# 《踩坑日记》

## 2019.12.15

编号：2019.12.15.1

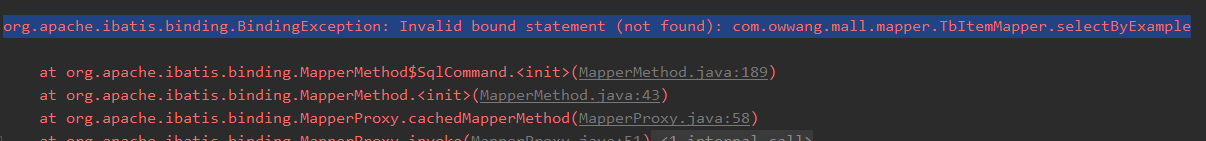
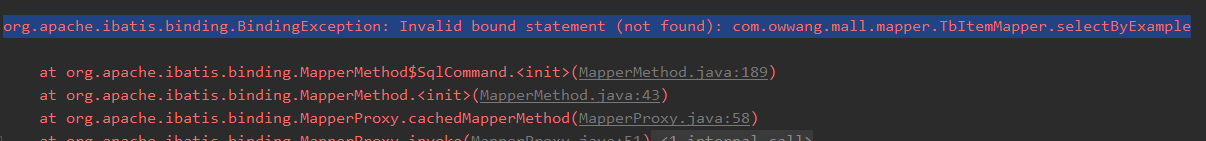
### 标题：Maven工程MyEclipse转移到IDEA后报错

**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

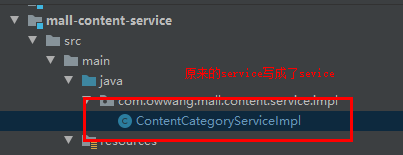
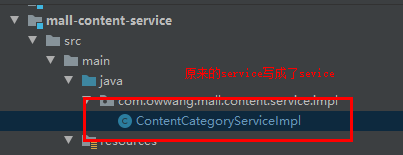
content-service 服务注入失败

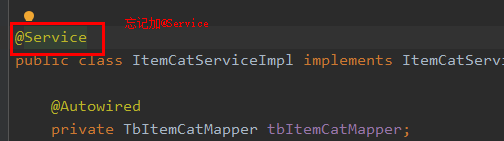
**2.原因**

（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**

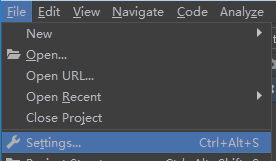
编写dao中的sql时,xml文件中背景一大片黄色，看着不舒服

**2.原因**

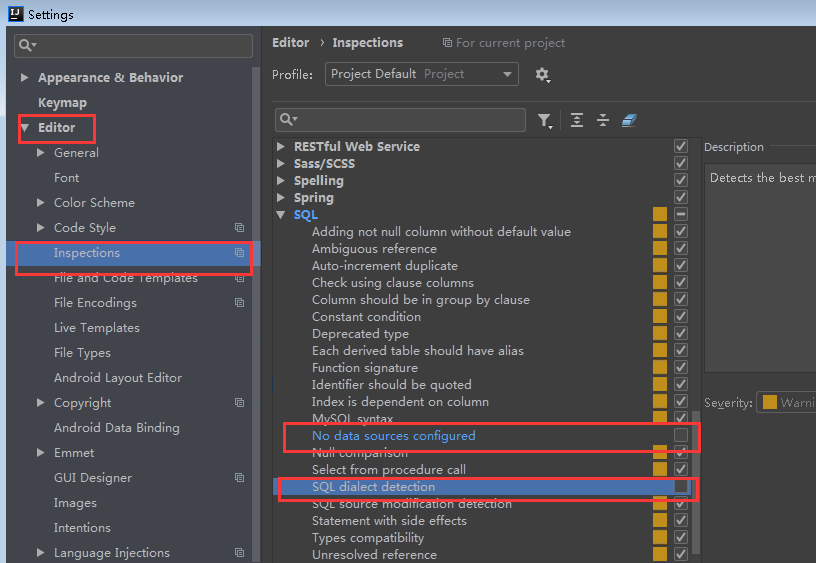
改设置即可

**3.解决方法**

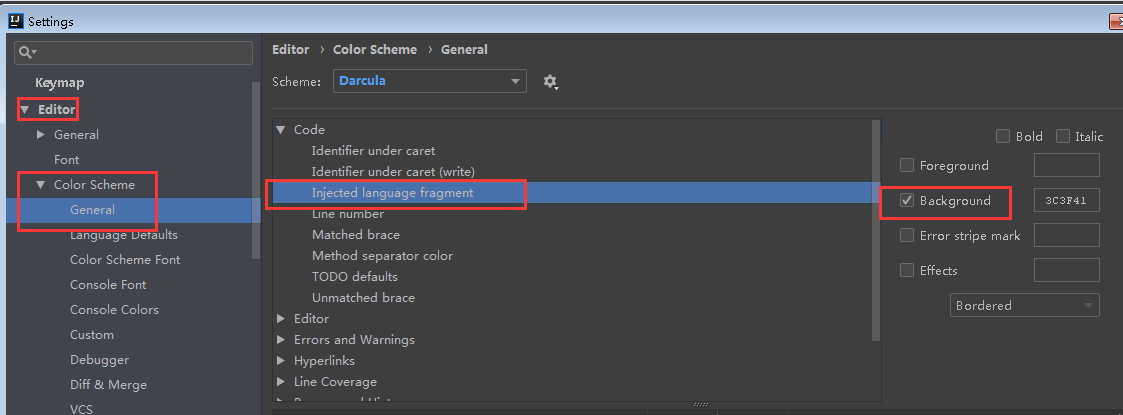
（1） File -> Settings...



（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.XX.XX

编号：2020.XX.XX.X

### 标题：XXXXX

**1.问题**

略

**2.原因**

略

**3.解决方法**

（1）略

（2）略

（3）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