启明星辰人力资源系统

登录、个人信息设计文档

版本：X.X

|  |  |  |  |
| --- | --- | --- | --- |
| **作者：** |  | **日期：** |  |
| **审批：** |  | **日期：** | 2016-09-13 |

目录

[简介 3](#_Toc461963323)

[1.1、目的 3](#_Toc461963324)

[1.2、相关注意事项 3](#_Toc461963325)

[项目数据库信息 3](#_Toc461963326)

[2.1、具备Mysql数据库条件 3](#_Toc461963327)

[2.2、具备静态页面条件 5](#_Toc461963328)

[项目的业务流程 6](#_Toc461963329)

[3.1、登录页面流程图 6](#_Toc461963330)

[3.2、修改信息页面流程图 7](#_Toc461963331)

[3.3、修改密码流程图 7](#_Toc461963332)

[项目开发步骤 8](#_Toc461963333)

[4.1、开发步骤提要 8](#_Toc461963334)

[4.2、类的命名设计 8](#_Toc461963335)

[4.2.1、目的 8](#_Toc461963336)

[4.2.2、该模块的类命名规定 8](#_Toc461963337)

[4.3、创建数据库对应实体类、接口、接口实现类等 9](#_Toc461963338)

[4.3.1、实体类的创建方式一 9](#_Toc461963339)

[4.3.2、实体类的创建方式二 9](#_Toc461963340)

[4.3.3、创建完成之后，如下图所示： 9](#_Toc461963341)

[4. 4、模块需要的方法开发说明 10](#_Toc461963342)

[4.4.1、开发功能及方法说明（注：只附注了相关的方法，并不是所有的方法） 10](#_Toc461963343)

[4.4.2、方法所需要的参数： 10](#_Toc461963344)

[4.5、结合.jsp页面功能流程 11](#_Toc461963345)

[4.5.1、登录界面： 11](#_Toc461963346)

[4.5.2、个人信息修改： 14](#_Toc461963347)

[4.5.3、密码修改界面： 15](#_Toc461963348)

# 简介

## 1.1、目的

本文档是[启明星辰人力资源管理系统详细设计文档](启明星辰人力资源管理系统详细设计文档.docx)的子文档，是对详细设计文档的补充和说明，通过本文档，我们可以规范的设计和开发当前模块——登录、个人信息模块。相关开发说明，详情请查阅[启明星辰人力资源管理系统详细设计文档](file:///C:\Users\Administrator\Desktop\启明星辰项目设计文档\文档规范\启明星辰人力资源管理系统详细设计文档.docx)。

## 1.2、相关注意事项

该模块与“系统管理”模块有相应的关联之处，在整合项目的时候，需要与做“系统管理”组员协同沟通。

1）、项目必须使用统一编码格式“utf-8”。

2）、必须使用SVN整合代码，每天提交、整合。

3）、数据库必须使用统一编码格式“utf-8”。

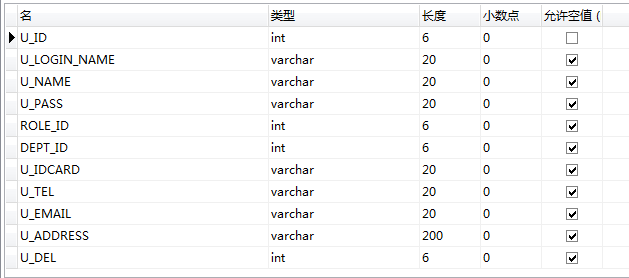
# 项目数据库信息

## 2.1、具备Mysql数据库条件

2.1.1、要有相应的用户表Users，用户表Users中有相应的字段信息，以及外键关联一个角色role\_Id和部门dept\_Id。可以通过Users、Role表中间表进行关联，也可以在Users表role\_id中关联了一个Role的Id。同样适用于Users和Deptment。

2.1.2、如下是Mysql数据库Users、Role、Deptment表图

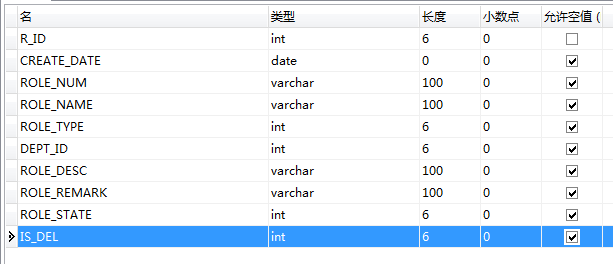
表一：Users表的结构



注：其中role\_id是Role表的主键字段，在Users表中role\_id是外键

其中dept\_id是Deptment表的主键字段，在Users表中dept\_id是外键

表二：Role表的结构



注：其中r\_id是Role表的主键字段。

其中dept\_id是Deptment表的主键字段，在Role表中dept\_id是外键。

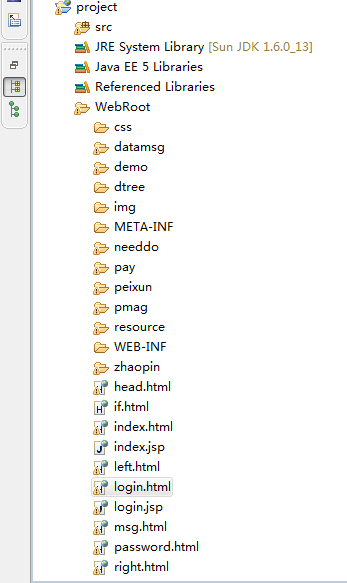
分清楚主键字段与外键字段所对应的关系，以及外键字段dept\_id是所属于哪个表

表三：Deptment表的结构

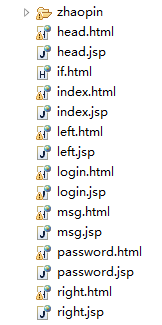


注：如图所示的d\_pid字段是表示该部门所属于某个部门的id，也就是说该字段的值是取决于上图中的d\_id，但是可以为空或者规定某一个值（如：0）不属于任何一个部门，因为最高级部门应该没有上级。

## 2.2、具备静态页面条件



2.2.1登陆、个人信息设计所涉及的页面主要是上面用方框框定的页面。在了解了静态页面的情况下，我们可以参考[启明星辰人力资源管理系统详细设计文档](file:///C:\Users\Administrator\Desktop\启明星辰项目设计文档\文档规范\启明星辰人力资源管理系统详细设计文档.docx)中将静态页面转换为.jsp页面的方法，转为.jsp页面。如下图所示：



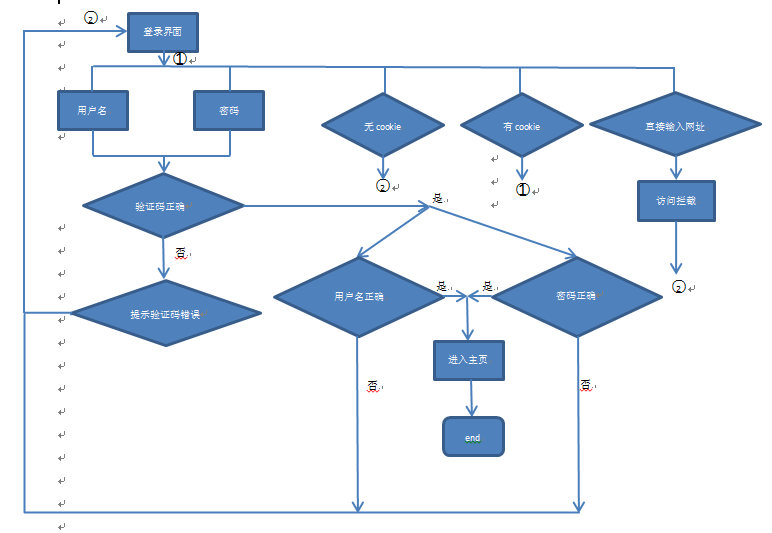
2.2.2、转换完成之后，如上图所示，方框中框定的.jsp页面就是本模块应该具备的。

2.2.3、在具备了上述的项目框架、数据库表的条件之后，这个项目应当准备的前期

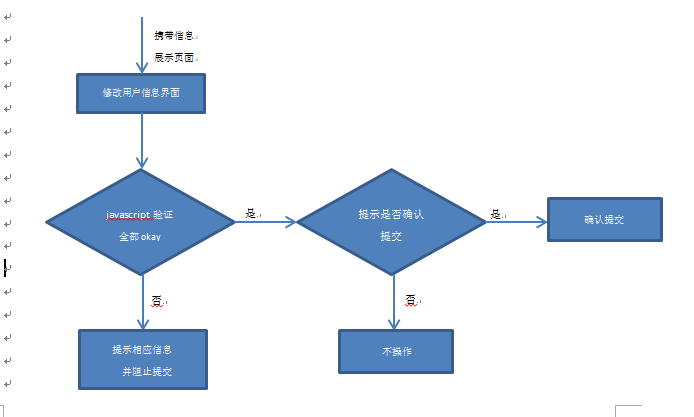
工作算是完成了。至此，我们才可以进行下一步的操作。

# 项目的业务流程

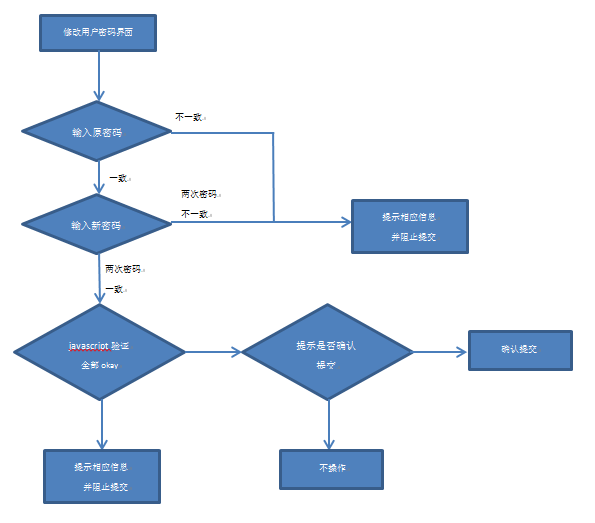
## 3.1、登录页面流程图



## 3.2、修改信息页面流程图



## 3.3、修改密码流程图



# 项目开发步骤

## 4.1、开发步骤提要

4.1.1、在框架分层搭建完成、jar包导入、所有应具备的条件的情况下，可以进行下一步项目的开发。如果不了解应该具备的条件，请参考[启明星辰人力资源管理系统详细设计文档](file:///C:\\Users\\Administrator\\Desktop\\启明星辰项目设计文档\\文档规范\\启明星辰人力资源管理系统详细设计文档.docx)中的环境搭建模块。

## 4.2、类的命名设计

### 4.2.1、目的

类的命名设计主要是为了统一设计实体类、接口、接口实现类名称，预防在项目组中每个人员各自创建实体类、接口类、接口实现类的时候出现重复的命名情况。

### 4.2.2、该模块的类命名规定

|  |  |  |  |
| --- | --- | --- | --- |
| 数据库表 | USERS | ROLE | DEPTMENT |
| 实体类 | User、UserBean、UserPo | Role | Deptment |
| 业务层接口 | UserService | RoleService | DeptService |
| 业务层接口实现类 | UserServiceImpl | RoleServiceImpl | DeptServiceImpl |
| Dao层接口 | UserDao | RoleDao | DeptDao |
| Dao层接口实现类 | UserDaoImpl | RoleDaoImpl | DeptDaoImpl |
| 控制层Servlet | UserServlet、LoginServlet | RoleServlet | DeptServlet |

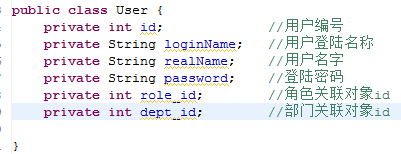
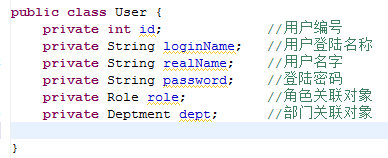
注：上表中的Users实体类可以使用三种命名方式，而控制层User则规定使用两个Servlet，UserServlet一个是用来处理与用户有关的操作，而LoginServlet则只是用来处理登陆与注销登陆的操作。

需要说明的是，该类命名规定默认采用的是驼峰式的规范，也可以用其他命名规范，请自行决定。但是类的首字母必须是大写的字母。

## 4.3、创建数据库对应实体类、接口、接口实现类等

### 4.3.1、实体类的创建方式一

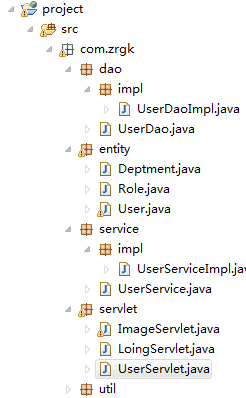
手动创建一个类，然后填写相应的属性名称，命名方式采用JAVA所规定的命名规范。在关联了外键关系的数据库表字段时，建议采用关联对象的形式创建关联对象。而不是简单的使用外键关联字段，这样后期使用起来更加方便。如下图所示：

### 4.3.2、实体类的创建方式二

采用框架结构中的逆向工程，如果是第二阶段的项目，建议不采用这种方式。

### 4.3.3、创建完成之后，如下图所示：



接口

控制层

验证码

实体类

接口实现类

注：绿色方框中的ImageServlet.java主要是用来生成验证码的类。

暂时只需要使用一个User表所对应的控制层、接口和接口实现类。

## 4. 4、模块需要的方法开发说明

### 4.4.1、开发功能及方法说明（注：只附注了相关的方法，并不是所有的方法）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能 类 | Servlet | 控制层方法 | 业务层接口方法 | 业务层实现类方法 | Dao层接口方法 | Dao层实现类方法 |
| 验证码实现及验证 | ImageServlet | imageCode | imageCode | 无 | 无 | 无 |
| 用户名密码验证 | LoginServlet | loginIn | loginIn | loginIn | loginIn | loginIn |
| 用户注销 | LoginServlet | loginOut | loginOut | 无 | 无 | 无 |
| 角色菜单展示 | UserServlet | userRoleInfo | userRoleInfo | userRoleInfo | userRoleInfo | userRoleInfo |
| 用户头像的显示 | UserServlet | loginIn | loginIn | loginIn | loginIn | loginIn |
| 数据信息的展示 | UserServlet | loginIn | loginIn | loginIn | loginIn | loginIn |
| 修改密码 | UserServlet | updatePass | updatePass | updatePass | updatePass | updatePass |
| 修改用户信息 | UserServlet | updateInfo | updateInfo | updateInfo | updateInfo | updateInfo |

### 4.4.2、方法所需要的参数：

方法的参数，只规定控制层的参数，其他层的参数由开发人员选择。除去request对象和response对象参数。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 控制层方法 | 访问控制修饰符 | 返回值 | 参数一 | 参数二 |
| imageCode | public或者(private) | void | 无 | 无 |
| loginIn | public或者(private) | void | username | password |
| loginOut | public或者(private) | void | 无 | 无 |
| userRoleInfo | public或者(private) | Role | userId | 无 |
| updatePass | public或者(private) | void | userId | password(新) |
| updateInfo | public或者(private) | void | user对象 | 无 |

1、Servlet中的方法

doGet、doPost方法除外

1）、LoginServlet中的方法：

登陆方法：private void loginIn(String username, String password){};

注销方法：private void loginOut(){};

2）、UserServlet中的方法：

获取用户信息的方法：private Role userRoleInfo (int userId){};

修改用户密码的方法：private void updatePass(int userId, String password){};

获取用户信息的方法：private User getUserInfoById(int userId){};

修改用户信息的方法：private void updateInfo(User user){};

2、业务层中的方法（只定义接口）

1）、UserService接口中的方法：

获取用户角色的方法：private Role userRoleInfo (int userId){};

修改用户密码的方法：private int updatePass(int userId, String password){};

获取用户信息的方法：private User getUserInfoById(int userId){};

修改用户信息的方法：private int updateInfo(User user){};

登录用户的方法：private User loginIn(String username, String password){};

3、Dao层中的方法（只定义接口）

1）、UserDao接口中的方法：

获取用户角色的方法：private Role userRoleInfo (int userId){};

修改用户密码的方法：private int updatePass(int userId, String password){};

获取用户信息的方法：private User getUserInfoById(int userId){};

修改用户信息的方法：private int updateInfo(User user){};

登录用户的方法：private User loginIn(String username, String password){};

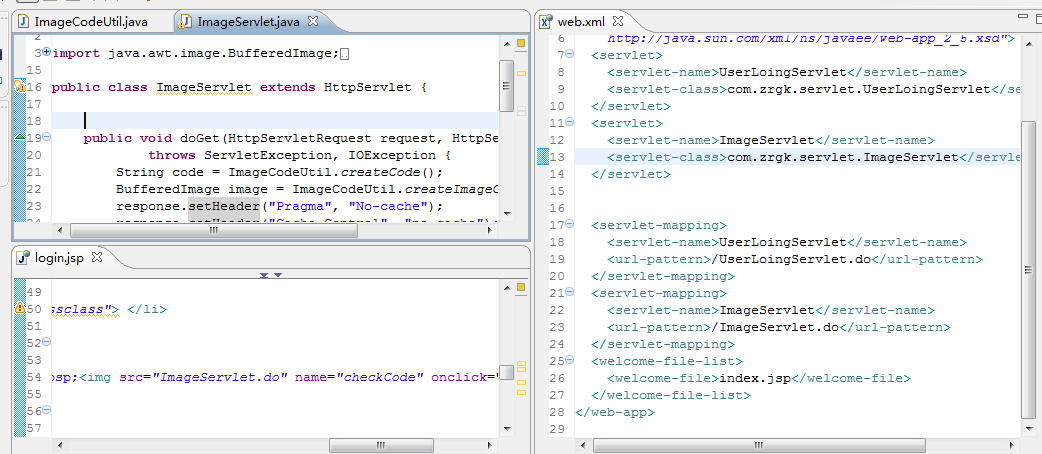
## 4.5、结合.jsp页面功能流程

### 4.5.1、登录界面：



1）、登录界面中，应该有验证码和自动登录的功能。

自动登录的Cookie默认是不保存，验证码是从后台调用ImageSerlvet中的方法，获取到的。相应的技术可以参考老师的课堂笔记。



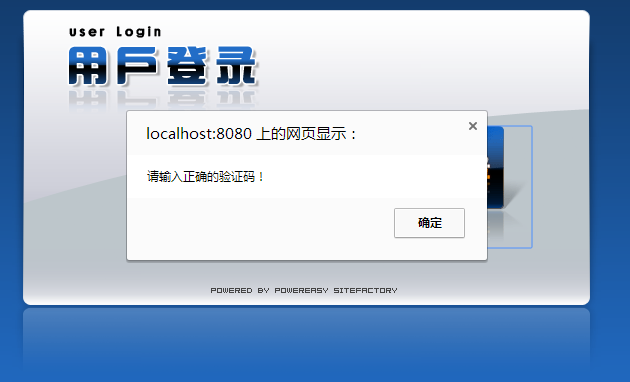
图片验证的相应执行流程就是如上图所示。

2）、要实现的功能：

1、要实现登录拦截的功能，二期项目使用的是过滤器来拦截没有登录的用户访问某些页面，三期项目使用的是拦截器来拦截没有登录的用户。当url路径中输入未登录没有权限的页面时，应该返回到login.jsp页面当中来。如下图：



2、当输入错误的验证码，无论输入正确或者错误的用户名和密码，都首先应该提示，验证码错误，并刷新验证码。如图：（可以采用如下两种方式）

3、当输入错误的用户名和密码，输入正确的验证码的时候，当点击登录的时候，应该出现用户名密码错误的提示。如图，（可以采用如下两种方式）

4、输入正确的用户名密码和验证码之后，就可以登陆到后台主页了。登录后界面如下



维护信息

用户头像

用户信息

用户登录名

注：上图红框中的信息都是数据库中的信息，应该展示在当前页面当中。

用户头像的路径地址，应该放在数据库当中，实际的文件存储是将文件存储到服务器的头像文件夹下。

3）、如下图所示，是本文档所涉及的主页模块：



pupdate

loginOut

updatePass

都是密码修改

接下来是对每个模块的介绍。

### 4.5.2、个人信息修改：

1）、在个人信息的修改页面应该携带当前登录人的登录信息，并展示出来，如下图所示：



不为空

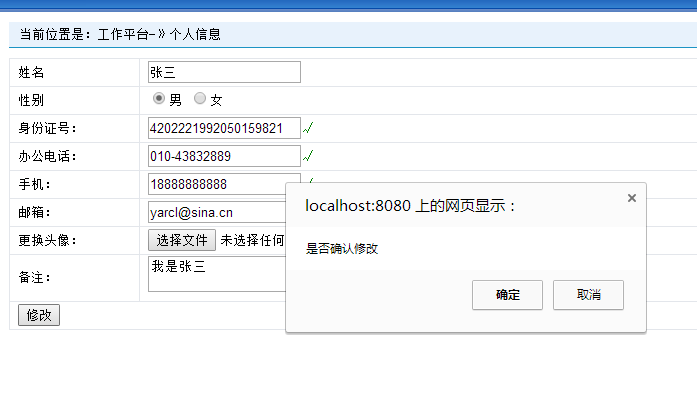
可为空

注：规定身份证和性别不允许修改，或者身份证选定的时候，自动选择性别。

应当有相应的js验证，包括姓名、身份证、手机、邮箱、电话的验证。其他验证自己选择。

当验证如上有错误的时候，点击修改的时候，不允许提交信息。

2）、当如下正确情况的时候，才可以点击提交。提交之前应该弹出“是否确认修改提示框”，如下图所示：

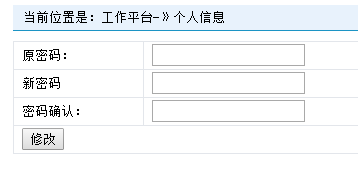
 

3）、“返回”链接，如下图所示：

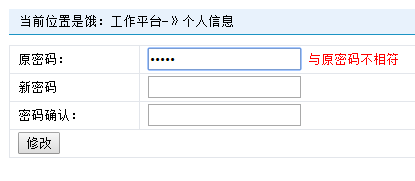
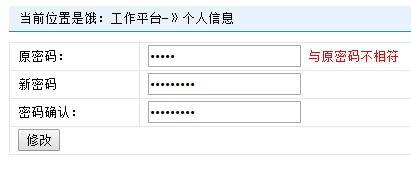


点击这个链接的时候，我们应该返回到进入当前页面之前的那个页面，而不是该页面所示的主页面。

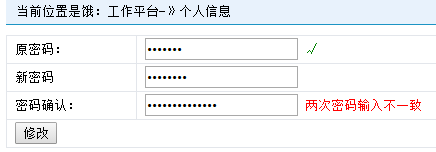
### 4.5.3、密码修改界面：



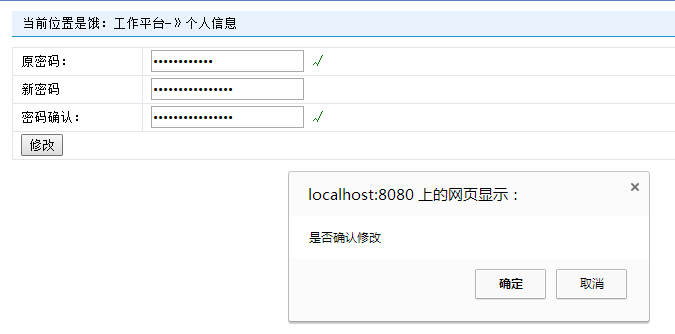
注：在输入的原密码与实际的密码不相符的时候，应该“出现与原密码不符的情况”，或者点击“修改”之后，出现“与原密码不相符”提示，并不允许提交。

当输入正确的原密码，但是输入两次不一样的新密码的时候，应该提示“两次输入密码不一致”提示，并不能提交。如下图所示：

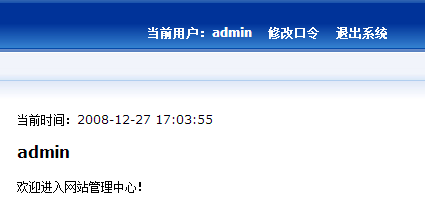


只有当原密码输入正确、两次新密码输入正确之后，才允许提交，并弹出友好提示框，提示“是否确认修改”。



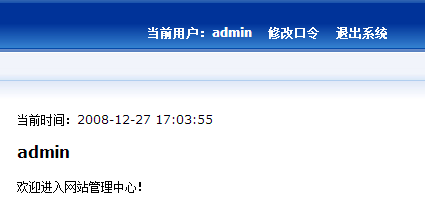
修改成功之后，应该退出登录，并销毁session，返回登录界面。

3、下图所示的“密码修改”：



操作请参考上面的密码修改。

4、退出系统，如下图所示。



应当进入LoginServlet，调用loginOut方法，并销毁session对象。