**My Walker平台开发指南**

版本：V1.0

## 运行准备

### 获得基础依赖包

在“walkerweb”程序中已经加入了这些包，第三章《开发步骤》中可以参考。

### 获得一份初始化SQL文件

系统可以在没有任何数据的情况下启动，之后可以通过超级管理员登陆，导入数据。

用户名：supervisor，密码：123456

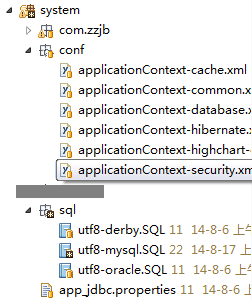
进入“部署🡪数据库管理🡪数据导入”功能中，把此SQL文件导出。

这个通常会在开发过程中，通过“备份”功能导出保留一份作为后面初始化使用。

### 导入SQL文件

## 业务包结构描述

以下以“系统管理”子系统为例

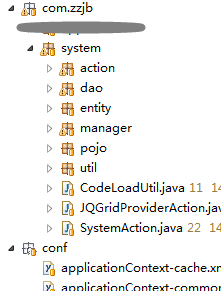


通常每个子系统新建一个“Source Folder”，其结构如下：

### 业务类的路径

路径命名规则为：**com.carmelasoft.子系统**。

子包命名规则如下：



### Conf文件夹

Conf文件夹下面全部都是spring配置文件，业务中需要配置的bean定义信息，因为系统中大部分使用了注解，因此这里的配置文件会很少或者没有，但仍可能会用到。

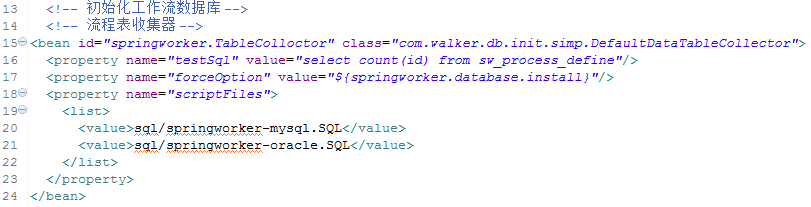
配置文件命名为：applicationContext-子系统-模块-名字.xml

### Sql文件夹

存放子系统用到的初始化表结构的SQL文件，如果需要支持不同数据库，就分别存放，文件命名中需要包含mysql、oracle、sqlserver关键字，如图：



同时，需要在业务配置文件中，加入脚本初始化加载器，以便系统启动时会自动初始化数据库表结构，配置示例：



当然，如果你不想让系统初始化自动创建表结构，就不需要此配置。

### Properties配置文件

在“Source Folder”的根目录下，可以放置子系统需要的java配置文件，目前还未实现在jar包中加载文件方式，暂时可以把所有属性文件放在工程的classes下面。

## 开发步骤

### 加入系统核心包

当前系统使用的基础包有两类，一种是基础类库，一种是业务基础模块包。

|  |  |  |
| --- | --- | --- |
| 序号 | 名字 | 描述 |
| 1 |  | 框架基础设施包  里面是所有基础代码和可重用类库，例如：数据库跨平台、抽象Action、解析配置文件的解析器等 |
| 2 |  | 流程引擎包  工作流引擎的定义与实现，不包括界面和设计器 |
| 3 |  | 框架提供的消息提醒包  内置实现了邮件消息方式，后续可以实现其他方式，如：短信等 |
| 4 |  | 框架的安全包  目前考虑的只有授权、注册等，需要更新，暂时无法使用 |
| 5 |  | 系统权限包  实现了组织机构、人员、角色、功能权限、参数设置等功能 |
| 6 |  | 系统流程设计包  包括：流程设计器web版实现、流程管理。  依赖于walker-flow.jar |

注：这些核心包后续会不断更新。

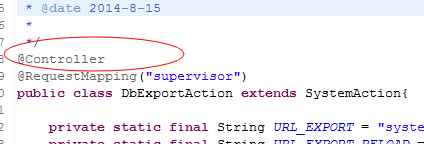
### 创建业务子系统包结构

参考第一章《运行准备》

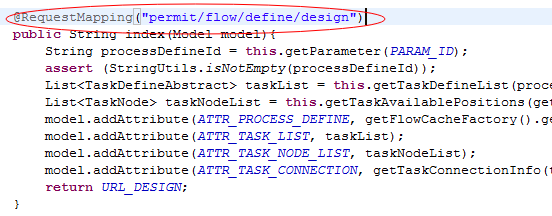
### 编写Action类

1）在action包中创建新的Action对象，记住必须继承“SystemAction”，否则无法使用系统提供的基础功能（如：获得登录用户信息）。

2）添加注解，这样系统启动会扫描到该Action并注册到系统中，如图：

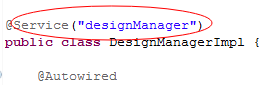
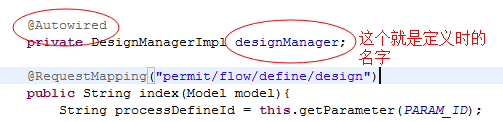


3）编写URL请求时对应的方法，方法名可以任意指定，但必须要通过注解配置请求的URL地址，如图：



上面配置URL就是用户要访问的地址，不要后缀名；

4）注入需要的Manager对象，定义的变量名字一定要和配置的名字一样，如图：



### 编写Dao类

集成系统提供的各种DAO默认实现，系统提供的默认DAO实现包括以下几个：

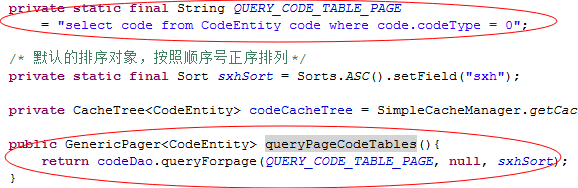
|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 描述 |
| 1 |  | 支持不同数据库类型的DAO实现  该DAO提供了很多基础方法，不过通常不会被直接使用，经常使用下面两个 |
| 2 |  | 支持POJO对象简单映射的DAO  可以使用普通的java对象来实现各种查询的自动映射，也不需要hibernate的实现。  **只能用户单表的环境中** |
| 3 |  | 支持直接使用SQL语句DAO实现  提供了各种通过sql语句查询的方法，包括：分页查询，以及批量sql更新等。 |

通常使用第三个，可以得到更多支持。

**Dao也使用注解来定义，不再用配置文件。**

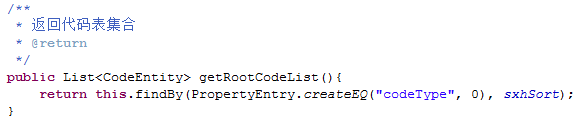
Dao常用方法列举：

1）通过Hibernate对象查询，提供HQL语句，分页返回数据集合：



在Dao中，可以通过this.xxx方式找到很多方法。

2）通过Hibernate的对象传递条件，返回集合对象：



示例中，返回了codeType = 0，并且按照sxhSort顺序定义的排序规则的代码集合。

3）通过HQL查询某一列的最大值：

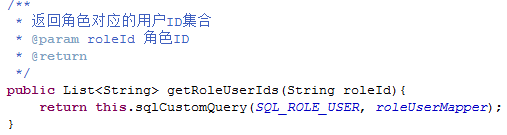


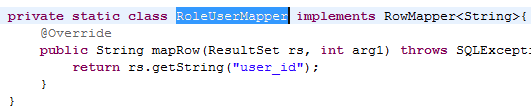
4）通过Hibernate分页返回数据，包括多个排序条件：



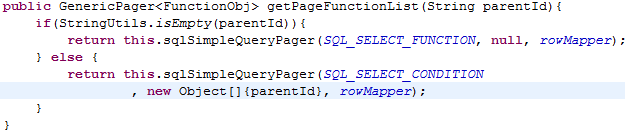


5）通过SQL语句返回对象集合，提供映射RowMapper实现：

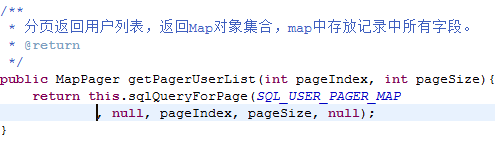




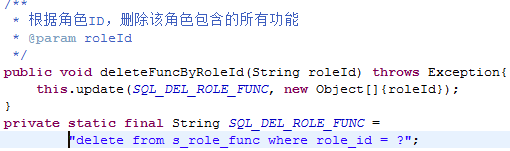
6）通过SQL语句分页返回对象集合，提供RowMapper实现：



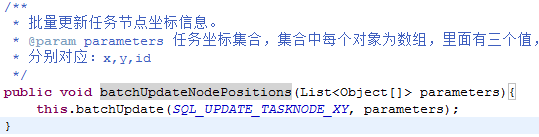
7）通过SQL语句分页返回Map集合：



8）执行普通SQL更新：



9）批量更新SQL，通过编译的SQL语句效率更高，提供参数集合：



### 编写manager类

1）manager类必须以\*\*\*ManagerImpl结尾，因为事务扫描的规则是这样定义的。

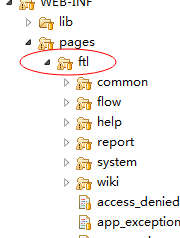
2）manager使用注解来配置，不再定义配置文件，通常定义名字为\*\*\*Manager，如图：



3）对于支持回滚的事务方法必须使用exec开头，只读事务使用query开头。

### 编写freemarker模板页面

1）在WEB-INF/pages/ftl中是所有页面模板文件，这是freemarker的根目录，如图：

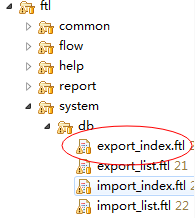
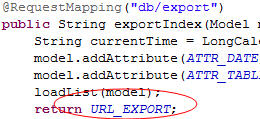


2）在下面根据子系统来新建文件夹，如：系统管理是system，流程是flow。至于子系统下面如何管理页面，可以根据需要来定，系统不做管理。

3）在Action中返回的页面URL必须与文件夹对应，但不需要写后缀名，例如：数据库导出功能的首页地址是：



在Action中直接返回该模板路径即可，系统会按路径查找页面，并自动加上.ftl后缀名，如图：



### 设置配置文件（可选）

## 注意事项

### 安全拦截

**对于新加入菜单的URL路径，是通过角色->权限来分配的，其他URL如果不想通过角色权限配置，必须使用permit/前缀，否则系统会报错：找不到页面或者权限未验证。**

## 问题与解答