●快速开发平台架构选型

主框架:SpringMVC 4.0+ OK

安全框架: Apache Shiro OK

工作流引擎框架:Activiti5.22 待评估

工作流设计器: Activiti Designer 5.18.0 Activiti-Modeler 待评估

JDK：1.7（Dynamic Web Module3.0） OK

Web容器:Tomcat7\Jetty9 OK

编译工具:Maven3.0+ OK

CSS模板: AdminLTE-2.3.7+\ OK

ORM框架: Spring JDBC +Hibernate OK

前后端交互: JSP+Ajax OK

jquery Validation Engine 验证框架（配合spring的验证框架，页面验证简化不少）

●开发工具

eclipse-jee-neon-3-win32-x86\_64 OK

mysql-5.7.16-winx64 OK

Navicat Premium\_11.1.8简体中文版 OK

ORM辅助工具: 配置文件\映射文件维护工具 待调查,目前手工维护

代码生成器:Jeesite代码生成器 待评估

Nodepad++7.3.3 (正则表达式) OK

●技术长期演进

-------------------------------------------------- 前端跨平台支持 -----------------------------------------------------------

移动APP、平板、智能设备

建立后端接口标准

表单交互控件的类型可由后端提供 JSON数据机构扩展 基础数据值：数据附加属性 使用JSON复杂结构

-------------------------------------------------- (模块化热插拔技术集成) -----------------------------------------------------------

OSGI: Eclipse  equinox\ Eclipse Virgo 待评估

-------------------------------------------------- (标准化业务模块/产品集成探索) -----------------------------------------------------------

集成市面上现有的标准产品可否集成到本平台

-------------------------------------------------- (先进技术追踪) -----------------------------------------------------------

分布式架构:??? 高压力 高吞吐量

Nosql与大数据:??? 非机构化海量数据处理与分析

●架构设计H+V+I+S+D

H：html +css负责页面描述UI静态结构和样式渲染 一个页面对应一个用户操作场景 页面迁移参考移动app

V：javascript展示数据已经动态的样式定义 只负责从服务端获取结构化数据并为H层提供动态展示，与H层1：1

I：提供数据获取的接口，与前端请求1：1，只负责提供结构化数据给V层，组织数据，安全验证。。。

S：逻辑处理层 负责进行业务逻辑处理，接收数据，返回数据，以业务模型为单元进行构造

D：数据操作层

一.ajax的优点：

    1.开发过程中前端与后端脱离，交互通过JSON传输来实现

    2.跨平台能力更强，依托于浏览器的支持

    3.使后台数据接口能够得到复用

二.ajax的缺点：

    1.开发难度大，考虑浏览器的兼容性

    2.页面请求过多

    3.属于后加载，无法被爬虫爬到

    4.接口代码需要新增很多

    5.无法直接显示java实体类对象，需要转换为json格式

三.jsp的优点：

    1.可被爬虫爬到

    2.减少请求次数

    3.不用考虑浏览器的兼容性

四.jsp的缺点：

    1.增大了服务器的压力

    2.前端与后端未脱离，拖慢开发进度

    3.过于依赖java运行环境

    4.复用较低。

●平台设计原则:

1. 四优原则:平台的学习曲线和+开发成本要优于纯代码开发; 使用平台开发的代码质量优于纯代码开发;平台灵活/扩展性和定制性优于市售封装化平台;平台演进和生命周期优于市售封装化平台
2. 模块化设计,任何模块和方便快捷的拆分出来作为独立项目运行,后续整合考虑OSGI技术或其思想的实现
3. 前后端低耦合:服务器不负责任何跳转与展示内容和形式,仅负责返回JSON数据,或者处理业务后返回提示信息,前台通过ajax请求数据,通过js控制业务与画面迁移
4. 后端服务可同时为移动客户端和PC端提供数据服务,
5. Action层暴露对外接口,业务处理在Service进行处理
6. 根据业务复杂度考虑引入前端js框架(TBD)

●数据库设计规范

数据库表名:全部小写,下划线分割,表名和对象名命名符合下列规则 例如 :f00\_user\_role b02\_zsml\_zszy(招生目录\_招生专业表)

数据库字段名:全部小写,下划线分割,避免使用id,name,key等简单单词,避免使用系统保留字 例如: record\_id user\_name xwbt(新闻标题) fbsj(发布时间)

bxx\_xxxx 业务相关表 bxx的xx用于区分模块,下划线后的部分用于命名表名,可用多个下划线分割,推荐采用拼音命名 例如 b01\_yjfx\_fxds(研究方向表\_方向导师表)

fxx\_xxxx 框架专用表 fxx的xx用于区分模块.下划线后的部分用于命名表名,,可用多个下划线分割,采用英文命名,含义与表名应一致 例如:f01\_dict\_data(字典数据项表)

f00\_xxxx 权限认证安全管理相关表

f01\_xxxx 系统通用功能相关表

●架构选型待办

SpringMvc框架精读官网案例和原理

基本框架搭建Spring+ Hibernate+maven+Hello Word 验证基本配置和目录结构 V0.1 完成

基本样例业务实现(CRUD),导入CSS框架\验证发和部署 V0.2 完成

实现通用多级字典管理功能 完成

验证和重构\评估前后端交互方式(Ajax\RESTful) V0.3 完成

异常和日志处理设计 完成

安全框架导入,结合Apache Shiro实现用户\角色\权限控制 V0.5 开发中

开发环境集成整合和绿色化

集成代码生成器,使用代码生成器实现一个基本业务(BUG管理CRUD) V0.6

架构测试,调查和评审架构的完备性以及和扩展性,进行调整

架构试用:第一个项目试用,试用完成改进问题后发布正式版本 V1.0

工作流导入,实现业务流程可视化定制

实现一个工作流业务(BUG审批流程),验证工作流的工程化使用 V1.1

架构试用:流程试用完成改进问题后发布正式版本 V1.2

工作流设计界面需要改进至可达到客户自定义使用的程度! V2.0

●可行性论证

小公司的优势：敏捷、专注、管理效率高、有细分市场

软件公司必须有自己的核心技术积累

有管理和技术双重经验的人做技术考虑会更全面，避免陷入为技术而技术的场景，可以平衡技术与成本、灵活与封装

技术发展目前正在转型（前后端隔离、重客户端、后端云化）和爆发，意味着有机会重新布局信息化市场

●Eclipse安装class反编译插件:Decompiler

点击菜单Help——Eclipse Marketplace,搜索Decompiler,安装Eclipse class Decompiler ,重启



●Oracle函数:结果集转化为逗号分割字符串函数wm\_concat()

SELECT DISTINCT

mlfx.xxfs

,mlfx.YJFX\_ID

,nvl(fx.fxqc,fx.yzfxmc) fxmc

,MAX(mlfx.order\_num) order\_num

,wm\_concat(dm.caption)

FROM

B02\_MLZYYJFX mlfx

,B00\_YJFX fx

,DM\_CODETABLE\_DATA dm

WHERE mlfx.YJFX\_ID=fx.RECORDID

and mlfx.mlzysx\_id='8c16c7de-2066-467c-bfc3-162bee995212'

and mlfx.zsfw=dm.pindex

and dm.codetablename='DM\_U262'

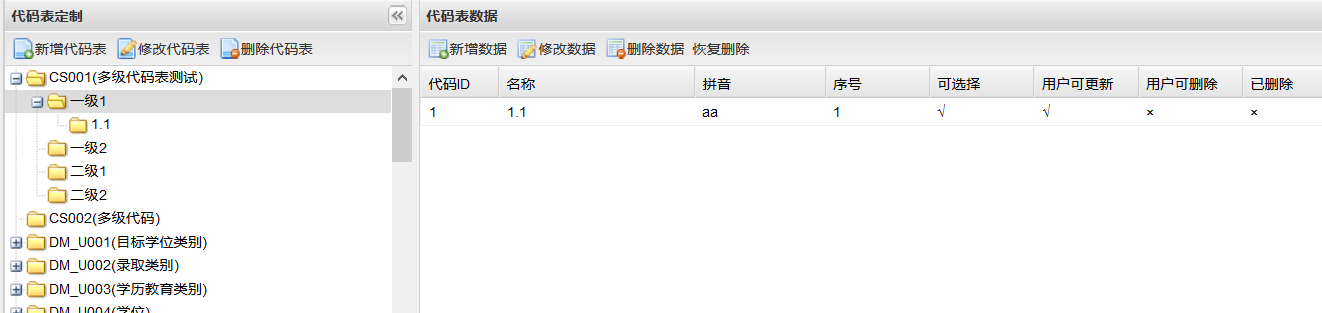
GROUP BY

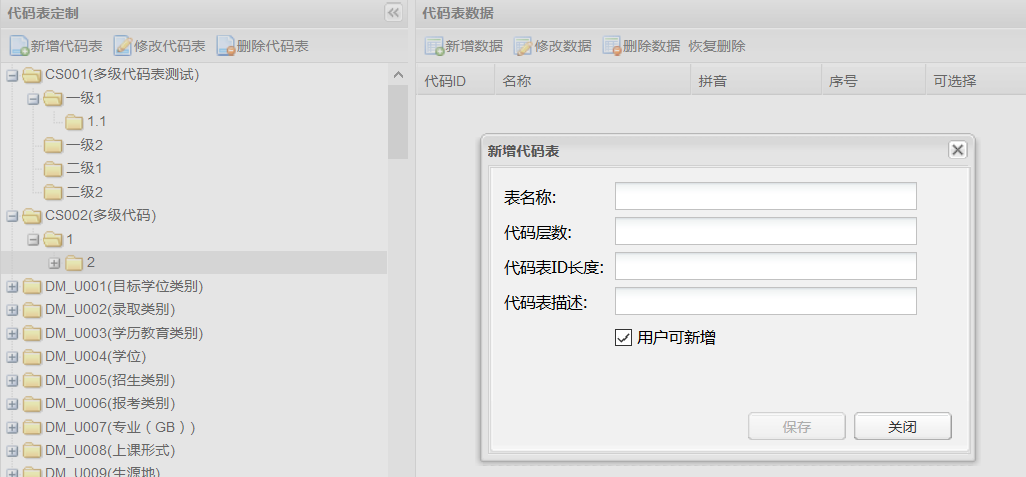
mlfx.xxfs,mlfx.YJFX\_ID,nvl(fx.fxqc,fx.yzfxmc),

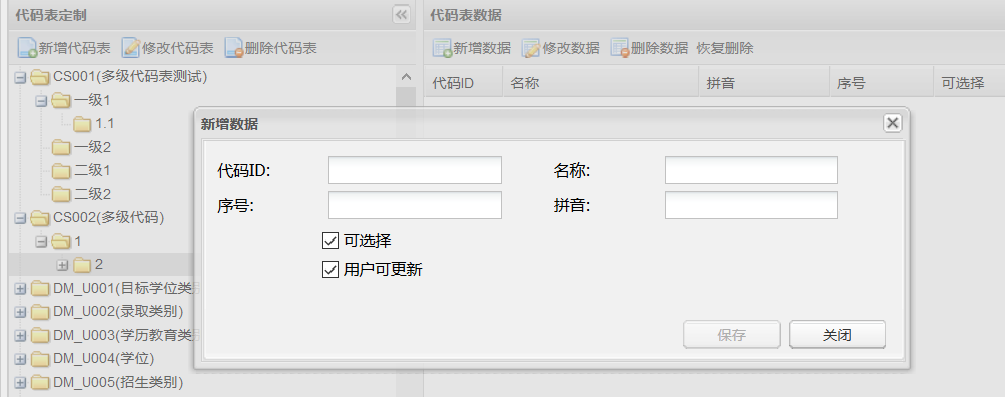
ORDER BY order\_num

●通用字典设计









●Oracle数据库导入导出

导出数据库备份

exp PKUDBNEW/PKUDBNEW@localhost:1521/orcl file=d:\aaa.dmp INDEXES=n STATISTICS=none

●通过数据库生成数据库设计文件的查询 Mysql

查询所有表

select TABLE\_NAME,NUM\_ROWS from user\_tables order by TABLE\_NAME

查询所有列

select c.TABLE\_NAME,c.column\_name,c.data\_type ,c.data\_length,c.data\_precision,c.data\_scale,c.DATA\_DEFAULT

from user\_tab\_columns c order by c.TABLE\_NAME

●Oracle数据库死锁解锁

1.下面的语句用来查询哪些对象被锁：

select object\_name,machine,s.sid,s.serial#

from v$locked\_object l,dba\_objects o ,v$session s

where l.object\_id　=　o.object\_id and l.session\_id=s.sid;

2.下面的语句用来杀死一个进程：

alter system kill session '24,111'; (其中24,111分别是上面查询出的sid,serial#)

●Oracle数据库操作日志分析工具使用



●设置Eclipse格式化代码换行宽度

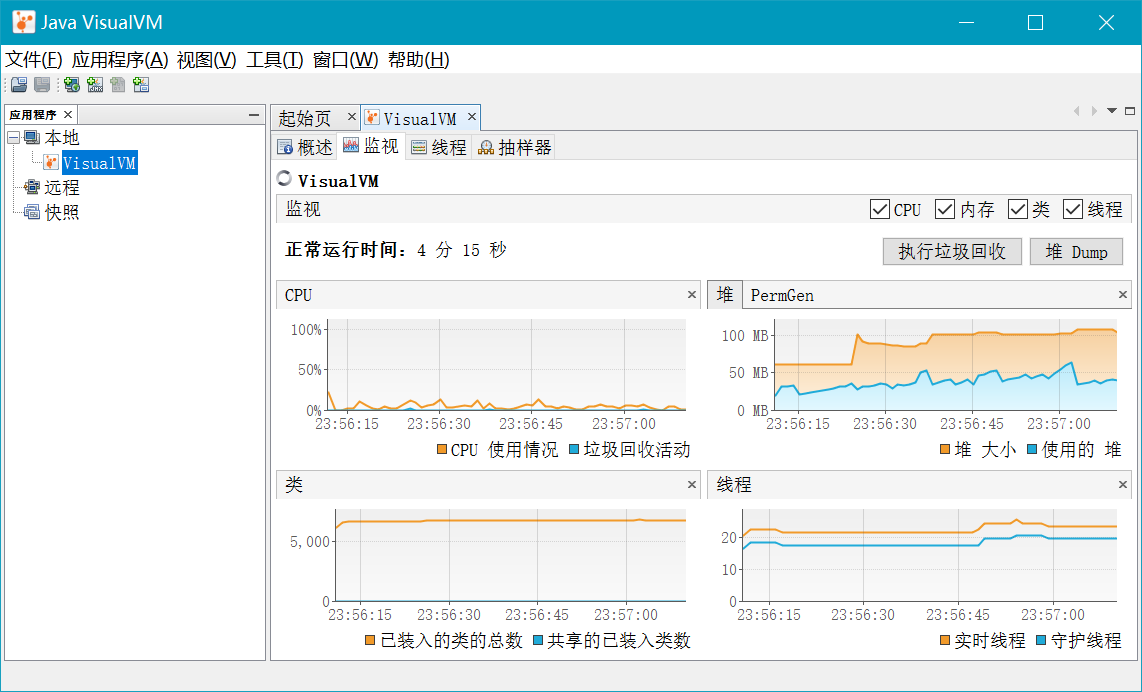
java---code style ---formatter

首先新建一个，如zxm。因为eclipse自带的不能编辑

编辑，在tab的Line Wrapping设置 最大的line width 即可。默认为80.这样以后代码自动格式化时不会给换行了。

●JVM性能监控

在命令行执行jvisualvm，在弹出的窗口 工具->插件->visual gc，安装



●数据库字段插入空值的时候可以插入空字符串””,不推荐插入null,字段中包含null则无法发挥索引优势

●JAVA字符串工具类

org.apache.commons.lang.StringUtils

使用介绍

http://blog.csdn.net/hu\_xinxin/article/details/9389385

●Mysql绿色版数据库初始化

1.命令行执行: mysqld --initialize --user=mysql –console 执行以上命令, 生成库（data目录）. 生成临时密码, 要记下来

2.命令行登录Mysql：mysql -uroot –p 用上一步的临时密码登录

3.修改root用户密码： set password for root@localhost = password('root');

4.打开mysql安装目录的data文件夹，删除以下2个文件：ib\_logfile0和ib\_logfile1

5.重启Mysql服务

●Mysql命令行启动及WINDOWS服务安装卸载和启动停止

命令行开启mysql服务：在Mysql的安装路径bin目录下执行: mysqld –console

安装服务：mysqld --install RADE-MYSQL-3308

卸载服务：mysqld --remove RADE-MYSQL-3308

启动、停止服务：

set SERVICE\_NAME="RADE-MYSQL-3308"

for /f "skip=3 tokens=4" %%i in ('sc query %SERVICE\_NAME%') do set "zt=%%i" &goto :next

:next

if /i "%zt%"=="RUNNING" (

echo 【%SERVICE\_NAME%】服务正在运行，即将停止该服务。。。

net stop %SERVICE\_NAME%

) else (

echo 【%SERVICE\_NAME%】服务已经停止，即将启动该服务。。。

net start %SERVICE\_NAME%

)

●Mysql日志及性能调优

●ORACLE数据库查询数据库表和视图的结构信息

select \* from user\_tab\_columns where table\_name='VIEW\_LQXX\_SS'