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

项目进度周报表格如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目编号 | 汇报人 | 汇报日期 | 项目进度详细描述 | 其他问题 |
|  |  |  |  |  |

# 3、命名规范

## 3.1 项目编号命名规范

命名方式：项目英文/中午拼音名称\_开发组编号\_序列号

序列号由3为数字组成，不足的用’0’补齐。

## 3.2 文档命名规范

命名方式：项目英文/中午拼音名称\_文档名称\_序列号\_编写人名称/编号

## 3.3 jsp/html命名规范

1、jsp与html文件名全部小写，并遵循如下的规范：

a.数据/内容显示页:

名词，多个单词用下划线分隔，要求能说明显示内容的信息，为避免冲突，可加上 “\_list”。 例如：

new\_message.html 或 my\_file\_list.jsp

b.操作处理页

　 命名格式：名词\_下划线\_动词，例如：file\_delete.jsp。

c.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

单词之间用下划线分隔且全部小写，例如： var file\_size。

<form>表单name属性

　 统一以“form\_”开头，其后加该表单所需收集的信息的作用或动作，例如：form\_file\_upload 或 form\_send\_mail。

表单elements

表单element的名称以element需收集的信息标示命名（与struts中的属性名对应），其他element单词之间使用下划线分隔且全部小写，例如：

<input type=”text” name=”username”> -username为struts中的属性名username

<input type=”radio” name=”file\_type”> -不与struts对应表单元素名

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。

## 3.4数据库命名规范

数据库表命名均遵循以下规范：

模块名\_存储信息名词(多个单词用下划线分隔)，全部小写，例如：ebwebmail\_message。

数据库字段命名遵循以下规范：

存储信息名词(多个单词用下划线分隔)，全部小写，例如：message\_id。

# 4、文档规范

1. 编写文档目前主要使用的工具是Word（项目计划文档例外，需要使用Project制作），辅助使用的工具有Visio，PowerPoint，Excel等。
2. 文档务必保持段落格式整齐，文字字体，颜色，大小统一。
3. 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。
4. 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。
5. 如果需要在文档中插图，不要使用word自带的绘图工具。可以选择使用PowerPoint或visio。使用PowerPoint，选择“插入”->“对象”->“Microsoft PowerPoint 幻灯片”。使用visio，需要另外建立visio文档，绘制完以后粘贴到word中。
6. 数据建模使用powerdesigner工具，文档中仅需要粘贴数据模型的逻辑视图（logical view），另外还要附上表结构描述。
7. 修改他人文档务必使用修订模式，以便保留备修改的内容。使用修订模式，选择“工具”->“修订”->“突出显示修订”，勾上“编辑时标记修订”。

附：一些模版

Java类设计文档(class documentation)

声明：

描述：

创建人：

创建时间：

属性：1. property\_name

声明：

说明 ：

创建人：

创建时间：

方法： 1. method\_name

声明 ：

说明 ：

参数 ：

返回值 ：

异常：

创建人：

创建时间：

如需修改加上：修改人：修改时间：修改原因：三个描述信息

注：

　方法声明中，如果遇到重载的方法，可以使用“|”表示可以接受不同类型的参数，例如：“String value | int value”；可以使用“[]”表示可选的参数，例如：“Object message[,Throwable t]”

Servlet设计(servlet document)

　类名 接受参数列表 描述

数据表设计(data modeling)

　表名：some\_table

说明：description

表结构：

　字段名 类型 说明

# 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：

　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

\*

\* 2002/01/29 Biz 创建文件

\*

\* 2002/02/19 kevin 增加XX功能

\*

\*2002/03/01 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)

关于缩进

缩进使用4个连续空格，不要在源文件中保存tab字符， 请注意调整所用的IDE工具，打开将tab转换为空格功能。

页宽

页宽应该设置为80字符. 源代码一般不会超过这个宽度, 并导致无法完整显示, 但这一设置也可以灵活调整. 在任何情况下, 超长的语句应该在一个逗号或者一个操作符后折行. 一条语句折行后, 应该比原来的语句再缩进4个空格。

操作符

　操作符左右各用一个空格分隔。

　例如：

　int a = b;

　if (a > 0);

SQL语句

　代码中书写的sql语句要求sql关键字全部大写，表名和字段名小写。例如：

　SELECT user\_id, name FROM account WHERE user\_id > ? AND depart = ? ORDER BY name

类和方法定义

　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

　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

　}

### 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变量并编写get/set方法，例如：

　protect String myName;

public String getMyName()

　{

　return myName;

　}

public void setMyName(String myName)

　{

　this.myName=myName;

　}

## 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：

<%--

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

　\*

　\* NAME : file\_download.jsp

　\*

　\* PURPOSE : 下载文件提示

　\*

　\* PARAMETERS :

　\* file\_id - 文件ID号

　\* force ?C 是否强制下载

　\*

　\* HISTORY

　\*

　\* 2002/04/05 Hafele 创建文件

　\*

　\* 2002/06/07 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=gb2312" %>

　<%@ 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=gb2312">

　<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

3. html页面一般需要设置一个背景色（一般是#FFFFFF）。

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VARCHAR(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） |
| |\_cn | 域名倒置 |
| |\_farmers | 域名 |
| |\_common | 公用包 |
| |\_util | 工具包 |
| |\_db | 数据库访问类 |
| |\_… | 其他公用包 |
| |\_action | 公用的action类包 |
| |\_模块名 | 模块功能实现包 |
| |\_dao | 数据库访问类包 |
| |\_biz | 模块业务实现类包 |
| |\_action | 模块专用action类包 |

### 6.1.2 Web应用程序目录规范

|  |  |
| --- | --- |
| **目录名** | **说明** |
| webpage | （Web页面源代码） |
| |\_images | 页面图片 |
| |\_js | Javascript源代码 |
| |\_css | 样式表文件 |
| |\_admin | 系统管理功能 |
| |\_模块名 | 系统管理模块功能实现页面 |
| |\_模块名 | 业务模块功能实现页面 |
| |\_web-inf | Web.xml等文件存放目录 |
| |\_classes | Java类class文件目录 |
| \*.properties | 配置文件 |
| |\_lib | 应用程序所需类包 |
| |\_config | 系统配置信息 |
|  |  |

# 7、其他规范

对于IDE的使用

目前暂不限制开发工具的使用，但是最后提交的代码必须不依赖任何IDE，而需要可以使用ant完成所有的编译工作。一般提交的代码目录格式如下：

|\_dist（存放输出的文件class文件）

　|\_lib（使用的库）

　|\_src（源代码）

　|\_docs（文档）

　|\_wwwroot（页面文件）

　|\_build.xml（ant的build文件）

　|\_changes.log（代码版本和修改的日志）

此条内容，暂时不执行。

限制session的使用

在代码中使用session需要听取项目经理的意见，项目经理需要在设计文档中登记项目中所有使用到的session的名字和作用。

限制外部包的使用

开发员如果需要使用一个外部包需要听取项目经理的意见。在项目经理批准以前，严禁擅自使用一个外部的包。