# Springmybatis工程说明

## 文档说明

普通文档内容

|说明文字

在工程运行流程中F为Flow简写

## 目录结构

Springmybatisdemo |工程根目录

├─.settings |myeclipse配置相关工程配置文件

├─sqlscript |数据库脚本

├─src |源代码根目录

│ └─com

│ └─hs |合商拼音缩写

│ ├─bean |实体类

│ ├─controller |控制层

│ ├─dao |持久化接口

│ │ └─mapper |mybatis映射文件

│ ├─service |服务接口

│ │ └─impl |服务接口实现

│ └─util |工程公共工具包

├─techdocument |工程信息相关文档

└─WebRoot |web工程根目录

├─css |工程样式目录

│ └─global |全局样式

│ └─default |全局默认样式

├─img |工程图片

├─js |js脚本

├─META-INF |编译相关属性

└─WEB-INF |工程相关配置,视图,编译后类,工程用到的jar包

├─classes

│ └─com

│ └─hs

│ ├─bean

│ ├─controller

│ ├─dao

│ │ └─mapper

│ ├─service

│ │ └─impl

│ └─util

├─configs |配置文件

├─lib |用到的jar包

└─views |视图

└─user |用户视图

## 工程文件说明

Springmybatisdemo

│ .classpath |工程内源码目录设置与运行环境

│ .project |工程配置信息与myeclipse用到的插件

├─.settings

│ .jsdtscope

│ org.eclipse.jdt.core.prefs

│ org.eclipse.wst.common.component

│ org.eclipse.wst.common.project.facet.core.xml

│ org.eclipse.wst.jsdt.ui.superType.container

│ org.eclipse.wst.jsdt.ui.superType.name

├─sqlscript

│ init.sql |初始化数据库的表与序列脚本

├─src

│ └─com

│ └─hs

│ ├─bean

│ │ BaseUserT.java |用户实体类

│ ├─controller

│ │ UserController.java |用户控制器

│ ├─dao

│ │ │ BaseDao.java |通用持久化接口 未使用

│ │ │ BaseUserTDao.java |用户持久化接口

│ │ └─mapper

│ │ BaseUserTMapper.xml |用户持久化接口映射文件

│ ├─service

│ │ │ BaseService.java |通用服务接口

│ │ │ BaseUserTService.java |用户服务接口

│ │ └─impl

│ │ BaseUserTServiceImpl.java |用户服务接口实现类

│ └─util

│ BaseController.java |通用控制器

│ BaseLog.java |通用日志处理

│ BaseValidate.java |通用验证类

│ GlobalControllerPath.java |全局控制器路径名称

├─techdocument

│ eclipseplug.txt

│ env.txt

│ 工程说明.docx

└─WebRoot

│ index.jsp |进入工程默认主页

│

├─css

│ └─global

│ └─default

├─img

├─js

├─META-INF

│ MANIFEST.MF

└─WEB-INF

│ web.xml |工程架构配置文件

├─classes

│ └─com

│ └─hs

│ ├─bean

│ │ BaseUserT.class |源码编译后的类

│ ├─controller

│ │ UserController.class

│ ├─dao

│ │ │ BaseDao.class

│ │ │ BaseUserTDao.class

│ │ └─mapper

│ │ BaseUserTMapper.xml

│ ├─service

│ │ │ BaseService.class

│ │ │ BaseUserTService.class

│ │ └─impl

│ │ BaseUserTServiceImpl.class

│ └─util

│ BaseController.class

│ BaseLog.class

│ BaseValidate.class

│ GlobalControllerPath.class

│

├─configs

│ jdbc.properties |spring框架中数据库连接配置信息

│ log4j2.xml |日志配置文件

│ spring-config.xml |spring配置文件

│ spring-mvc.xml |spring mvc配置文件

│ sqlMapConfig.xml |mybatis配置文件

│

├─lib

│ aopalliance-1.0.jar |面向切面编程

│ aspectjweaver-1.7.4.jar |spring面向切向支持jar包

│ commons-dbcp2-2.0.1.jar |apache dbcp数据库连接池

│ commons-logging-1.2.jar |apache logging日志记录

│ commons-pool2-2.2.jar |apache连接池支持

│ jstl-1.2.jar |jsp通用标签库

│ jstl-impl-1.2.jar |jsp通用标签库实现

│ log4j-api-2.0.1.jar |log4j2日志api

│ log4j-core-2.0.1.jar |log4j2核心日志

│ log4j-over-slf4j-1.7.7.jar |slf4j接口的log4j调用支持

│ mybatis-3.2.7.jar |mybatis框架

│ mybatis-spring-1.2.2.jar |mybatis与spring集成支持

│ ognl-2.6.11.jar |ognl表达式

│ ojdbc6.jar |oracle数据库驱动

│ slf4j-api-1.7.7.jar |slf4j日志接口api

│ spring-aop-4.0.6.RELEASE.jar |spring切面编程

│ spring-beans-4.0.6.RELEASE.jar |spring bean管理

│ spring-binding-2.4.0.RELEASE.jar

│ spring-context-4.0.6.RELEASE.jar

│ spring-core-4.0.6.RELEASE.jar

│ spring-expression-4.0.6.RELEASE.jar

│ spring-jdbc-4.0.6.RELEASE.jar

│ spring-js-2.4.0.RELEASE.jar

│ spring-js-resources-2.4.0.RELEASE.jar

│ spring-tx-3.2.9.RELEASE.jar |spring事务支持

│ spring-web-4.0.2.RELEASE.jar

│ spring-webflow-2.4.0.RELEASE.jar |spring webflow支持

│ spring-webmvc-4.0.2.RELEASE.jar |spring webmvc支持

│ standard.jar |jstl支持

│

└─views

│ index.jsp |二级默认主页

│

└─user

login.jsp |用户登陆界面

userop.jsp |用户操作界面

usersList.jsp |用户列表界面

## 工程运行流程

### F1 工程部署后加载web.xml文件(从上到下看)

#### ->F1a1 http方法拦截过滤器用来启用方法支持

#### ->F1a2 http方法拦截映射

#### ->F1b1编码过虑器,将交换内容转为utf-8

#### ->F1b2 编码过滤器映射

#### ->F1c1 spring配置文件位置监听器

#### ->F1d1 spring web mvc配置文件servlet

#### ->F1c2 spring监听映射

#### ->F1d2 spring web mvc servlet映射

#### ->F1e1 应用内会话超时时间单位分钟

#### ->F1f1 log4j2初始化参数

#### ->F1g1 spring与log4j监听

### F2 用户列表与前置主页

#### ->F2a1工程加载默认主页标题为0

#### ->F2a2 通过let’s go链接到达userc控制器的toIndex

#### ->F2a3 到达二级默认主页标题为1

#### ->F2a4 用户列表链接到userc控制器中usersList

#### ->F2a5 userList调用服务类,服务类调用持久化接口,持久化接口调用映射文件,映射文件为mybatis通过spring连接oracle数据库得到用户列表信息.通过层层返回userList中并放入请求属性名称为dataList

#### ->F2a6跳转页面到视图目录中user目录的usersList.jsp页面,spring会自动添加.jsp所以控制器中只写usersList即可

#### ->F2a7 用户列表页标题为2usersList通过JSTL标签遍历请求属性中dataList用户对象列表

## 代码详解

### 实体类

**package** com.hs.bean; |包名

**import** java.io.Serializable; |导入序列化

**import** java.sql.Blob; |导入数据类型支持

**import** javax.persistence.Column; |导入持久化注解支持

@Entity |实体类注解

@Table(name="baseUserT") |表名注解

**public** **class** BaseUserT **implements** Serializable{ |类名 实现序列化

**private** **static** **final** **long** *serialVersionUID* = 6024801528883494913L;

**private** **int** buId; |唯一编号

………….省略

**private** String buLoginName; |登陆名称

………….省略

@Id |唯一编号注解 获取方法

**public** **int** getBuId() {

**return** buId;

}

**public** **void** setBuId(**int** buId) { |设置唯一编号

**this**.buId = buId;

}

………….省略

@Column(name="buLoginName") |列名注解

**public** String getBuLoginName() {

**return** buLoginName;

}

………….省略

**public** BaseUserT() { |空构造方法

**super**();

}

| 全部字段构造方法 一定要有 用来与mybatis对象封装作对应

**public** BaseUserT(**int** buId, **int** buSeq, String buLoginId, String buLoginName,

String buLoginMobile, String buLoginEmail, String buLoginPass,

Blob buAvata, String buAvataUrl, **int** buStatus, Date buRegTime,

Date buLastLoginTime, String buLastLoginIp, String buMemo) {

**super**();

**this**.buId = buId;

**this**.buSeq = buSeq;

**this**.buLoginId = buLoginId;

**this**.buLoginName = buLoginName;

**this**.buLoginMobile = buLoginMobile;

**this**.buLoginEmail = buLoginEmail;

**this**.buLoginPass = buLoginPass;

**this**.buAvata = buAvata;

**this**.buAvataUrl = buAvataUrl;

**this**.buStatus = buStatus;

**this**.buRegTime = buRegTime;

**this**.buLastLoginTime = buLastLoginTime;

**this**.buLastLoginIp = buLastLoginIp;

**this**.buMemo = buMemo;

}

}

### 控制器类

@Controller |控制器注解

@RequestMapping(value=GlobalControllerPath.*userC*) |控制器名称映射

**public** **class** UserController **extends** BaseController{

@Autowired |自动注入支持

BaseUserTService buts; |自动注入的对象是用户服务接口注意是接口

@RequestMapping(value="toEdit") |控制器二级映射名

**public** String toEdit(HttpServletRequest r){ |可以传入如实体类,请求,响应等

r.setAttribute("method", "edit");

|调用验证并使用服务类将结果封装进请求属性

**if**(*isNotNullInParameter*(r, "buId") && BaseValidate.*isNumber*(r.getParameter("buId")));

r.setAttribute("u", buts.queryById(Integer.*parseInt*(r.getParameter("buId"))));

**return** "user/userop"; |返回用户操作视图

}

### 服务接口

**public** **interface** BaseUserTService **extends** BaseService { |服务接口

**public** BaseUserT queryById(**int** buId); |服务方法

………….省略

}

### 服务接口实现

@Service |服务注解

@Transactional |事务注解

**public** **class** BaseUserTServiceImpl **implements** BaseUserTService{

@Autowired

**private** BaseUserTDao d;

@Override

**public** BaseUserT queryById(**int** buId) {

**return** d.queryById(buId);

}

………….省略

}

### 持久化接口

**public** **interface** BaseUserTDao { |持久化接口

**public** BaseUserT queryById(**int** buId); |持久化方法

………….省略

}

### 持久化映射

<?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.hs.dao.BaseUserTDao"*> |结果集声明

<resultMap type=*"BaseUserT"* id=*"baseUserTList"*>

<result property=*"buId"* column=*"buId"*/>

<result property=*"buSeq"* column=*"buSeq"*/>

<result property=*"buLoginId"* column=*"buLoginId"*/>

<result property=*"buLoginName"* column=*"buLoginName"*/>

<result property=*"buLoginMobile"* column=*"buLoginMobile"*/>

<result property=*"buLoginEmail"* column=*"buLoginEmail"*/>

<result property=*"buLoginPass"* column=*"buLoginPass"*/>

<result property=*"buAvataUrl"* column=*"buAvataUrl"*/>

<result property=*"buStatus"* column=*"buStatus"*/>

<result property=*"buRegTime"* column=*"buRegTime"*/>

<result property=*"buLastLoginTime"* column=*"buLastLoginTime"*/>

<result property=*"buLastLoginIp"* column=*"buLastLoginIp"*/>

<result property=*"buMemo"* column=*"buMemo"*/>

</resultMap>

|查询语句声明 对应dao接口

<select id=*"queryById"* parameterType=*"int"* resultType=*"BaseUserT"*>

select buId,buSeq,buLoginId,buLoginName,buLoginMobile,buLoginEmail,buLoginPass,buAvataUrl,buStatus,buRegTime,buLastLoginTime,buLastLoginIp,buMemo from baseUserT

where buId = #{buId}

</select>

<select id=*"queryByName"* parameterType=*"String"* resultMap=*"baseUserTList"*>

select u.buId as buId,u.buLoginName as buLoginName, u.buLoginPass as buLoginPass

from baseUserT u

where u.buLoginName like '%' || #{buLoginName} || '%'

</select>

<select id=*"login"* parameterType=*"java.util.HashMap"* resultType=*"BaseUserT"*>

select u.buId as buId,u.buLoginName as buLoginName, u.buLoginPass as buLoginPass

from baseUserT u

where u.buLoginName=#{buLoginName} and u.buLoginPass=#{buLoginPass}

</select>

<select id=*"queryAll"* resultMap=*"baseUserTList"*>

select buId,buSeq,buLoginId,buLoginName,buLoginMobile,buLoginEmail,buLoginPass,buAvataUrl,buStatus,buRegTime,buLastLoginTime,buLastLoginIp,buMemo

from baseUserT u

</select>

|添加数据配置 对应dao接口

<insert id=*"add"* parameterType=*"BaseUserT"*>

insert into baseUserT (buId,buSeq,buLoginId,buLoginName,buLoginMobile,buLoginEmail,buLoginPass,buStatus,buRegTime,buLastLoginTime,buLastLoginIp,buMemo)

values(seqForBaseUserT.nextval,#{buSeq},#{buLoginId},#{buLoginName},#{buLoginMobile},#{buLoginEmail},#{buLoginPass},#{buStatus},sysdate,sysdate,#{buLastLoginIp},#{buMemo})

</insert>

<update id=*"update"* parameterType=*"BaseUserT"*>

update baseUserT set buSeq=#{buSeq},buLoginId=#{buLoginId},buLoginName=#{buLoginName},buLoginMobile=#{buLoginMobile},buLoginEmail=#{buLoginEmail},buLoginPass=#{buLoginPass},buStatus=#{buStatus},buLastLoginIp=#{buLastLoginIp},buMemo=#{buMemo}

where buId = #{buId}

</update>

<delete id=*"delete"* parameterType=*"int"*>

delete from baseUserT where buId=#{buId}

</delete>

</mapper>

### 视图

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"utf-8"*%>

|标签库引入

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/functions"* prefix=*"fn"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/fmt"* prefix=*"fmt"*%>

|工程路径定义

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

request.setAttribute("basePath", basePath);

%>

|html版本声明

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<base href=*"*${basePath }*"*>

<title>2usersList</title>

<meta http-equiv=*"charset"* content=*"utf-8"*>

<meta http-equiv=*"pragma"* content=*"no-cache"*>

<meta http-equiv=*"cache-control"* content=*"no-cache"*>

<meta http-equiv=*"expires"* content=*"0"*>

<meta http-equiv=*"keywords"* content=*""*>

<meta http-equiv=*"description"* content=*""*>

<!--

<link rel="stylesheet" type="text/css" href="styles.css">

-->

</head>

<body>

<table>

<caption>用户列表</caption>

<tr><td>用户id</td><td>用户名</td><td>密码</td><td>邮箱</td><td>最后登录IP</td><td>用户说明</td><td>编辑</td>

</tr>

|c foreach 标签使用

<c:forEach items=*"*${dataList==null?'':dataList}*"* var=*"u"* varStatus=*"status"*>

<tr>

<td>${u.buId }</td>

<td>${u.buLoginName }</td>

<td>${u.buLoginPass }</td>

<td>${u.buLoginEmail }</td>

<td>${u.buLastLoginIp }</td>

<td>${u.buMemo }</td>

<td><a href=*"*${basePath }*userc/toEdit?buId=*${u.buId }*"*>编辑</a></td>

</tr>

</c:forEach>

</table>

</body>

</html>