Mybatis-spring-paging分页组件使用指南

|  |  |  |  |
| --- | --- | --- | --- |
| 修改原因 | 修改日期 | 修改人 | 审核人 |
| 新建 | 2013-4-3 | 朱赋 |  |
| 应宙爷要求修改在common项目中的包路径 | 2013-4-3 | 朱赋 |  |
| 修改依赖的必要包  修改了配置文件等 | 2013-6-19 | 朱赋 |  |

目录

[Mybatis-oracle分页组件使用指南 1](#_Toc359405106)

[1. 目的 4](#_Toc359405107)

[2. 组件的作用 4](#_Toc359405108)

[3. 使用组件的必要条件 4](#_Toc359405109)

[4. 使用步骤 4](#_Toc359405110)

[4.1. 添加依赖 4](#_Toc359405111)

[4.2. 关键配置 5](#_Toc359405112)

[4.2.1. Spring配置文件中 5](#_Toc359405113)

[4.2.2. Mybatis配置文件 6](#_Toc359405114)

[4.2.3. 代码调用方式 6](#_Toc359405115)

[4.2.4. 运行输出结果 8](#_Toc359405116)

[5. 遗留问题 8](#_Toc359405117)

[6. FAQ 8](#_Toc359405118)

# 目的

本文档用于指导使用pay项目中的分页组件。

# 组件的作用

本组件用来自动添加分页查询，并且默认查询总数。

# 使用组件的必要条件

必须基于mybatis3.2（3.2和3.1差别很大）。

由于使用了com.cyou.fz.common.spring.SpringContextAware，故要加入对spring-context的依赖。

# 使用步骤

## 添加依赖

|  |
| --- |
| <groupId>com.cyou.fz.common</groupId>  <artifactId>mybatis-spring-paging</artifactId>  <version>0.0.1</version> |

注:Pay项目组可以忽略此步骤，此组件源码已经单独复制在common项目中。

## 关键配置

### Spring配置文件中

需要配置一个sqlsessionFactoryBuilder用于sqlSessionFactoryBean。需要配置mybatis扫描类MapperScannerConfigurer的annotationClass为MyBatisRepository。

|  |
| --- |
| <bean id=*"sqlSessionFactoryBuilder"* class=*"com.cyou.fz.common.mybatis.paging.pagination.proxy.MySqlSessionFactoryBuilder"*></bean>    <!-- MyBatis配置 -->  <bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <property name=*"dataSource"* ref=*"dataSource"* />  <!-- 自动扫描entity目录, 省掉Configuration.xml里的手工配置 -->  <property name=*"typeAliasesPackage"* value=*"com.cyou.fz.common.paging.po"* />  <property name=*"mapperLocations"* value=*"classpath:mybatis/\*.xml"* />  <property name=*"configLocation"* value=*"classpath:mybatis.xml"* />  <property name=*"sqlSessionFactoryBuilder"* ref=*"sqlSessionFactoryBuilder"*/>  </bean>  <!-- 扫描basePackage下所有以@MyBatisRepository标识的 接口 -->  <bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <property name=*"basePackage"*  value=*" com.cyou.test.dao,com.cyou.fz.common.mybatis.paging.pagination.dao"* />  <property name=*"annotationClass"* value=*"com.cyou.fz.common.mybatis.MyBatisRepository"* />  <property name=*"annotationClass"* value=*"com.cyou.fz.common.mybatis.MyBatisRepository"*/>  </bean> |

在basePackage中的红色部分是必填项目，用户只需在前后加入自己的包路径（逗号分隔）即可。

### Mybatis配置文件

需要配置一个拦截器，以及用来添加分页语句的方言和需要拦截的dao方法表达式。

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  <plugins>  <plugin  interceptor=*"com.cyou.fz.common.mybatis.paging.pagination.interceptor.PaginationInterceptor"*>  <property name=*"dialectClass"*  value=*"com.cyou.fz.common.mybatis.paging.pagination.dialect.db.OracleDialect"* />  <property name=*"sqlPattern"* value=*".\*****findAll****.\*"* />  </plugin>  </plugins>  </configuration> |

### 代码调用方式

略过建立PO对象。

首先建立一个DAO并且加上注解，入参必须是map，返回值是分页对象。

|  |
| --- |
| @MyBatisRepository  **public** **interface** PayUserDAO {  PaginationSupport<PayUserPO> findAll(Map<String, Object> map);  } |

建立一个mybatis的mapper。这里没有特别需要注意的，和正常的写法一样

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  <mapper namespace=*"com.cyou.fz.common.paging.dao.PayUserDAO"*>  <select id=*"findAll"* parameterType=*"map"* resultType=*"PayUserPO"*>  select  id as userId,  account, name,  pass\_word as passWord,  bp\_id as bpId,  status,  description  from pay\_user  where 1=1  <if test=*"orderFlag=='useridasc'"*>order by userId asc</if>  </select>  </mapper> |

创建Controller。其中必须将paginationsupport对象设置到作为查询参数的map中，并且Key必须是page。

|  |
| --- |
| @Controller  **public** **class** HomeController {  **private** **static** **final** Logger *logger* = LoggerFactory  .*getLogger*(HomeController.**class**);  @Autowired  **private** PayUserDao payUserDao;  @RequestMapping(value = "/test")  **public** @ResponseBody  PaginationSupport<PayUser> findAll(**int** pageNo, **int** pageSize) {  Map<String, Object> paramMap = **new** HashMap<String, Object>();  PaginationSupport<PayUser> payUserList = **new** PaginationSupport<PayUser>();  payUserList.setPageNo(pageNo);  payUserList.setPageSize(pageSize);  paramMap.put("page", payUserList);  payUserList = payUserDao.findAll(paramMap);  **return** payUserList;  }  } |

### 运行输出结果

|  |
| --- |
|  |

# 遗留问题

由于使用了com.cyou.fz.common.spring.SpringContextAware，将可能导致部分分页查询的单元测试无法进行，必须启动web容器测试。后续改进中。。。。。。

# FAQ