# 《踩坑日记》

## 2019.12.15

编号：2019.12.15.1

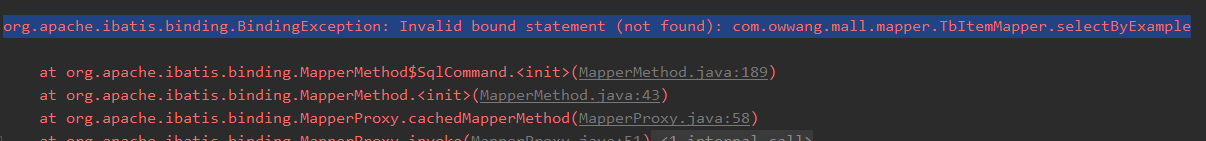
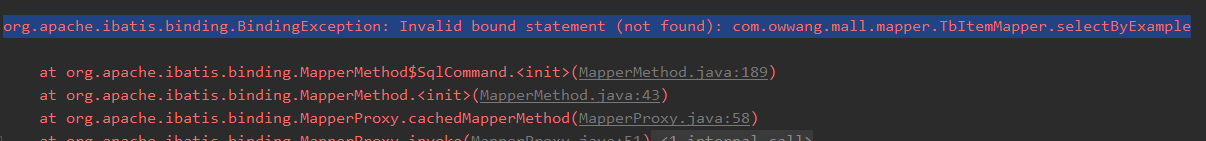
### 标题：Maven工程MyEclipse转移到IDEA后报错

**org.apache.ibatis.binding.BindingException: Invalid bound statement (not found): com.owwang.mall.mapper.TbItemMapper.selectByExample 错误**

**1.问题**

项目从myEclipse转移到IDEA后，出现了无法找到配置资源，造成无法注入对象

org.apache.ibatis.binding.BindingException: Invalid bound statement (not found): com.owwang.mall.mapper.TbItemMapper.selectByExample



**2.原因**

如果我们的mapper.xml文件没有放置到src-main-resources下面，是不会被maven build plugin给默认扫描到的。此时需要修改启动的模块的pom文件，在build标签里面加入：（解决方法中）

**参考文章链接：**[**https://www.jianshu.com/p/a9516bcd3cb0**](https://www.jianshu.com/p/a9516bcd3cb0)

**3.解决方法**

加入：

<build>

<resources>

<resource>

<directory>src/main/java</directory>

<includes>

<include>\*\*/\*.xml</include>

<include>\*\*/\*.properties</include>

</includes>

<filtering>true</filtering>

</resource>

<resource>

<directory>src/main/resources</directory>

<includes>

<include>\*\*/\*.xml</include>

<include>\*\*/\*.properties</include>

</includes>

<filtering>true</filtering>

</resource>

</resources></build>

编号：2019.12.15.2

### 标题：bubbo注入失败

Caused by: org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'contentCategoryServiceImpl' is defined 注入失败

**1.问题**

Caused by: org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'contentCategoryServiceImpl' is defined

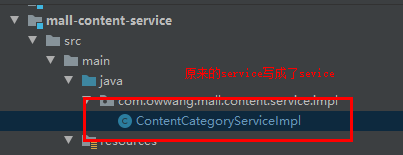
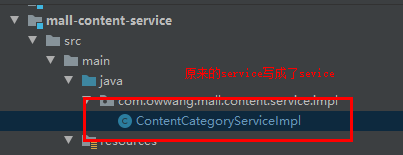
content-service 服务注入失败

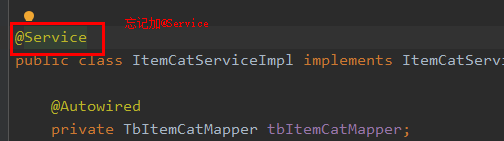
**2.原因**

（问题1）实现类的方法名写错了  引以为戒！！！

（问题2）忘记加上@Service

**3.解决方法**

****



编号：2019.12.15.3

### 标题：新版IDEA中文显示不正常

**1.问题**

****

文字显示不规整

**2.原因**

设置问题

**3.解决方法**

Settings-->Editor-->Font-->Fallback font-->SimHei

## 2019.12.16

编号：2019.12.16.1

### 标题：编写dao中的sql时，xml文件中背景大片黄色

**1.问题**

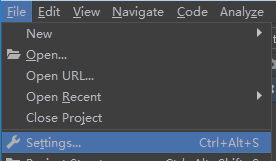
编写dao中的sql时,xml文件中背景一大片黄色，看着不舒服

**2.原因**

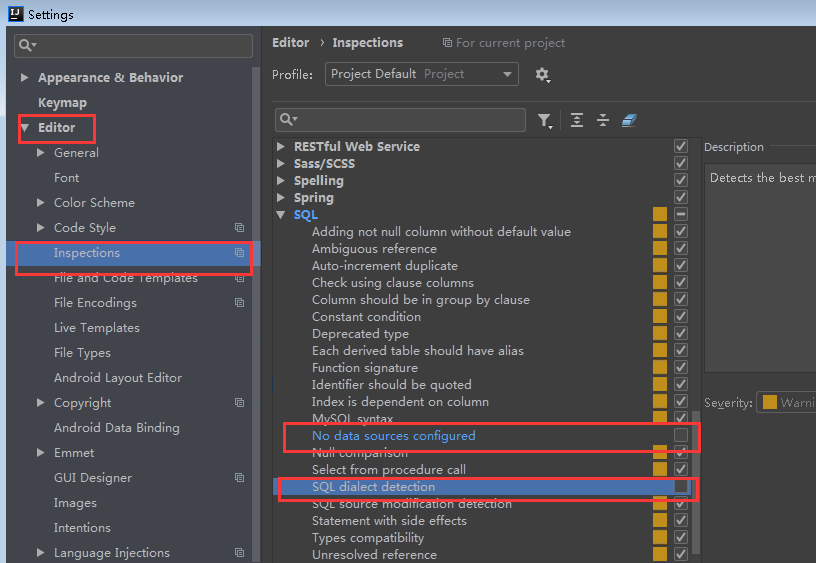
改设置即可

**3.解决方法**

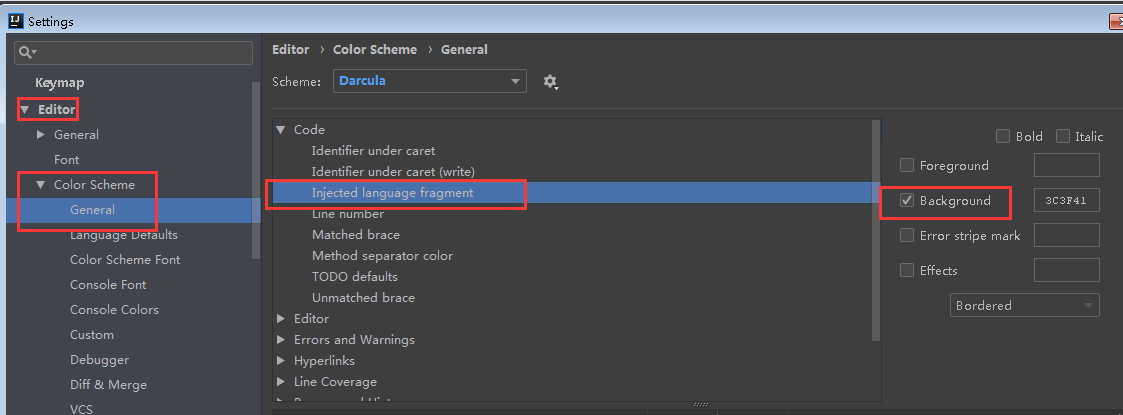
（1） File -> Settings...



（2）去消以下两项勾选    (Inspections    -- 如果找不到，直接在左上角查询框中搜索)



（3）取消勾选 Background    (Injected language fragment    -- 如果找不到，直接在左上角查询框中搜索)



（4）点击保存

编号：2019.12.16.2

### 标题：easyui中的datagrid 分页传入参数

**1.问题**

使用easyui中的datagrid 分页时，无法理解为什么：向后台传参数page 和rows

**2.原因**

不了easyUI的API

**3.解决方法**

传到后台有两个参数：  
page 当前页码  
rows 传当前页  
  
后台传回前台一个json字符串  
{  
    total: 15,//数据库中记录数条数  
    rows:[  
        {},  //本页第一条记录  
        {},  //本页第二条记录  
        {},  
        …  
        {},  
        {}   //本页第十条记录  
    ]  
}

Service中中可创建类进行回传：

public class EasyUIDataGriResult implements Serializable {  
 private Integer total;  
 private List<?> rows;}  
注：rows中只要放一页数据就好，翻页时，datagrid 会重新传 page 到后台的,收到新的page时，再重新组织另一页数据的json字符串

而service中的功能使用pagehelper实现

编号：2020.12.16.3

### 标题：PageHelper使用方法

**1.问题**

Pagehelper使用方法

**2.原因**

略

**3.解决方法**

**（1）依赖：**

<!-- pagehelper分页工具 -->

<dependency>

<groupId>com.github.miemiedev</groupId>

<artifactId>mybatis-paginator</artifactId>

</dependency>

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

</dependency>

**（2）Mybatis分页插件 - PageHelper说明**

如果你也在用Mybatis，建议尝试该分页插件，这个一定是最方便使用的分页插件。

该插件目前支持Oracle,Mysql,MariaDB,SQLite,Hsqldb,PostgreSQL六种数据库分页。

**（3）使用方法**

第一步：在Mybatis配置xml中配置拦截器插件:

<plugins>

<!-- com.github.pagehelper为PageHelper类所在包名 -->

<plugin interceptor="com.github.pagehelper.PageHelper">

<!-- 设置数据库类型 Oracle,Mysql,MariaDB,SQLite,Hsqldb,PostgreSQL六种数据库-->

<property name="dialect" value="mysql"/>

</plugin>

</plugins>

第二步：在代码中使用

1、设置分页信息：

//获取第1页，10条内容，默认查询总数count

PageHelper.startPage(1, 10);

//紧跟着的第一个select方法会被分页

List<Country> list = countryMapper.selectIf(1);

2、取分页信息

//分页后，实际返回的结果list类型是Page<E>，如果想取出分页信息，需要强制转换为Page<E>，

Page<Country> listCountry = (Page<Country>)list;

listCountry.getTotal();

1. 取分页信息的第二种方法

//获取第1页，10条内容，默认查询总数count

PageHelper.startPage(1, 10);

List<Country> list = countryMapper.selectAll();

//用PageInfo对结果进行包装

PageInfo page = new PageInfo(list);

//测试PageInfo全部属性

//PageInfo包含了非常全面的分页属性

assertEquals(1, page.getPageNum());

assertEquals(10, page.getPageSize());

assertEquals(1, page.getStartRow());

assertEquals(10, page.getEndRow());

assertEquals(183, page.getTotal());

assertEquals(19, page.getPages());

assertEquals(1, page.getFirstPage());

assertEquals(8, page.getLastPage());

assertEquals(true, page.isFirstPage());

assertEquals(false, page.isLastPage());

assertEquals(false, page.isHasPreviousPage());

assertEquals(true, page.isHasNextPage());

## 2019.12.17

编号：2019.12.17.1

### 标题：IDEA 对MAVEN聚合工程DBUG

**1.问题**

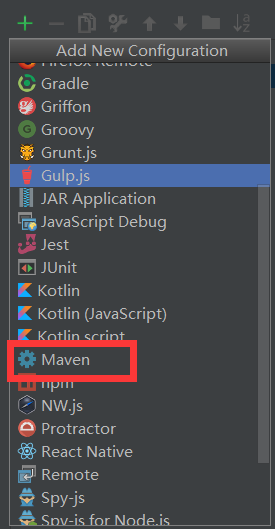
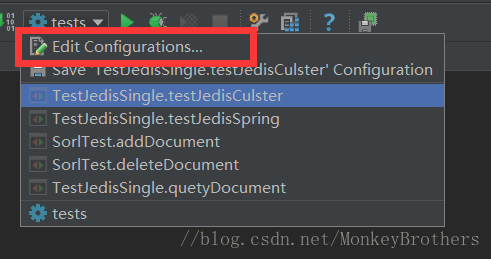
Eclipseclipse转到IDEA，对于如何对聚合进行DBUG不够熟悉

**2.原因**

略

**3.解决方法**

（1）首先进入编辑构造

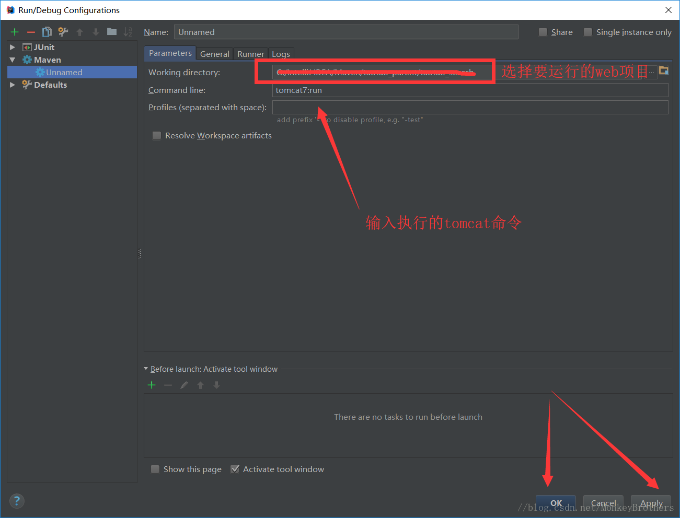


**第二步**

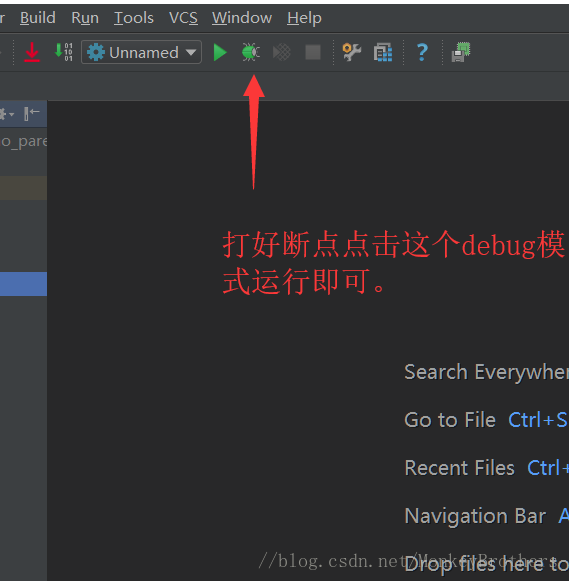
**第一步**

（2）选择绿色的“+”号往下拉找到“Maven”这个选项点进去

（3）选择要debug运行的项目

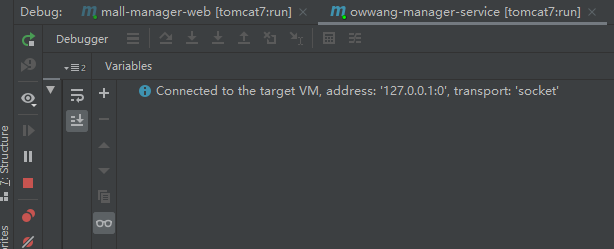


（4）debug运行即可



（5）如果多个maven项目需要同时dbug，只需按照以上步骤配置。然后再逐一启动即可





（6）DEBUG快捷键：

F7：一行一行测，进入嵌套方法中

F8：一行一行测，不进入嵌套方法

F9：一个断点一个断点测，跳到下一个断点

## 2019.12.18

编号：2019.12.18.1

### 标题：dubbo的Multipart文件上传问题

**1.问题**

采用dubbo对MultipartFile文件进行传输时，会报invoke method 错误

[Fail to decode request due to: RpcInvocation](https://www.cnblogs.com/driveliu/p/10578138.html)

**2.原因**

因为dubbo并不能跨系统传递这个对象，可以采用Hessian

**3.解决方法**

（1）引入Hessian依赖，消费者和提供者中都要引入。注意，jetty包中要排除servlet-api包的依赖，否则会报错，Tomcat不能正常启动。

<!-- Hessian -->

<dependency>  
 <groupId>com.caucho</groupId>  
 <artifactId>hessian</artifactId>  
 <version>4.0.7</version>  
</dependency>  
<dependency>  
 <groupId>org.mortbay.jetty</groupId>  
 <artifactId>jetty</artifactId>  
 <version>6.1.26</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.mortbay.jetty</groupId>  
 <artifactId>servlet-api</artifactId>  
 </exclusion>  
 </exclusions>  
</dependency>

（2）消费者和提供者SpringMVC、Service配置接口暴露。主意消费者要设置一下timeout，不然会连接超时报错

<dubbo:referenceprotocol="hessian"interface="com.owwang.mall.

service.PictureService" id="pictureService" timeout="300000"/>

<dubbo:service protocol="hessian" interface="com.owwang.mall.service.PictureService" ref="pictureServiceImpl"/>

（3）另外，dubbo跨系统传递流，会使得流为空，最简单有效的方法是将流转变为byte数组，然后再将byte数组转化为流，即用byte为中介传递

@RequestMapping("/pic/upload")

@ResponseBody

public PictureResult upload(@RequestParam MultipartFile uploadFile) throws IOException {

System.out.println(uploadFile.getSize());

byte[] pic = uploadFile.getBytes();

PictureResult result = pictureService.uploadPicture(pic, uploadFile);

return result;

}

---------------------------------------------

@Override

public PictureResult uploadPicture(byte[] pic, MultipartFile uploadFile) {

if(pic == null || pic.length == 0) {

return PictureResult.error("空文件");

}

String oldName = uploadFile.getOriginalFilename();

String newName = UUID.randomUUID() + oldName.substring(oldName.lastIndexOf("."));

String filePath = new DateTime().toString("/yyyy/MM/dd");

boolean res;

res = FtpUtils.uploadFile(FTP\_ADDRESS, FTP\_PORT, FTP\_USER\_NAME, FTP\_PASSWORD,

FTP\_BASE\_PATH, filePath, newName, new ByteArrayInputStream(pic));

if(!res) {

return PictureResult.error("上传失败");

}

String url = FTP\_BASE\_PATH + filePath + "/" + newName;

return PictureResult.ok(url);

}

编号：2019.12.18.2

### 标题：IDEA maven 打包时去掉 test 选项

**1.问题**

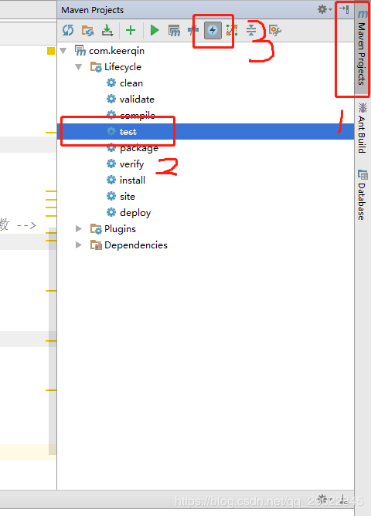
每次安装install的时候都test，非常慢，非常浪费时间

**2.原因**

略

**3.解决方法**

简单粗暴

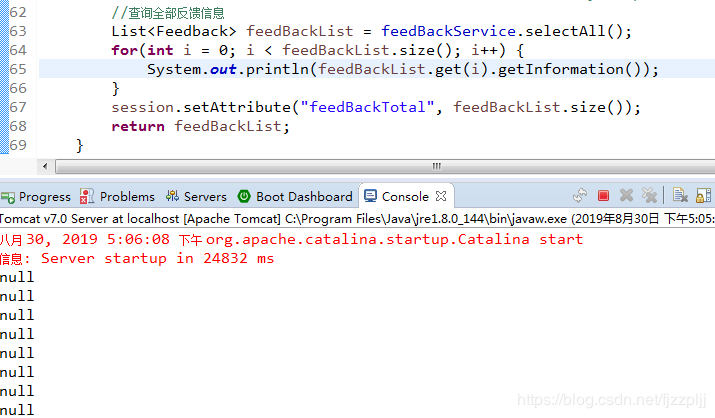


编号：2019.12.18.3

### 标题：逆向工程的dao获取数据库中的text类型数据

**1.问题**

在用ssm框架为layui数据表格写接口时，发现我的数据库text数据类型的数据在后台一直获取不到。而且数据库的时间过来的是乱的数字。

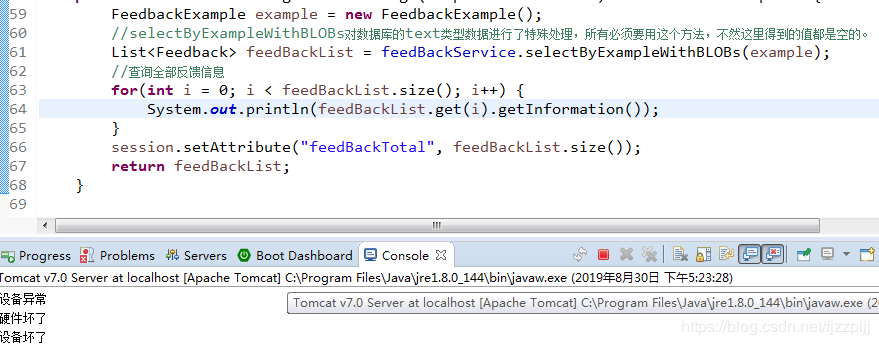


**2.原因**

使用的是mybatis逆向工程生成代码，ssm获取text类型的数据要通过selectByExampleWithBLOBs(example)来实现

**3.解决方法**

获取text类型的数据要通过selectByExampleWithBLOBs(example)来实现



## 2019.12.19

编号：2020.12.19.1

### 标题：链表下标越界异常

**1.问题**

在IDEA上开发springboot项目时遇到一个链表下标越界的异常

|  |
| --- |
| 严重: Servlet.service() for servlet [mall-manager-web] in context with path [] threw exception [Request processing failed; nested exception is java.lang.IndexOutOfBoundsException: Index: 0, Size: 0] with root cause  java.lang.IndexOutOfBoundsException: Index: 0, Size: 0 |

**2.原因**

当程序执行到if的情况时，if条件设置错误，进入tbItemParams.get(0)

的size为0，是个空链表，若此时再调用get方法，就会报错

**3.解决方法**

设置正确if条件，当没有查询到数据的时候，不要get(0)获取数据

|  |
| --- |
| if(tbItemParams!=null&&tbItemParams.size()>0){  return MallResult.ok(tbItemParams.get(0)); } |

**4.总结**

在往数据库添加数据需要判断数据库中是否已有记录，判断的返回结果通常是List。在List为空的情况下，调用其方法需要格外注意，例如：调用get()则会报下标越界的异常。当然还可以联想到其他情况，当判断数组、集合为空后，注意不要取其中内容，不然就会报上述错误。

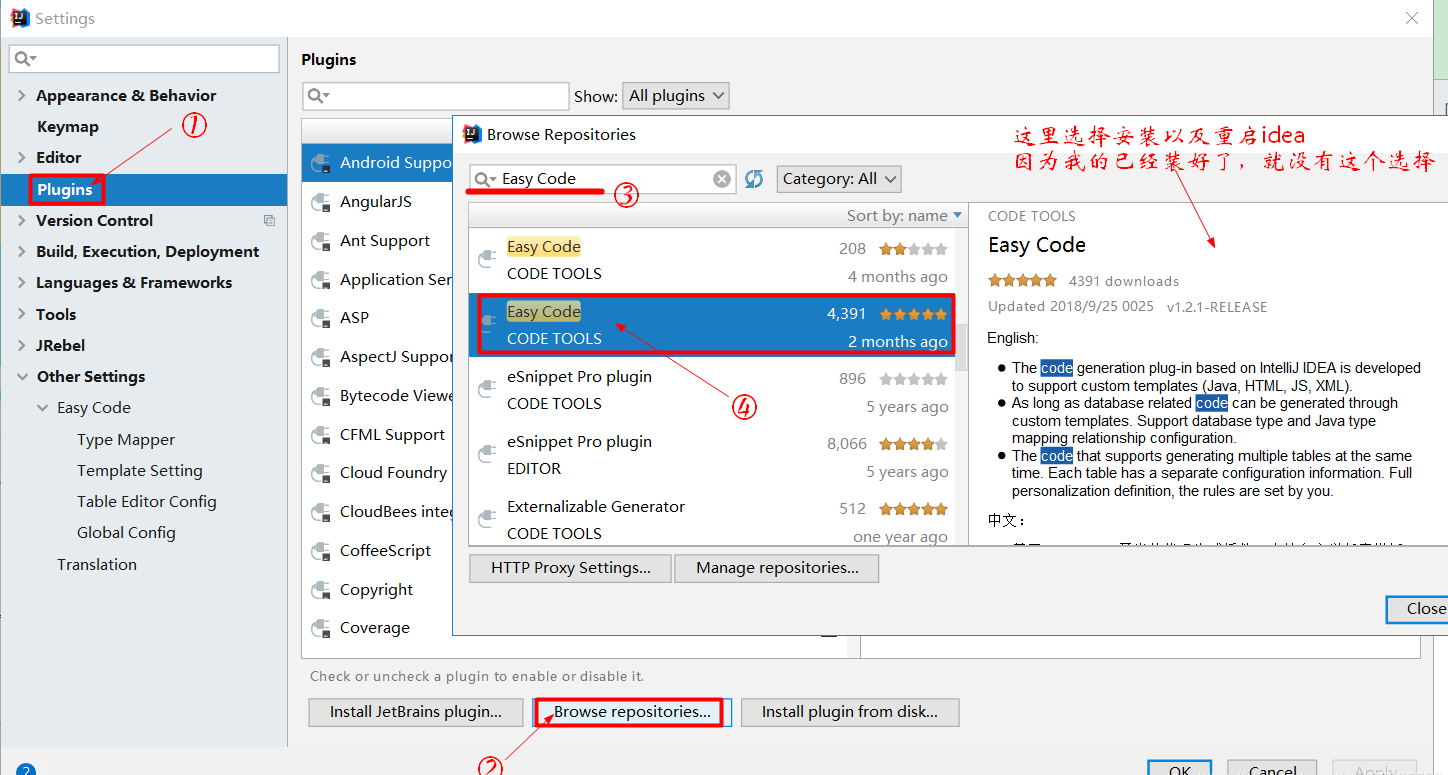
## 2019.12.20

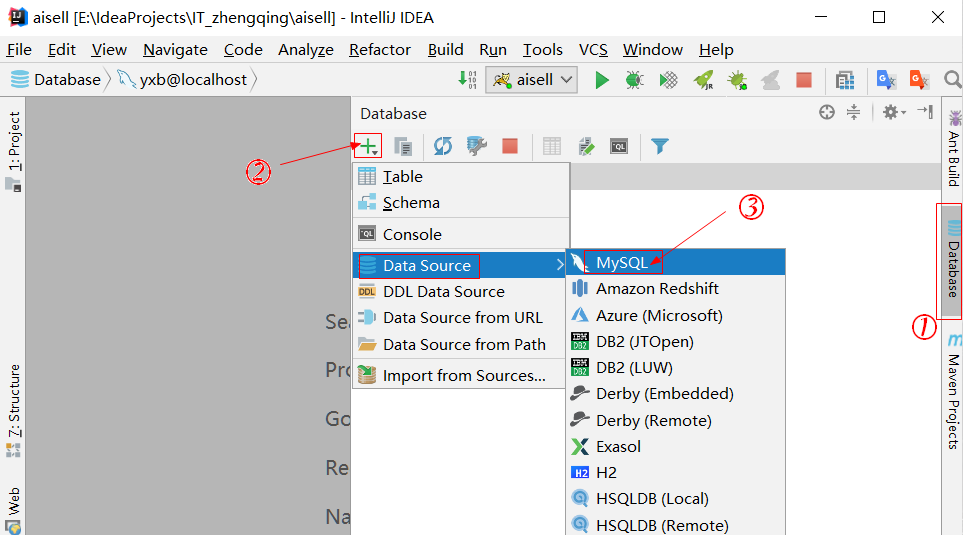
编号：2020.12.20.1

### 标题：EasyCode使用方法

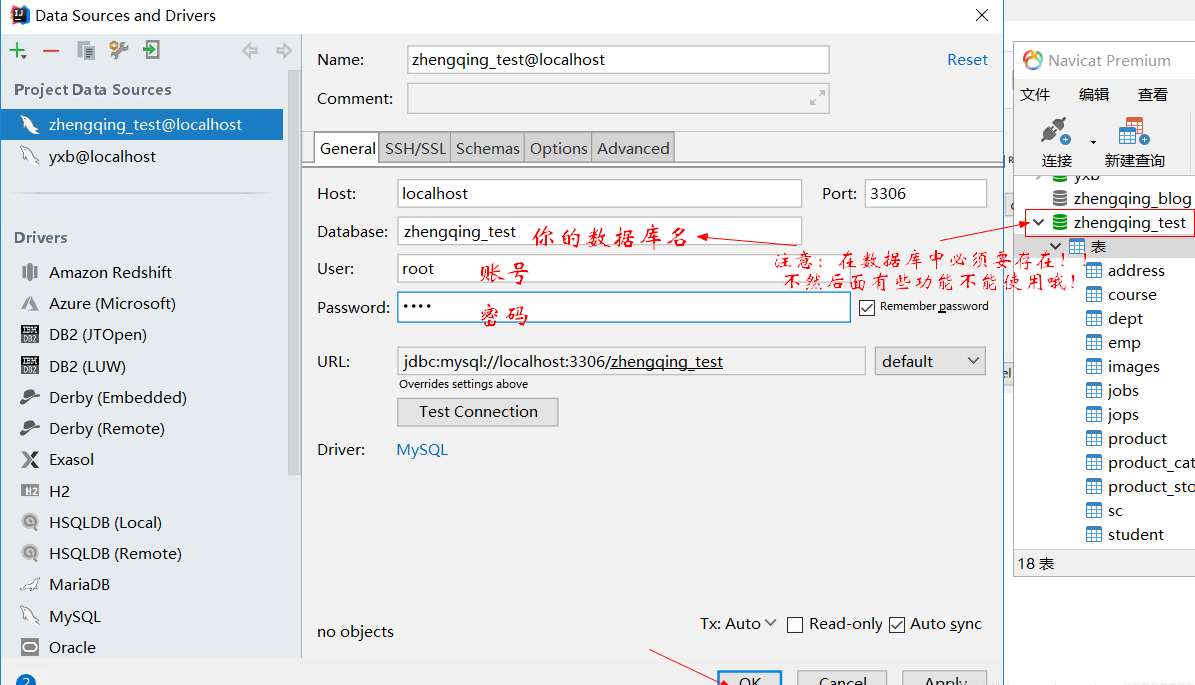
**1.idea安装EasyCode**

ctrl+alt+s --> 选择Plugins --> 安装Easy Code插件   -->  安装之后注意重启idea！



**2.配置数据库添加数据源**

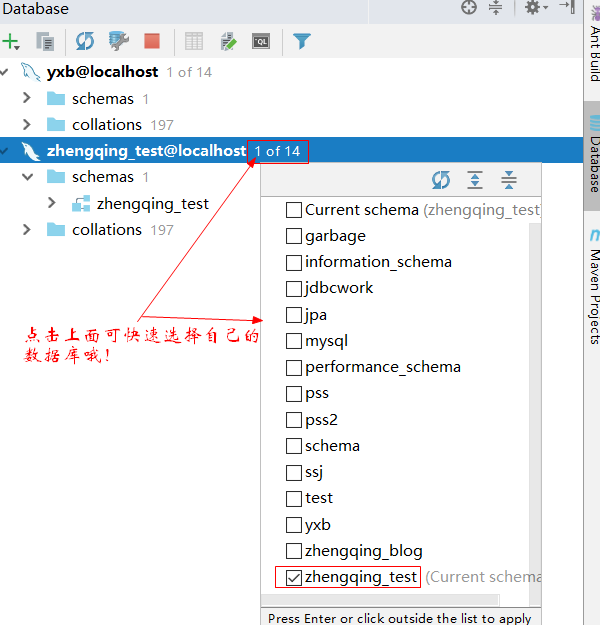
连接数据库，注意下面的Database要么不填写，要么就必须填写你数据库中对应的数据库名哦！！



之后等待idea最下面的显示加载完成

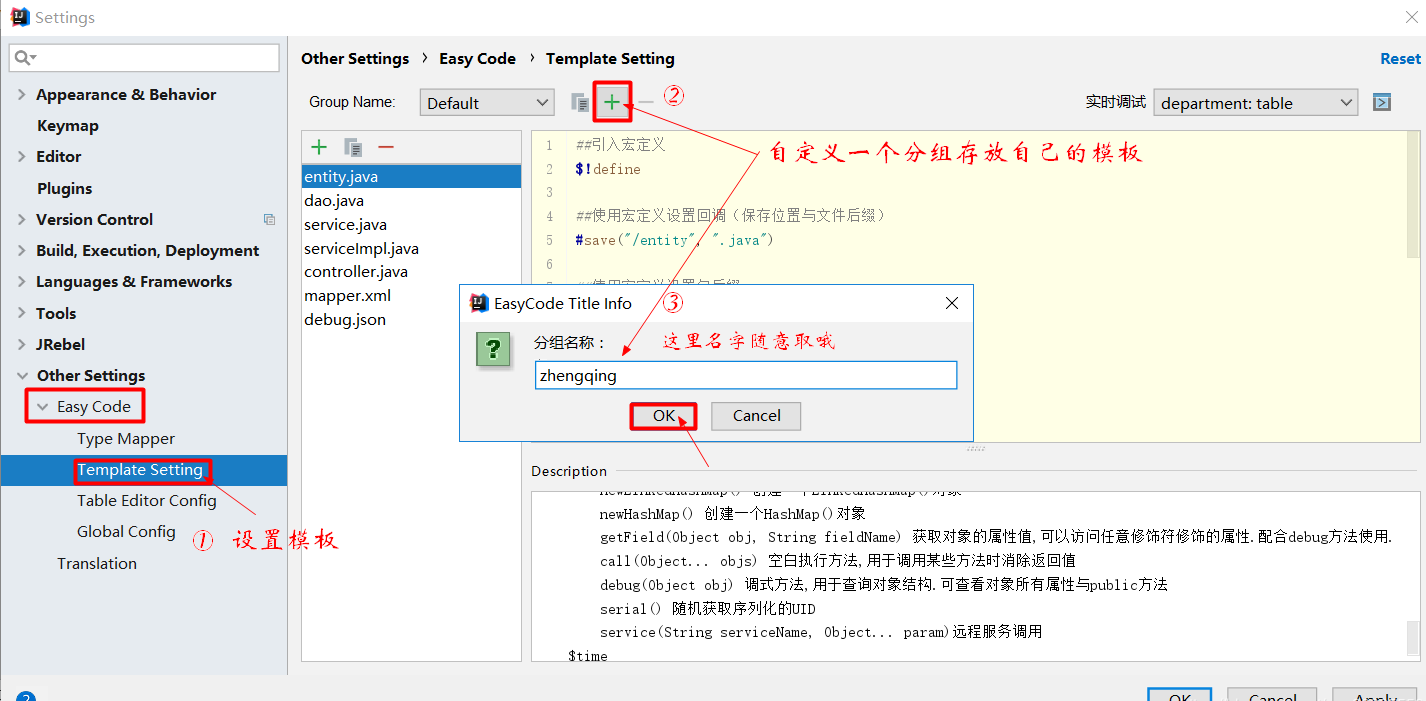


最后就把数据源添加好了哦~~

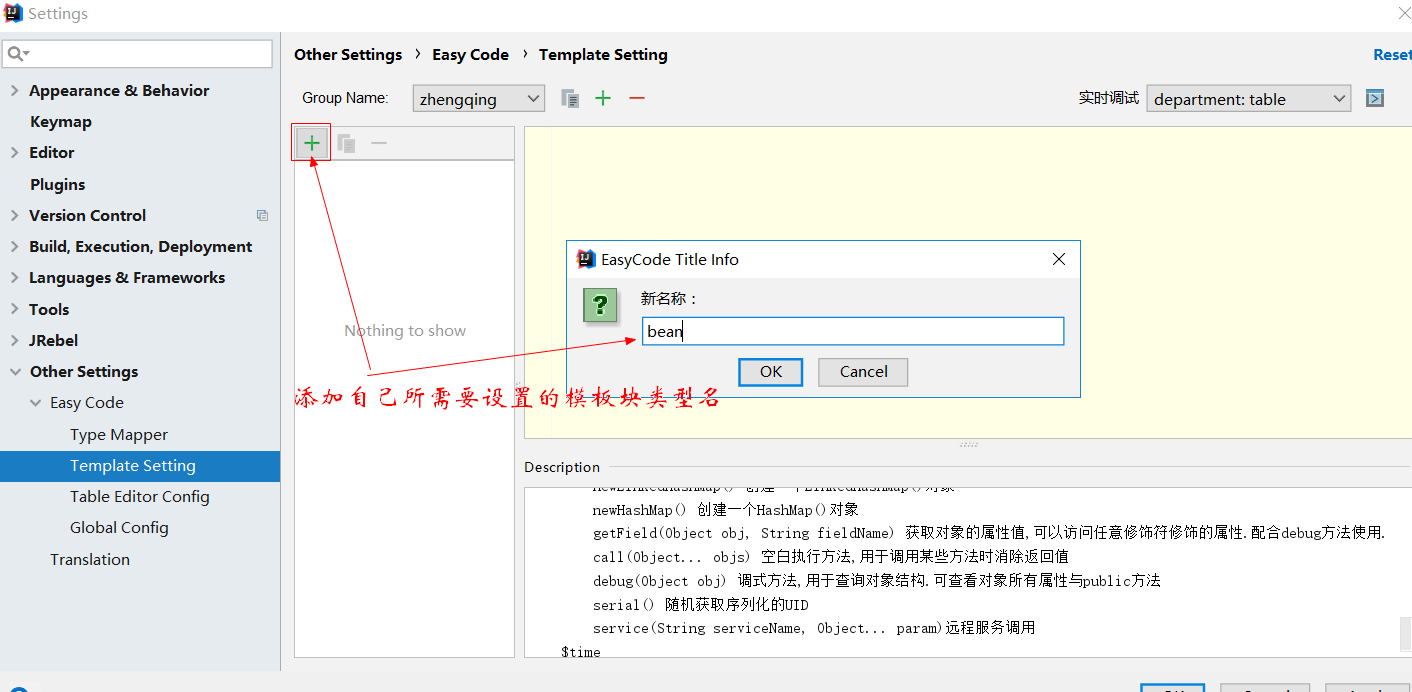


**3.配置代码生成器**

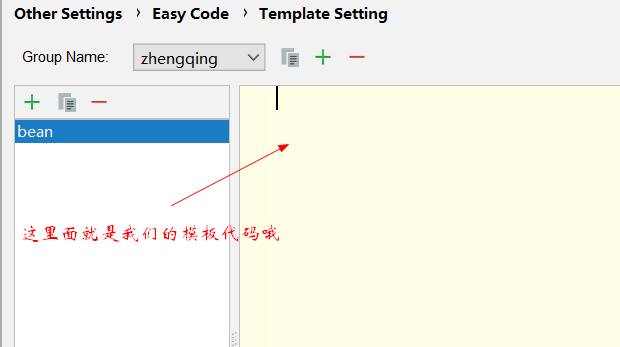
ctrl+alt+s --> 设置模板 , 先自定义一个分组存放自己需要设置的模板



添加所需要设置的模板块类型名



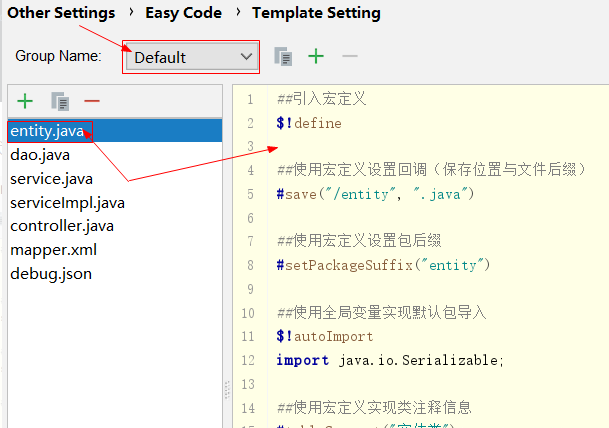
右边部分就是我们需要设置的模板代码内容



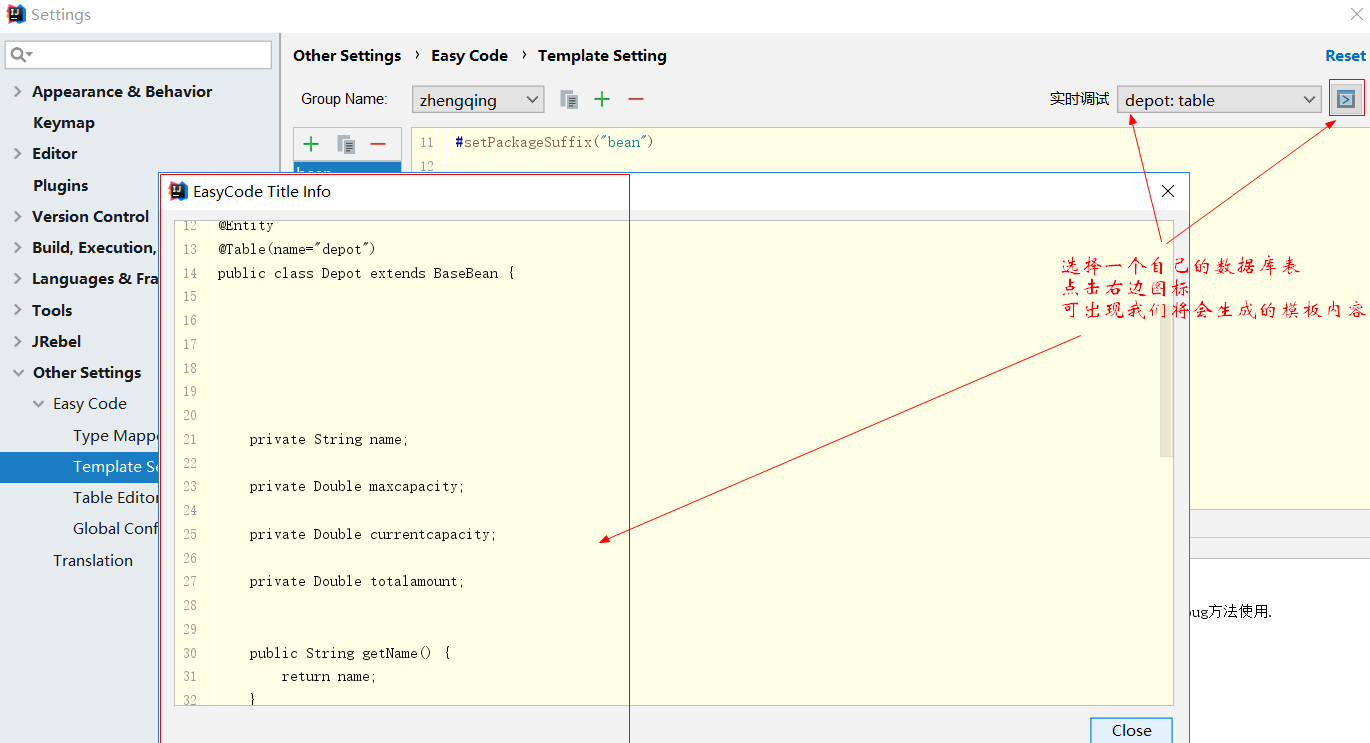
下面是我的一个实体类的模板代码，你们可以选择参考哦

|  |
| --- |
| ##引入宏定义  $!define  ## 首字母小写表名  #set($lowerTableName= $tool.firstLowerCase($!{tableInfo.name}))    ##使用宏定义设置回调（保存位置与文件后缀）  #save("/main/java/com/zhengqing/aisell/bean", ".java")    ##使用宏定义设置包后缀  #setPackageSuffix("bean")    ##使用全局变量实现默认包导入  $!autoImport  import javax.persistence.Entity;  import javax.persistence.Table;    ##使用宏定义实现类注释信息  #tableComment("实体类")  @Entity  @Table(name="$!{lowerTableName}")  public class $!{tableInfo.name} extends BaseBean {  ##实现列进行排除(去掉不需要的那一列 这里去除的是id列)==================  #set($temp = $tool.newHashSet("id"))  #foreach($item in $temp)  #set($newList = $tool.newArrayList())  #foreach($column in $tableInfo.fullColumn)  #if($column.name!=$item)  ##带有反回值的方法调用时使用$tool.call来消除返回值  $tool.call($newList.add($column))  #end  #end  ##重新保存  $tableInfo.setFullColumn($newList)  #end  ##========================================================================  #foreach($column in $tableInfo.fullColumn)  #if(${column.comment})//${column.comment}#end    private $!{tool.getClsNameByFullName($column.type)} $!{column.name};  #end    #foreach($column in $tableInfo.fullColumn)  ##使用宏定义实现get,set方法  #getSetMethod($column)  #end  } |

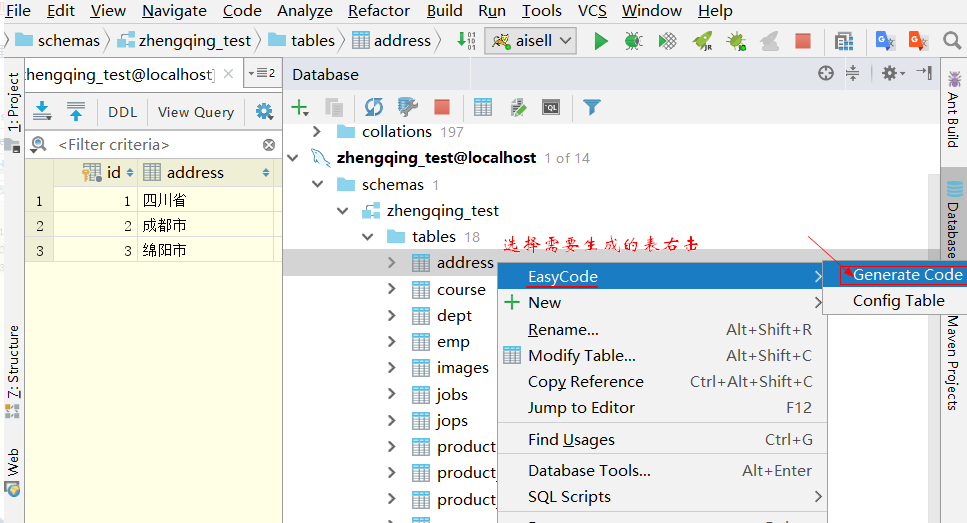
也可以去看看它自带的模板代码格式进行学习



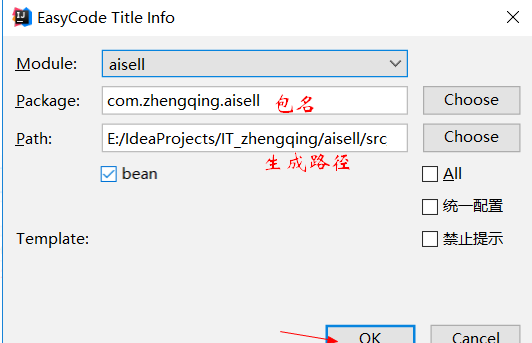
右边选择一个表即可查看将会生成的模板内容



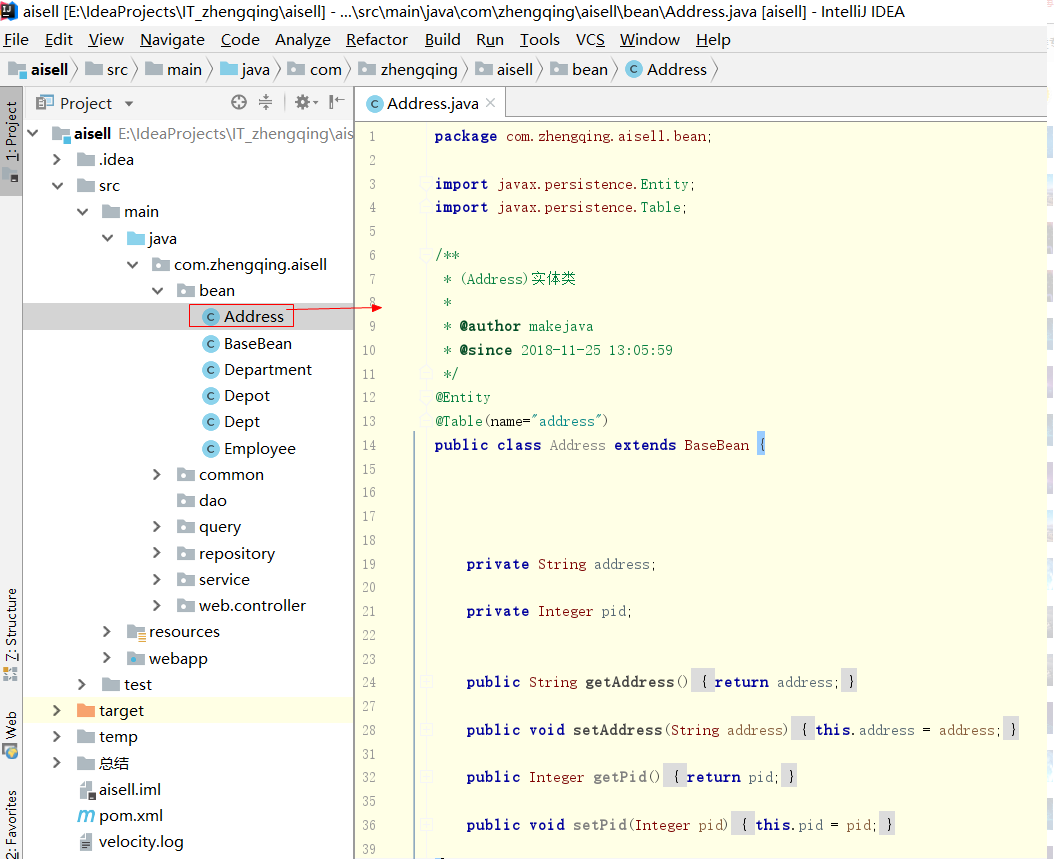
**4.进行代码自动生成**



这里我选择一个adress表生成对应的实体类

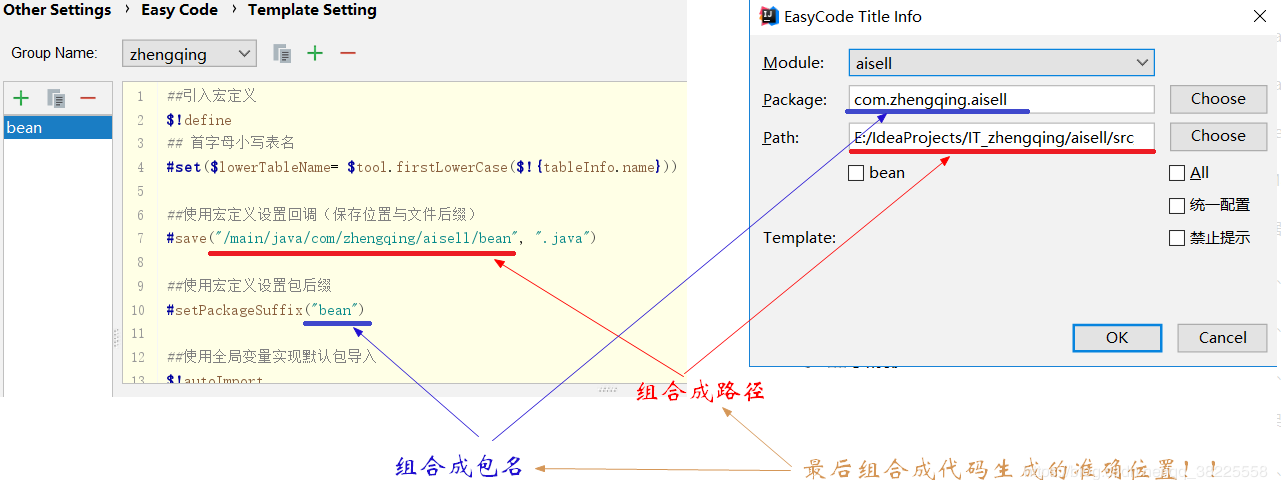


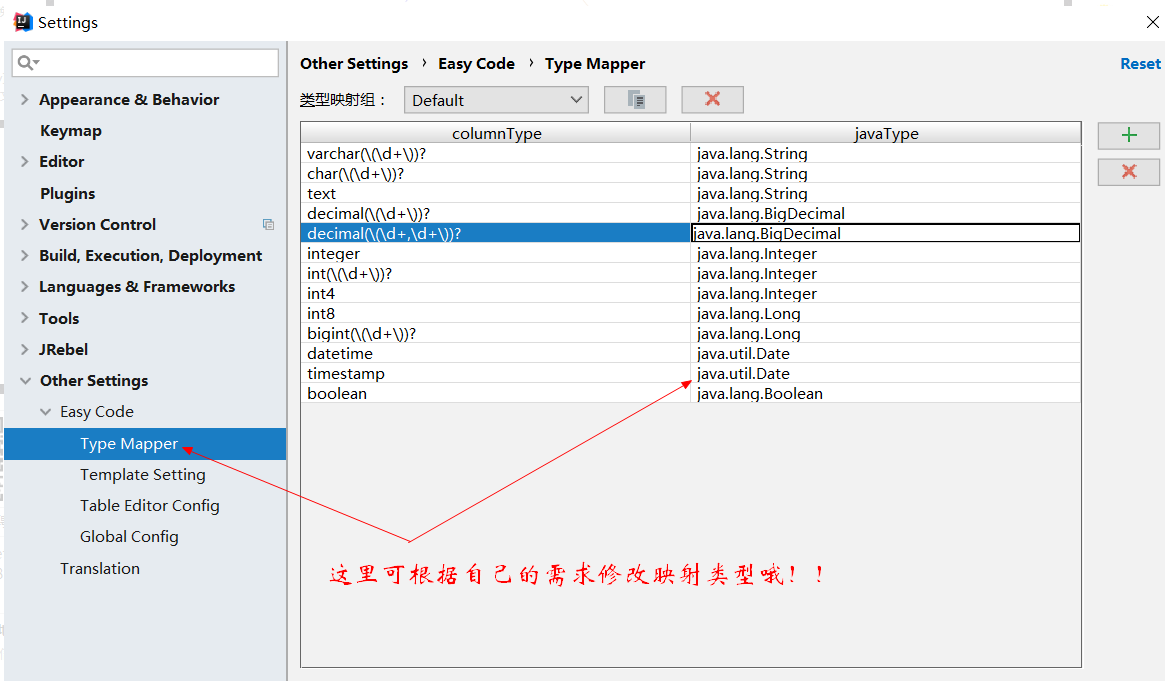
最后我们就可以看见在我们的项目下生成了一个Address实体类



更多模板配置方法如上，掌握操作方法即可！！

温馨小提示：





更多的需要可参考

EasyCode文档说明：<https://gitee.com/makejava/EasyCode/wikis/pages>

编号：2019.12.20.2

### 标题：Dubbo导入造成无法启动Tomcat

**1.问题**

略

Failed to execute goal org.apache.tomcat.maven:tomcat7-maven-plugin:2.2:run (default-cli) on project owwang-manager-service: Could not start Tomcat

**原因一**

Maven项目导入Dubbo没有排除

dubbox版本2.8.4

zookeeper版本3.4.13

jdk 1.8 tomcat7

开始以为是jdk或者tomcat版本的问题,试过几个版本没有解决

终于找到是由于dubbox依赖的servlet-api是spring不支持的

为dubbo的依赖排除servlet-apijar包

排除javax.servlet

|  |
| --- |
| <dependency>  <groupId>com.alibaba</groupId>  <artifactId>dubbo</artifactId>  <version>2.8.4</version>  <exclusions>  <exclusion>  <groupId>javax.servlet</groupId>  <artifactId>javax.servlet-api</artifactId>  </exclusion>  </exclusions>  </dependency> |

**原因二**

项目已经在运行，需要关闭项目，找不到就去关闭进程



**原因三**

Target里面有东西用不了，把整个target删了

## 2019.12.28

编号：2019.12.28.1

### 标题：安装activeMQ后无法启动

**1.问题**

安装activeMQ后无法启动

**2.原因**

这是一个坑，很大的坑。搞了半天才知道，原来ActiveMQ与java的JDK是有版本对应匹配的

**3.解决方法**

匹配信息如下：

MQ版本号 Build-Jdk 依赖JDK

apache-activemq-5.0.0 1.5.0\_12 1.5+

apache-activemq-5.1.0 1.5.0\_12 1.5+

apache-activemq-5.2.0 1.5.0\_15 1.5+

apache-activemq-5.3.0 1.5.0\_17 1.5+

apache-activemq-5.4.0 1.5.0\_19 1.5+

apache-activemq-5.5.0 1.6.0\_23 1.6+

apache-activemq-5.6.0 1.6.0\_26 1.6+

apache-activemq-5.7.0 1.6.0\_33 1.6+

apache-activemq-5.8.0 1.6.0\_37 1.6+

apache-activemq-5.9.0 1.6.0\_51 1.6+

apache-activemq-5.10.0 1.7.0\_12-ea 1.7+

apache-activemq-5.11.0 1.7.0\_60 1.7+

apache-activemq-5.12.0 1.7.0\_80 1.7+

apache-activemq-5.13.0 1.7.0\_80 1.7+

apache-activemq-5.14.0 1.7.0\_80 1.7+

apache-activemq-5.15.0 1.8.0\_112 1.8+

编号：2019.12.28.2

### 标题：Linux系统查看所有端口占用情况

**1.问题**

查看端口占用情况方法

**2.原因**

略

**3.解决方法**

netstat -ntlp //查看当前所有tcp端口



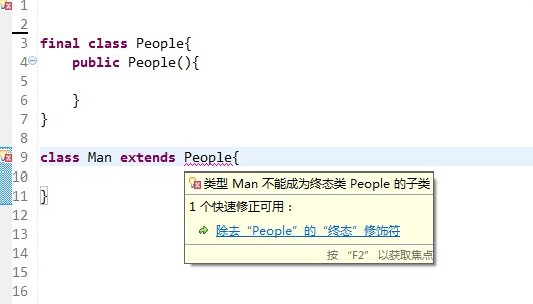
## 2019.12.29

编号：2019.12.12.29

### 标题：final修饰符作用

**1、修饰类**

　  当用final修饰一个类时，表明这个类不能被继承。也就是说，如果一个类你永远不会让他被继承，就可以用final进行修饰。final类中的成员变量可以根据需要设为final，但是要注意final类中的所有成员方法都会被隐式地指定为final方法。



在使用final修饰类的时候，要注意谨慎选择，除非这个类真的在以后不会用来继承或者出于安全的考虑，尽量不要将类设计为final类。

**2、修饰方法**

     下面这段话摘自《Java编程思想》第四版第143页：

　 “使用final方法的原因有两个。第一个原因是把方法锁定，以防任何继承类修改它的含义；第二个原因是效率。在早期的Java实现版本中，会将final方法转为内嵌调用。但是如果方法过于庞大，可能看不到内嵌调用带来的任何性能提升。在最近的Java版本中，不需要使用final方法进行这些优化了。“

　　因此，如果只有在想明确禁止 该方法在子类中被覆盖的情况下才将方法设置为final的。即父类的final方法是不能被子类所覆盖的，也就是说子类是不能够存在和父类一模一样的方法的。

     final修饰的方法表示此方法已经是“最后的、最终的”含义，亦即此方法不能被重写（可以重载多个final修饰的方法）。此处需要注意的一点是：因为重写的前提是子类可以从父类中继承此方法，如果父类中final修饰的方法同时访问控制权限为private，将会导致子类中不能直接继承到此方法，因此，此时可以在子类中定义相同的方法名和参数，此时不再产生重写与final的矛盾，而是在子类中重新定义了新的方法。（注：类的private方法会隐式地被指定为final方法。）

public class B extends A {

public static void main(String[] args) {

}

public void getName() {

}

}

class A {

/\*\*

\* 因为private修饰，子类中不能继承到此方法，因此，子类中的getName方法是重新定义的、

\* 属于子类本身的方法，编译正常

\*/

private final void getName() {

}

/\* 因为pblic修饰，子类可以继承到此方法，导致重写了父类的final方法，编译出错

public final void getName() {

}

\*/

}

[复制代码](javascript:void(0);)

**3、修饰变量**  
      修饰变量是final用得最多的地方，也是本文接下来要重点阐述的内容。

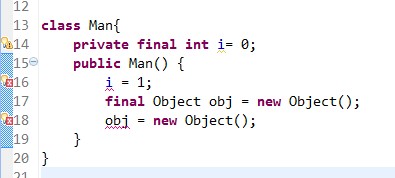
      final**成员变量表示常量，只能被赋值一次，赋值后值不再改变。**

　　当final修饰一个基本数据类型时，表示该基本数据类型的值一旦在初始化后便不能发生变化；如果final修饰一个引用类型时，则在对其初始化之后便不能再让其指向其他对象了，但该引用所指向的对象的内容是可以发生变化的。本质上是一回事，因为引用的值是一个地址，final要求值，即地址的值不发生变化。

**final修饰一个成员变量（属性），必须要显示初始化。这里有两种初始化方式**，一种是在变量声明的时候初始化；第二种方法是在声明变量的时候不赋初值，但是要在这个变量所在的类的所有的构造函数中对这个变量赋初值。

　　当函数的参数类型声明为final时，说明该参数是只读型的。即你可以读取使用该参数，但是无法改变该参数的值。

      举个例子：



     上面的一段代码中，对变量i和obj的重新赋值都报错了。

## 2019.12.30

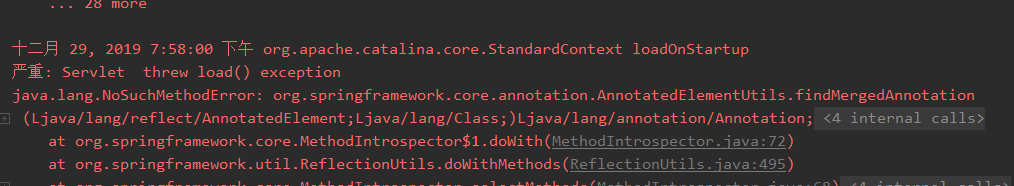
编号：2019.12.30.1

### 标题：导入activeMQ jar包后工程报错

**1.问题**

报错信息：

java.lang.NoSuchMethodError: org.springframework.core.annotation.AnnotatedElementUtils.findMergedAnnotation(Ljava/lang/reflect/AnnotatedElement;Ljava/lang/Class;)Ljava/lang/annotation/Annotation;



**2.原因**

由于自带的有日志jar文件，造成maven导入activemq-all的时候，会出现jar包冲突

slf4j-api.jar

slf4j-log4j12.jar

log4j-1.2.17.jar

|  |
| --- |
| <!-- activemq -->  <dependency>  <groupId>org.apache.activemq</groupId>  <artifactId>activemq-all</artifactId>  <version>${activemq.version}</version>  </dependency> |

**3.解决方法**

剔除其中的jar包文件没能实现，通过一个一个引入jar包实现

|  |
| --- |
| <dependency>     <groupId>org.apache.activemq</groupId>     <artifactId>activemq-broker</artifactId>     <version>5.9.0</version> </dependency>  <dependency>     <groupId>org.apache.activemq</groupId>     <artifactId>activemq-client</artifactId>     <version>5.9.0</version> </dependency>  <dependency>     <groupId>org.apache.activemq</groupId>     <artifactId>activemq-spring</artifactId>     <version>5.9.0</version> </dependency>  <dependency>     <groupId>org.apache.activemq</groupId>     <artifactId>activemq-kahadb-store</artifactId>     <version>5.9.0</version> </dependency> |

## 2019.12.29

编号：2020.XX.XX.X

### 标题：XXXXX

**1.问题**

xxxxx

**2.原因**

xxxxx

**3.解决方法**

（1）xxxxx

（2）xxxxx

（3）xxxxx

**4.总结**

略

编号：2020.XX.XX.X

### 标题：XXXXX

**1.问题**

xxxxx

**2.原因**

xxxxx

**3.解决方法**

（1）xxxxx

（2）xxxxx

（3）xxxxx

**4.总结**

略