Java web开发规范

# 1、前言

为了使软件开发过程有章可循，保证软件质量，加强开发管理。

# 2、开发管理

无

# 3、命名规范

## 3.1 项目编号命名规范

项目简称pilottery ，纸质即开票

## 3.2 文档命名规范

命名方式：项目英文\_阶段\_功能\_版本

（pilottery\_）单元测试 （目录）新增用户V1.doc.

## 3.3 jsp/html命名规范

a.普通页面命名方式

　 命名格式：（模块：目录组织）\_对象\_动词（list,delete,add,modify,search），

例如：库存管理新增仓库 （stockmanage）stock\_add.jsp。

b.含frame页面

<frameset>中<frame>的name属性命名的格式是①xxx\_②xxx\_③xxx

①xxx部分用来标识当前页面隶属于整个系统中的哪一功能模块。

如：属于ebwebmail模块则写ebwebmail，其它情况依次类推。

②xxx部分标识当前页面所要完成的功能。

如：完成用户登录的功能则被标识为login，其它情况依次类推。

③xxx部分用来用来表示页面在浏览器窗口所处的位置。

处于浏览器窗口的顶部则标识为top，其它情况依次类推。

例如：ebwebmail\_inbox\_top.jsp

<frame> src属性相应的文件名根据情况建议在原命名规范上用下划线加上所处窗口的位置。

2、javascript脚本文件命名方式：

脚本函数都以①xxx\_②xxx的方式命名。

①xxx对应页面隶属的模块。

②xxx表示函数所要实现的功能(动宾结构)，多个单词用下划线连接。

例如：ebwebmail\_send\_mail()

模块通用的脚本函数必须集合于一个js文件中，在页面上通过<script language=”javascript” src=”url”></script>形式导入。js文件名命名使用模块名，例如：ebwebmail.js。

如果项目已经提供了公共js脚本，则优先使用公共js脚本中提供的函数。

所有定义方法的<script>元素定义在<head></head>中或</body>后。

3、javascript脚本内部变量与参数

全部采用驼峰式，例如： var fileSize。

<input type=”text” name=”userName”>

4、cookie命名

命名格式：模块名\_存储信息名词(多个单词用下划线分隔) ，全部大写，例如：EBWEBMAIL\_SORT\_TYPE。

5、window.open中name参数的命名

javascript的window.open方法中有一个name的参数，浏览器约定同样的名字的窗口只能打开一个，如果程序间名字重复将相互冲突。如果不限制打开窗口数，可以指定‘’或“”（不是null），否则需要加上模块名，例如ebwebmail\_viewmail。

# 4、文档规范

1. 如果需要摘引html页面中的内容，不能直接从html中粘贴过来（会在word文档中留下html格式），而必须先去除格式，例如：先粘贴到notepad中。
2. 务必注意中英文标点符号，文档正文一概使用中文标点符号。
3. 代码及其页面全部采用英文标点符号。

# 5、代码规范

## 5.1 Java源代码规范

### 5.1.1 命名

1、Package 的命名

Package 的名字应该都是由一个小写单词组成，例如：net.ebseries.modules。

此外，对于包名我们做如下约定：

1. 工具函数类包名前缀为.util

2. Servlet类包名前缀为.servlet

3. test case类包名前缀为.test

2、Class 的命名

　Class 的名字必须由大写字母开头而其他字母都小写的单词组成，例如：DataFile或InfoParser。

3、Class 变量的命名

　变量的名字必须用一个小写字母开头。后面的单词用大写字母开头，例如：debug 或 inputFileSize。

4、Static Final 变量的命名

　Static Final 变量的名字应该都大写，并且指出完整含义，例如：MAX\_UPLOAD\_FILE\_SIZE=1024。

5、参数的命名

　参数的名字必须和变量的命名规范一致。使用有意义的参数命名，如果可能的话，使用和要赋值的字段一样的名字：

　Void setSize(int size)

　{

　 this.size = size;

　}

6、数组的命名

　数组应该总是用下面的方式来命名：

　byte[] buffer;

　而不是：

　byte buffer[];

7、方法的命名

方法的命名以一个小写字母开头。后面的单词用大写字母开头，例如：debugMethod() 或 inputFile()。

### 5.1.2 代码格式

　文件头声明

　源文件的头部需要一个history段，对于每次对源文件的重大改动，都需要在history段中注明。该段定义在package和　　import之间，例如：

／\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* HISTORY

\*

\* 2015/09/01 Biz 创建文件

\*

\* 2015/09/01 kevin 增加XX功能

\*

\*2015/03/01 ben 增加xx功能

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import顺序

　import包按一下顺序：

　1. jdk标准包

　2. java扩展包（例如servlet，javamail，jce等）

　3. 使用的外部库的包（例如xml parser）

　4. 使用的项目的公共包

　5. 使用的模块的其他包

　每一类import后面加一个换行。

例如：

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.mail.\*;

import org.apache.xml.\*;

import net.ebseries.\*;

import net.ebseries.util.\*;

import net.ebseries.ebwebmail.\*;

代码块书写格式

　可以选择以下任意一种代码块的书写方式：

if (true){

　//body

}

或

if (true)

{

　//body

}

建议使用第二种书写方式。如果是修改他人的代码，必须使用代码原来的书写方式。

对于代码块过长，超过1屏以上，}后面要说明属于那个代码块，例如：

if (i > 100)

　{

　//too many lines more than one screen

　}// if (i > 100)

**操作符**

　操作符左右各用一个空格分隔。

　例如：

　int a = b;

　if (a > 0);

**类和方法定义（供参考）**

　类定义或方法定义过长需要换行书写，例如：

public class CounterSet

　extends Observable

　implements Cloneable

private PortletSet getPortlets( Portlets portlets,

　RunData rundata,

　boolean application,

　boolean applicationsOnly )

　{

　}

### 5.1.3 注释

　1、public 和 protected的成员变量和方法必须写javadoc注释。超过1句以上的注释使用中文书写。对于代码多于10行的private方法也要写javadoc注释。对外的接口也要注释，同时说清楚返回值及其参数定义。

2、状态和枚举定义的属性必须提供注释，标注状态含义。

3、对于代码中的逻辑分支或循环条件需要书写注释，例如：

if (some condition)

　{

　//符合某个条件，应该这样处理

　}

　else

　{

　//否则应该那样处理

　}

4、复杂业务流程及其算法写注释。

##### 5.1.4 其他

　关于属性

　类中的属性不能定义为public变量直接存取，而是定义成protect或者private变量并编写get/set方法，作为继承父类属性就protect，否则private。 例如：

protect String myName;

public String getMyName()

{

　 return myName;

}

public void setMyName(String myName)

{

　 this.myName=myName;

}

##### 5.1.5 日志

日志：日常信息用debug，重要业务信息info，出错一定要用error

在Controller层编写

例如:

@RequestMapping(method = RequestMethod.GET)

public String doLogin(HttpServletRequest request,HttpServletResponse response) {

try {

User user = UserSession.getUser(request);

// 记录debug级别的信息

logger.debug("This is debug message.");

// 记录info级别的信息

logger.info("This is info message.");

if(user == null) {

return "login";

}

} catch (Exception e) {

// 记录error级别的信息

logger.error("This is error message." + e.getErrorMessage());

e.printStackTrace();

}

return "\_main/index";

}

## 5.2 jsp/html代码规范

1、jsp/html描述注释

　jsp/html页面顶部必须存在一个基本描述注释，包含功能描述、参数列表和历史修改信息，例如：

<%--

　/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

　\*功能：xxxxxx

　\*

　\*-修改记录（不做强制要求，建议复杂功能页面增加）

　\* 2015/04/05 Hafele 创建文件

　\* 2015/06/07 tmp 增加强制下载功能

　\*

　\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

　--%>

2、jsp头格式

　jsp头部一般需要遵循以下格式：

　<%@ page contentType="text/html;charset=utf-8" %>

　<%@ page import="java.io.\*" %> // jdk标准包

　<%@ page import="javax.mail.\*" %> // java扩展包

　<%@ page import="org.apache.xml.\*" %> //使用的外部库的包

　<%@ page import="com.sunrise..\*" %> //使用的项目的公共包

　<%@ page import=" com.sunrise.applications.\*" %> //使用的模块的其他包

　<%@ include file="some.jsp" %> //include其他的jsp

　<%

　response.setHeader("Pragma","No-cache");

　response.setHeader("Cache-Control","no-cache");

　response.setHeader("Expires","0");

　%> //一般jsp都需要防止缓存

2、html格式

　1. html头一般需要遵循以下格式：

　<head>

　<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

　<title>some title</title>

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

　<script language="javascript">

　//some javascript

　</script>

　</head>

注意：必须指定一个有意义的<title>，严禁出现“Untitled”或“未命名”之类的<title>。

2. 所有html标签使用小写

html语法校验

　所有的jsp/html页面需要能够使用DreamWeaver正确打开（即html语法正确，没有错误的标记）。

注释

　一般不使用html注释，除非是有必要让最终用户看到的内容。对于包含JSP代码的html块，必须使用JSP注释。对于没有必要的注释，在发行版本中必须移除。

form属于域的maxlength

　对于text类型的输入域，必须根据数据库字段的长度设置相应的maxlength，例如数据库类型是NVARCHAR2(64)，那么maxlength是32（因为中文浏览器对于中文也认为是一个字符）。

# 6、项目规范

## 6.1应用程序目录规范

### 6.1.1 Java源文件目录

|  |  |
| --- | --- |
| **目录名** | **说明** |
| src | （Java源代码） |
| |cls | 域名-公司 |
| |pilottery | 域名-项目名称 纸质即开票 |
| |common | 公用包 |
| |utils | 工具包 |
| |filter | 过滤器包 |
| |… | 其他公用包 |
| |modulename | 业务模块名称 |
| |controller | 控制器功能实现包 |
| |dao | Dao层访问类包 |
| |service | 业务实现类包 |
| |model | 实体类包 |
| |mapper | Mybaits xml包 |

### 6.1.2 Web应用程序目录规范

|  |  |
| --- | --- |
| **目录名** | **说明** |
| webroot | （Web页面源代码） |
| |images | 页面图片 |
| |js | 公共Javascript源代码，单独的放在模块下 |
| |css | 样式表文件 |
| |views | 试图层 |
| |模块 | 模块页面 |

# 7、其他规范

菜单模块功能编码规则如下：

菜单支持四级：每个级别两个数字。具体如下：

|  |  |
| --- | --- |
| 模块 | 编码 |
| 系统管理-菜单 | 01 00 00 00 |
| 用户管理-菜单 | 01 01 00 00 |
| 角色管理-菜单 | 01 02 00 00 |
| 新增角色-功能 | 01 02 00 00 01  （增加01 修改 02 删除 03 启用 04 禁用 05 授权 06 查询 99） |
| 修改密码-菜单-功能 | 01 03 00 00 （01） |
| 数据维护 | 02 00 00 00 |
| 销售管理 | 03 00 00 00 |
| 物品库存 | 04 00 00 00 |
| 彩票库存 | 05 00 00 00 |
| 兑奖管理 | 06 00 00 00 |
| 资金结算 | 07 00 00 00 |
| 市场专员 | 08 00 00 00 |

