**Java程序开发规范**

**<V1.0>**

作者：常育新

**变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **变化状态** | **修改点说明** | **变更人** | **时间** |
| V1.0 | C | 创建文档 | 常育新 | 2016-10-28 |
|  |  |  |  |  |

\*变化状态：C――创建，A——增加，M——修改，D——删除

**目 录**

[1. 文档介绍 4](#_Toc439333104)

[**1.1.** 编写目的 4](#_Toc439333105)

[**1.2.** 适用范围 4](#_Toc439333106)

[**1.3.** 定义 4](#_Toc439333107)

[2. 基本规范 5](#_Toc439333108)

[3. 目录规范 5](#_Toc439333109)

[**3.1.** Maven目录 5](#_Toc439333110)

[4. 版本规范 6](#_Toc439333111)

[**4.1.** 命名规则 6](#_Toc439333112)

[**4.2.** 版本规则 6](#_Toc439333113)

[**4.3.** 版本的命名 6](#_Toc439333114)

[5. 部署规范 7](#_Toc439333115)

[6. 数据库规范 7](#_Toc439333116)

[7. 代码规范 7](#_Toc439333117)

[**7.1.** Java代码的格式要求 7](#_Toc439333118)

# 文档介绍

## 编写目的

本文编写Java项目规范的目的主要是为Java开发人员提供一个标准的、规范的方式进行代码的设计和编写。通过遵循此开发规范，以使开发人员养成良好的编码风格和习惯，形成开发小组编码约定，最终提高程序的可靠性、可读性、可修改性、可维护性和一致性等，增进团队间的交流，从而进一步提高软件开发团队的效率。

## 适用范围

本文档对所有采用Java作为开发语言的项目适用。

## 定义

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 开发服务器 | 开发人员用来调试程序的服务器 |
| 测试服务器 | 测试人员跟业务人员用来测试开发人员的实现是否符合业务需求的服务器 |
| 集成服务器 | 业务部署到生产服务器前用来测试业务的服务器，跟生产服务器数据一致 |
| 生产服务器 | 正式面向实际用户提供服务的服务器，通常以集群的方式呈现 |
| snapshot版 | 还处在开发过程中，可能变化很大的不稳定版本 |
| beta版 | 已经实现了全部业务需求，可以提供给测试人员跟业务人员测试的版本 |
| release版 | 已经通过所有测试，可以正式发布到集成服务器的版本 |
| service pack版 | 已经发布到生产服务器运行的release版本，发现有bug，修改bug以后的版本。 |
| jar包 | 使用Java语言开发的，打包形成的.jar文件 |
| pom.xml文件 | Maven管理的项目对象模型文件，用来描述一个项目的各个方面信息 |

# 基本规范

开发人员只在本机、开发服务器调试开发，不跟测试服务器、集成服务器、生产服务器发生交互。

开发人员工作使用统一开发工具：Eclipse、Maven、Tomcat、Jdk、SecureCRT、SSH Secure，版本有项目组制定，必须保证同一项目组的人员使用的开发环境是相同的。

搭建Maven私服Nexus统一管理依赖。

代码统一使用Maven管理，采用 UTF8编码。

服务器统一使用CentOS 6.5 64bit作为操作系统。

Jar包不允许双向互相引用，如果存在，需要重新划分jar包中java文件。

Jar包中java文件一律以"com.iss.gbg.项目名" 开头作为package名。

# 目录规范

## Maven目录

由于采用了maven来管理java项目，项目的目录需要遵循maven的要求，如下图所示：

|  |  |
| --- | --- |
| **Java工程** | **Web 工程目录** |
|  |  |

bin目录下放置Windows/Linux命令行脚本，项目的安装说明等。

docs目录下放置项目相关文档。

sql目录放置项目相关的sql脚本。

src目录下,放置项目源代码跟资源文件。

其中main目录放置项目实际源代码跟资源文件。

java目录放置java源文件。

resources目录放置资源文件。

webapp目录放置jsp等web项目文件。

test目录放置项目测试源代码跟资源文件。

java目录放置测试java源文件。

resources目录放置测试资源文件。

target目录放置执行maven各种命令生成的文件，自动生成。

pom.xml文件是maven项目的描述文件。

# 版本规范

## 命名规则

Jar包的命名，一律采用小写字母，为模块名-版本名.jar的命名方式。

## 版本规则

开发过程中存在4种版本，分别是:snapshot、beta、release、release service pack。

snapshot版本是一个项目开始时候的最初版本，表明项目还处在开发过程中，可能存在很大的变化。

beta是已经实现了全部业务需求，可以提供给测 试人员跟业务人员测试的版本。

release版是已经通过所有测试，可以正式发布到引导服务器的版本。

release service pack是已经发布到生产服务器运行的release版本，发现有bug，修改bug以后的版本

## 版本的命名

模块名-版本名-SNAPSHOT.jar

模块名-版本名-BETA.jar

模块名-版本名-RELEASE.jar

模块名-版本名—RELEASE-sp+number.jar

以mod\_cms3\_content模块，3.9.9版本为例，命名应该是：

mod\_cms3\_content-3.9.9-SNAPSHOT.jar

mod\_cms3\_content-3.9.9-BETA.jar

mod\_cms3\_content-3.9.9-RELEASE.jar

在release版发布以后，打了两次补丁的版本

mod\_cms3\_content-3.9.9-RELEASE-sp2.jar

对于同一个版本号的4种版本，存在固定的先后顺序， snapshot > beta > release > release service pack 也就是说当一个版本已经处于release版本状态，再做的修改，不能回退到snapshot、beta、release只能是新的版本号的snapshot、beta、release。

# 部署规范

程序版本分为四个版本状态，分别是snapshot、beta、release、release service pack版本，只有处于release跟release service pack状态的版本，才允许部署到集成服务器跟生产服务器。

部署之前流程控制部门要确认项目相关文档完备才允许开始部署。

部署首先是部署到集成服务器，只有在集成服务器测试运行没有问题以后，才允许部署到生产服务器，不允许越过集成服务器直接部署到生产服务器。

部署由专门的人员按照部署文档来操作，不允许开发人员未经批准私自部署。

部署之前必需对原有系统进行备份，不允许未经备份直接部署。

部署必需按照规定的时间点开始跟结束，不允许私自更改部署的时间。

# 数据库规范

数据库总的来说分为两部分，开发数据库跟线上数据库。开发人员在通常情况下，是不允许连接线上数据库进行开发的。

线上数据库是有版本的，从一个版本升级到下一个版本，由运维部门根据数据库文档进行。

数据库文档保存在svn服务器上，由运维部门来维护。

数据库文档包括以下几个部分：设计文件及相关文档，这部分由运维部门要求开发部门提供。

数据库相关脚本，包括当前版本的全部脚本、优化、调整的脚本，表、索引、约束、序列、存储过程等，所有文档应该有统一的版本标识，一致性保证由运维部门负责。

线上数据库表结构、索引、约束、序列、存储过程的增删改，由运维部门DBA进行操作。开发人员在特殊情况下，需要对线上数据库表数据进行增删改查的，由开发人员提出申请，申请通过后，由运维部门进行操作开通。

# 代码规范

## Java代码的格式要求

1. protected跟public类变量需要添加注释说明。
2. public类方法需要添加注释说明。
3. 必须导入类的完整路径，即不能使用\*导入所需的类。（对整个项目进行替换操作可以在项目文件夹右键菜单选择 Source > Organize Imports）。
4. 不导入使用sun.\*开头的类，在一个java文件里面不多次导入同一个类，不导入不使用的类。
5. 一个java文件最大行数不超过2000行，每行不超过120字符。
6. 一个方法最大行数不超过100行，匿名类不超过60行，检查方法和构造子参数的最大个数不超过7个，
7. 在发布版本中不包含main方法。
8. 不在程序中使用System.out.print、System.exit、printStackTrace，不抛出多于的异常。
9. 嵌套的层次不超过6层。
10. 定义一个常量时,使用大写的L来代替小写的l（原因是小写的l和数字1很象）

**详细规范参见《Java编码规范v0.1.docx》**