JAVA 达摩班

实训 之*B2B2C*

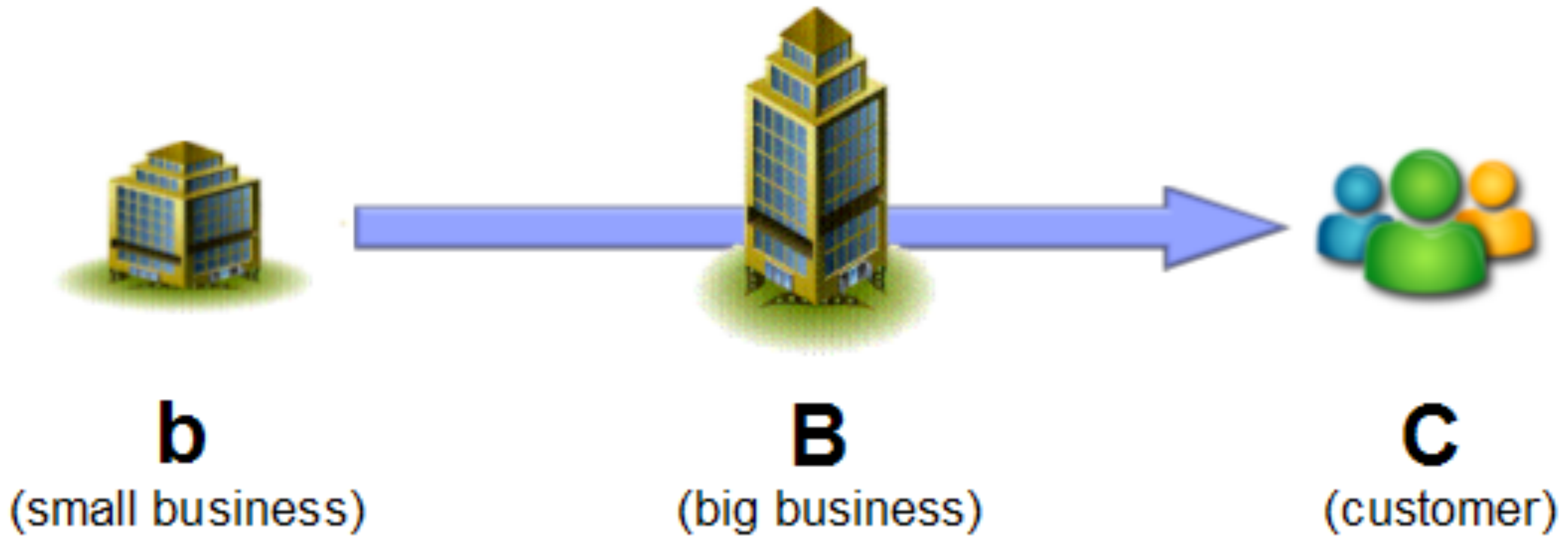
如何理解B2B2C

B2B2C can help a company market its product or service more effectively by entering a B2B relationship with a company whose expertise is selling online -- a B2C (Business2Consumer or Business-to-Consumer) company. In return, the B2C company is able to offer its customers more options.

第一个BUSINESS，并不仅仅局限于品牌供应商、影视制作公司和图书出版商，任何的商品供应商或服务供应商都能可以成为第一个BUSINESS；第二B是B2B2C模式的电子商务企业，通过统一的经营管理对商品和服务、消费者终端同时进行整合，是广大供应商和消费者之间的桥梁，为供应商和消费者提供优质的服务，是互联网电子商务服务供应商。C表示消费者，在第二个B构建的统一电子商务平台购物的消费者；
通过B2B2C模式的电子商务企业构建自己的物流供应链系统，提供统一的服务。

B2B的定义：企业跟企业之间的电子商务运作方式。
B2C的定义：企业跟消费者之间的电子商务运作方式。
B2G的定义：企业跟政府之间的合作关系。
C2C的定义：消费者跟消费者之间的电子商务运作方式。

如何理解B2B2C



Small business selling "through" a big business

如何理解B2B2C

There are three simple, but important takeaways:

1. Don’t wait for customers to come to you
2. Build emotional connection
3. Make it easy to transact

A relatable example from the retail industry is JCWhitney.com selling automotive parts through marketplace at Sears.com. According to the Internet Retailer,

“Last September Whitney began supplying Sears.com with more than 130,000 items for its parts and accessories inventory for cars, trucks, motorcycles and other vehicles. Sears will process all transactions, but order fulfillment and customer service will be the responsibility of individual retailers.”

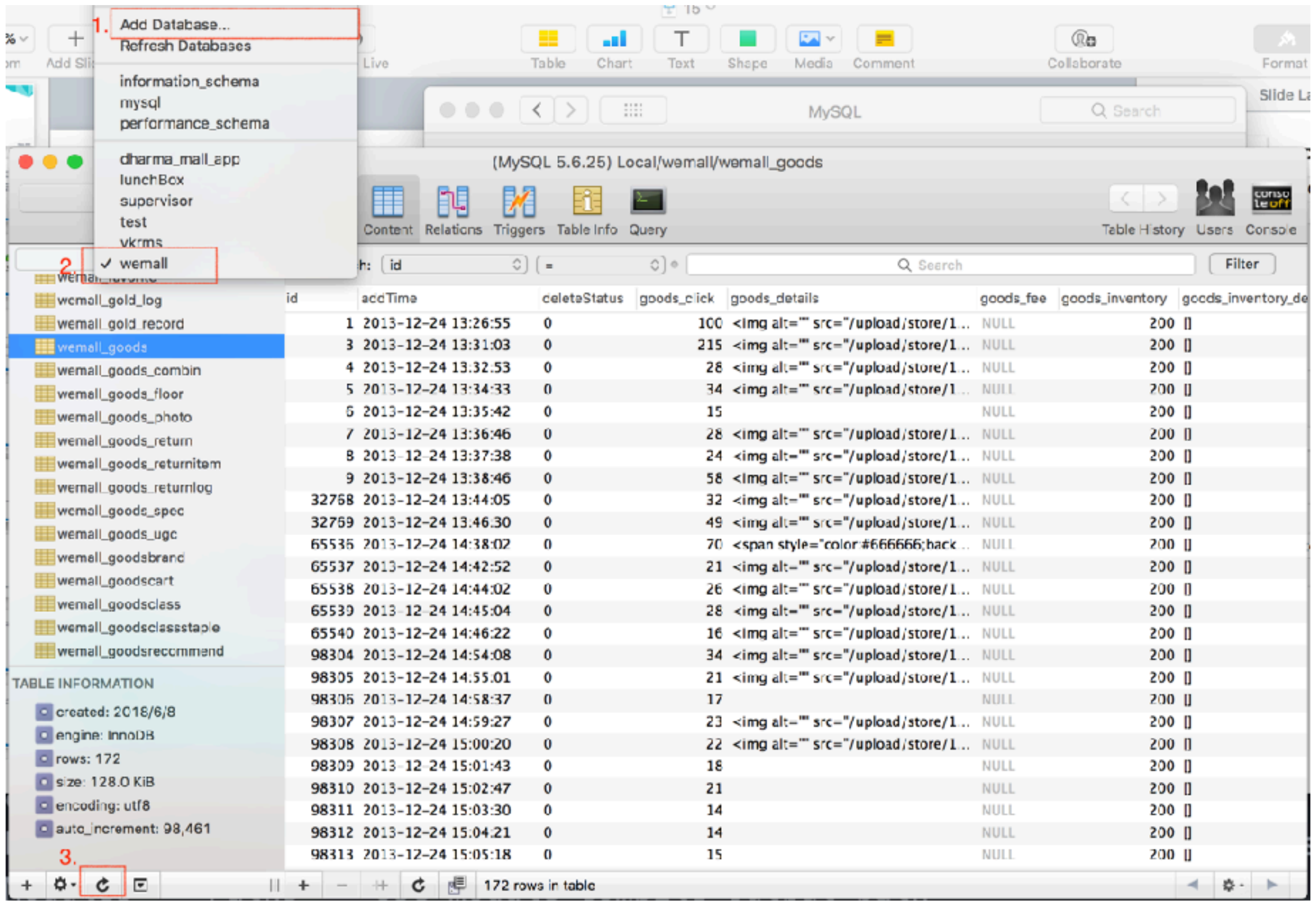
So now, for example, you can buy a Rotor for 1985 Ford Ranger from Sears.com which is typically sold in auto parts stores like JC Whitney and others.

Wemall

安装

前提

JDK 8
Tomcat 8
Intellij Idea 2018
MySQL 5.7.*
Memcached
Sequel Pro



wemall_gold_log
wemall_gold_record
wemall_goods
wemall_goods_combin
wemall_goods_floor
wemall_goods_photo
wemall_goods_return
wemall_goods_returnitem
wemall_goods_returnlog
wemall_goods_spec
wemall_goods_ugc
wemall_goodsbrand
wemall_goodscart
wemall_goodsclass
wemall_goodsclassstaple
wemall_goodsrecommend

TABLE INFORMATION
created: 2018/6/8
engine: InnoDB
rows: 172
size: 128.0 KiB
encoding: utf8
auto_increment: 98,461

id	addTime	deleteStatus	goods_click	goods_details	goods_fee	goods_inventory	goods_inventory_de
1	2013-12-24 13:26:55	0	100	<img alt="" src="/upload/store/1...	NULL	200	[]
3	2013-12-24 13:31:03	0	215	<img alt="" src="/upload/store/1...	NULL	200	[]
4	2013-12-24 13:32:53	0	28	<img alt="" src="/upload/store/1...	NULL	200	[]
5	2013-12-24 13:34:33	0	34	<img alt="" src="/upload/store/1...	NULL	200	[]
6	2013-12-24 13:35:42	0	15		NULL	200	[]
7	2013-12-24 13:36:46	0	28	<img alt="" src="/upload/store/1...	NULL	200	[]
8	2013-12-24 13:37:38	0	24	<img alt="" src="/upload/store/1...	NULL	200	[]
9	2013-12-24 13:38:46	0	58	<img alt="" src="/upload/store/1...	NULL	200	[]
32768	2013-12-24 13:44:05	0	32	<img alt="" src="/upload/store/1...	NULL	200	[]
32769	2013-12-24 13:46:30	0	49	<img alt="" src="/upload/store/1...	NULL	200	[]
65536	2013-12-24 14:38:02	0	70	<span style="color:#666666;back...	NULL	200	[]
65537	2013-12-24 14:42:52	0	21	<img alt="" src="/upload/store/1...	NULL	200	[]
65538	2013-12-24 14:44:02	0	26	<img alt="" src="/upload/store/1...	NULL	200	[]
65539	2013-12-24 14:45:04	0	28	<img alt="" src="/upload/store/1...	NULL	200	[]
65540	2013-12-24 14:46:22	0	16	<img alt="" src="/upload/store/1...	NULL	200	[]
98304	2013-12-24 14:54:08	0	34	<img alt="" src="/upload/store/1...	NULL	200	[]
98305	2013-12-24 14:55:01	0	21	<img alt="" src="/upload/store/1...	NULL	200	[]
98306	2013-12-24 14:58:37	0	17		NULL	200	[]
98307	2013-12-24 14:59:27	0	23	<img alt="" src="/upload/store/1...	NULL	200	[]
98308	2013-12-24 15:00:20	0	22	<img alt="" src="/upload/store/1...	NULL	200	[]
98309	2013-12-24 15:01:43	0	18		NULL	200	[]
98310	2013-12-24 15:02:47	0	21		NULL	200	[]
98311	2013-12-24 15:03:30	0	14		NULL	200	[]
98312	2013-12-24 15:04:21	0	14		NULL	200	[]
98313	2013-12-24 15:05:18	0	15		NULL	200	[]

172 rows in table

导入数据库

1. 在Sequel Pro中创建名为wemall的数据库
2. 进入jshop-v3.0.sql所在目录，执行语句 ***/usr/local/mysql/bin/mysql -u root -p wemall < jshop-v3.0.sql***
3. 在Sequel Pro中刷新看到表正确导入

安装

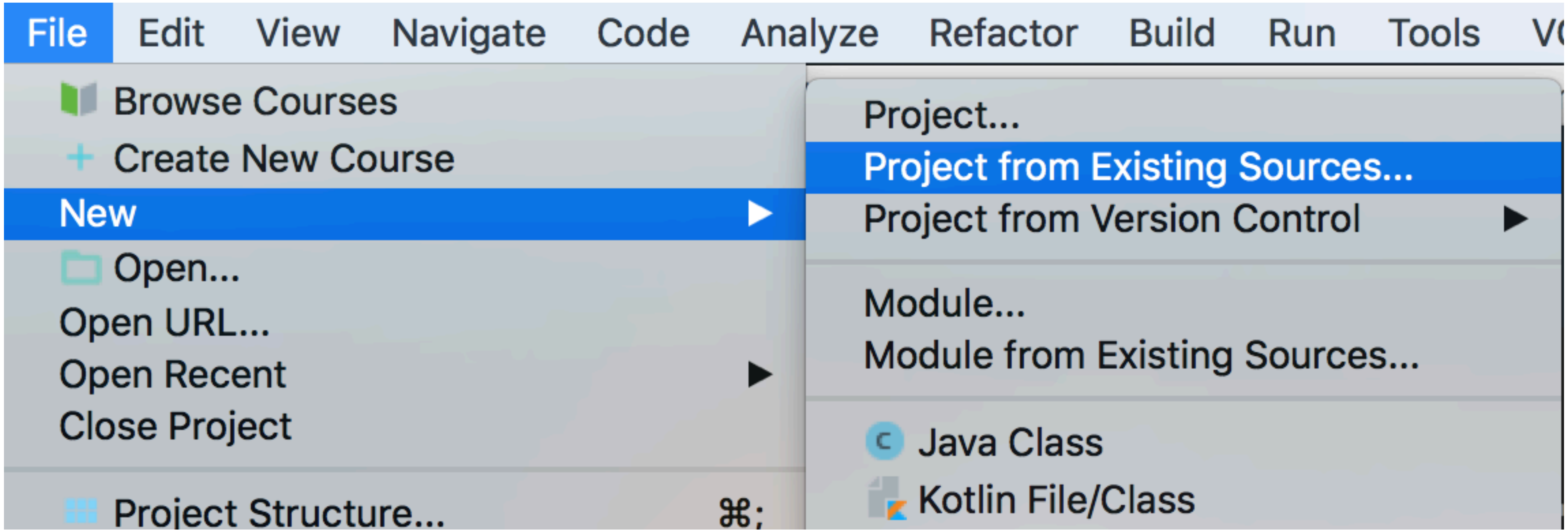
导入项目到Intellij IDEA

1. File -> New -> Project from existing sources，选择wemall目录，选择eclipse工程，下一步一直到结束
2. 快捷键cmd+enter打开工程配置，按照如下配置

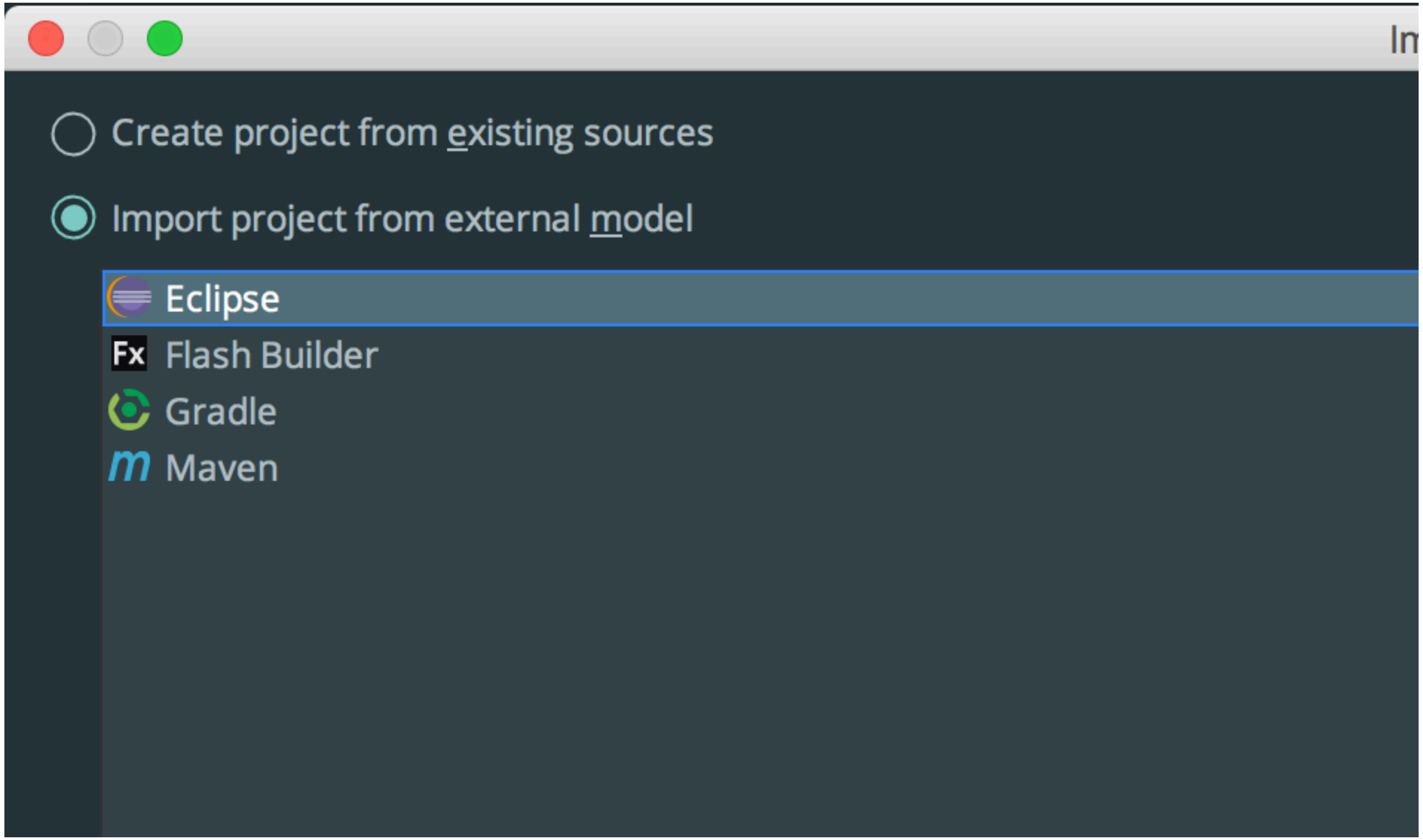
*Project -> project compiler output -> ****/wemall/WebRoot/WEB-INF/classes*
*Modules -> Paths -> Use module compile output path -> ****/wemall/WebRoot/WEB-INF/classes*
*Modules -> + -> Web -> Type -> ****/wemall/WebRoot/WEB-INF/web.xml*
*-> Web Resource Directory -> ****/wemall/WebRoot*
*Libraries -> + -> Java -> ****/wemall/WebRoot/WEB-INF/lib*
Artifacts -> + -> Web Application: Exploded -> from module -> wemall

3. 回到Intellij IDEA，添加tomcat

安装


File
Edit
View
Navigate
Code
Analyze
Refactor
Build
Run
Tools
V
Browse Courses
Create New Course
New
Open...
Open URL...
Open Recent
Close Project
Project Structure... ⌘;
Project...
Project from Existing Sources...
Project from Version Control
Module...
Module from Existing Sources...
Java Class
Kotlin File/Class

安装



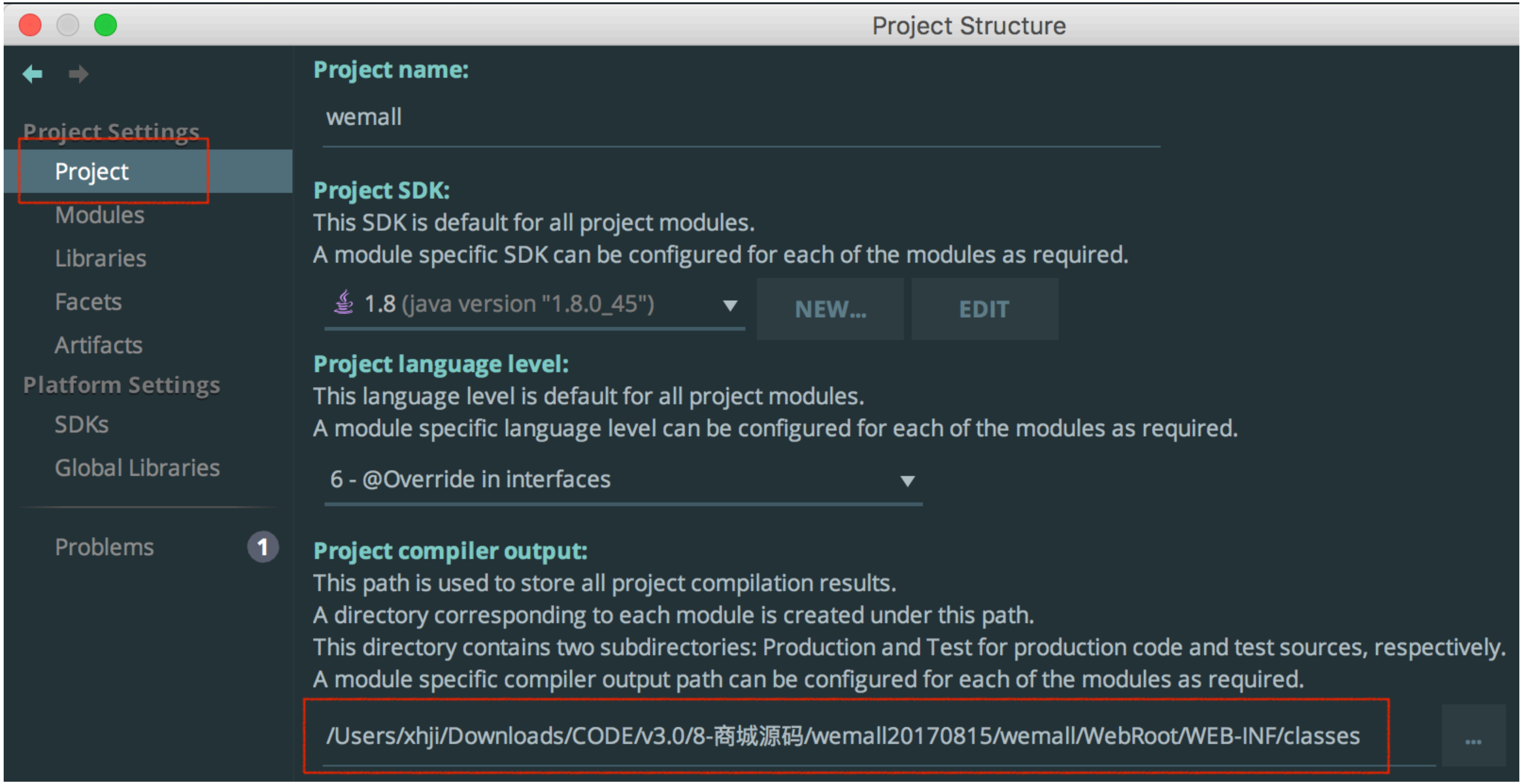
Im

○ Create project from existing sources

◉ Import project from external model

- Eclipse
- Flash Builder
- Gradle
- Maven

安装


Project Structure

Project Settings
Project
Modules
Libraries
Facets
Artifacts
Platform Settings
SDKs
Global Libraries
Problems 1

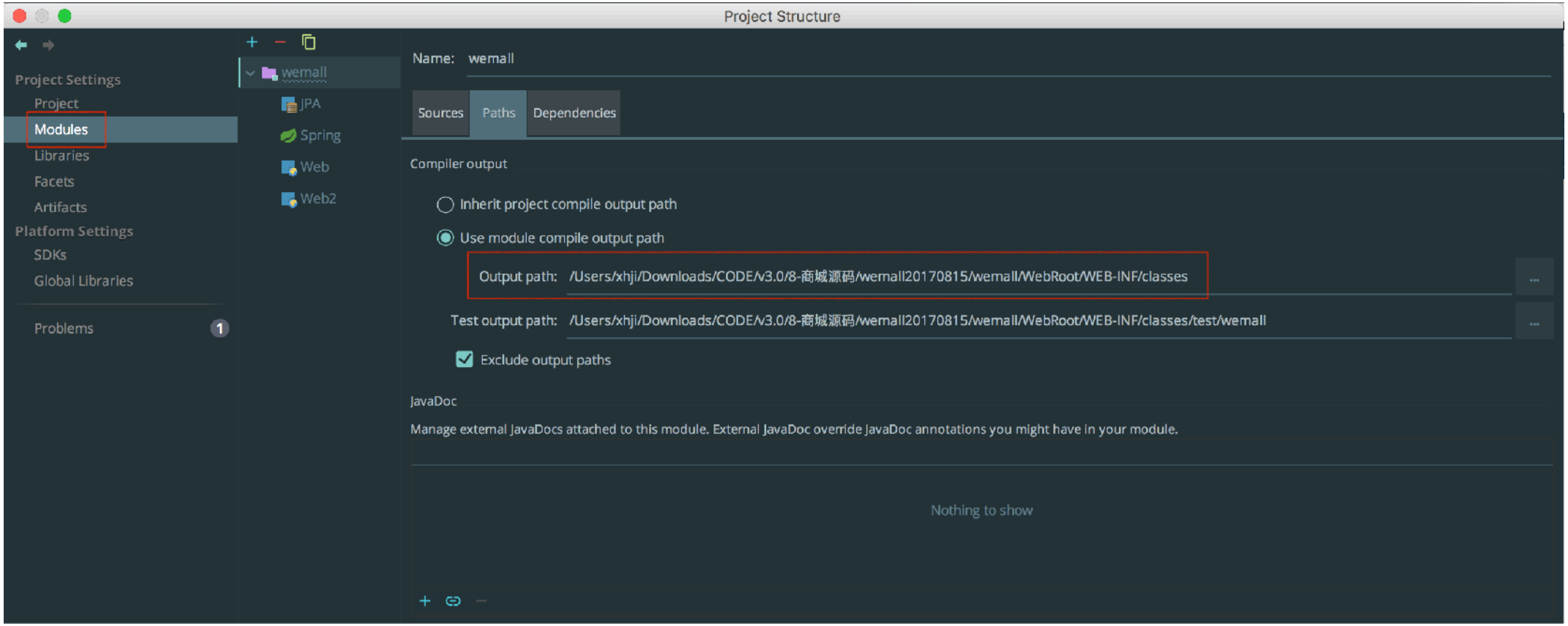
Project name:
wemall

Project SDK:
This SDK is default for all project modules.
A module specific SDK can be configured for each of the modules as required.
1.8 (java version "1.8.0_45") NEW... EDIT

Project language level:
This language level is default for all project modules.
A module specific language level can be configured for each of the modules as required.
6 - @Override in interfaces

Project compiler output:
This path is used to store all project compilation results.
A directory corresponding to each module is created under this path.
This directory contains two subdirectories: Production and Test for production code and test sources, respectively.
A module specific compiler output path can be configured for each of the modules as required.
/Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/WebRoot/WEB-INF/classes

安装



Name: wemall

Compiler output

○ Inherit project compile output path

● Use module compile output path

Output path: /Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/WebRoot/WEB-INF/classes

Test output path: /Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/WebRoot/WEB-INF/classes/test/wemall

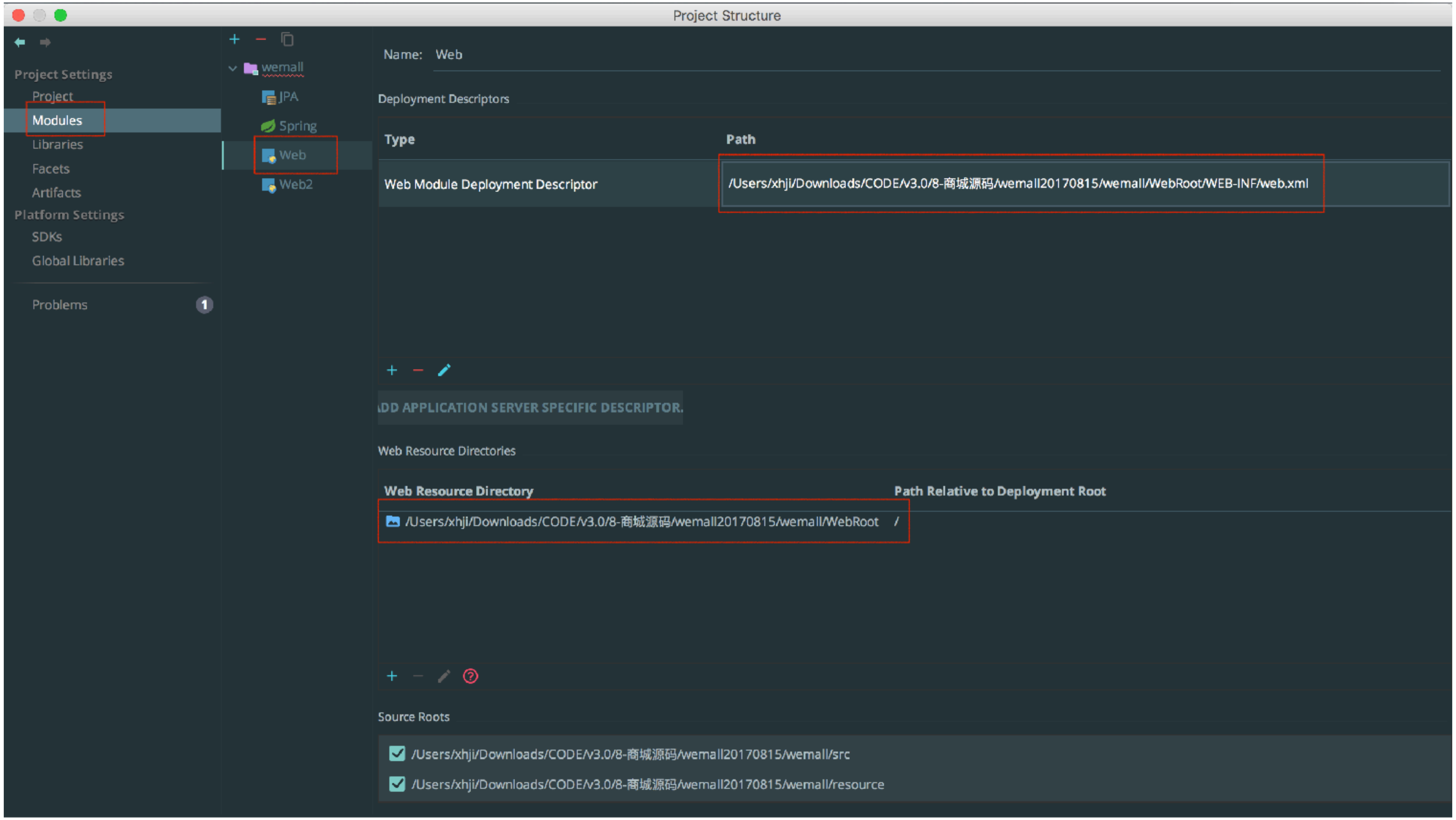
☑ Exclude output paths

JavaDoc

Manage external JavaDocs attached to this module. External JavaDoc override JavaDoc annotations you might have in your module.

Nothing to show

安装



Name: Web

Deployment Descriptors

Type	Path
Web Module Deployment Descriptor	/Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/WebRoot/WEB-INF/web.xml

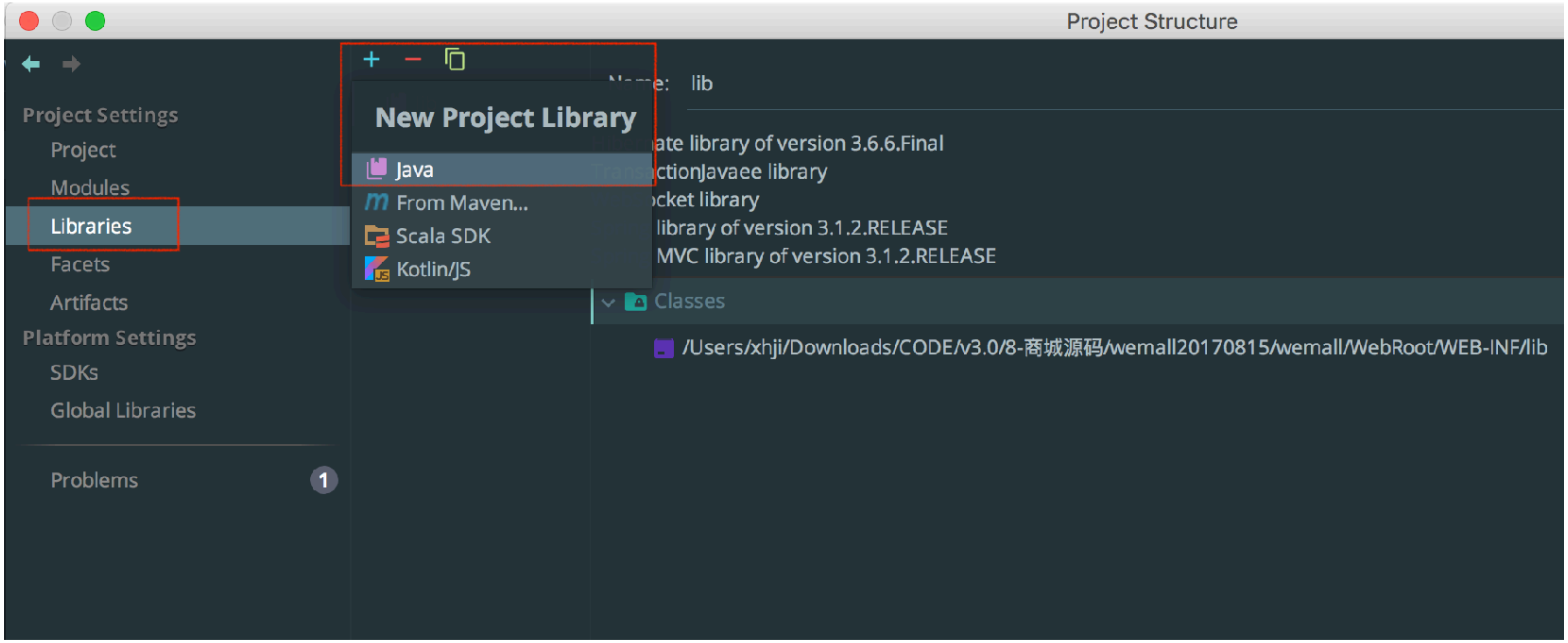
Web Resource Directories

Web Resource Directory	Path Relative to Deployment Root
/Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/WebRoot	/

Source Roots

- ✓ /Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/src
- ✓ /Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/resource

安装

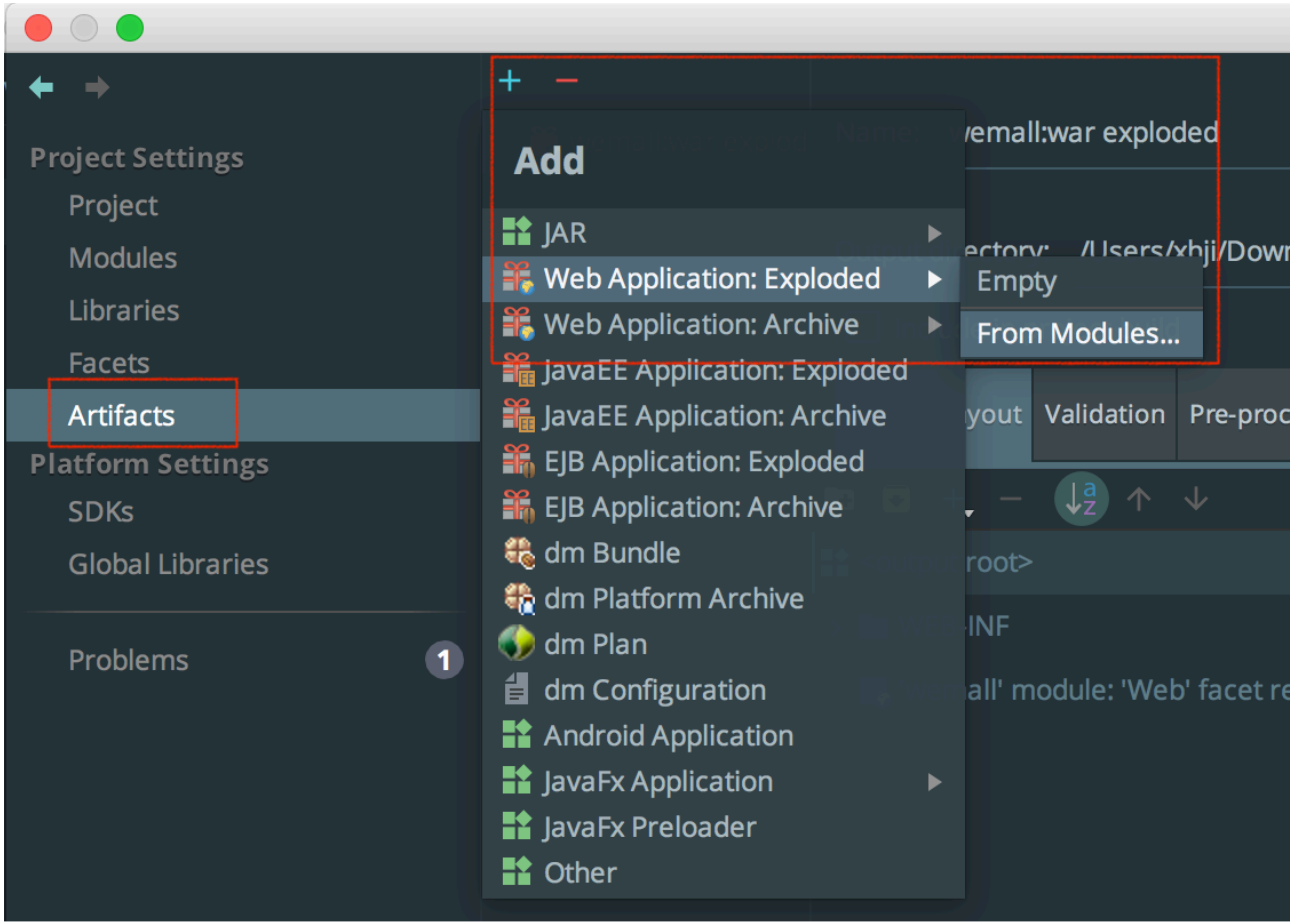


Name: lib

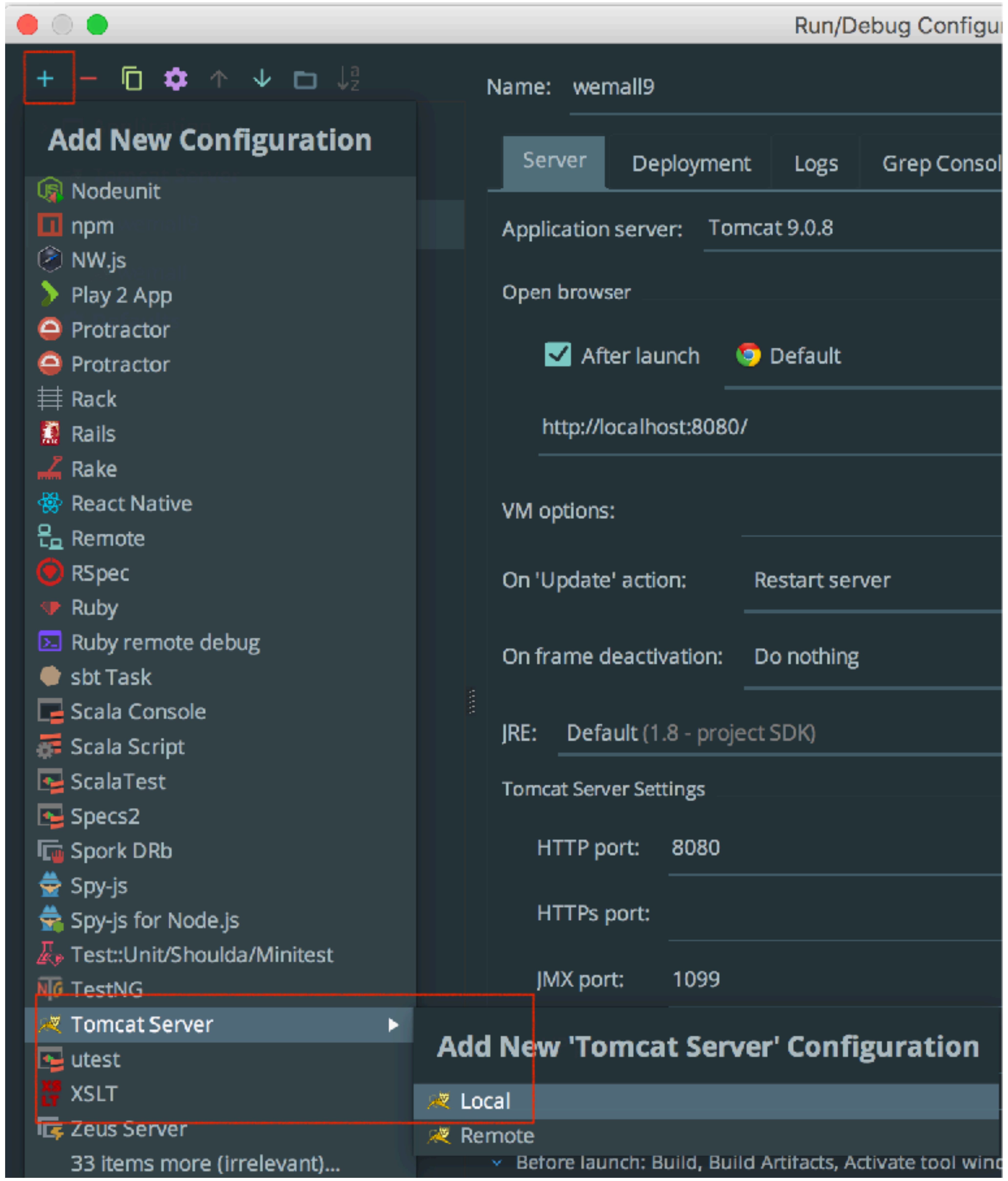
ate library of version 3.6.6.Final
ctionJavaee library
cket library
library of version 3.1.2.RELEASE
MVC library of version 3.1.2.RELEASE

/Users/xhji/Downloads/CODE/v3.0/8-商城源码/wemall20170815/wemall/WebRoot/WEB-INF/lib

安装


Project Settings
Project
Modules
Libraries
Facets
Artifacts
Platform Settings
SDKs
Global Libraries
Problems
1
Add
JAR
Web Application: Exploded
Web Application: Archive
JavaEE Application: Exploded
JavaEE Application: Archive
EJB Application: Exploded
EJB Application: Archive
dm Bundle
dm Platform Archive
dm Plan
dm Configuration
Android Application
JavaFx Application
JavaFx Preloader
Other
Empty
From Modules...
emall:war exploded
ectory: /Users/xbji/Dowr
yout
Validation
Pre-proc
root>
INF
all' module: 'Web' facet re

安装



Run/Debug Configu

Add New Configuration

Nodeunit
npm
NW.js
Play 2 App
Protractor
Protractor
Rack
Rails
Rake
React Native
Remote
RSpec
Ruby
Ruby remote debug
sbt Task
Scala Console
Scala Script
ScalaTest
Specs2
Spork DRb
Spy-js
Spy-js for Node.js
Test::Unit/Shoulda/Minitest
TestNG
Tomcat Server
utest
XSLT
Zeus Server
33 items more (irrelevant)...

Add New 'Tomcat Server' Configuration

Local
Remote

Name: wemall9

Server | Deployment | Logs | Grep Consol

Application server: Tomcat 9.0.8

Open browser

After launch ✓ Default

http://localhost:8080/

VM options:

On 'Update' action: Restart server

On frame deactivation: Do nothing

JRE: Default (1.8 - project SDK)

Tomcat Server Settings

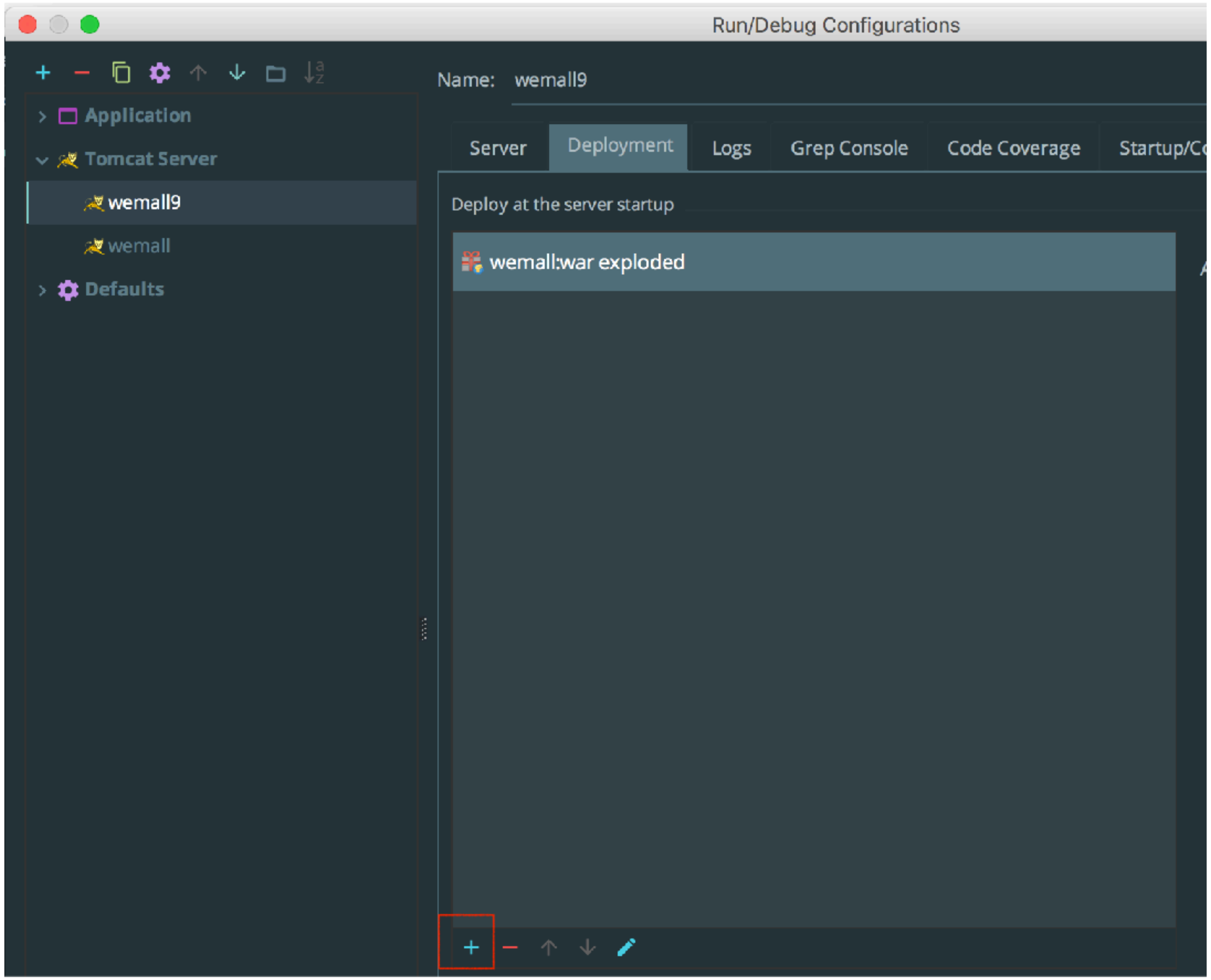
HTTP port: 8080

HTTPs port:

JMX port: 1099

Before launch: Build, Build Artifacts, Activate tool wind

安装


Run/Debug Configurations

Name: wemall9

- Application
- Tomcat Server
  - wemall9
  - wemall
- Defaults

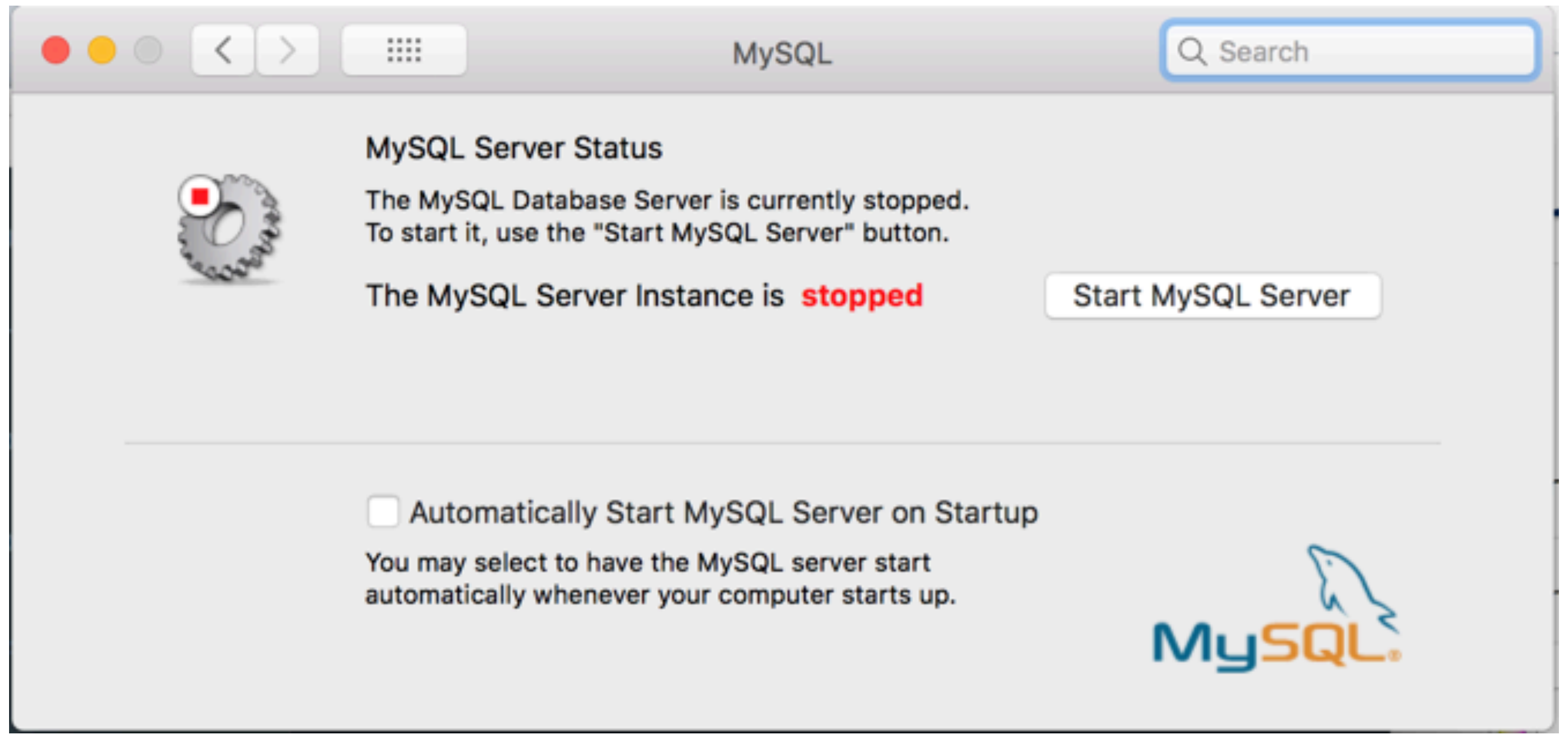
Deploy at the server startup

wemall:war exploded

安装

启动项目

1. 在控制台输入memcached，启动memcached
2. 确认MySQL启动
3. 启动Tomcat



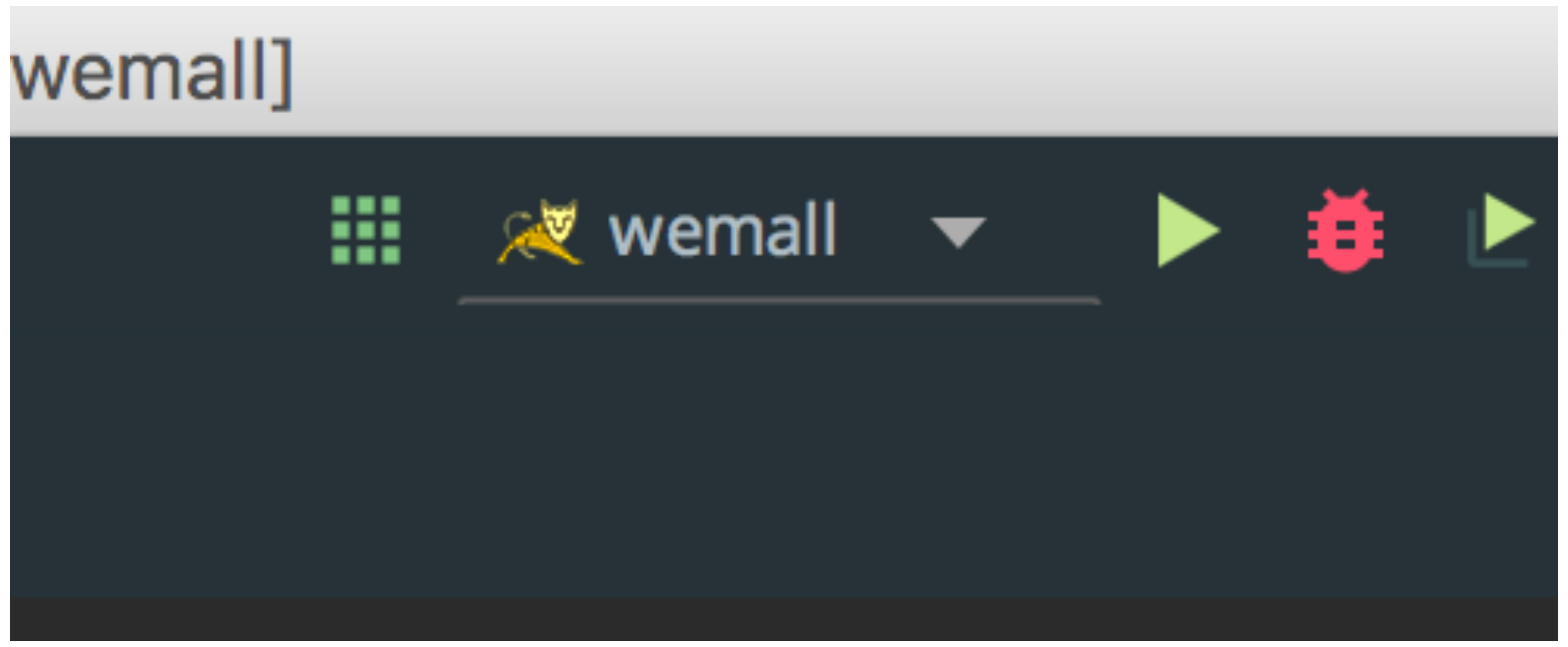
MySQL Server Status

The MySQL Database Server is currently stopped.
To start it, use the "Start MySQL Server" button.

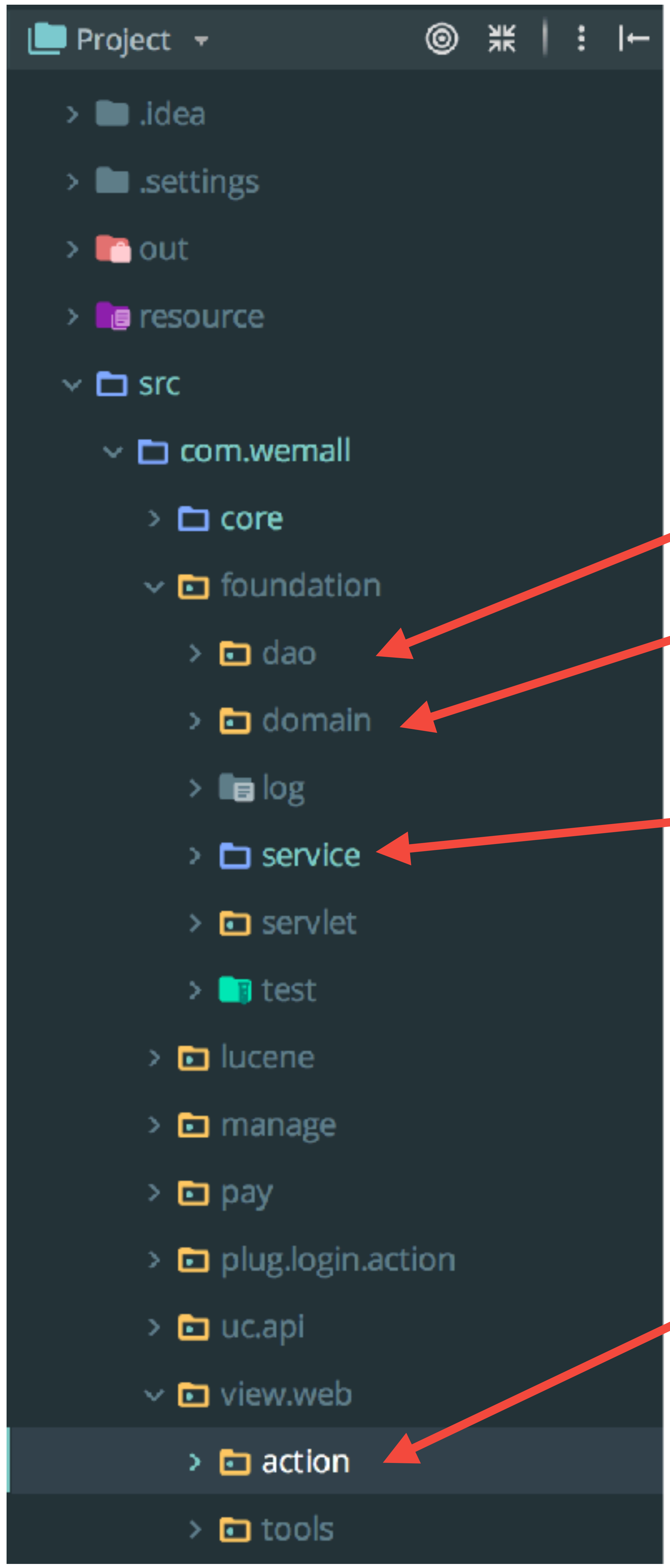
The MySQL Server Instance is **stopped**

Automatically Start MySQL Server on Startup

You may select to have the MySQL server start automatically whenever your computer starts up.



项目架构



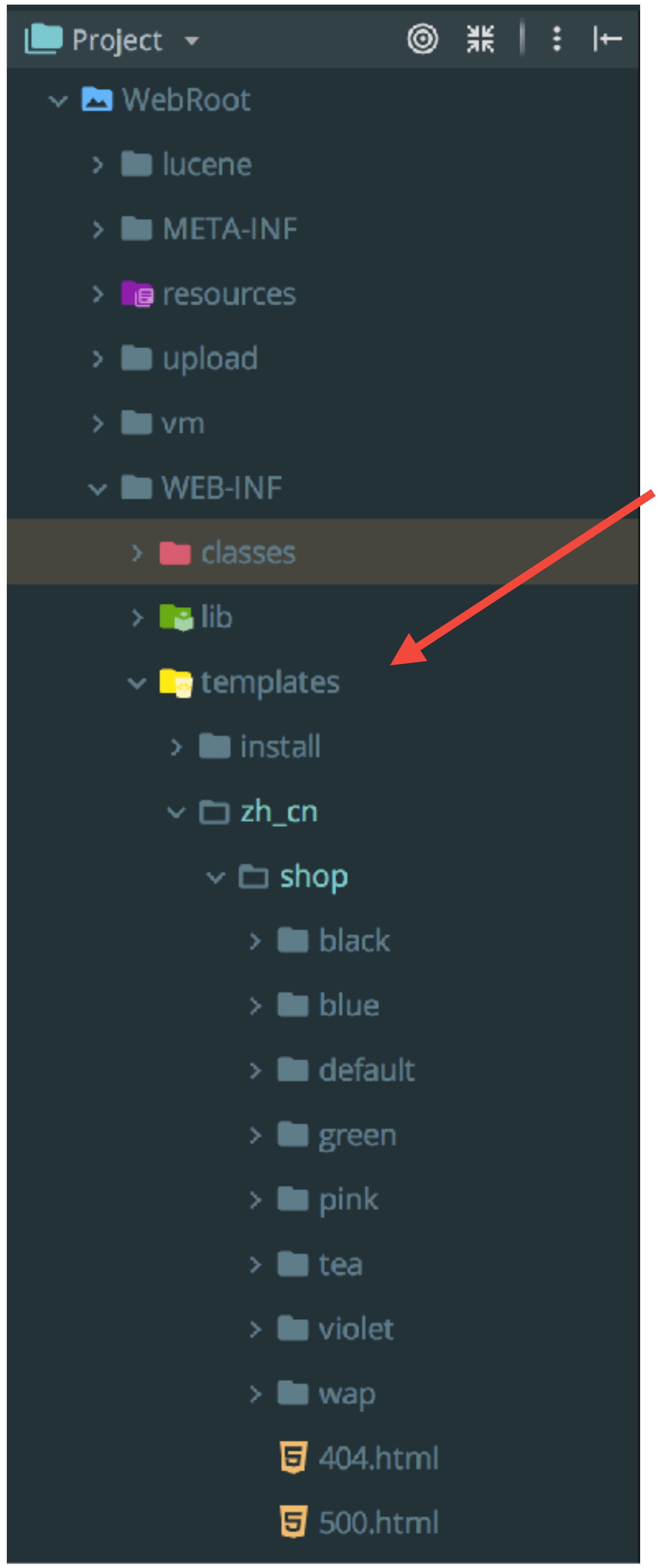
```
Project
 > .idea
 > .settings
 > out
 > resource
 v src
   v com.wemall
     > core
     v foundation
       > dao
       > domain
       > log
       > service
       > servlet
       > test
     > lucene
     > manage
     > pay
     > plug.login.action
     > uc.api
     v view.web
       > action
       > tools
```

数据访问实现，对应domain

领域模型，和表对应

服务类，实现业务，调用DAO

控制器，处理请求，调用服务
得到数据，跳转到视图



```
Project
 v WebRoot
   > lucene
   > META-INF
   > resources
   > upload
   > vm
   v WEB-INF
     > classes
     > lib
     v templates
       > install
       v zh_cn
         v shop
           > black
           > blue
           > default
           > green
           > pink
           > tea
           > violet
           > wap
           404.html
           500.html
```

视图类，组装页面

过滤器

Servlet filter用于在web应用中拦截requests和responses

```
<filter>
  <filter-name>errorHandlerFilter</filter-name>
  <filter-class>com.dharma.mall.ErrorHandleFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>errorHandlerFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
public class ErrorHandleFilter implements Filter {
    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
                                throws IOException, ServletException {
        try {
                chain.doFilter(request, response);
        } catch (Exception ex) {
                request.setAttribute("errorMessage", ex);
                request.getRequestDispatcher("/WEB-INF/views/jsp/error.jsp").forward(request, response);
        }
    }
}
```

URL 重写

UrlRewriteFilter是一个用于改写URL的Web过滤器，类似于Apache的mod_rewrite。适用于任何Web应用服务器（如Resin，Orion，Tomcat等）。其典型应用就把动态URL静态化，便于搜索引擎爬虫抓取你的动态网页。

***JSP*页面地址--> 服务器*Filter*过滤 --> 调用*urlrewrite.xml*映射规则 --> 服务器响应 --> 转换成伪地址**

```
<filter>
    <filter-name>UrlRewriteFilter</filter-name>
    <filter-class>org.tuckey.web.filters.urlrewrite.UrlRewriteFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>UrlRewriteFilter</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
</filter-mapping>
```

查看配置：http://127.0.0.1:8080/rewrite-status

参考：https://cdn.rawgit.com/paultuckey/urlrewritefilter/master/src/doc/manual/4.0/guide.html

Forward vs Redirect

- 重定向设置响应状态码302，新URL在Location头里，发送响应到浏览器。浏览器发送新URL请求。
- 转发都发生在服务器。servlet容器仅仅转发相同的请求到目标url，使用相同的请求属性，相同的请求参数，浏览器并不知道。
- 转发只有一次HTTP请求，在HttpServletRequest操作，这样在浏览器URL的地址栏不变，而重定向发了两个HTTP请求，由HttpServletResponse操作，在浏览器URL的地址栏改变

执行流程

Browser(request) ——>

Action(Controller) ——>

Service ——>

DAO(JDBC, JPA,HIB) ——> DB ——>

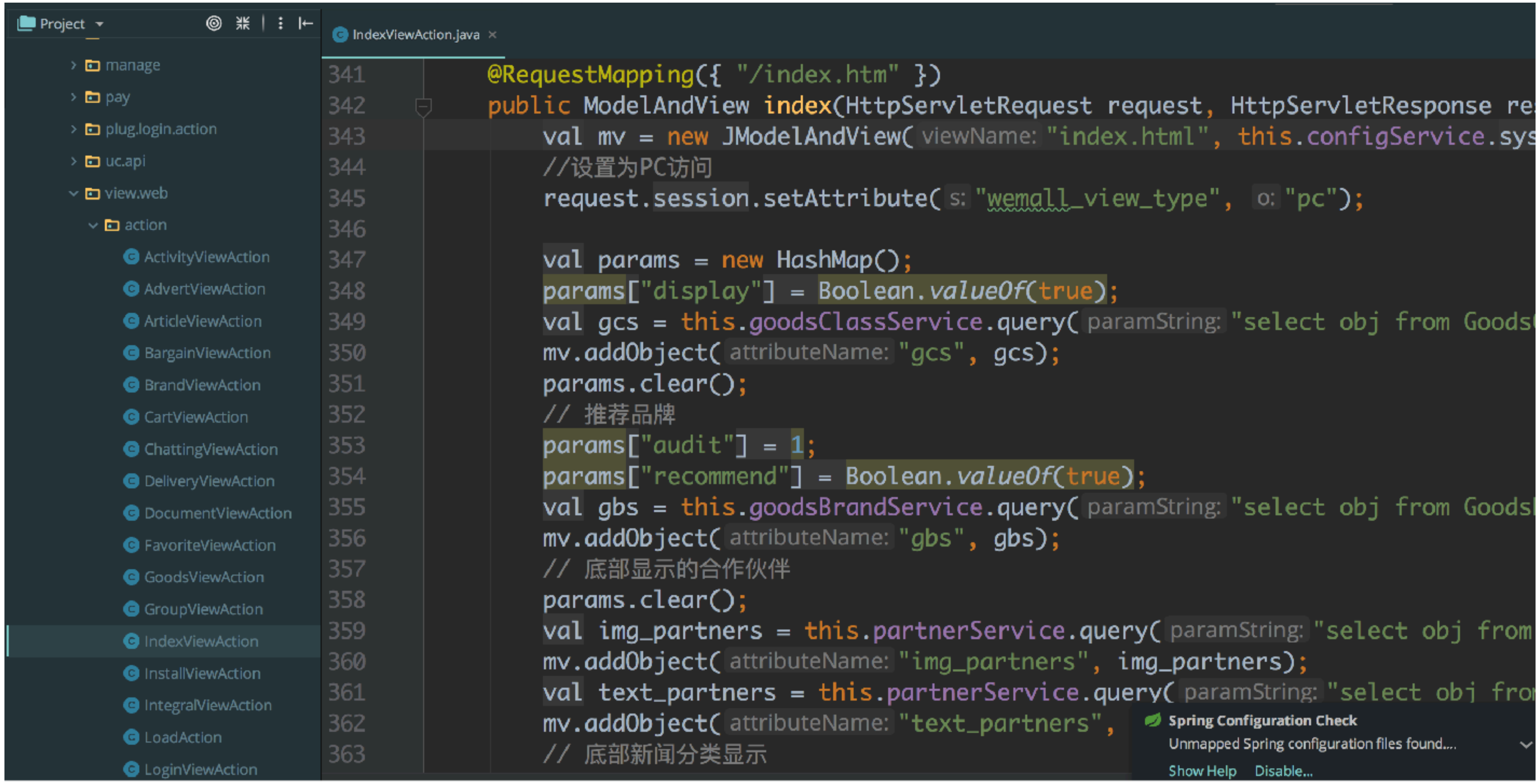
Service (Domain, Model) ——>

Action(ViewName + ModelAndView) ——>

View ——>

Browser(HTML)

执行流程

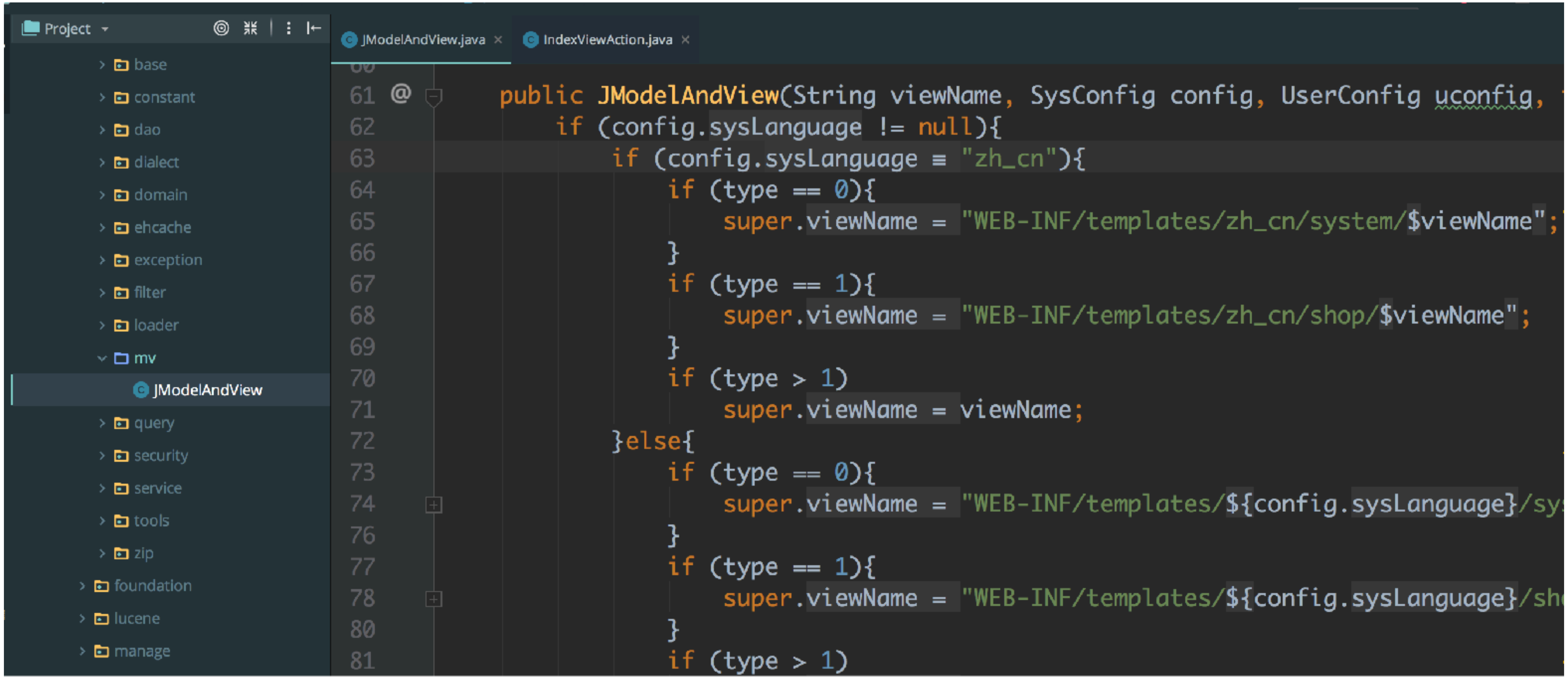


```
    @RequestMapping({ "/index.htm" })
    public ModelAndView index(HttpServletRequest request, HttpServletResponse re
        val mv = new JModelAndView( viewName: "index.html", this.configService.sys
        //设置为PC访问
        request.session.setAttribute( s: "wemall_view_type",  o: "pc");

        val params = new HashMap();
        params["display"] = Boolean.valueOf(true);
        val gcs = this.goodsClassService.query( paramString: "select obj from Goods
        mv.addObject( attributeName: "gcs", gcs);
        params.clear();
        // 推荐品牌
        params["audit"] = 1;
        params["recommend"] = Boolean.valueOf(true);
        val gbs = this.goodsBrandService.query( paramString: "select obj from Goods
        mv.addObject( attributeName: "gbs", gbs);
        // 底部显示的合作伙伴
        params.clear();
        val img_partners = this.partnerService.query( paramString: "select obj from
        mv.addObject( attributeName: "img_partners", img_partners);
        val text_partners = this.partnerService.query( paramString: "select obj fro
        mv.addObject( attributeName: "text_partners",
        // 底部新闻分类显示
```

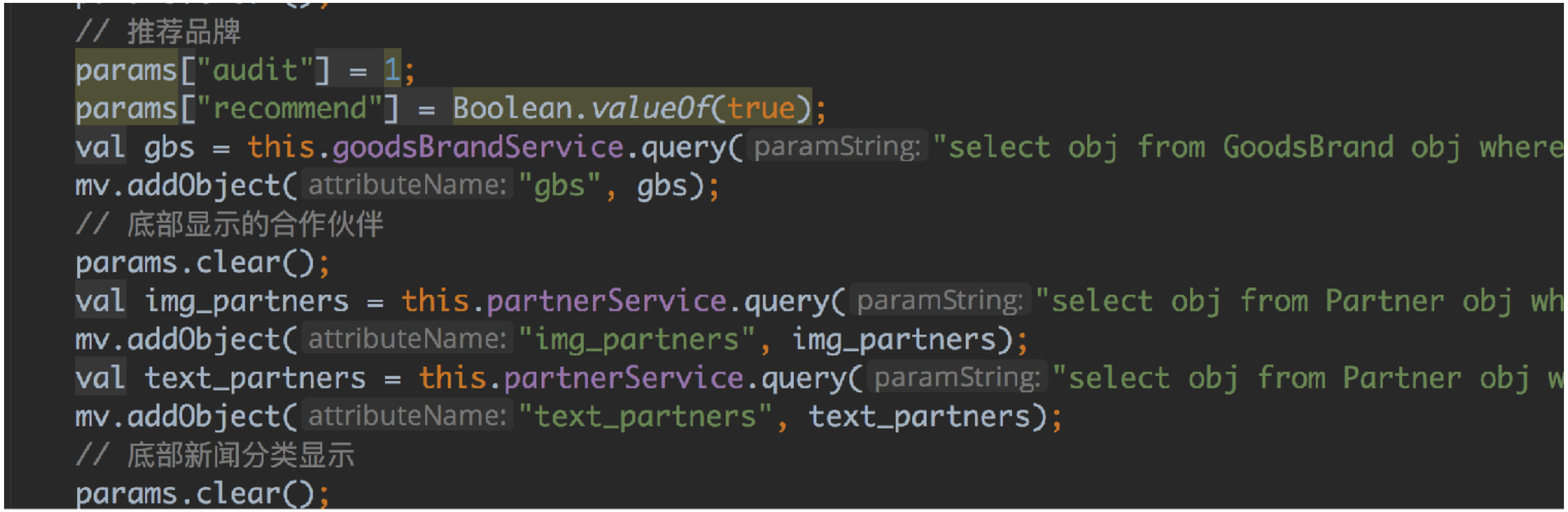
Spring Configuration Check
Unmapped Spring configuration files found....
Show Help Disable...

执行流程



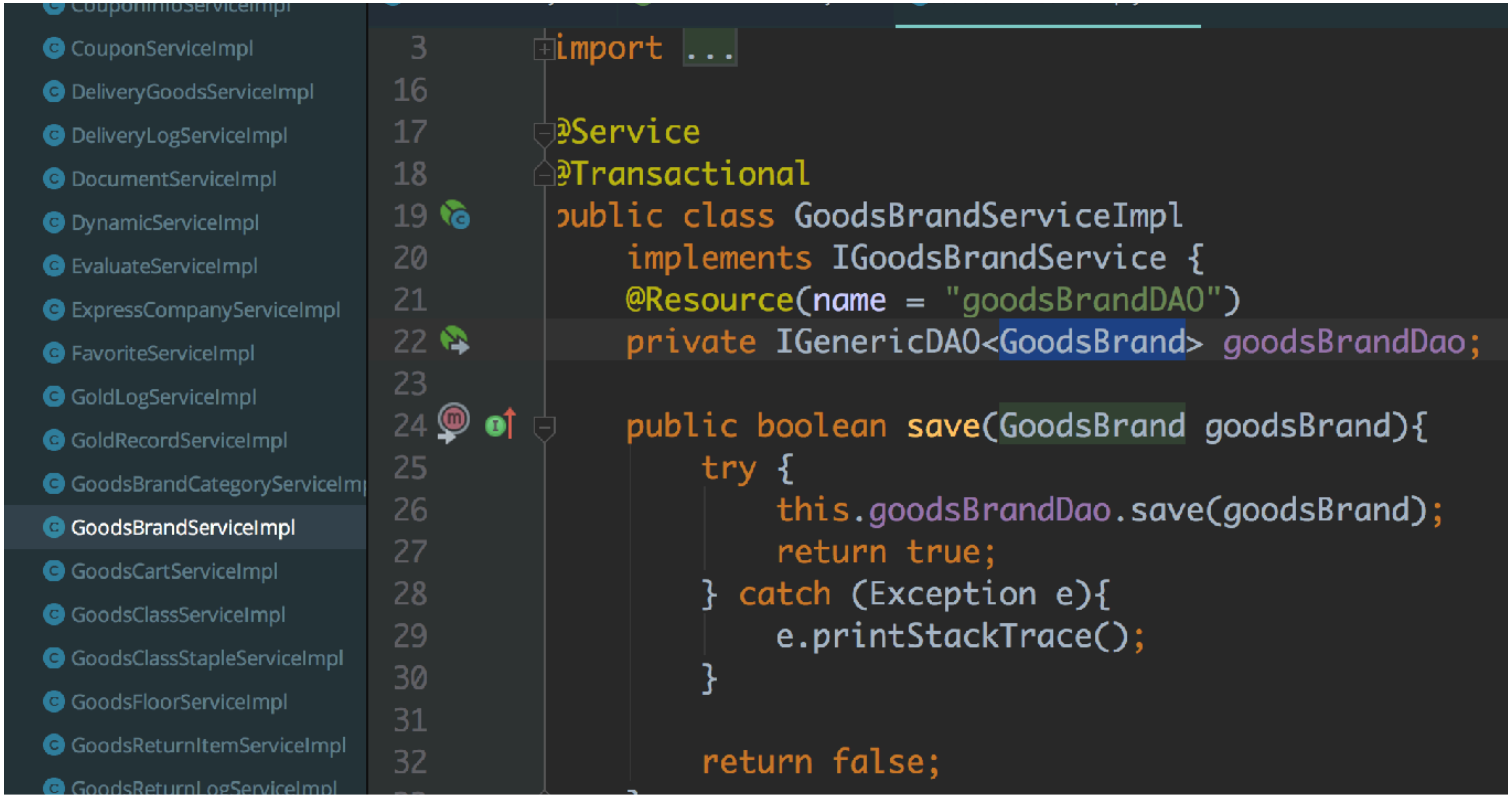
```
    public JModelAndView(String viewName, SysConfig config, UserConfig uconfig,
        if (config.sysLanguage != null){
            if (config.sysLanguage = "zh_cn"){
                if (type == 0){
                    super.viewName = "WEB-INF/templates/zh_cn/system/$viewName";
                }
                if (type == 1){
                    super.viewName = "WEB-INF/templates/zh_cn/shop/$viewName";
                }
                if (type > 1)
                    super.viewName = viewName;
            }else{
                if (type == 0){
                    super.viewName = "WEB-INF/templates/${config.sysLanguage}/sy
                }
                if (type == 1){
                    super.viewName = "WEB-INF/templates/${config.sysLanguage}/sh
                }
                if (type > 1)
```

执行流程



```
// 推荐品牌
params["audit"] = 1;
params["recommend"] = Boolean.valueOf(true);
val gbs = this.goodsBrandService.query( paramString: "select obj from GoodsBrand obj where
mv.addObject( attributeName: "gbs", gbs);
// 底部显示的合作伙伴
params.clear();
val img_partners = this.partnerService.query( paramString: "select obj from Partner obj wh
mv.addObject( attributeName: "img_partners", img_partners);
val text_partners = this.partnerService.query( paramString: "select obj from Partner obj w
mv.addObject( attributeName: "text_partners", text_partners);
// 底部新闻分类显示
params.clear();
```

执行流程



CouponInfoServiceImpl
CouponServiceImpl
DeliveryGoodsServiceImpl
DeliveryLogServiceImpl
DocumentServiceImpl
DynamicServiceImpl
EvaluateServiceImpl
ExpressCompanyServiceImpl
FavoriteServiceImpl
GoldLogServiceImpl
GoldRecordServiceImpl
GoodsBrandCategoryServiceImp
GoodsBrandServiceImpl
GoodsCartServiceImpl
GoodsClassServiceImpl
GoodsClassStapleServiceImpl
GoodsFloorServiceImpl
GoodsReturnItemServiceImpl
GoodsReturnLogServiceImpl

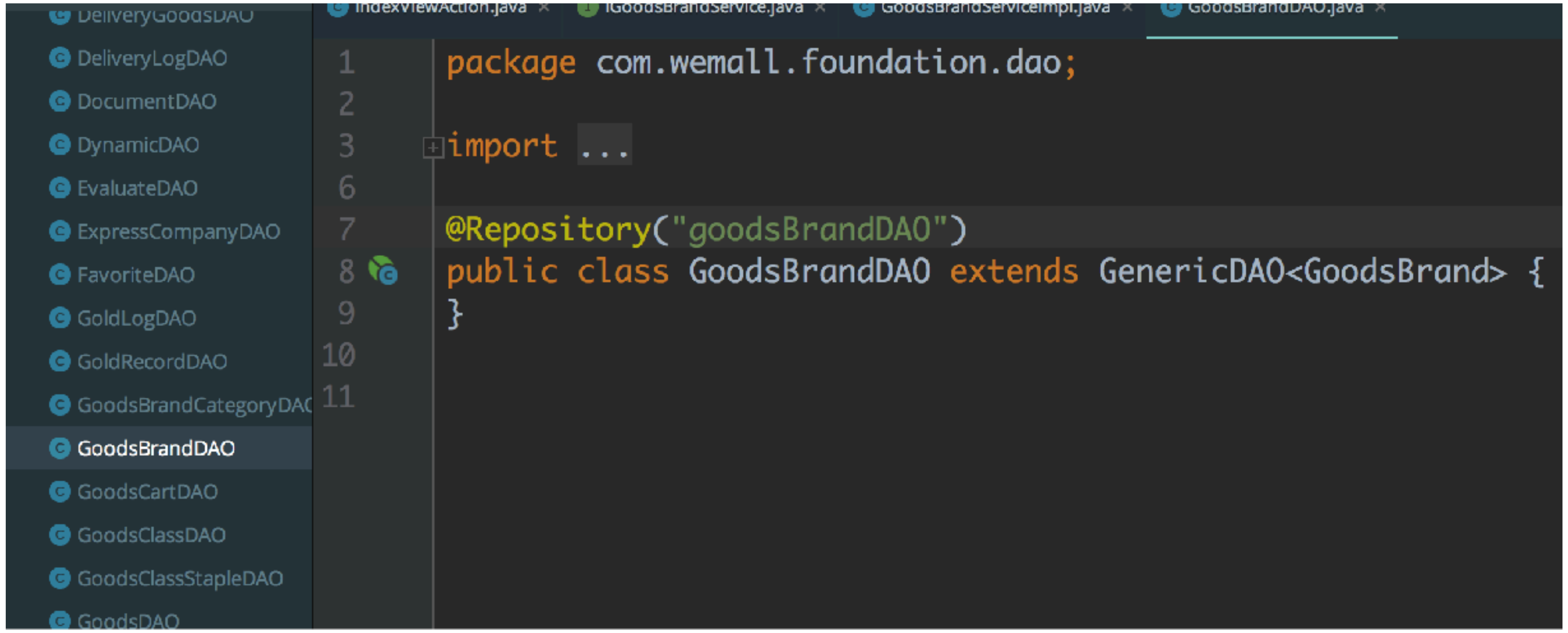
```java
import ...

@Service
@Transactional
public class GoodsBrandServiceImpl
    implements IGoodsBrandService {
    @Resource(name = "goodsBrandDAO")
    private IGenericDAO<GoodsBrand> goodsBrandDao;

    public boolean save(GoodsBrand goodsBrand){
        try {
            this.goodsBrandDao.save(goodsBrand);
            return true;
        } catch (Exception e){
            e.printStackTrace();
        }

        return false;
```

执行流程

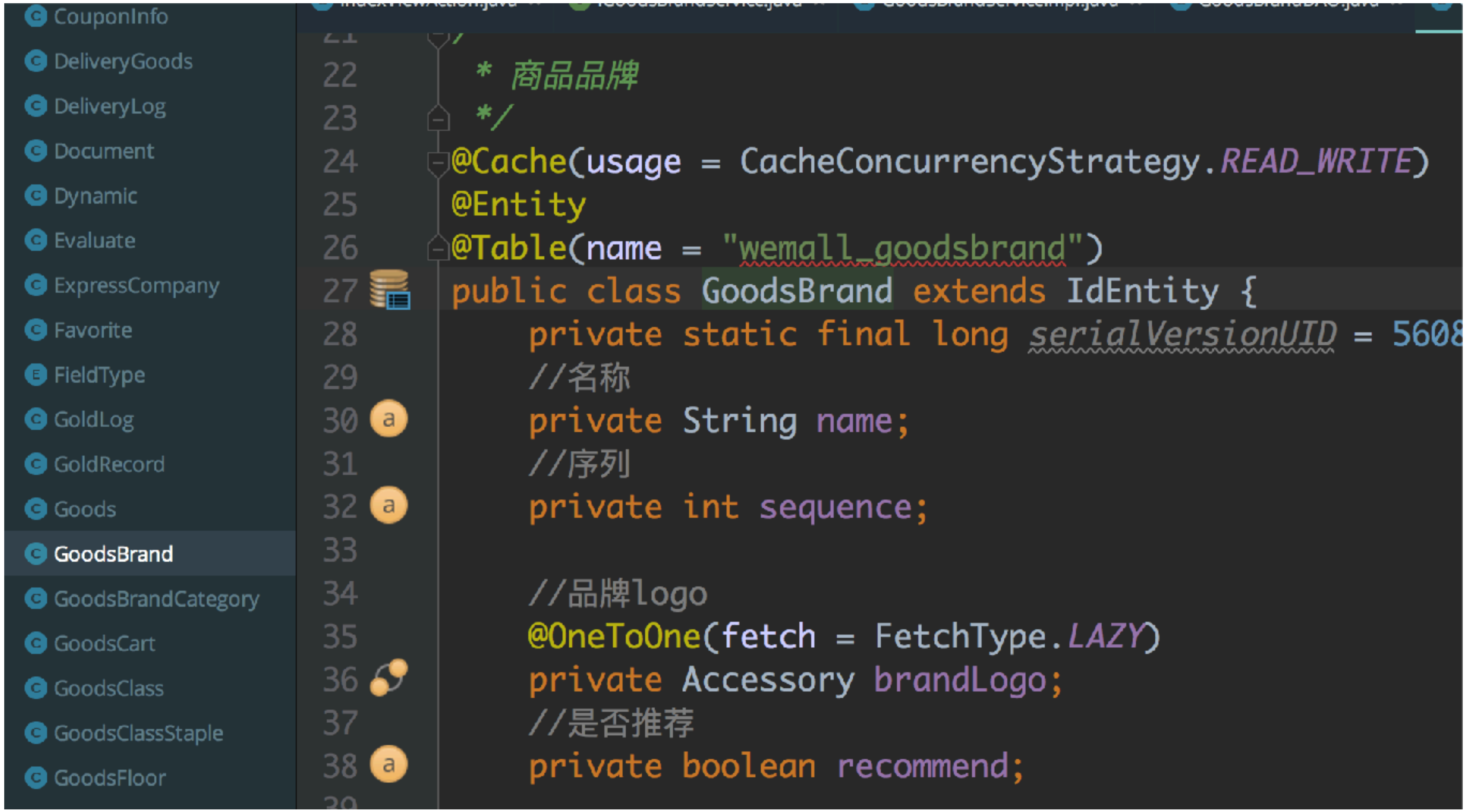


```java
package com.wemall.foundation.dao;

import ...

@Repository("goodsBrandDAO")
public class GoodsBrandDAO extends GenericDAO<GoodsBrand> {
}
```

执行流程

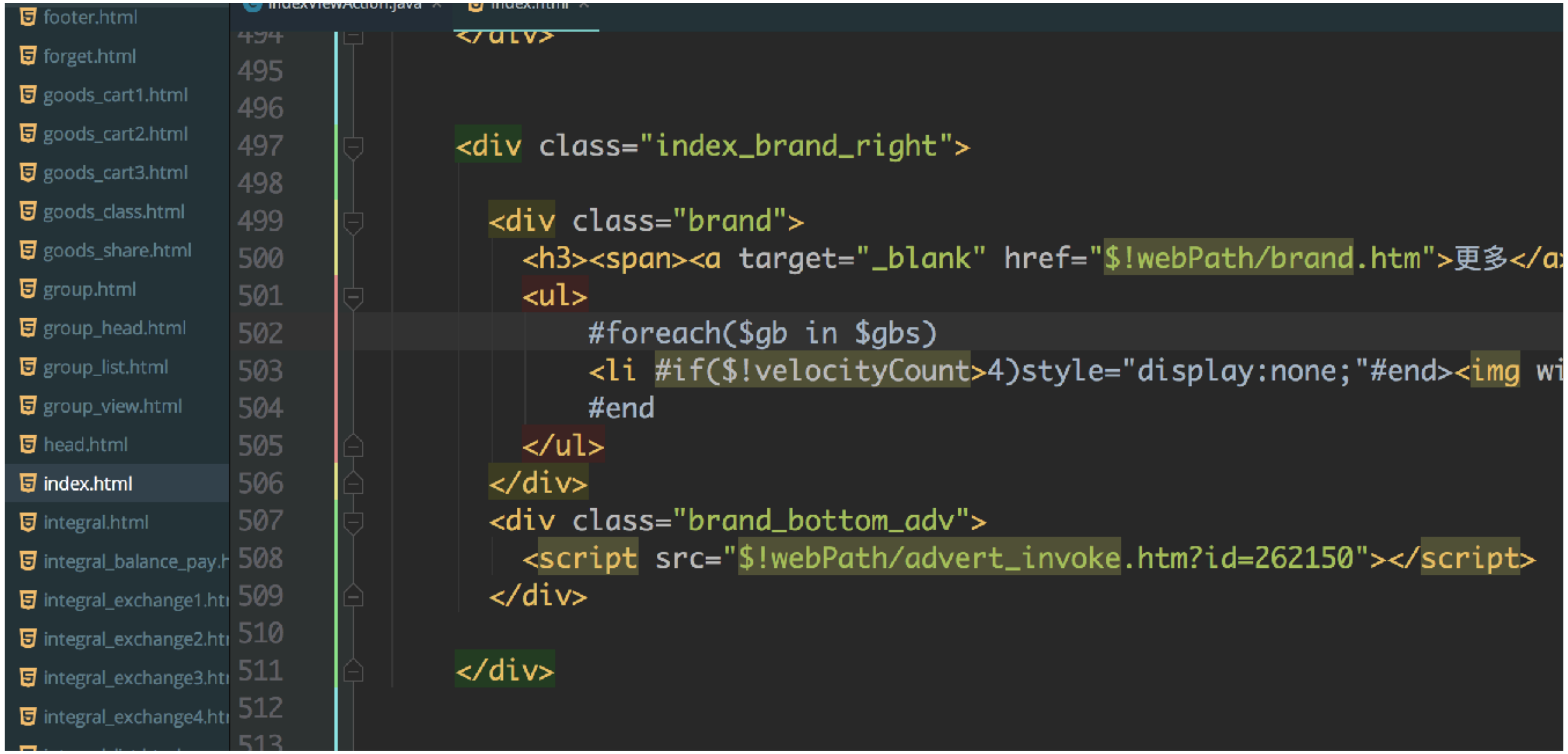


CouponInfo
DeliveryGoods
DeliveryLog
Document
Dynamic
Evaluate
ExpressCompany
Favorite
FieldType
GoldLog
GoldRecord
Goods
GoodsBrand
GoodsBrandCategory
GoodsCart
GoodsClass
GoodsClassStaple
GoodsFloor

```java
 * 商品品牌
 */
@Cache(usage = CacheConcurrencyStrategy.READ_WRITE)
@Entity
@Table(name = "wemall_goodsbrand")
public class GoodsBrand extends IdEntity {
    private static final long serialVersionUID = 5608
    //名称
    private String name;
    //序列
    private int sequence;

    //品牌logo
    @OneToOne(fetch = FetchType.LAZY)
    private Accessory brandLogo;
    //是否推荐
    private boolean recommend;
```

执行流程



```
    </div>


    <div class="index_brand_right">

      <div class="brand">
        <h3><span><a target="_blank" href="$!webPath/brand.htm">更多</a
        <ul>
            #foreach($gb in $gbs)
            <li #if($!velocityCount>4)style="display:none;"#end><img wi
            #end
        </ul>
      </div>
      <div class="brand_bottom_adv">
        <script src="$!webPath/advert_invoke.htm?id=262150"></script>
      </div>

    </div>
```

DEMO

Thanks!

Any questions?