EMVC web开发框架说明

https://60.173.194.42/svn/GovInfo/MLFrame

# 框架概述

## 功能目标

* + 安全策略（目录安全设定、用户身份认证和授权）
  + 系统级异常处理
  + Action基类（返回分页数据json，返回对象数据json，实现页面跳转，其它）
  + 基本功能：用户、角色、权限、用户登录界面、用户修改密码
  + 配置管理
  + 日志管理
  + 主界面，可配置的系统名称、功能菜单、版权、技术支持等
  + 工作区创建或打开tabpage的方法$.tabs()
  + 页面窗囗基本样式(使用EasyUI的组件。
  + 版本管理
  + 使用本框架，可以运行在平台、健康档案等系统的数据环境下，那么，要求其基础数据保持与已有数据的一致。

## 框架数据库

* Ehealth\_system\_configzt
* Ehealth\_role
* Ehealth\_resource
* ROLE\_MENU
* Ehealth\_role\_resource
* Ehealth\_user\_role
* T\_GOVINFO
* Hys\_user
* Hys\_user\_ztext



## 框架使用

* 确定新开发的项目名称和字母标识（如xxx人事管理系统，RSSYS）；
* 建立基于本框架的分支；
* 修改ActionConstants.app\_name=”RSSYS”;
* 新建项目数据库，把框架系统表建立，建立用户扩展表，（如果需要在已有系统基础上运行，那么建立机构表T\_GOVINFO，用户表HYS\_USER的“同义词”）；
* 修改配置文件后，可在服务器中运行；
* 此时，系统名称、技术支持、版权、电话信息都是NULL;
* 登陆系统，进行配置，维护相关数据；或者执行insertData.sql（需用新系统字母标识替换’FRAME’）建立配置数据；
* 重新登陆系统，系统名称、技术支持、版权、电话信息都是将出现。
* 用户维护功能的开发，根据新的用户扩展表的字段需要做开发。
* 菜单定义：WebContent\HYSAPP\MenuData.js文件中维护菜单项目，（请保留基本菜单项目：1 密码修改；2 用户管理 3 机构维护 4 角色资源定义 5 角色功能分配）；
* 更多开发，请参考以下内容；

## 基础

|  |  |
| --- | --- |
| jQuery1.11.1  jquery-easyui1.4  ztree.js | Struts 1.2.9  Spring 2.0.6  Hibernate 3.2.5 |
| 扩展：  extjs  bootstrap angular.js javascript面向对象 | Webservice  Xml |

# 组织

## 基础

* Oracle
* Ssh，security,log4j,
* Jquery 、jqueryui 、pqgrid、ztree、layout、bootstrap、extjs、
* Javascript面向对象编程
* 模块模型开发
* Js-mvc模式
* freemarker

## FreeMakerUtil

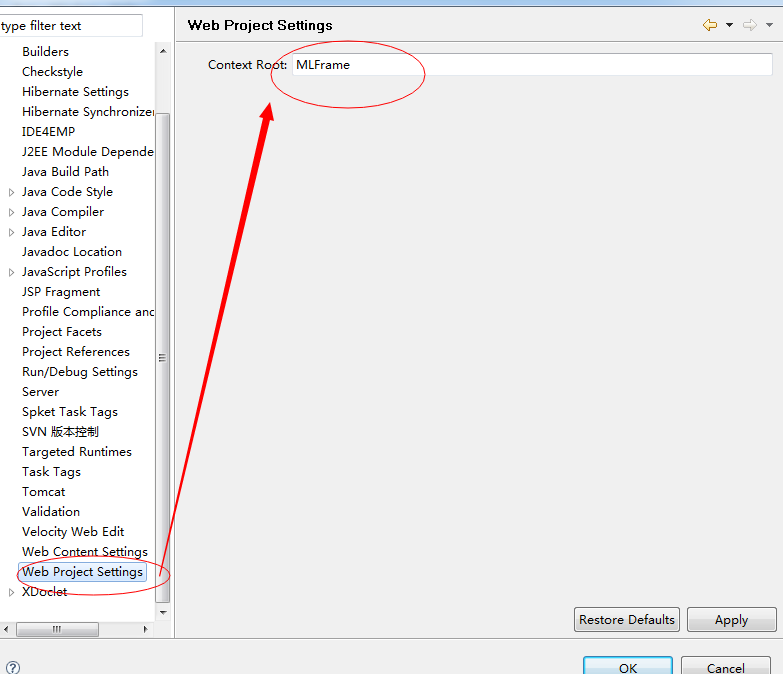
模板生成工具，为指定数据表生成前后台代码及xml配置文件；

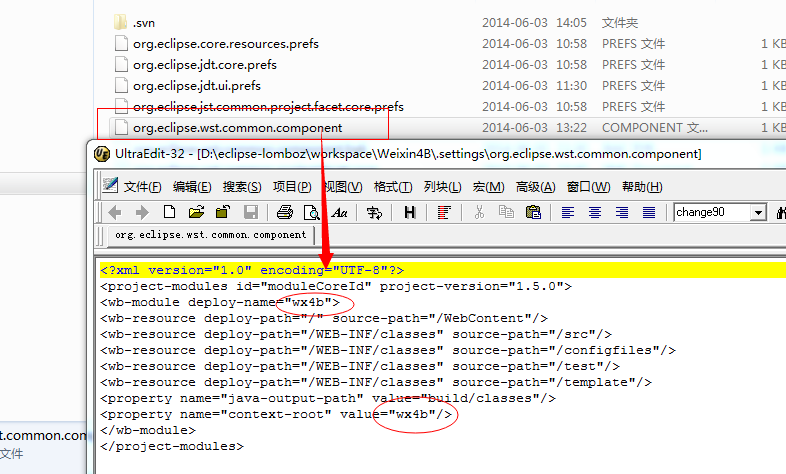
|  |  |
| --- | --- |
| 1. Manager.ftl 2. Model.ftl 3. Facade.ftl 4. FacadeImp.ftl 5. ListAction.ftl 6. DetailJSONAction.ftl 7. DetailAction.ftl 8. DeleteAction.ftl 9. AddeditAction.ftl 10. hbm.ftl 11. spring.ftl 12. struts.ftl 13. JSModel.ftl 14. JSView.ftl 15. JSControler.ftl 16. Html\_list.ftl 17. Html\_list\_js.ftl 18. \_list\_html.ftl 19. \_list\_js.ftl 20. \_detail\_jsp.ftl 21. \_detail\_js.ftl | Java  Hibernate配置  Manager/Façade 的Spring bean配置  .do struts配置  前端model  前端view  前端controler  前端页面html  前端页面js  列表页面，包含增删改查功能按钮（ext grid）  明细页面（ext） |

## 操作过程

### 开始

1. 确定应用标识（原始的为FRAME）；
2. 新建数据库用户
3. 在新的数据库用户下建立表、序列、执行写入数据的sql（根据应用标识修改）
4. 修改model.xml的数据库连接配置
5. ActionConstants.APP\_NAME设置应用标识；
6. 开发环境，修改web标识





1. 启动web服务
2. 启动成功，登录请使用用户admin,密码111111;
3. 进入系统后需要做以下功能的操作：

* 配置
* 角色资源定义
* 角色功能分配
* 机构维护

## 服务器中启动过程：

|  |  |  |
| --- | --- | --- |
|  |  | 结果显示 |
| SecureResourceFilterInvocationDefinitionSource  .afterPropertiesSet() | 初始化urlMatcher |  |
| ServletContextLoader. contextInitialized() | addr\_config.properties  读配置文件 |  |
| InitContextFilter.init() | 设置encoding，failure页面地址等 |  |
| EhealthActionServlet.init() | * 为struts整合设置springBeanContext   （ActionHelper.getBean()就是使用这个实现的）   * 读入资源🡪角色   servletContext.setAttribute(“urlAuthorities”)  [/zt/\*\*: xxx1角色，xxx2角色, ]   * 读入角色🡪资源   servletContext.setAttribute(“resourceRoleEdit”)   * 读取配置   写入ActionHelper.configSet  和APP\_NAME | 角色资源对照修改后，需要重新启动应用 |

## 访问一个url的过程

|  |  |  |
| --- | --- | --- |
|  |  |  |
| SecureResourceFilterInvocationDefinitionSource  .getAttribute() | 获取url对应的角色列表 |  |
| CustomVoter | 检查当前用户的角色是否在url的角色列表中?在，允许访问 |  |
| InitContextFilter.doFilter() |  |  |

## 目录定义和安全设置

Struts action 目录的定义

Web的目录

安全的目录设置

资源的定义

## Log4j配置

每个项目设置自己的文件名；

## 实体类，继承自EhealthPersistentEntityWithICPCODE，

其不能被其它机构用户删除；

当需其它机构删除时，不要继承自这个；

# 前端开发

在新建动态web工程中，

粘贴HYSAPP 、jquery-easyui 1.4 、 mainlayout 3个目录和mainlayout.html文件到webroot目录；

## 依赖

jQuery1.11.1

jquery-easyui1.4

ztree.js

扩展：

extjs

bootstrap

## 主页结构

1 内容包括

master.html

HYSAPP.common.\* //前端应用的基础

HYSAPP.master.\* //布局页面的mvc

[必须先加载HYSAPP/common/NameSapce.js]

2 布局元素设置

在html文件中调用MasterControl.show()方法，将请求后端服务getUserData.do,获得并显示以下内容：

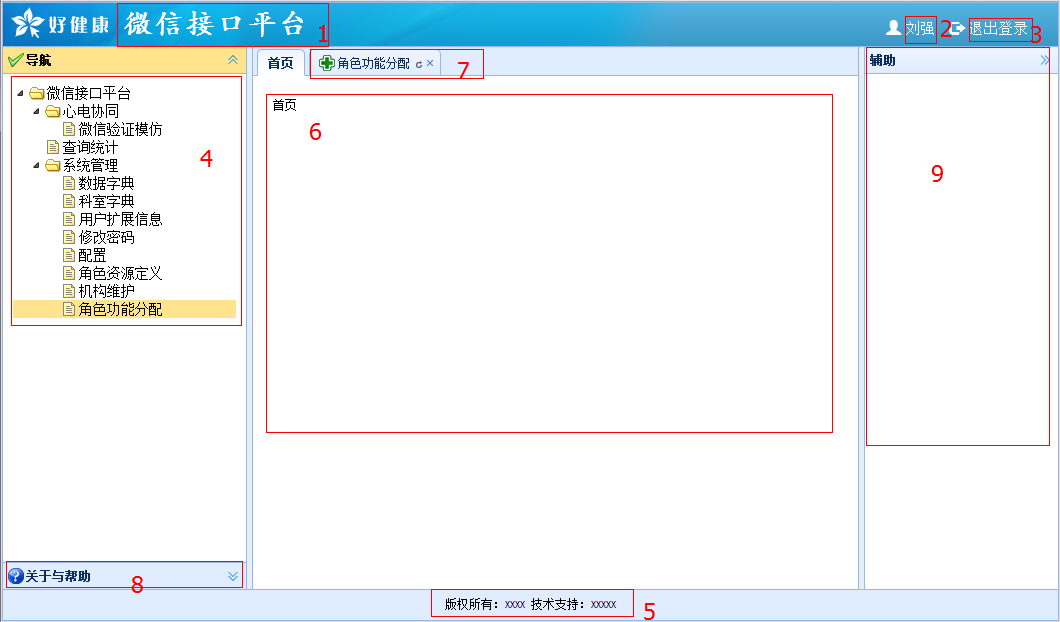
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Dom标识 |  |
| 1 | 系统名称 | Class=divTopTitle | 来自配置 |
| 2 | 当前用户姓名，点击打开修改密码的window | Id= userInfo |  |
| 3 | 退出登录 ，点击退出，返回到登录页面； | Id=userLogout | location.href ="exitLogin.do"; |
| 4 | 动态个人角色对于的导航菜单，点击在tab中新建或打开tabpage； | Id= westNav  Id=navTree |  |
| 5 | 版权和技术支持 电话 |  | 来自配置 |
| 6 | 指定首页显示页面 |  | Master.html指定首页的href |
| 7 | 增加或打开tabpage的方法，$(“#centerTabs”).tabs(), | ID=centerTabs |  |
| 8 | 关于和帮助可自行定义，并在master.html中增加js代码 |  |  |
| 9 | 辅助区域可去掉或自行定义，并在master.html中修改 |  |  |

后台有两个服务提供：

menuUrl:"common/user/getRoleMenuJSON.do",

userUrl:"common/user/getCurrentUserInfo.do",

首页布局如下：



**另外还有一个隐藏的div id=win，用来作为弹出的window使用；如：**

**$('#win').window({title:"修改密码"}).window('open');**

主页已经使用的id不能被其它页面使用，包括：

Class = divTopTitle

Workspace

centerTabs

userInfo

userLogout

westNav

navTree

Win

## 导航菜单

* 请修改HYSAPP/common/MenuData.js 并将其中的数组数据部分复制到/MenuData.json文件中; -----admin用户登录或未设置menuUrl时,将使用此数据，作为原始菜单数据使用
* 后台服务支持：common/user/getRoleMenuJSON.do
* MenuData ：
  + Independent，用来表示页面显示方式：
    - True,新建tab，并加入<iframe> src=action
    - False,新建tab，直接把action的页面内容写入主页内
    - “win”，用$.window(“open”)
  + iconclass,没有定义的，以文件夹、文件作为图标；点击菜单，在tabs中新增的tabpage将加载相同的图标，没有定义的，图标为icon-add;

## 布局

使用easyui-layout对页面内容分割排列

## JS面向对象编程

即时函数，闭包，避免冲突；

构造函数，原型；

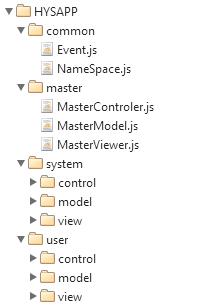
## 命名空间

HYSAPP = {}

HYSAPP.common={}

HYSAPP.

HYSAPP.master={}



## CRUD页面实例

* 参看：用户管理
* 采用easyui的html

（菜单定义independent=tue，那么需要在html文件中引入css文件，引入jquery.js jquery-easyui.js；否则不用引入）

* Js文件：

代码包裹：Jquery(document).ready(function($){});

全局函数的设置:HYSAPP.页面名称.函数名=function(){};

## MVC

网页（html+js）

表格

[按钮]

Controler

view

Model

Mvc,只是与“数据”相关的处理；如上图，网页中的表格由mvc驱动，按钮则不是mvc中的，按钮使用controler的方法执行，controler调用model的方法（因为真正与服务端数据交互是model完成的），model的数据变化，通过view表现到网页上；controler作为model和view的容器，将model方法、view事件（如表格单击一行）包裹，统统由其管理；

1. Model

构造函数：参数指定curd的url

属性：items存储相关数据，

事件：retireved：检索后触发，

方法：retrieve：向服务端发送请求获得数据，并使用retrieved.notify()通知已完成检索

1. view

构造函数：包含model和selector

          为model.retrieved的事件附加处理方法，即响应retrieved事件；

属性：$wrapper 用$包裹的dom节点的jquery对象；

注意，当前容器是dom元素定位的基础。

HYSAPP.panelContainer代表了tabs内的当前tabpage的容器；应当以$(“domid”,HYSAPP.panelContainer)的方法对dom元素操作；

事件：itemClick 点击一个内容是触发  
方法：rebulid show，根据model.items展现内容到$wapper，并为每个数据项的元素添加click事件；

1. control（可不创建对象，而直接在页面中书写）

建立model  
建立view  
为view.itemClick事件附加响应处理过程  
执行model.retrieve的方法检索数据（同时实现页面上展现出数据内容）；

**目录要求：**

以html文件的路径+文件名作为mvc的包路径

如,页面/common/system/rolemenu.html

那么对应的mvc应该是 :

HYSAPP.common.system.rolemenu.xxxxModel.js

HYSAPP.common.system.rolemenu.xxxxViewer.js

HYSAPP.common.system.rolemenu.xxxxControler.js

[在master.html中加载mvc的js文件，调试是才能看到源码]

4 关于 web前端开发

需求分析

平面设计，尽量详细；

（粗糙设计将导致开发后期的时间花费）

页面逻辑编码

结合页面，设计服务端接口

Java端编码

功能设计，数据设计

服务端架构设计，

以业务实现为目标的接口、类设计

逻辑分层，模块划分，开发组件

## 表单处理

Css，采用bootstrap的表单样式；

栅格系统

## 前端API介绍

### HYSAPP的基本属性

HYSAPP={

application:”zt”,

titleText:””

namespace:function(){}}

### HYSAPP\common\NameSpace.js

在HYSAPP下提供了若干方法：

### HYSAPP.namespace()

### HYSAPP.$import()

### ~~top.HYSAPP.ajaxSetting()~~

处理ajaxStart()显示提示信息；ajaxStop()关闭提示信息；ajaxSuccess()当后台处理有异常时提示异常result.fault；ajaxError()当服务器端错误显示其信息，当403时跳转到login.jsp;

调用HYSAPP.showLoading();加载提示

调用HYSAPP.closeLoading();关闭提示

调用top.HYSAPP.messagebox(),显示异常、错误信息；

### top.HYSAPP.messagebox()

消息提示框

### HYSAPP.closeTabPage();

~~//关闭当前tabpage~~

### HYSAPP.openTabPage(options);

~~//新建立一个tabpage~~

### HYSAPP.getParameter()

//取url中的参数

### HSYAPP\common\下的控件

Event.js

ImgButton.js

MenuData.js

~~NavAccordion.js:主页左侧导航菜单~~

~~Tabs.js：tabpage页~~

TreeDiv.js：树形控件；

~~DataLIst.js:分页表格控件~~

### //在当前tabpage中执行前一tabpage的iframe窗口的一个方法

**top.getOldPanel().find("iframe")[0].contentWindow**.XXXXX();

### 表单的异步请求

表单序列化

Var formData = $(\_form).serialize();

异步请求

$.ajax({data:formData});

# 后端Ssh，基本框架

User,role,menu,resource, 机构，科室，配置管理，安全，日志，AbstractAction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户 | 角色 | 角色功能点菜单 | 机构 | 科室 | 资源 |
| 表 |  |  |  |  |  |
| HYS\_USER  HYS\_USER\_ZTEXT  HYS\_USER\_CSHISEXT  HYS\_USER\_HISEXT  HYS\_USER\_JKDAEXT  HYS\_USER\_JXKHEXT  HYS\_USER\_PTEXT  HYS\_USER\_XNHEXT  HYS\_USER\_YKTEXT  HYS\_USER\_ETLEXT  HYS\_USER\_HPMHEXT  HYS\_USERS  HYS\_USER\_NETWORK  Ehealth\_user 视图 | EHEALTH\_ROLE | ROLE\_MENU  [EHEALTH\_USER\_MENUROLE  健康档案系统使用的] | T\_GOVINFO | DW\_DEPART；  DW\_DEPART\_BASE | EHEALTH\_RESOURCE |
| Model对象 |  |  |  |  |  |
| Ehealthuser  Hysuser  HysUserZText  Jkdauserext | EhealthRole | RoleMenu  [Menurole健康档案使用的] | TGov | Depart  DepartBase | EhealthResource |
|  |  |  |  |  |  |

## Spring

### 安全

Hysuser，要实现UserDetails接口

**Web.xml的安全配置**

* 加入listener，如ServletContextLoaderListener，加载“角色、url资源定义”，（角色资源对照关系，启动web服务时加载，保存到

servletContext.setAttribute("urlAuthorities", urlAuthorities);）

* springSecurityFilterChain，启动Spring Security过滤器链进行各种身份验证和授权服务，可以通过参数<url-pattern>/j\_spring\_security\_check</url-pattern>设置表单验证的地址名称
* OpenSessionInViewFilter，是hibernate相关；有这个后，需要配置事务的边界

**Application-context-security.xml**

* concurrent-session-control
  + max-sessions="1"
  + exception-if-maximum-exceeded="false" ：

相同的用户名登陆将使前一个使用此用户登陆的session无效

* intercept-url，顺序很重要（不需要认证的目录写在前面）filter=”none”：不进入认证的url；以下是access的设置

1. IS\_AUTHENTICATED\_ANONYMOUSLY允许匿名用户进入
2. IS\_AUTHENTICATED\_FULLY 允许登录用户进入
3. IS\_AUTHENTICATED\_REMEMBERED 允许登录用户和rememberMe用户进入
4. IS\_AUTHENTICATED\_FULLY：是则满足以下情况返回通过:
   * \*\*.既不是RememberMeAuthentication也不是AnonymousAuthenticationToken的实例
5. IS\_AUTHENTICATED\_REMEMBERED：是则满足以下任一情况返回通过:
   * a\*.Authentication是RememberMeAuthenticationToken的实例
   * b\*.既不是RememberMeAuthentication也不是AnonymousAuthenticationToken的实例
6. IS\_AUTHENTICATED\_ANONYMOUSLY：是则满足以下任一情况返回通过:
   * a\*.Authentication是AnonymousAuthenticationToken的实例
   * b\*.既不是RememberMeAuthentication也不是AnonymousAuthenticationToken的实例
   * c\*.Authentication是RememberMeAuthenticationToken的实例

AccessDecissionManager 用于授权一个特定的操作

三个实现类分别为：

AffirmativeBased 只需有一个投票赞成即可通过(最常用）；allowIfAllAbstainDecisions属性值默认为false，意思是如果所有的授权投票是都是弃权，则通不过授权检查。

ConsensusBased 需要大多数投票赞成即可通过；

UnanimousBased 需要所有的投票赞成才能通过(悲观吧）

* 认证提供者，实现UserDetailsService接口（登录中，根据提供的用户名获取用户对象）
* URL资源定义提供者（用户密码验证通过后被调用）
* authentication-provider，使用“认证提供者”；
* authenticationManager，使用了authentication-provider
* 授权器 ，其中可以自定义投票器
* URL资源拦截器，使用：
  + authenticationManager
  + 授权器
  + URL资源定义提供者

**Spring-context-purview.xml:**

配置bean

userDetailsService

messageSource（登陆失败的提示—异常信息国际化）

#### 我们的策略：

* 特殊服务.do，\*.js，图片等，不做filter；其它所有都需要filter;(在\*security.xml的<http>中配置)
* 所有目录分别设置为“资源”，为角色配置需要的资源；用户登录后，其角色将参与验证使用
* Admin用户是超级用户，可以无条件访问（用户获得角色时，admin用户自动加入ROLE\_ADMIN;请求URL进行filter时，都将加入ROLE\_ADMIN;）

### Spring struts整合

Struts-config.xml

Spring 和 struts 整合的四种方式。

1.使用Spring 的 ActionSupport   
2.使用Spring 的 DelegatingRequestProcessor 类。  
3.全权委托。org.springframework.web.struts.DelegatingActionProxy

4.使用 org.springframework.web.struts.AutowiringRequestProcessor 类,只需配置controller，不需要在spring中配置。自动装载能力强

本框架使用AutowiringRequestProcessor的方法，能够自动将façade注入到action

### spring hibernate整合

主要在appliction-context-model.xml，事务bean使用HibernateTransactionManager，还要配置事务边界等；

Web.xml 中OpenSessionInViewFilter，是hibernate相关；有这个后，需要配置事务的边界；

本框架是在\*Façade中自动实现事务管理的，要求其中的方法满足配置中的要求，名称以以下内容开始的：get set cancel find load save add update remove delete

## 日志，

### Web服务器日志：

Common-logging.jar和log4j.jar是必须的

log4j.properties文件需要放到web-inf/classess（创建出Log4J的运行环境），（策略：相关信息输出到控制台，同时写入文件）配置如下：

#记录器的级别DEBUG < INFO < WARN < ERROR < FATAL

#设置输出到stdout和file(还可以有更多)

log4j.rootLogger=**warn**,stdout,file

#=========

#stdout Appender， 日志输出：

#ConsoleAppender，控制台

#PatternLayout，格式自定义

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=[HYS] %d %5p[%t] %l - %m%n

#=========

#file DailyRollingFileAppender 每日日志输出（不同web目录，设置不同的文件名）

log4j.appender.file=org.apache.log4j.DailyRollingFileAppender

log4j.appender.file.File=${catalina.home}/logs//hys\_info\_log.txt

log4j.appender.file.DatePattern='\_'yyyyMMdd-HH'.txt'

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=[HYS] %d - %-5p %x - %m - %c%l -%-4r [%t] %n

### 业务系统日志记录：

EHEALTH\_SYSTEM\_RECORDLOG

com.hys.system.model.RecordLog

## Struts 基础--AbstractAction

ListAction,加入字典数据的方法，把page转成map，再把字典数据put进去：

Map preparedData = ActionHelper .*extractPaginationInfoMap*(dataPage);

preparedData.put("userlist", userlist);

return preparedData;

## 基础功能

### 配置

* 配置数据（ehealth\_system\_configzt）：分基本配置和扩展配置；id<=100为基本配置，开发阶段设置；id〉100为扩展配置，可修改在应用阶段有管理员配置；配置修改后需要重写
* 应用标识名称，在配置中有记录configname=’app\_name’设置本系统的模块标识，一般为项目拼音简写，如：jkda、zt等；均为小写；
* 本系统配置：owner=应用标识名称的配置数据为本系统的配置；
* ActionConstants.APP\_NAME
* ActionHelper.getConfigSet(Key)

### 已有服务

|  |  |  |  |
| --- | --- | --- | --- |
|  | url | 入参 | 结果 |
| 取当前用户的信息以及系统名称、技术支持信息等 | common/user/getCurrentUserInfo.do |  |  |
| 检索配置 | common/system/EhealthSystemConfigzt\_list.do |  |  |
| 删除配置 | common/system/EhealthSystemConfigzt\_delete.do |  |  |
| 增加或修改配置 | common/system/EhealthSystemConfigzt\_addedit.do |  |  |
| 检索角色（根据ActionConstants.app\_name） | common/user/ehealthrole\_list.do | 无 |  |
| 删除角色 | common/user/ehealthrole\_delete.do |  |  |
| 增加或修改角色 | common/user/ehealthrole\_addedit.do |  |  |
| 取角色的权限菜单 | common/user/getRoleMenuJSON.do | Roleid,  如果没有设置参数，则取当前用户的角色（可以有多个） | Json Array |
| 保存角色的权限菜单 | common/user/menurole\_addedit.do |  |  |
| 检索资源数据 | common/user/ehealthresource\_list.do |  |  |
| 删除资源 | common/user/ehealthresource\_delete.do |  |  |
| 增加或修改资源 | common/user/ehealthresource\_addedit.do |  |  |
| 根据角色取资源 | common/user/getResourceListByRoleid.do |  |  |
| 删除角色资源对照 | common/user/roleResourceDelete.do |  |  |
| 增加或修改角色资源对照 | common/user/roleResourceSave.do |  |  |
| 根据用户检索其所有的角色 | common/user/userRolelist.do | Userid |  |
| 删除用户角色对照 | common/user/userRoleDelete.do |  |  |
| 增加或修改用户角色对照 | common/user/userRoleSave.do |  |  |
| 列举机构树需要的数据 | common/user/tgov\_tree.do |  |  |
| 列举子机构数据 | common/user/tgovinfo\_list.do | parent\_jgbm |  |
| 删除机构 | common/user/tgovinfo\_delete.do |  |  |
| 增加或修改机构 | common/user/tgovinfo\_addedit.do |  |  |
|  |  |  |  |
|  |  |  |  |

## 扩展功能

使用模板生成单表操作代码

# 版本管理

应用程序，页面代码中写入发布版本号1.0.0001;

数据库中，保存数据库版本1.0;

版本要求，应用程序版本必须大于等于数据库版本；

管理要求：

# 扩展开发

## 用户管理

Hys\_user表是同其它系统模块共享的；在已有的系统中已经有此表数据了。在开发新的模块时，会增加用户的扩展信息表如hys\_user\_xxxxExt;扩展表与hys\_user表以username字段一一对照。确定增加用户扩展表时，将需要完成以下工作：

1 页面修改。表单中增加扩展表需要的字段输入内容；

2 用户保存的.do及相关后端代码修改，以保存到扩展表；

3 其它需要的位置使用扩展表；