

JavaEE技术

üFTP服务器上经常会增加或修改一些资源，需要你去关注！

ü细节决定成败：在学习开发过程中，较多的同学遇到运行报错、网页打开不正常等问题，大部分都是因为工程结构混乱、参数配置有误、名称错了、引用有误、或者开发工具使用不当等这些小问题造成的。

ü培养良好的习惯。

ü学习过程中，细心+耐心+勤劳=成功——完成JavaEE项目的开发。

ü前端开发：精通JS，能熟练应用jQuery，懂CSS，能熟练运用这些知识，进行交互效果的开发。

ü后端开发：会写Java代码，会写SQL语句，能做简单的数据库设计，会使用框架，懂一些设计模式等。

所有资源用于JavaEE 相关课程教学和实训—2017

所有教学视频因录制时的屏幕分辨率比较高，只有全屏观看才能看到所有操作。

教学内容

更新时间：2017/3/15 12:02:24

理论篇

可行性与计划研究阶段

需求分析阶段

设计阶段

实现阶段

测试阶段

运行维护阶段

准备篇

下载资源

安装JTM

设置环境变量

视频0-JTM安装

安装MyEclipse

常用工具

考核任务1

入门篇

1、借鉴前端页面

在线考试系统的功能需求

2、准备Jar包和js库

3、设计数据库

4、设计工程结构

视频1-入门准备

5、30分钟把项目跑起来

新建项目

创建MainModule类

主模块上支持的注解

配置web.xml

添加数据源

添加Pojo类

新建MainSetup类

视频2-项目跑起来

考核任务2

初级篇

修改index.jsp

视频3-登录页面

用户登录模块

子模块上支持的注解

入口函数上支持的注解

Dao 接口的基本操作

增强安全

登录成功
登出系统
视频4—登录登出
考核任务3
用户信息处理（后端）
用户添加
用户删除
用户修改
用户查询
Tree菜单封装
Tree菜单数据的组织
数据封装成json
DataGrid数据封装
用户信息处理（前端）
后台布局
用户列表（查询+删除）
用户添加
用户修改
视频5—用户管理
考核任务4
提高篇
添加数据库表News，生成相应pojo
新闻信息处理（后端）
新闻信息处理（前端1）
新闻信息处理（前端2）
考核任务5
添加数据库表Tik，生成相应pojo
题库信息处理（后端）
题库信息处理（前端）
考核任务6
添加数据库表userpk，生成相应pojo
添加数据库表scores，生成相应pojo
考试信息处理（后端）
考核任务7
考试（前端）
考核任务8
整体测试与部署
考核任务9

理论篇

软件开发流程，即软件设计思路和方法的一般过程，包括设计软件的功能和实现的算法和方法、软件的总体结构设计和模块设计、编程和调试、程序联调和测试以及编写、提交软件的使用手册等一系列操作。

软件开发标准流程分为六个阶段：

可行性与计划研究阶段

❶可行性研究报告：在可行性研究与计划阶段内，要确定该软件的开发目标和总的要求，要进行可行性分析、投资-收益分析、制订开发计划，并完成应编制的文件。

❷项目开发计划：用文件的形式，把对于在开发过程中各项工作的负责人员、开发进度、所需经费预算、所需软、硬件条件等问题作出的安排记载下来，以便根据本计划开展和检查本项目的开发工作。

需求分析阶段

❶相关系统分析员和用户初步了解需求，然后用WORD列出要开发的系统的大功能模块，每个大功能模块有哪些小功能模块，对于有些需求比较明确相关的界面时，在这一步里面可以初步定义好少量的界面。

❷系统分析员深入了解和分析需求，根据自己的经验和需求用WORD或相关的工具再做出一份文档系统的功能需求文档。这次的文档会清楚列出系统大致的大功能模块，大功能模块有哪些小功能模块，并且还列出相关的界面和界面功能。

❸系统分析员和用户再次确认需求。

❹实施步骤:调查研究,分析综合,书写需求分析报告,评审。

❺原则是:分解,化简,抽象,自顶向下,逐步细化。

设计阶段

○概要设计:需要对软件系统设计,包括系统的基本处理流程、系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等,为软件的详细设计提供基础。

○详细设计:描述实现具体模块所涉及到的主要算法、数据结构、类的层次结构及调用关系,需要说明软件系统各个层次中的每一个程序(每个模块或子程序)的设计,以便进行编码和测试。应当保证软件的需求完全分配给整个软件。详细设计应当足够详细,能够根据详细设计报告进行编码。

○数据库设计:设计软件所需数据库的所有标识、逻辑结构和物理结构。

实现阶段

○开发者根据软件系统详细设计中对数据结构、算法分析和模块实现等方面的设计要求,开始具体的编码工作,采用结构化设计方法,自顶向下逐步细化,分别实现各模块的功能,从而实现对目标系统的功能、性能、接口、界面等方面的要求。

测试阶段

○测试人员和用户,采用多种测试方法测试软件系统的所有功能模块,在线系统还要测试浏览器的兼容性、大量用户并发访问时的系统处理能力(压力测试),记录测试的结果、发现及分析作为测试报告。

运行维护阶段

○软件交付准备:在软件测试证明软件达到要求后,软件开发者应向用户提交开发的目标安装程序、数据库的数据字典、《用户安装手册》、《用户使用指南》、需求报告、设计报告、测试报告等双方合同约定的产物。《用户安装手册》应详细介绍安装软件对运行环境的要求、安装软件的定义和内容、在客户端、服务器端及中间件的具体安装步骤、安装后的系统配置。《用户使用指南》应包括软件各项功能的使用流程、操作步骤、相应业务介绍、特殊提示和注意事项等方面的内容,在必要时还应举例说明。

○用户验收。软件系统在验收后,正式上线运行。

○软件维护:包括纠错性维护、适应性维护、完善性维护和可预防性维护,保存维护记录,评价维护效果。编写维护修改报告。

○在整个软件开发过程中,所占费用最多的就是软件的维护,而且评价一款软件的好坏,可维护性也占有很大的份额,所以,我们在做软件开发时需要做好软件维护。

准备篇

下载资源

从<ftp://10.1.13.253> 下载全部资源

安装JTM

JTM是Windows下绿色的JDK + Tomcat + MySQL环境集成(包),用户无需对JDK、Tomcat、MySQL进行任何安装和配置即可迅速搭建支持JavaEE项目 + MySQL的服务器运行环境。

●新建服务器目录 d:\jtmlrw

jdk 解压得到

tomcat 解压得到

mysql 解压得到

webapps 新建

StartMySQL.bat 安装并运行MySQL

StartTomcat.bat 安装并运行tomcat

●保证批处理文件中的端口号(3306,8080)与配置文件 (mysql\my.ini, tomcat\conf\server.xml)中的端口号一致。

设置环境变量

注意变量值中的分号

JAVA_HOME= D:\JTMLRW\jdk

CLASSPATH=.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar;

在**Path**最前面添加**%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;**

——也可以一键设置以上3个环境变量,运行setJDKenv.bat,然后按提示操作。——

●右键以管理员身份运行Startsql.bat,右键以管理员身份运行Starttomcat.bat

●复制tomcat\webapps\ROOT到d:\jtmlrw\webapps

●在浏览器输入localhost:8080,看到图像cat,即判定tomcat安装成功

●安装过程中发现的问题:

☆安装目录有空格

☆JTM的目录名与批处理文件中的不一致

☆以前安装的版本没有完全卸载，冲突

☆环境变量冲突

☆端口号冲突。Dos命令 **netstat -ano** 配合任务管理器 能帮你识别占用端口的进程

视频0--JTM安装

安装MyEclipse

●安装原版，安装路径和工作空间workspace尽量选择磁盘中一级目录，目录选择短小英文名

●作为学习、研究用途时的破解：

运行crack.bat--输入 UserCode内容--点击 SystemId... --点击 Active--Tools -> 2.SaveProperties--Tools -> 1.ReplaceJarFile--选择 Myeclipse2015安装目录中的plugins--打开--稍等，Replaced...Done,完成激活。

如果出现没激活，则重启电脑重新激活。

MyEclipse常规设置：

●windows--preferences--General--Appearance--Colors and Fonts:

字体设置稍大一点，如14

●windows--preferences--General--Web Browser:

选用系统默认的外部浏览器，通常为便于调试的**极速模式浏览器**

●**windows--preferences--General--Workspace:**

Encoding为UTF-8★非常重要★

●windows--preferences--myeclipse--maven4myeclipse:

为了避免长时间联网更新，只勾选offline、Do not automatically update...；mavenjdk改为JTM中的jdk

●windows--preferences--myeclipse--Files and Editors:

设置各类文件的Encoding为UTF-8

常用工具

1、HeidiSQL

HeidiSQL是一款用于简化 MySQL 服务器和数据库管理的图形化界面。HeidiSQL软件允许浏览数据库，管理表，浏览和编辑记录，管理用户权限，从文本文件导入数据，运行 SQL查询，在两个数据库之间同步表以及导出选择的表到其它数据库或者 SQL 脚本当中。HeidiSQL提供了一个用于在数据库浏览之间切换 SQL 查询和标签带有语法突出显示的简单易用的界面。其它功能包括BLOB 和 MEMO 编辑，大型SQL脚本支持，用户进程管理等。HeidiSQL免费开源。

●通过HeidiSQL工具管理MySQL数据库、创建表、编辑数据，**设置字符集为UTF8★非常重要★**

2、Notepad++

程序员必备的文本编辑器，软件小巧高效，支持27种编程语言，通吃C,C++,Java,C#,XML,HTML,PHP,JS等，可完美地取代微软的记事本。

●配置Notepad++到右键菜单

3、MagicalTool

代码自动生成工具，以前版本的名字叫：HiberanteTools 和 hbmTools，现在更名为：MagicalTools 功能更多，支持模板，可以直接生成dao,controller,service等瞬间完成项目表的增删查改；

●由建好的数据库直接快速生成pojo类

考核任务1

JTMM的安装与配置

入门篇

1、借鉴前端页面

充分利用浏览器的极速模式或chrome浏览器中的开发者工具（F12或Ctrl+Shift+I）：

通过从网页上的元素，右键"审查元素"在**Elements**中定位标签元素、内容、样式CSS

从**Console**找到加载出错的原因，如缺少文件、JS错误...

从**Sources**中找到网页中直接引用的文件：如图片、JS、CSS...

借助Notepad++对保存的网页文件、CSS文件、js文件进行适当的改、删、换...最终得到一个合适的漂亮页面

在线考试系统的功能需求

考生：考生以自己的学号/工号作为考号，登录系统；登录成功后可以看到考试相关信息，如考试须知、考试内容、考试时间等；在规定的考试时间范围内，在线答题、交卷。

管理员：可以对考生信息实现增删查改；可以对题库信息实现增删查改；可以指定考生、指定科目，安排考试，发布考试信息；可以查看考试成绩、导出成绩单。查看登录日志。

2、准备Jar包和js库

本次课程使用以下的Jar包，基本上满足需求了。如果在开发其它项目时，需要其它的Jar包，基本上都可以从[Maven](#)查找和下载各种Jar包，有很多Jar包还提供各种历史版本。

■**Nutz**—组轻便小型的框架的集合，最大限度的提高Web开发人员的生产力是这个框架唯一的目标

■**mysql-connector-java** Mysql数据库驱动

■**mdruid** 数据库连接池Druid，带强大的监控功能

■**Log4j** 用Java编写的可靠、快速和灵活的日志框架(API)

■**beetl** 一款6倍于Freemarker的超高性能的java模板引擎

■**zbus** 支持消息队列, 发布订阅, RPC, 代理(TCP/HTTP/DMZ)亿级消息堆积能力、支持HA高可用

■**jxl** 通过java操作excel表格的工具类库

■**shiro** 用于认证、授权、加密、会话管理、与Web集成、缓存等

.....

本次课程还将使用到一些常用的JS库文件，如jquery，easyui等。

3、设计数据库

借助HeidiSQL等管理MySQL图形界面的工具，创建数据库exam及数据表user，数据库及表的字符集确保为utf8。

Field	Type	Null	Key	Default	Comment
id	varchar(16)	NO	PRI	(NULL)	体现层级，如三级000100030010
name	varchar(50)	NO			栏目名称
url	varchar(50)	YES			链接
permission	varchar(50)	YES		(NULL)	权限标识
contest					
Field	Type	Null	Key	Default	Comment
id	varchar(17)	NO	PRI	(NULL)	yyyyMMddhhmmssSSS
title	varchar(255)	NO		(NULL)	试卷或考试名称
intro	varchar(500)	YES		(NULL)	任务要求
bjs	varchar(500)	NO		(NULL)	接受任务的班级
kmcode	varchar(4)	NO		(NULL)	考试科目
kstype	varchar(2)	NO		0	考试类型1正式0模拟，是否生成试卷
kch	varchar(2)	NO		1	考场号
pc	tinyint(2)	NO		1	批次
starttime	datetime	NO		(NULL)	开始时间，校正服务器时间
endtime	datetime	NO		(NULL)	结束时间
cruser	varchar(20)	NO		(NULL)	创建者
chkid	varchar(150)	NO		0	指定选择题，题号列表(含，分隔)或套号
handleid	varchar(150)	NO		0	指定操作题，题号列表
status	char(1)	NO		0	0未开始1考试中2结束
sentp	char(1)	NO		0	0未发卷1已发卷

hsp	char(1)	NO		0	0未回收1已回收
loginlog					
Field	Type	Null	Key	Default	Comment
id	int(10)	NO	PRI	(NULL)	
uid	varchar(12)	YES		(NULL)	考号
loginip	varchar(15)	YES		(NULL)	登录ip
logintime	datetime	YES		(NULL)	登录时间
bver	varchar(100)	YES			登录浏览器
news					
Field	Type	Null	Key	Default	Comment
id	int(11)	NO	PRI	(NULL)	
classid	int(4)	YES		0	所属分类
bjs	varchar(50)	YES		(NULL)	接收新闻的班级
title	varchar(255)	NO		(NULL)	新闻标题
content	text	NO		(NULL)	新闻内容
pic	varchar(100)	YES		(NULL)	新闻封面图片
istop	char(1)	NO		0	0不置顶1置顶
tjdate	datetime	NO		(NULL)	更新日期
cruser	varchar(50)	NO		ucte	作者
hitnum	int(10)	NO		0	点击量
scores					
Field	Type	Null	Key	Default	Comment
id	int(10)	NO	PRI	(NULL)	
ksid	varchar(17)	NO		(NULL)	考试号，组织的考试/试卷
uid	varchar(12)	NO		(NULL)	考号
score	float(5,1)	YES		0	总分，卷面成绩
score1	float(5,1)	YES		0	题目1，通常是选择题
score2	float(5,1)	YES		0	题目2
score3	float(5,1)	YES		0	技能成绩
score4	float(5,1)	YES		0	技能成绩
score5	float(5,1)	YES		0	技能成绩
score6	float(5,1)	YES		0	技能成绩
score7	float(5,1)	YES		0	技能成绩
paper	text	YES		(NULL)	试卷
donum	int(11)	YES		1	练习次数
jjtime	datetime	NO		(NULL)	交卷时间
subject					
Field	Type	Null	Key	Default	Comment
kbcode	varchar(50)	NO	PRI	(NULL)	科目&知识块代码，参照国考
pcode	varchar(50)	NO		(NULL)	知识块所属科目
kemu	varchar(50)	NO		(NULL)	科目英文简称
kmcn	char(100)	NO		(NULL)	科目中文名称
ticount	tinyint(4)	NO		0	抽题数量

ghandle	tinyint(4)	NO		0	操作题套数
gchk	tinyint(4)	NO		0	选择题套数
qtcode	varchar(2)	NO		(NULL)	题型
ctnolist	text	NO		(NULL)	待抽题号列表
htnolist	varchar(250)	NO		0	操作题待抽题号列表
ksNotice	text	NO		(NULL)	科目考试需知
tik					
Field	Type	Null	Key	Default	Comment
id	int(10)	NO	PRI	(NULL)	
gid	int(10)	NO		0	试卷套号
tiid	int(10)	NO		0	套内题号
kmcode	varchar(3)	NO		0	科目代码
kbcode	varchar(12)	NO		0	知识块代码
qtcode	varchar(2)	NO		0	题型代码1单选2多选4判断
qteasy	char(1)	NO		0	题目难度1易2中3难
fenvalue	float(4,1)	NO		0	分值
qname	text	NO		(NULL)	题干
qnote	text	YES		(NULL)	评析/提示
qans	text	YES		(NULL)	答案, 选项字母ABCDEFGH.操作无参考答案
ansA	text	YES		(NULL)	选项A+判断
ansB	text	YES		(NULL)	选项B+判断
ansC	text	YES		(NULL)	选项C
ansD	text	YES		(NULL)	选项D
ansE	text	YES		(NULL)	选项E
ansF	text	YES		(NULL)	选项F
ansG	text	YES		(NULL)	选项G
ansH	text	YES		(NULL)	选项H
ansI	text	YES		(NULL)	选项I
user					
Field	Type	Null	Key	Default	Comment
uid	varchar(8)	NO	PRI	(NULL)	考号
xm	varchar(20)	NO		(NULL)	姓名
pwd	varchar(50)	NO		(NULL)	密码
department	varchar(50)	NO		(NULL)	部门
role	varchar(1)	NO		(NULL)	角色1管理员5考生
userpk					
Field	Type	Null	Key	Default	Comment
id	int(11)	NO	PRI	(NULL)	
uid	varchar(12)	NO			考号
kemu	varchar(10)	NO		15	科目
ksid	varchar(17)	NO		0	考试号长度大于0时, 统一考试。否则自我练习
chkid	text	NO		(NULL)	套号,选择题题号集合
handleid	text	YES		(NULL)	操作题套号,题号集合

kch	varchar(20)	YES		0	考场号
pc	tinyint(2)	YES		0	批次
loginip	varchar(15)	YES			登录IP
starttime	datetime	YES		(NULL)	
endtime	datetime	YES		(NULL)	

4、设计工程结构

一个软件系统项目，需要用到多种资源，为了科学管理各类资源，建议分类管理，例如：

exam	项目名称
--src	源码文件夹
--cn.lrw.exam	Main包，主模块
--cn.lrw.exampoj	Pojo包，子模块，存放数据库表/视图对应的Java类
--cn.lrw.exam.module	Module包，子模块，存放业务逻辑类
--cn.lrw.exam.utils	Utils包，子模块，存放封装公用方法或属性的类
--conf	存放js、json、properties等类型的配置文件
--ioc	存放诸如数据源的配置文件
--WebRoot	web项目根目录
--include	存放网页中引用的js、css、图像类文件
--css	
--js	
--img	
--web	存放网站中网页文件，如htm、html、jsp等
--WEB-INF	Java的WEB应用的安全目录，自动存放class文件，客户端无法访问
--lib	存放项目需要的jar包
--web.xml	Web工程的配置文件
--index.jsp	Web工程默认的首页文件

java中package命名规则

Java的包名都由小写单词组成，类名首字母大写。

为了保障每个Java Package命名的唯一性，在最新的Java编程规范中，要求开发人员在自己定义的包名前加上唯一的前缀；包的路径符合所开发的系统模块的定义，见包名知模块。一般路径为“**com.公司名或者团队名或者个人名.项目名.模块名....**”。

视频1--入门准备

5、30分钟把项目跑起来

新建项目

创建Web Project，添加conf文件夹，把准备好的Jar包添加到WebRoot/WEB-INF/lib目录下

创建MainModule类

·添加模块类自动扫描：@Modules(scanPackage=true)

将自动搜索主模块所在的包（包括子包）下所有的类

·配置IocBy: @IocBy(type=ComboIocProvider.class,args={"*js","ioc/*","*anno","cn.lrw.exam","*tx"})

控制反转 IoC是Inversion of Control的缩写，是说对象的控制权进行转移，转移到第三方，比如转移交给了IoC容器，它就是一个创建工厂，你要什么对象，它就给你什么对象，有了IoC容器，依赖关系就变了，原先的依赖关系就没了，它们都依赖IoC容器了，通过IoC容器来建立它们之间的关系。

@IocBy - 设置应用所采用的 Ioc 容器

一个Mvc框架可以通过Ioc接口同一个Ioc容器挂接，挂接的方法很简单：在主模块上声明 @IocBy oComboIocProvider的args参数,星号开头的是类名或内置缩写,剩余的是各加载器的参数

o*js 是JsonLocLoader,负责加载js/json结尾的ioc配置文件
o*anno 是AnnotationLocLoader,负责处理注解式Ioc, 例如@IoBean
o*tx 是TransLocLoader,负责加载内置的事务拦截器定义

·配置SetupBy: @SetupBy(value=MainSetup.class)

@SetupBy - 应用启动以及关闭时的额外处理

在整个应用启动或者关闭时, 你想做一些额外的处理工作, 你可以实现一个 org.nutz.mvc.Setup 接口, 并将其配置在主模块上

·配置视图模板: @Views({BeetlViewMaker.class})

主模块上支持的注解

@Modules	整个应用有哪些子模块
@IocBy	整个应用, 应采用何种方式进行反转注入。如果没有声明, 整个应用将不支持 Ioc
@Localization	整个应用的本地地化字符串的目录
@SetupBy	应用启动和关闭时, 应该进行的处理。
@Views	扩展整个应用支持的视图模板类型
@Ok	整个应用默认的成功视图
@Fail	整个应用默认的失败视图
@AdaptBy	整个应用默认的 HTTP 参数适配方式
@Filters	整个应用默认的过滤器数组
@Encoding	整个应用默认的输入输出字符编码

到目前为止, MainModule中完整的代码如下:

```
package cn.lrw.exam;

import org.beetl.ext.nutz.BeetlViewMaker;
import org.nutz.mvc.annotation.IocBy;
import org.nutz.mvc.annotation.Modules;
import org.nutz.mvc.annotation.SetupBy;
import org.nutz.mvc.annotation.Views;
import org.nutz.mvc.ioc.provider.ComboIocProvider;

@SetupBy(value=MainSetup.class)
@IocBy(type=ComboIocProvider.class, args={"*js", "ioc/", "*anno", "cn.lrw.exam", "*tx"})
@Modules(scanPackage=true)
@Views({BeetlViewMaker.class})

public class MainModule {
}
```

配置 web.xml

在display-name节点和welcome-file-list节点之间, 添加nutz的filter, 正确设置主模块的包

```
<filter>
<filter-name>nutzfilter-name</filter-name>
<filter-class>org.nutz.mvc.NutFilterfilter-class</filter-class>
<init-param>
<param-name>modulesparam-name</param-name>
<param-value>cn.lrw.exam.MainModuleparam-value</param-value>
```

```

init-param>
filter>
<filter-mapping>
<filter-name>nutzfilter-name>
<url-pattern>/*url-pattern>
<dispatcher>FORWARDdispatcher>
<dispatcher>INCLUDEdispatcher>
<dispatcher>REQUESTdispatcher>
filter-mapping>
<welcome-file-list>
<welcome-file>index.htmlwelcome-file>
<welcome-file>index.htmwelcome-file>
<welcome-file>index.jspwelcome-file>

```

添加数据源

在conf/ioc创建dao.js，正确设置数据库的驱动、地址、端口、数据库名、用户名和密码。

```

var ioc = {
  dataSource : {
    type : 'com.alibaba.druid.pool.DruidDataSource',
    events : {
      create : 'init',
      dispose : 'close'
    },
    fields : {
      url : 'jdbc:mysql://127.0.0.1:3306/exam',
      username : 'root',
      password : '',
      testWhileIdle : true,
      validationQuery : 'select 1',
      maxActive : 100
    }
  },
  dao : {
    type : 'org.nutz.dao.impl.NutDao',
    args : [{refer : 'dataSource'}]
  }
};

```

一组 IocBean 的配置信息，这个JS文件的格式通常为

```

{
  "对象名称": {
    type: "对象类型",
    parent: "被继承的对象名称",
    events: {
      fetch: "触发器的类型或者函数名",
      create: "触发器的类型或者函数名",
      dispose: "触发器的类型或者函数名"
    },
    args: [

```

```

参数1, 参数2 ...
],
fields : {
"字段名称1": 字段值1 ,
"字段名称2": 字段值2 ,
...
},
factory : "xxx.yyy.zzz#create" //工厂类#工厂方法
}
}

```

添加Pojo类

POJO (Plain Ordinary Java Object)，是指简单的Java对象，实际就是普通JavaBeans，包含属性及其getter/setter方法的类,没有业务逻辑。可以理解为简单的实体类，便于程序员使用数据库中的数据表，将POJO类当作对象来进行使用，可以方便的调用其getter/setter方法。

创建user数据库表对应的pojo类User，包名：**cn.lrw.examp.pojo**。

借助MagicalTools自动生成数据库表对应的pojo类，将生成的代码复制到新建的User类中。对于正式开发软件项目，通常会有大量的数据库表，需要生成对应的pojo类，如果借助MagicalTools会减少写代码的时间，提高开发效率。

在pojo类中，标注Nutz所需的注解**@Table**，**@Column**，**@Id**，**@Name**

整数型主键	注解 @Id	声明在字段上
字符型主键	注解 @Name	声明在字段上
复合主键	注解 @PK	声明在类上

```

package cn.lrw.exam.pojo;

import org.nutz.dao.entity.annotation.Column;
import org.nutz.dao.entity.annotation.Name;
import org.nutz.dao.entity.annotation.Table;

@Table("user")
public class User {

    @Name
    private java.lang.String uid; //考号
    @Column
    private java.lang.String xm; //姓名
    @Column
    private java.lang.String pwd; //密码
    @Column
    private java.lang.String department; //部门
    @Column
    private java.lang.String role; //角色1管理员5考生

    //以下省略了属性对应的get/set方法
}

```

新建MainSetup类

实现Setup接口,并在其中初始化数据库表，如添加一条默认的用户信息

```

public class MainSetup implements Setup {

    public void init(NutConfig nc) {
        Loc ioc = nc.getLoc();

```

```

Dao dao = ioc.get(Dao.class);
Daos.createTablesInPackage(dao, "cn.lrw.exam", false);
// 初始化默认根用户
if(dao.count(User.class)==0){
    User user =new User();
    user.setUid("2953");
    user.setXm("罗如为");
    String salt="abcdefghijklmnopqrstuvw";
    user.setPwd(new Sha256Hash("123", salt, 1024).toBase64());
    user.setDepartment("信息学院");
    user.setRole("1");
    dao.insert(user);
}
}

public void destroy(NutConfig nc) { // webapp 销毁之前执行的逻辑
}
}

```

视频2--项目跑起来

考核任务2

正确运行的Nutx入门项目（首页使用了登录页面的，加分）

初级篇

修改index.jsp

在网站首页上引入获取的登录页面。重要的修改：

- 1、css和js文件分类存放，单独创建文件夹
- 2、用户名和密码所使用的input，修改其id/name值，以便后台获取提交的信息。

```

src="/include/js/jquery.min.js">
src="/include/js/login.js">
<script>var base="<%=basePath%>";
<formid="login_form"method="post"onsubmit="returnfalse">
<u>
<li><inputtype="text" id="uid" name="uid" placeholder="考号">li>
<li><inputtype="password" id="pwd" name="pwd" placeholder="密码">li>
<li><label><inputtype="radio" name="role" value="5" checked="checked">考生label>
<label><inputtype="radio" name="role" value="1">管理员label>li>
<liclass="last"><a href="javascript:;" id="login_submit">登录a>li>
u>

```

- 3、采用jquery+ajax提交用户信息；提交前在客户端校验用户名和密码是否为空；前端校验通过后，采用JSON形式只封装需要的数据，然后post到后端，后端验证后返回信息，前端显示相应提示或进行跳转请求。

```

var bver,role="5";
$(document).ready(function () {
$('#login_form input').keydown(function (e) {
if (e.keyCode == 13)
{
checkUserName();//$('#login_submit').click();
}
}

```

```

});
$("input:radio[name='role']").change(function () {
role=$("input:radio[name='role']:checked").val();
});
$("#login_submit").click(checkUserName);
});
function checkUserName()//登录前，校验用户信息
{
var a=$('#uid').val();
var b=$('#pwd').val();
if(a==""){alert("请输入考号");return;}
var re=/^d{4}$|^d{8}$/;
if(!re.test(a)){alert("考号格式错误");return;}
if(b=="'||b==undefined){alert("请输入登录密码");return;}
$.ajax({
url : base+"user/doLogin",
data :{"uid":a,"pwd":b,"role":role},
type:"POST",
success :function (res) {
if (res == "1") {
window.location.href=base+"user/goIndex";
} else{alert(res);}
return false;
},
error : function(res) {alert("系统错误！");}
});
}

```

视频3--[登录页面](#)

用户登录模块

创建第1个Nutz模块UserModule，在UserModule类中实现对用户信息的增删改查。比如，用户登录实际上就是查找用户信息。

配置Ioc相关注解及属性,即IocBean，Inject，At和Dao属性

@IocBean	AnnotationIocLoader 根据这个注解来判断哪些类应该被自己加载
@Inject	AnnotationIocLoader 根据这个注解来了解类中的字段，具体的注入方式

@IocBean

@At("/user")

@Filters({@By(type=LoginFilter.class)})

publicclass UserModule **extends** BaseSrv{//实现对用户信息的增删改查

@Inject

protected Dao dao;

//查--登录

@At

```

@Ok("raw")
@Filters()
public String doLogin(@Param("uid")String uid, @Param("pwd") String pwd,
HttpSession session,HttpServletRequest req){
    User user = dao.fetch(User.class, uid);
    if (user == null) return"考号不存在！";
    String p =lrwCode(pwd, "");
    if (!p.equals(user.getPwd())) return"密码不正确";
    session.setAttribute("me", user);
    return"1";
}
@At
@Ok("beetl:web/index.html")//考生登录成功，跳转到考试信息页面
publicvoid goIndex(HttpSession session,HttpServletRequest req){
    User loginuser=(User)session.getAttribute("me");
    req.setAttribute("user", loginuser);
}
}

```

子模块上支持的注解

@InjectName	loc 容器中，本模块对应的名称，如果不指明，表示这个模块是通过默认构造函数创建的
@At	模块所有入口函数的 URL 前缀
@Ok	模块默认成功视图
@Fail	模块默认失败视图
@AdaptBy	模块默认 HTTP 参数适配方式
@Filters	模块默认的过滤器数组
@Encoding	模块默认 HTTP 请求的输入输出字符编码

入口函数上支持的注解

@At	函数对应的 URL
@Ok	成功视图
@Fail	失败视图
@AdaptBy	HTTP 参数适配方式
@Filters	函数的过滤器数组
@Encoding	HTTP 请求的输入输出字符编码
@GET	限定函数接受 HTTP GET 请求
@POST	限定函数接受 HTTP POST 请求
@PUT	限定函数接受 HTTP PUT 请求
@DELETE	限定函数接受 HTTP DELETE 请求

在[Aop](#)使用中，可以自定义注解。

Dao 接口的基本操作

插入	Insert	一条 SQL 插入一条记录或者多条记录
插入	FastInsert	一条 SQL ,通过batch插入多条记录
删除	Delete	一条 SQL 删除一条记录
更新	Update	一条 SQL 更新一条或者多条记录
获取	Fetch	一条 SQL 获取一条记录
查询	Query	一条 SQL 根据条件获取多条记录
清除	Clear	一条 SQL 根据条件删除多条记录
建表	Create	根据实体建表
删表	Drop	根据实体/表名称进行删表
聚合	Func	执行sum,count等操作

增强安全

设置Filter防止无登录用户访问网站敏感信息。

```
public class LoginFilter implements ActionFilter {
    @Override
    public View match(ActionContext context) {
        HttpServletRequest request = context.getRequest();
        User user = (User) request.getSession().getAttribute("me");
        String contentType = request.getContentType();
        if (Strings.isNullOrEmpty(contentType).contains("application/x-www-form-urlencoded") && user == null) {
            context.getResponse().setHeader("sessionstatus", "timeout");
            return new RawView("");
        }
    }
}
```



```

}elseif (user == null) {
ServerRedirectView view = newServerRedirectView("/web/nologin.jsp");
return view;
}
request.setAttribute("me",user);
String basePath=request.getScheme()+ "://" + request.getServerName() + ":" +request.getServerPort()
+ MvcS.getServletContext().getContextPath()+ "/";
request.setAttribute("basePath",basePath);
returnnull;
}
}

```

登录密码加密处理

```

publicclass BaseSrv {
/*
* 获取Sha256Hash加密密码
*/

protected String lrwCode(String password,String salt){
if(salt==""){
/*RandomNumberGenerator rng = new SecureRandomNumberGenerator();
salt =rng.nextBytes().toBase64();//随机salt*/
salt="abcdefghijklmnopqrstuvwx";
}
returnnew Sha256Hash(password, salt, 1024).toBase64();
}
}

```

当用户没有登录或访问超时，但仍旧访问需要登录才能访问的网页时，系统自动弹出提示，并跳转到系统首页。

无效登录页面nologin.jsp，Body中的内容。Js代码实现了2秒钟内跳转到系统首页：

```

<bodystyle="text-align:center">
<h1>登陆失效h1>
<p>对不起，您登陆已超时。p>
<scripttype="text/javascript">
setTimeout(reDo,2000);
functionreDo(){ top.location.href = "<0%=basePath0%>"; }
script>
body>

```

登录成功

考生登录成功后跳转到页面web/index.html,此页显示一些与考试相关的新闻，如考试须知、考试题型分值等信息，阅读相关信息后，考生确认跳转到考试页面。页面中包含了倒计时代码，强制考生阅读考试须知，并确认已阅读考试须知。

管理员登录成功后跳转到页面web/admin.html,此页以easyui布局形式展现后台管理功能，布局左侧以easyui树形结构显示分级菜单，布局中心区域使用了easyui标签页，标签页使用了iframe框架，框架中引用的页面是真正互动实现对信息进行增删改查操作的界面，界面通常会用到一些输入框easyui TextBox、按钮LinkButton...

查看[Easyui布局demo](#),[Easyui布局教程](#),
[Easyui树demo](#),[Easyui树教程](#),
[Easyui标签页demo](#),[Easyui标签页教程](#),
...更多

由于使用了Beetl模板，页面中部分代码需遵守模板语法，比如：

```
<div style="float: right; color: yellow"><font>${user.xm}, 您好！欢迎登录</font><br><a href="${ctxPath}/user/doLogout">【退出】</a></div>
```

查看[Beetl模板在线文档](#)。

登出系统

在前端页面中请求/user/doLogout时，UserModule业务逻辑模块中添加响应该请求的方法，必须使得session中保存的信息失效：

```
@At
```

```
@Ok(">:/")
```

```
public void doLogout(HttpSession session) {  
    session.invalidate();  
}
```

视频4--登录登出

考核任务3

实现正确登录跳转，不正确登录有明确提示

用户信息处理（后端）

在UserModule添加用户信息处理的方法--增删改查

可以根据需要约定**方法的命名规则**，便于团队协作，例如：

!请求打开视图goXXX

!执行保存saveXXX

!执行特定操作doXXX

!查询数据列表listXXX

用户添加

```
@At
```

```
@Ok("beetl:web/useradd.html")
```

```
public void goAdd(HttpSession session, HttpServletRequest req) {  
}
```

```
@At
```

```
@Ok("raw")
```

```
public String saveAdd(@Param("..") User user, HttpSession session, HttpServletRequest req) {  
    try {  
        String pwd = user.getPwd();  
        pwd = lrwCode(pwd, "");  
        user.setPwd(pwd);  
        dao.insert(user);  
    } catch (Exception e) {  
        e.printStackTrace();  
        return "添加失败";  
    }  
    return "true";  
}
```

```
}
```

用户删除

```
@At
```

```
@Ok("raw")//删除指定uid的单个用户
```

```
public String doDel1(@Param("uid")String uid,HttpSession session,HttpServletRequest req){  
    User user=(User)(session.getAttribute("me"));  
    if(user.getId().equals(uid))return "false";  
    if(dao.delete(User.class,uid)>0)return "true";  
    return "false";  
}
```

```
@At
```

```
@Ok("raw")//删除指定uid的多个用户
```

```
public String doDelN(@Param("uids")String uids,HttpSession session,HttpServletRequest req){  
    //User user=(User)(session.getAttribute("me"));  
    String[] s=uids.split(",");  
    if(dao.clear(User.class, Cnd.where("uid", "in", s))>0)return "true";  
    return "false";  
}
```

用户修改

```
@At
```

```
@Ok("beetl:web/useredit.html")
```

```
public void goEdit(@Param("uid")String uid,HttpSession session,HttpServletRequest req){  
    //Cnd cnd=Cnd.where("uid", "=", uid);  
    User user=dao.fetch(User.class,uid);  
    req.setAttribute("user", user);  
}
```

```
@At
```

```
@Ok("raw")
```

```
public String saveEdit(@Param("..")User user,HttpSession session,HttpServletRequest req){  
    try{  
        String pwd=user.getPwd();  
        if(pwd.length()>0){//用户输入了新密码  
            pwd=lrwCode(pwd,"");  
            user.setPwd(pwd);  
        }else user.setPwd(null);//保留原密码  
        if(dao.updateIgnoreNull(user)==1)return "true";  
        else return "修改失败";  
    } catch (Exception e){  
        e.printStackTrace();  
        return "修改失败";  
    }  
}
```

用户查询

```
//查询单个用户--登录
```

```

@At
@Ok("json")
@Filters()
public Object doLogin(@Param("uid")String uid, @Param("pwd") String pwd,
@Param("role")String role, @Param("bver") String bver,
HttpSession session,HttpServletRequestreq){
    NutMap re = newNutMap();
    User user = dao.fetch(User.class, uid);
    if (user == null || !user.getRole().equals(role)) {
        re.put("ok", false);
        re.put("msg", "考号不存在");
        returnre;
    }
    String p =lrwCode(pwd, "");
    if (!p.equals(user.getPwd())) {
        re.put("ok", false);
        re.put("msg", "密码不正确");
        returnre;
    }
    ///登录成功后...记录登录日志
    Loginlog log = newLoginlog();
    log.setUid(user.getUid());
    log.setLoginip(getIpAddr(req));
    log.setLogintime(new Date());
    log.setBver(bver);
    dao.insert(log);
    session.setAttribute("me", user);
    ///登录成功后...不同角色将请求跳转到不同页面
    if(role.equals("5")) {
        re.put("ok", true);
        re.put("msg", "user/goIndex");
        returnre;
    }
    elseif(role.equals("1")) {
        re.put("ok", true);
        re.put("msg", "user/goAdmin");
        returnre;
    }
    else {
        re.put("ok", false);
        re.put("msg", "系统错误！");
        returnre;
    }
}
}

```

```

//查
@At
@Ok("beetl:web/userlist.html")//跳转到用户信息列表
public void goList(HttpSession session,HttpServletRequest req){
}
@At
@Ok("raw")//分页查询指定用户或所有用户
public String listUser(@Param("page") int curPage, @Param("rows") int pageSize,
@Param("s_name") String s_name, HttpSession session) {
Criteria cri = Cnd.cri();
if (!Strings.isBlank(s_name)) {
cri.where().andLike("uid", s_name).orLike("xm", s_name);
}
else cri.where().andEquals("1", 1);
cri.getOrderBy().asc("uid");
return listPageJson(dao, User.class, curPage, pageSize, cri);
}
@At
@Ok("beetl:web/index.html")//考生登录成功，跳转到考试信息页面
public void goIndex(HttpSession session,HttpServletRequest req){
User loginuser=(User)session.getAttribute("me");
req.setAttribute("user", loginuser);
}
@At
@Ok("beetl:web/admin.html")//管理员登录成功，跳转到管理信息页面
public void goAdmin(HttpSession session,HttpServletRequest req){
User loginuser=(User)session.getAttribute("me");
req.setAttribute("user", loginuser);
}
@At
@Ok(">>:/")//登出系统
public void doLogout(HttpSession session) {
session.invalidate();
}
///查看登录日志
@At
@Ok("beetl:web/loglist.html")
public void goLog() {
}
@At
@Ok("raw")//分页查询所有登录日志
public String listLog(@Param("page") int curPage, @Param("rows") int pageSize) {
Cnd cnd=(Cnd) Cnd.where("1", "=", "1").desc("logintime");
return listPageJson(dao, Loginlog.class, curPage, pageSize, cnd);
}

```

```
}
```

Tree菜单封装

Tree菜单数据的组织

存放于数据库中的结构参照前文《数据库设计》部分中的**cmenu**,数据的组织是按id内容标识层次关系,name值为显示在界面上菜单名称, url为请求打开指定功能的页面的路径, permission设定为指定角色的用户才能看到相应的菜单项。

id	name	url	permission
01	系统管理		1
0101	考生管理	/user/goList	1
0102	题库管理	/tik/goTik	1
0103	排考管理	/pkao/goPkao	1
0104	成绩管理	/score/goList	1
0105	登陆日志	/user/goLog	1

数据封装成json

```
@At
```

```
@Ok("raw")
```

```
public Stringmenutree(HttpServletRequest req, HttpSession session) {
```

```
User user = (User) session.getAttribute("me");
```

```
String role=user.getRole();
```

```
//父级菜单
```

```
List menulist=dao.query(Cmenu.class, Cnd.where("id", "like", "___").asc("id")); //id=01,02,03
```

```
List eList = new ArrayList<>();
```

```
if(menulist.size() != 0){
```

```
for (inti = 0; i < menulist.size(); i++) {
```

```
Cmenu t = menulist.get(i);
```

```
if(!t.getPermission().contains(role))continue; //无权限,不显示相应菜单
```

```
EasyUITree e = new EasyUITree();
```

```
e.setId(t.getId());
```

```
e.setText(t.getName());
```

```
List eList2 = new ArrayList<>();
```

```
List menu2 = dao.query(Cmenu.class, Cnd.where("id", "like", t.getId() + "___").asc("id"));
```

```
for (intj = 0; j < menu2.size(); j++) { //二级菜单
```

```
Cmenut2 = menu2.get(j);
```

```
if(!t2.getPermission().contains(role))continue;
```

```
Map, Object> attributes = new HashMap();
```

```
attributes.put("url", t2.getUrl());
```

```
attributes.put("role", t2.getPermission());
```

```
EasyUITree e1 = new EasyUITree();
```

```
e1.setAttributes(attributes);
```

```
e1.setId(t2.getId());
```

```
e1.setText(t2.getName());
```

```

e1.setState("open");
eList2.add(e1);
}
e.setChildren(eList2);
e.setState("closed");
eList.add(e);
}
}
return Json.toJson(eList);
}

```

easyUITree结构:

```

publicclass EasyUITree{
private String id;
private String text;
private Boolean checked = false;
private Map attributes;
private String state = "closed";
public List children;
public String getId() {
returnid;
}
publicvoid setId(String id) {
this.id = id;
}
public String getText() {
returntext;
}
publicvoid setText(String text) {
this.text = text;
}
public Boolean getChecked() {
returnchecked;
}
publicvoid setChecked(Boolean checked) {
this.checked = checked;
}
public MapgetAttributes() {
returnattributes;
}
publicvoidsetAttributes(Map attributes) {
this.attributes = attributes;
}
public String getState() {
returnstate;
}

```



```

}

public void setState(String state) {
    this.state = state;
}

public List getChildren() {
    return children;
}

public void setChildren(List children) {
    this.children = children;
}
}

```

DataGrid数据封装

为了能让前端easyui datagrid获取正确的数据，需要在后端从数据库按查询条件cnd获取数据后，再次封装成包含分页curPage、pageSize的JSON格式数据。

```

public String listPageJson(Dao dao, Class obj, int curPage, int pageSize, Condition cnd) {
    Map jsonobj = new HashMap<>();
    Pager pager = dao.createPager(curPage, pageSize);
    List list = dao.query(obj, cnd, pager);
    pager.setRecordCount(dao.count(obj, cnd)); //记录数需手动设置
    jsonobj.put("total", pager.getRecordCount());
    jsonobj.put("rows", list);
    return Json.toJson(jsonobj);
}

```

用户信息处理（前端）

能够查询显示用户信息userlist、添加新用户useradd、删除用户和修改用户信息useredit的前端页面。后台管理，基本上采用了easyui控件，所以网页中引用了相关的css和js文件，其中`${ctxPath}`为beetl模板中定义的全局变量ContextPath，例如“/exam”，它不包含末尾的“/”，所以在使用时，需要在其后补充：

```

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" type="text/css" href="${ctxPath}/include/css/themes/default/easyui.css">
<link rel="stylesheet" type="text/css" href="${ctxPath}/include/css/themes/icon.css">
<script type="text/javascript" src="${ctxPath}/include/js/jquery.min.js">script<
<script type="text/javascript" src="${ctxPath}/include/js/jquery.easyui.min.js">script<
<script type="text/javascript" src="${ctxPath}/include/js/easyui-lang-zh_CN.js">script<

```

后台布局

后台管理页admin.html采用easyui布局+easyui 标签页，所以在网页中除了引用上述文件外，又引用了tabpage.js。

布局中的north区域放置了系统logo、登录用户名称、退出；west区域加载树形菜单；south区域显示版权信息；center区域使用tab来展示内容窗口；暂时没有使用east区域。

```

<body class="easyui-layout">
<div data-options="region:'north',border:false"
style="background:#B3DFDA;padding:0 10px 0 10px;vertical-align: middle;">

<div style="float:right;line-height:50px;margin-right:10px;">
<a href="#" id="logout">退出</a>
</div>

```

```

<div style="float:right;line-height:50px;margin-right:10px;">登录用户: ${me.xml}|div>
div>
<div data-options="region:'west',split:true,title:'系统菜单'" style="width:180px;padding:10px;">
<ul id="menutree" class="easyui-tree">ul>
div>
<div data-options="region:'south',border:false"
style="height:30px;padding:5px;text-align:center;font-family: arial;
color: #40B1BB;">Copyright © 2017 JavaEE.All Rights Reserved.
div>
<div data-options="region:'center'">
<div id="tabs" class="easyui-tabs" fit="true" border="false">
<div title="" id="home">
<iframe id="mainframe" name="mainframe" scrolling="auto"
height="99%" width="99%" frameborder="0" src="">iframe>
div>
div>
div>
body>

```

使用以下的js代码，在网页左侧(west)加载树形菜单，菜单数据来源于后台以json形式封装的层级菜单数据。如果点击一个菜单项，则在tab中打开相应的页面\$("#mainframe").attr("src", "\${ctxPath}/" + node.attributes.url),并在标签上面显示相应的名称setTabName(node.text); 如果点击一个父级菜单，则收缩collapse其它打开的父级菜单，展开expand当前点击的菜单。

```

$("#menutree").tree(
{
url : "${ctxPath}/menutree",
animate :true,
onClick :function(node) {
if(!node.attributes) {
if(!opened_node) {
$("#menutree").tree('expand', node.target);
opened_node= node.target;
} else if (opened_node!= node.target) {
$("#menutree").tree('collapse', opened_node);
$("#menutree").tree('expand', node.target);
opened_node= node.target;
}
} else {
$("#mainframe").attr("src", "${ctxPath}/" + node.attributes.url);
setTabName(node.text);
}
},
onLoadSuccess :function(node, data) {
$("#menutree").tree('expandAll');
}
}

```

```
});
```

设置进入后台管理页面后，默认打开的标签页：

```
setTabName("考生管理");
```

```
$("#mainframe").attr("src", "${ctxPath}/user/goList");
```

退出系统功能的实现：

```
$("#logout").click(function() {
```

```
dlg_h(300, 140, "退出系统？", "您确认要退出系统吗？", dlg_ok1, null, "question");
```

```
});
```

```
functiondlg_ok1() {
```

```
top.location.href = "${ctxPath}/user/doLogout";
```

```
}
```

布局框架中隐藏了已经封装的两种形式的公用对话框，一种只有“确认”按钮，另一种有“确认”和“取消”按钮，以下代码封装在 **dlg.html** 页面文件中，在 admin.html 文件中引用它 `<%include("dlg.html"){}%>`

```
<div id="dlg" style="padding:10px;font-size:14pt;" >div>
```

```
<script type="text/javascript">
```

```
/*
```

```
messenger-icon:error,info,question,warning
```

```
*/
```

```
functiondlg_h(w,h,t,content,fun_ok,fun_cancel,icon){
```

```
if(icon)$("#dlg").html("
```

```
+icon+"\">
```

```
"+content);
```

```
else$("#dlg").html(content);
```

```
if(w>800)w=800;
```

```
elsew='auto';
```

```
if(h<400)h='auto';
```

```
$('#dlg').dialog({
```

```
width:w, height:h,title:t,
```

```
iconCls: 'icon-tip',resizable: false, modal: true, closed: true,
```

```
buttons: [{
```

```
text:'确认',iconCls:'icon-ok',
```

```
handler:function(){
```

```
$('#dlg').dialog('close');
```

```
if(fun_ok)fun_ok();
```

```
}
```

```
},{
```

```
text:'取消',iconCls:'icon-cancel',
```

```
handler:function(){
```

```
$('#dlg').dialog('close');
```

```
if(fun_cancel)fun_cancel();
```

```
}
```

```
}}]
```

```
});
```

```
$("#dlg").dialog("open");
```

```

$('#dlg').dialog('center');
}
function dlg_ok(w,h,t,content,fun_ok,icon){
if(icon)$("#dlg").html(
+icon+"\>
"+content);
else$("#dlg").html(content);
if(w>800)w=800;
elsew='auto';
if(h<400)h='auto';
$('#dlg').dialog({
width:w, height:h,title:t,
iconCls: 'icon-tip',resizable: false, modal: true, closed: true,
buttons: [{
text:'确认',iconCls:'icon-ok',
handler:function(){
$('#dlg').dialog('close');
if(fun_ok)fun_ok();
}
}
});
$("#dlg").dialog("open");
$('#dlg').dialog('center');
}

```

script>

用户列表（查询+删除）

userlist.html页面中用easyui datagrid显示列表数据，加载后端封装传输过来的分页json数据时，分析数据后显示“修改”、“删除”等操作信息；在页面的输入框中输入待查询的关键字（考号或姓名）可以进行模糊查询；可以删除指定的用户，删除之前必须提示确认；可以请求进入添加用户页面；可以请求进入修改指定用户的页面。

```

<table id="dg" cellpadding="2">table>
<div id="tb" style="padding:5px;">
<input id="s_name" class="easyui-textbox" data-options="prompt:'考号或姓名...' style="width:200px;height:32px">
<a id="s_user" href="#" class="easyui-linkbutton" data-options="iconCls:'icon-search'">查询a>
<a id="a_user" href="#" class="easyui-linkbutton" data-options="iconCls:'icon-add'">添加a>
div>
<script type="text/javascript">
var s_name="",uid="";
function loadGrid(){
s_name=$("#s_name").val();
$("#dg").datagrid({
width:800,height:500,nowrap:false,
striped:true,border:true,collapsible:false,
url:"${ctxPath}/user/listUser",
queryParams:{ "s_name":s_name},

```

```

pagination:true,
rownumbers:true,
fitColumns:true,pageSize:20,
loadMsg:'数据加载中...' ,
columns:[[
{title:'学院', field:'department',width:100},
{title:'考号', field:'uid',width:80},
{title:'姓名', field:'xm',width:80},
{title:'操作', field:'role',width:100, formatter: function(value,row,index){
var p="+row.uid+"\">修改";
if($ {me.uid}==row.uid)p+="";
else p+="|"+row.uid+"\">删除";
returnp;
}},
]],
rowStyler:function(index, row){
if($ {me.uid}==row.uid){ return 'background-color:#FFA042;font-weight:bold';}
},
toolbar:'#tb'
});
}

functioneditUser(uid){
parent.$("#mainframe").attr("src","${ctxPath}/user/goEdit?uid="+uid);
var tab=parent.$("#tabs").tabs("getSelected");
parent.$("#tabs").tabs("update",{tab:tab,options:{title:"修改考生信息"}});
}

functiondelUser(ph){
uid=ph;
parent.dlg_h(300,140,"确认删除？","您确认要删除："+ph+"吗？",dlg_ok,null,"question");
}

functiondlg_ok(){
$.ajax({
url:"${ctxPath}/user/doDel?uid="+uid,
type:"post",
success:function(res){
if(res=="true"){
parent.dlg_h(300,140,"删除成功","您已删除考生："+uid,null,null,"info");
uid="";s_name="";
loadGrid();
} else {
parent.dlg_h(300,140,"删除失败",res,null,null,"error");
}
return false;
},

```

```

error:function(res){
parent.dlg_h(300,140,"系统提示","系统错误",null,null,"error");
}
})
}
$(function(){
loadGrid();
$("#s_user").click(function(){
s_name=$("#s_name").val();
loadGrid();
});
$("#a_user").click(function(){
parent.$("#mainframe").attr("src","${ctxPath}/user/goAdd");
var tab=parent.$("#tabs").tabs("getSelected");
parent.$("#tabs").tabs("update",{tab:tab,options:{title:"添加考生信息"}});
});
$("#tb").bind("keydown",function(e){
var theEvent = e || window.event;// 兼容FF和IE和Opera
var code = theEvent.keyCode || theEvent.which || theEvent.charCode;
if (code == 13) { //回车执行查询
$("#s_user").click();
}
});
})

```

script>

用户添加

useradd.html页面中的表单采用了自动校验form('enableValidation').form('validate')，如果标记为*required:true*的项没填写，则在保存时会自动提示，并用不一样的颜色提醒；设定登录密码时，需要输入两次，以确认输入密码的一致性。保存成功后或者返回，则自动跳转到用户列表页。

```

<div class="easyui-panel" style="width:580px;padding:30px 60px">
<form id="adduser_form" method="post">
<div style="margin-bottom:20px">
<input id="department" name="department" class="easyui-textbox" data-options="prompt:'班级',required:true" style="width:100%;height:32px">
div>
<div style="margin-bottom:20px">
<input name="xm" class="easyui-textbox" data-options="prompt:'姓名',required:true" style="width:100%;height:32px">
div>
<div style="margin-bottom:20px">
<input id="uid" name="uid" class="easyui-textbox" data-options="prompt:'考号',required:true" style="width:100%;height:32px;">
div>
<div style="margin-bottom:20px">

```

```

<div>登录密码:div>
<inputname="pwd"id="pwd"class="easyui-validateboxeasyui-textbox"type="password" data-options="{iconCls:'icon-lock'}",required:true"style="width:100%;height:32px" >
div>
<divstyle="margin-bottom:20px">
<div>确认密码:div>
<inputid="pwd2"validType="equals['#pwd']"class="easyui-validatebox easyui-textbox"type="password" data-options="{iconCls:'icon-lock'}"style="width:100%;height:32px" >
div>
<divstyle="margin-bottom:20px">
<div>角色:div>
<inputtype="radio"name="role"value="5"checked="checked"/>考生
<inputtype="radio"name="role"value="1"/>管理员
div>
form>
<divstyle="text-align:center;">
<a id="saveuser"href="#"class="easyui-linkbutton"iconCls="icon-ok"style="width:132;height:32px">保存a>
<a id="back"href="#"class="easyui-linkbutton"iconCls="icon-back"style="width:132;height:32px">返回a>
div>
div>
<script>
varuid;
varopened_node=null;
$(function(){
$('#saveuser').click(function(){
if(! ($("#adduser_form").form('enableValidation').form('validate'))){return;
varp=$('#pwd').val(), p2=$('#pwd2').val();
if(p!=p2){
parent.dlg_h(300,120,"校验失败！","密码不一致！",null,null,"warning");
return;
}
uid=$("#uid").textbox('getValue');
jQuery.ajax({
type: 'POST',
url :"$ {ctxPath}/user/saveAdd",
data : ($("#adduser_form").serialize(),
success :function (res) {
if(res=="true"){
parent.dlg_h(300,140,"添加成功！","你已添加用户:"+uid,dlg_close,null,"info");
} else{
parent.dlg_h(300,140,"添加失败！","添加失败！",null,null,"error");
}
return false;
},

```



```

error: function(res) {parent.dlg_h(300,140,"系统提示","系统错误！",null,null,"error");}
});
});
$('#back').click(function() {dlg_close();});
})
functiondlg_close(){
parent.$("#mainframe").attr("src","${ctxPath}/user/goList");
vartab = parent.$("#tabs").tabs("getSelected");
var title1="用户信息";
parent.$("#tabs").tabs("update", {tab:tab,options: {title: title1}});
}
script>

```

用户修改

修改用户信息的页面**usereedit.html**与添加新用户的页面内容基本相同，不相同的是，需要加载指定用户的信息到页面中相应的位置；修改后保存时，提交数据的路径不一样。html的代码中，使用了beetl语法，如：*`${user.uid}`*，尤其是在判断用户角色时，使用了if结构。

```

<divclass="easyui-panel"style="width:580px;padding:30px 60px">
<formid="edituser_form"method="post"action="#">
<divstyle="margin-bottom:20px">
<divstyle="margin-bottom:20px">
<inputid="uid"name="uid"value="${user.uid}"readonly="readonly"class="easyui-textbox"data-options="prompt:'考
号',required:true"style="width:100%;height:32px;">
div>
<divstyle="margin-bottom:20px">
<inputid="department"name="department"value="${user.department}"class="easyui-textbox"data-
options="prompt:'部门',required:true"style="width:100%;height:32px;">
div>
<divstyle="margin-bottom:20px">
<inputname="xm"value="${user.xm}"class="easyui-textbox"data-options="prompt:'姓
名',required:true"style="width:100%;height:32px">
div>
<divstyle="margin-bottom:20px">
<div>登录密码(不修改密码请留空):div>
<inputname="pwd"id="pwd"class="easyui-validateboxeasyui-textbox"type="password"data-options="icons:
[{iconCls:'icon-lock'}]"style="width:100%;height:32px" >
div>
<divstyle="margin-bottom:20px">
<div>确认密码:div>
<inputid="pwd2"validType="equals['#pwd']"class="easyui-validatebox easyui-textbox"type="password"data-
options="icons:[{iconCls:'icon-lock'}]"style="width:100%;height:32px" >
div>
<divstyle="margin-bottom:20px">
<div>角色:div>
<inputtype="radio"name="role"value="5"

```

```

<% if (user.role=='5'){ %>
checked="checked"
<% } %>
/>考生

<input type="radio" name="role" value="1"
<% if (user.role=='1'){ %>
checked="checked"
<% } %>
/>管理员

div>
form>
<div style="text-align:center;">
<a id="saveuser" href="#" class="easyui-linkbutton" iconCls="icon-ok" style="width:132px; height:32px">保存a>
<a id="back" href="#" class="easyui-linkbutton" iconCls="icon-back" style="width:132px; height:32px">返回a>
div>
div>
<script>
var uid="${user.uid}";
$(function(){
$('#saveuser').click(function(){
//提交前先校验
if(! ($("#edituser_form").form('enableValidation').form('validate'))){return;
var p=$('#pwd').val(), p2=$('#pwd2').val();
if(p!=p2){
parent.dlg_h(300,120,"校验失败！","密码不一致！",null,null,"warning");
return;
}
jQuery.ajax({
type: 'POST',
url: "${ctxPath}/user/saveEdit",
data: ($("#edituser_form").serialize(),
success: function (res) {
if(res=="true"){
parent.dlg_h(300,140,"修改成功！","你已修改用户:"+uid,dlg_close,null,"info");
} else {
parent.dlg_h(300,140,"修改失败！","修改失败！",null,null,"error");
}
return false;
},
error: function (res) {parent.dlg_h(300,140,"系统提示","系统错误！",null,null,"error");}
});
$('#back').click(function(){dlg_close();});
})

```

```
function dlg_close(){
parent.$("#mainframe").attr("src","${ctxPath}/user/goList");
var tab = parent.$("#tabs").tabs("getSelected");
var title1="用户信息";
parent.$("#tabs").tabs("update", {tab:tab,options: {title: title1}});
}
script>
```

视频5—用户管理

考核任务4

正常实现用户信息的添加、修改、删除、查询，无中文乱码

提高篇

添加数据库表News，生成相应pojo

新闻信息处理（后端）

创建第2个Nutz模块NewsModule，实现对考试新闻的增删改查

新闻信息处理（前端1）

能够添加、删除、修改考试新闻信息的前端页面

新闻信息处理（前端2）

呈现考试新闻信息的前端页面

考核任务5

正常实现新闻信息的添加、修改、删除、查询，无中文乱码

添加数据库表Tik，生成相应pojo

题库信息处理（后端）

创建第3个Nutz模块TikModule，实现对题库中题目的增删改查

题库信息处理（前端）

能够添加、删除、修改题库信息的前端页面

考核任务6

正常实现题库信息的添加、修改、删除、查询，无中文乱码

添加数据库表userpk，生成相应pojo

用于存储考生抽题信息

添加数据库表scores，生成相应pojo

用于存储考生成绩及试卷文件链接

考试信息处理（后端）

创建第4个Nutz模块ExamModule，实现随机抽题算法、交卷评分、生成试卷

考核任务7

正确实现考试模块的方法

考试（前端）

呈现考试题目的前端页面，缓存考生已作答案

考核任务8

正确实现考试页面的方法

整体测试与部署

考核任务9

整个系统脱离MyEclipse调试环境运行；整个考试系统的各项功能都可以正常使用，可以抽题、发布考试，可以正常登录考试、交卷，可以正常评卷、生成试卷、查看成绩和试卷。

