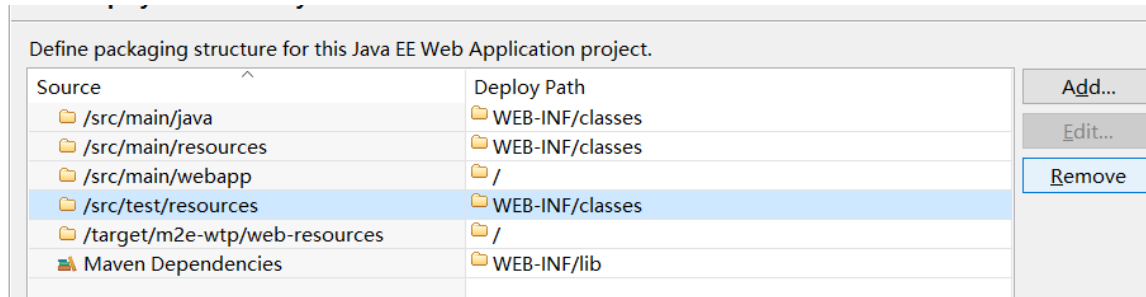
2.6.2.

Java Editor	<input type="checkbox"/>	Cloud Foundry Standalone Application	1.0		A
Javadoc Location	<input type="checkbox"/>	CXF 2.x Web Services	1.0		u:
JavaScript	<input checked="" type="checkbox"/>	Dynamic Web Module	2.3	▼	
JSP Fragment	<input type="checkbox"/>	EAR	6.0	▼	
Maven	<input type="checkbox"/>	EJB Module	3.1	▼	
Project Facets	<input type="checkbox"/>	EJBDoclet (XDoclet)	1.2.3	▼	
Project References	<input checked="" type="checkbox"/>	Java	1.8	▼	
Run/Debug Settings	<input checked="" type="checkbox"/>	JavaScript	1.0		
Server	<input type="checkbox"/>	JavaServer Faces	2.2	▼	
Service Policies	<input type="checkbox"/>	JAX-RS (REST Web Services)	1.1	▼	
Targeted Runtimes	<input type="checkbox"/>	JAXR	2.2	▼	

2.7.7.修改默认部署配置

2.7.1. 右键-properties-Deployment Assembly ,移除倒数第三个

2.7.2.



2.8.8.配置服务器依赖

2.8.1. <plugins>

<!--Jetty 服务器 -->

<!-- <plugin>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-maven-plugin</artifactId>

<version>9.4.11.v20180605</version>

</plugin> -->

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

<executions>

<execution>

<phase>package</phase>

<goals>

<goal>run</goal>

</goals>

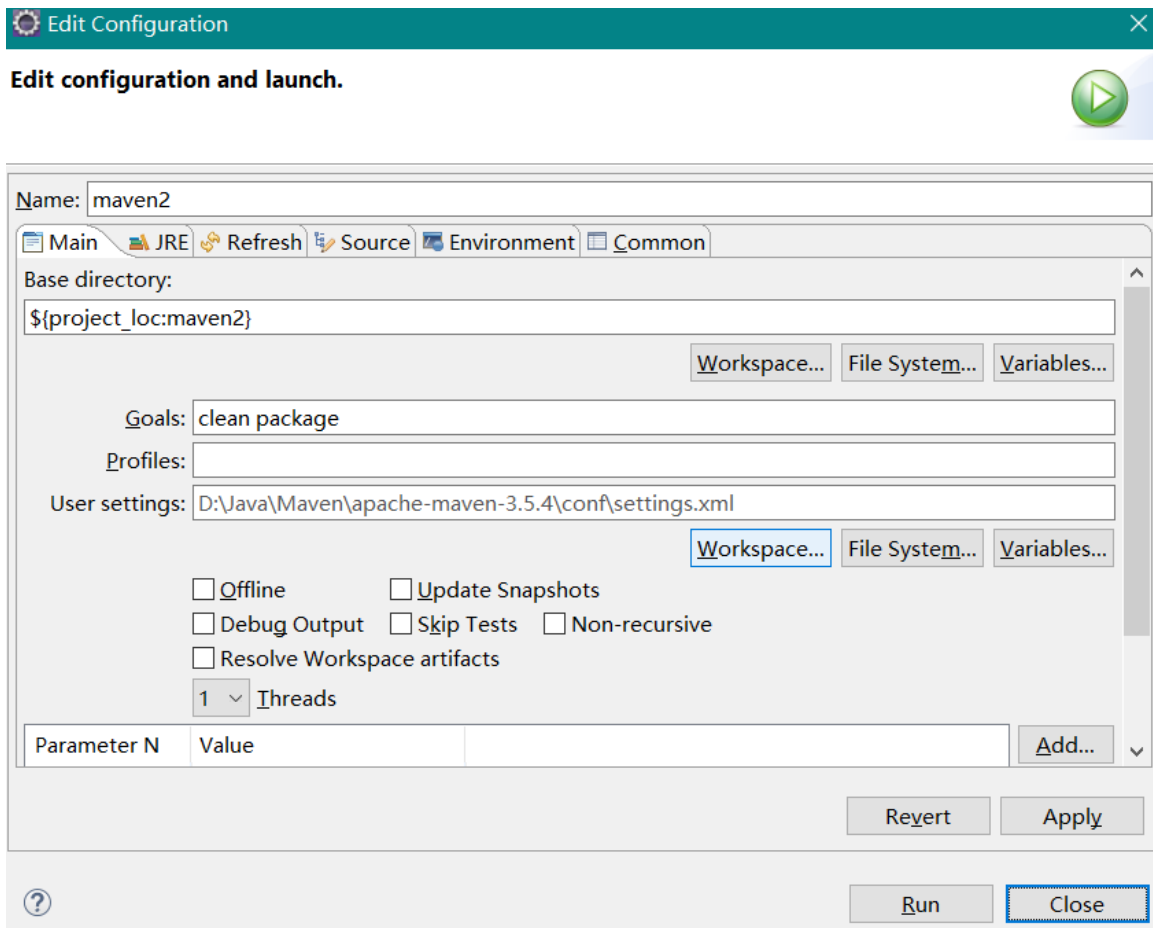
</execution>

```
</executions>
</plugin>
</plugins>
```

2.9.9.运行

2.9.1. 右键-run as -maven //: clean package

2.9.2.







































2.10. 10.浏览器输入

2.10.1. [INFO] Running war on http://localhost:8080/maven2

3. Maven Web的目录结构

3.1.

- ✓  maven03
 - >  Deployment Descriptor: <web ap
 - ✓  Java Resources
 - ✓  src/main/java
 - >  com.maven.dao
 - >  com.maven.dto
 - >  com.maven.entity
 - >  com.maven.enums
 - >  com.maven.exception
 - >  com.maven.service
 - >  com.maven.service.impl
 - >  com.maven.web
 - ✓  src/main/resources
 - >  mapper
 - >  spring
 -  jdbc.properties
 -  logback.xml
 -  mybatis-config.xml
 - ✓  src/test/java
 - >  com.maven
 - >  com.maven.dao
 - >  com.maven.service.impl
 - >  com.maven.web
 - >  Libraries
 - >  JavaScript Resources
 - >  Deployed Resources
 - ✓  src
 - ✓  main
 - >  java
 - >  resources
 - >  sql
 - >  webapp
 - ✓  test
 - >  java
 - >  resources
 - >  target

3.2. Java

Resources文件夹下面的代码都是以包的形式展现的，主要是为了开发人员更好的找到修改的代码文件。Java

Resources文件夹下的代码和**src**文件夹下面的代码其实就是一份。

这里面写Java代码,在**src/main/webapp**下写**jsp**页面,这个目录下的**jsp**就相当于**web Content**目录下的**JSP**

3.3. `<import resource="classpath:spring-mybatis.xml" />`。

这里的这个**spring-**

mybatis.xml文件是怎么找到的呢，其实它生成后，不就是在根目录下的**WEB-INF**下的**classes**的路径下嘛，这个就是**classpath**啦。

3.4. **src/main/java** :放业务包和业务java类

src/main/resources: 放资源文件(这个就是**classpath**地址)

src/main/resources/mapper:放数据库xml

src/test/java: 放测试包和测试java类

4. Junit

4.1. 1. **@Test**: 测试方法

a)(**expected=XXException.class**)如果程序的异常和**XXException.class**一样，则测试通过

❑ b)(**timeout=100**)如果程序的执行能在**100**毫秒之内完成，则测试通过

2.**@Ignore**: 被忽略的测试方法：加上之后，暂时不运行此段代码

3.**@Before**: 每一个测试方法之前运行

4.**@After**: 每一个测试方法之后运行

5.**@BeforeClass**: 方法必须要是静态方法（**static**

声明），所有测试开始之前运行，注意区分**before**，是所有测试方法

6.@AfterClass: 方法必须要是静态方法（static 声明），所有测试结束之后运行，注意区分 @After

4.2. ①测试方法上必须使用@Test进行修饰

②测试方法必须使用public void 进行修饰，不能带任何的参数

③新建一个源代码目录来存放我们的测试代码，即将测试代码和项目业务代码分开

④测试类所在的包名应该和被测试类所在的包名保持一致

⑤测试单元中的每个方法必须可以独立测试，测试方法间不能有任何的依赖

⑥测试类使用Test作为类名的后缀（不是必须）

⑦测试方法使用test作为方法名的前缀（不是必须）

4.3. 断言

4.3.1. assertEquals断言:比较实际的值和用户预期的值是否一样

5. 常见问题

5.1.1. EL表达式无法解析

5.1.1. Maven的web 版本2.5 默认忽视EL表达

A.在jsp页面添加<%@ page isELIgnored="false" %>

B.修改web.xml 成新的 版本 3.0

5.2.2.ojdbc 依赖需要本地仓库

5.2.1. mvn install:install-file -Dfile=D:\Java\ojdbc7.jar -DgroupId=com.oracle -

DartifactId=ojdbc7 -Dversion=12.2.0 -Dpackaging=jar -DgeneratePom=true

<dependency>

<groupId>com.oracle</groupId>

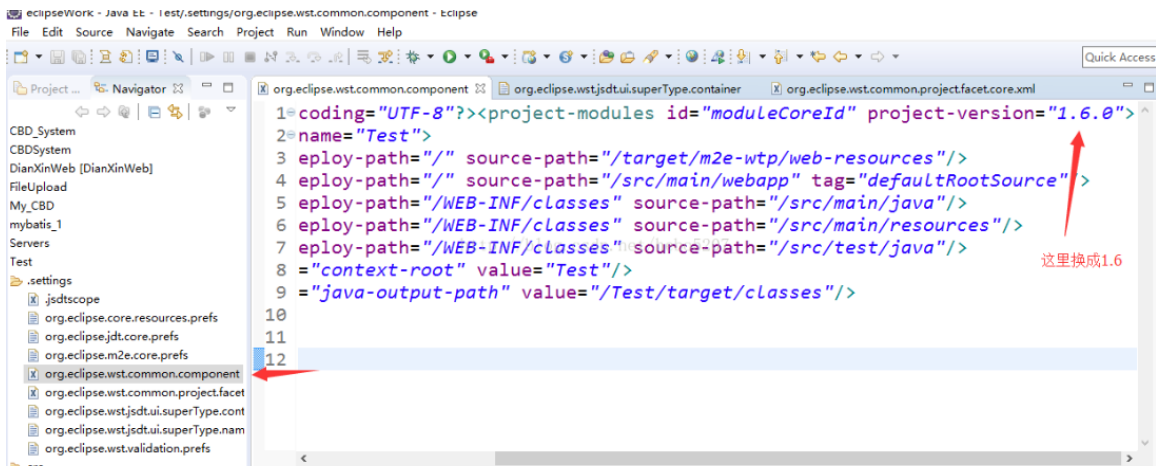
<artifactId>ojdbc7</artifactId>

<version>12.2.0</version>

</dependency>

5.3.3.修改web dynamic module

5.3.1.



5.3.2.

