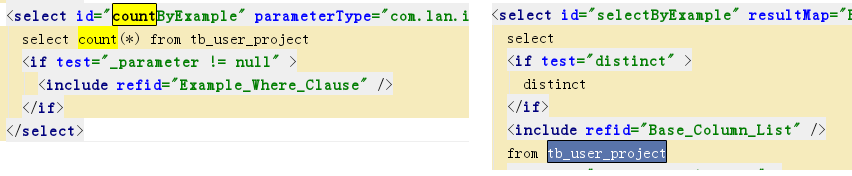
# PageHelper

就是写一个查询的sql语句，封装进去后就自动调用，底层是2条sql语句，一句你要查询的列，一句统计总行数，这是逆向工程的东西



## 分页插件

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。

https://github.com/pagehelper/Mybatis-PageHelper/blob/master/README.md

## 使用方法

### Pom.xml

把PageHelper依赖的jar包添加到工程中。

<**dependency**>  
 <**groupId**>com.github.pagehelper</**groupId**>  
 <**artifactId**>pagehelper</**artifactId**>  
 <**version**>${pagehelper.version}</**version**>  
</**dependency**>

### SqlMapConfig

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>

### java

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());

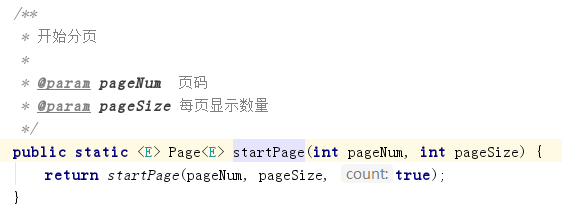
## 分页测试

|  |
| --- |
| @Test  **public** **void** testPageHelper() **throws** Exception {  //初始化spring容器  ApplicationContext applicationContext = **new** ClassPathXmlApplicationContext("classpath:spring/applicationContext-\*.xml");  //获得Mapper的代理对象  TbItemMapper itemMapper = applicationContext.getBean(TbItemMapper.**class**);  //设置分页信息  PageHelper.*startPage*(1, 30);  //执行查询  TbItemExample example = **new** TbItemExample();  List<TbItem> list = itemMapper.selectByExample(example);  //取分页信息  PageInfo<TbItem> pageInfo = **new** PageInfo<>(list);  System.***out***.println(pageInfo.getTotal());  System.***out***.println(pageInfo.getPages());  System.***out***.println(pageInfo.getPageNum());  System.***out***.println(pageInfo.getPageSize());  } |

# 常用类

## PageHelper

### 方法startPage



## PageInfo

### 构造方法

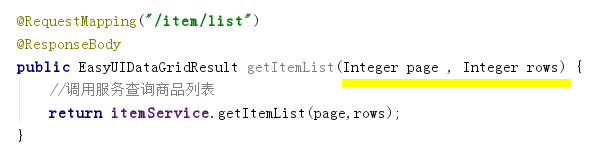


# Easyui实例

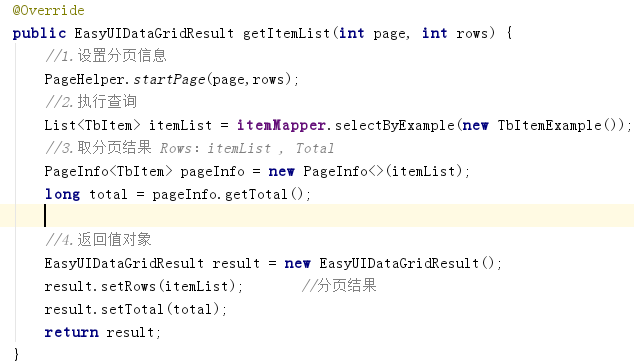
## 返回结果pojo

*/\*\*  
 \* 用于接受easyUI 分页表格数据，也可以用map代替  
 \*/***public class** EasyUIDataGridResult **implements** Serializable{  
 **private** Long **total**;  
  
 **private** List<?> **rows**;  
  
 **public** Long getTotal() {  
 **return total**;  
 }  
  
 **public void** setTotal(Long total) {  
 **this**.**total** = total;  
 }  
  
 **public** List<?> getRows() {  
 **return rows**;  
 }  
  
 **public void** setRows(List<?> rows) {  
 **this**.**rows** = rows;  
 }  
}

## Controller接受参数



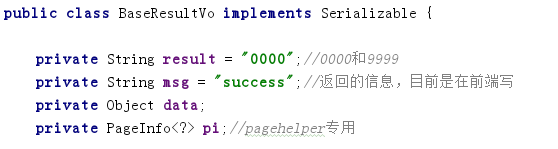
## Service



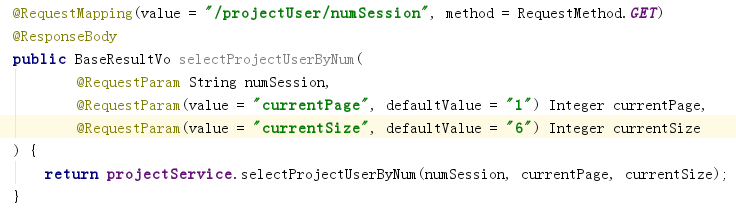
**public** EasyUIDataGridResult getItemList(**int** page, **int** rows) {  
 *//1.设置分页信息* PageHelper.*startPage*(page,rows);  
 *//2.执行查询* List<TbItem> itemList = **itemMapper**.selectByExample(**new** TbItemExample());  
 *//3.取分页结果 Rows：itemList , Total* PageInfo<TbItem> pageInfo = **new** PageInfo<>(itemList);  
 **long** total = pageInfo.getTotal();  
  
 *//4.返回值对象* EasyUIDataGridResult result = **new** EasyUIDataGridResult();  
 result.setRows(itemList); *//分页结果* result.setTotal(total);  
 **return** result;  
}

# 分页和数据不是一个pojo（复杂点）

## 返回结果vo



## Controller接受参数



## Service

