文档编号: TF-TS-K131029

文档密级: 3

文档版本: V1.0

**Java Web项目开发**

**技术规范文档**

.

****2013年12月30日

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本 | 修订章节 | 描述 | 作者 |
| 2013.10.29 | V0.1 | 初版模版 |  | 李建 |
| 2013.12.30 | V1.0 | 初版规范 |  | 唐梦岚 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[目录 I](#_Toc376178609)

[**第一章 引言** 1](#_Toc376178610)

[1.1 目的 1](#_Toc376178611)

[1.2 XXdsdss 1](#_Toc376178612)

[**第二章 总体框架** 2](#_Toc376178613)

[2.1 框架概述 2](#_Toc376178615)

[2.2 框架规范 2](#_Toc376178619)

[2.2.1 Struts2规范 2](#_Toc376178620)

[2.2.2 Hibernate规范 2](#_Toc376178621)

[2.2.3 Spring规范 3](#_Toc376178622)

[2.2.4 web配置文件规范 3](#_Toc376178623)

[**第三章 命名规范** 4](#_Toc376178624)

[3.1 项目编号命名规范 4](#_Toc376178626)

[3.2 文档命名规范 4](#_Toc376178627)

[3.3 Java文件名规范 4](#_Toc376178628)

[3.3.1 文件名后缀 4](#_Toc376178629)

[3.3.2 常用文件名 4](#_Toc376178630)

[3.4 JSP/Html文件命名规范 4](#_Toc376178631)

[3.4.1 JSP与html文件命名规范 4](#_Toc376178632)

[3.4.2 JavaScript脚本文件命名规范 4](#_Toc376178633)

[3.4.3 JavaScript脚本内部变量与参数 5](#_Toc376178634)

[3.4.4 Cookie命名规范 5](#_Toc376178635)

[3.4.5 Window.open中name参数的命名 5](#_Toc376178636)

[3.5 数据库命名规范 5](#_Toc376178637)

[3.5.1 数据库表命名 5](#_Toc376178638)

[3.5.2 数据库字段命名 5](#_Toc376178639)

[3.5.3 索引命名 5](#_Toc376178640)

[3.5.4 主键、外键命名 6](#_Toc376178641)

[3.5.5 Sql语句编写 6](#_Toc376178642)

[3.5.6 其他 7](#_Toc376178643)

[**第四章 文档规范** 8](#_Toc376178647)

[4.1 使用工具 8](#_Toc376178649)

[4.2 风格要求 8](#_Toc376178653)

[4.3 其他 8](#_Toc376178654)

[**第五章 代码规范** 9](#_Toc376178655)

[5.1 Java源代码规范 9](#_Toc376178657)

[5.1.1 命名规范 9](#_Toc376178658)

[5.1.2 编程规范 9](#_Toc376178659)

[5.1.3 语句规范 10](#_Toc376178660)

[5.1.4 注释规范 10](#_Toc376178661)

[5.1.5 缩进排版规范 10](#_Toc376178662)

[5.1.5 声明规范 11](#_Toc376178663)

[5.1.6 空白规范 11](#_Toc376178664)

[5.2 JSP/Html代码规范 11](#_Toc376178668)

[5.2.1 JSP/html描述注释 11](#_Toc376178669)

[5.2.2 JSP头格式 12](#_Toc376178670)

[5.2.3 Html格式 12](#_Toc376178671)

[**第六章 项目目录规范** 14](#_Toc376178672)

[6.1 Java源文件目录 14](#_Toc376178674)

[6.2 Web应用程序目录 14](#_Toc376178678)

[6.3 数据库文件目录 15](#_Toc376178679)

[6.4 资源文件目录 15](#_Toc376178680)

[**第七章 其他规范** 16](#_Toc376178681)

[7.1项目打包部署规范 16](#_Toc376178682)

**第一章 引言**

1.1 目的

随着公司的发展和规模的扩大，项目越来越多，本文档拟在引领技术部开发规范，缩短开发周期，进一步提高软件项目质量，以便于其他开发人员（包含新同事和继续开发人员）能快速读懂代码，以及为后期测试维护人员和维护人员降低难度，使得本技术部的开发能以统一、标准的、规范的方式设计和编码。通过建立开发规范，使得每个开发人员养成良好的开发风格和习惯；提高程序的可靠性、可读性、可维护性和一致性等，提高程序员的开发水平，并保证软件产品的质量。

本规范适用于本公司所有Java Web软件项目、产品等的设计、开发和维护等的工作，所有软件开发人员在整个软件开发过程中必须遵循此规范。

1.2 XXdsdss

151sd5f13sd1f23

Fsd

F

Ds

F

Sd

F

Ds

**第二章 总体框架**



2.1 框架概述



本公司Java Web开发的软件产品多为B/S结构，涉及多个层面的开发，为了便于业务逻辑与界面显示的分离以及项目的分工，开发的软件产品应当符合MVC设计模式，基础框架使用基于MVC的SSH(struts2+spring+hibernate)框架，其中struts2充当前端控制器，主要用于接收请求和处理请求；hibernate主要实现对数据库的访问；spring通过配置文件管理struts2和bibernate技术，改善程序结构，降低耦合度，提高系统的灵活性，便于扩展和维护。

2.2 框架规范

2.2.1 Struts2规范

Struts2框架主要完成MVC架构，该架构包括Model层、View层和Controller层。Model层主要是项目的业务逻辑类JAVA，View层主要就是视图文件JSP，Controller层主要就是Actoin类（Action类中调用Model层的相关类）。

其中，Model层包括包括基本的业务对象类的构建或者一些其他的通用功能类的构建；View层主要实现界面的显示工作，只负责界面接收用户输入的数据或者界面显示呈现给用户的数据；Controller层包括DAO和Service的接口类和实现类以及Action类，实现对业务逻辑的处理。

编写好这些相关类后，Struts2中另外的一部分重要内容就是对xml配置文件的配置。首先需要在WEB-INF下web.xml配置文件中配置struts2过滤器的配置以及相应的映射文件的配置，另外，就是对struts.xml文件的配置，包括对公共配置信息的配置和对其他一些配置文件的包含，例如对struts-xxx.xml配置文件的包含，而该文件中包含了对<action>的配置，即决定用户做完具体的操作后所要跳转的页面。

对于该框架的相关文件的存放位置、文件命名、编程规范等内容将在后面部分分类介绍。

2.2.2 Hibernate规范

Hibernate框架主要完成对数据库的访问，它的实现思想是将关系数据库中表的数据映射成为对象，以对象的形式展现，从而使得对数据库的操作转化为对这些对象的操作。

该框架在编程过程中首先需要配置数据库驱动，进而将数据库中的表自动生成相对应的对象java类以及对应的xxx.hbm.xml文件，然后修改Hibernate属性配置文件，添加一些必要的属性，如添加对数据库表对象生成的xxx.hbm.xml文件进行资源的映射属性等。

完成以上这些准备工作后，就可以进行通过DAO接口类和实现类以及service接口类和实现类进行对数据库的操作。

对于该框架的相关文件的存放位置、文件命名、编程规范等内容将在后面部分分类介绍。

2.2.3 Spring规范

Spring是对Struts2框架和Hibernate框架进行整合的框架。

Spring对于struts2框架的整合中，可以使得struts2通过使用Spring容器创建bean实例，Struts2的Action实例交给Spring框架的IOC容器进行装配和管理，要做到Spring对struts2对整合，首先，需要在web.xml文件中配置一个listener来加载spring配置文件的功能，另外，通过context-param指定特定的配置文件；然后，还需在struts.properties文件中添加struts.objectFactory = spring配置。

而在Spring对Hibernate的整合时，主要就需要在applicationContext.xml文件文件中配置数据库连接池、添加对Hibernate事务管理的配置，配置SessionFactory工厂，引入一些其他的一些基本类的配置文件（dao、service、action）等。

对于该框架的相关文件的存放位置、文件命名、编程规范等内容将在后面部分分类介绍。

2.2.4 web配置文件规范

Web.xml是整个web项目中非常重要的一部分，该文件对项目中的一些配置信息进行了初始化设置，如对welcome页面、servlet、filter、listener等进行了初始化配置，其中，在web.xml文件配置时，所有的UPLOAD-PATH都应该在该文件中定义，页面显示时都应该获取此路径作为url的开头部分，若此路径在webapp路径下，应以"/"开头，若在其他服务器，应写全路径，如<http://10.168.1.253:81/its_upload>。

**第三章 命名规范**



3.1 项目编号命名规范

命名方式：项目英文名/中文名拼音\_开发组编号\_序列号；

序列号由3位数字组成，不足用‘0’补齐

3.2 文档命名规范

命名方式：项目英文名/中文名拼音\_文档名称\_序列号\_编写人名称/编号

3.3 Java文件名规范

3.3.1 文件名后缀

Java源文件后缀为”.java”

Java字节码文件名后缀为”.class”

3.3.2 常用文件名

README：概述特定目录下所包含内容的文件的首选文件名，如Readme.txt

3.4 JSP/Html文件命名规范

3.4.1 JSP与html文件命名规范

文件名全部小写，并遵循如下规范：

1. 数据内容显示页:

文件名由英文名词组成，多个单词用下划线隔开，要求文件名能见名之意，能说明显示内容的信息。

1. 操作处理页

命名方式：名词\_动词,例如 user\_regist.jsp,其中，名词为处理的对象，动词为对应的操作。

1. 含有frame页面

<frame>中的name属性的命名格式为①xxx\_②xxx\_③xxx,其中,

①xxx部分用来说明当前页面隶属于整个系统的哪一功能模块，例如，属于用户模块则写成users；

②xxx部分用来说明当前页面所要完成的功能，例如，属于登录功能则写成login；

③xxx部分用来表示当前页面所在浏览器窗口所处的位置，例如，位于浏览器顶部则标识为top；

3.4.2 JavaScript脚本文件命名规范

脚本函数都已①xxx\_②xxx的方式命名，其中，

①xxx对应页面隶属的模块；

②xxx标识函数所要实现的功能，多个单词用下划线连接；

注意：

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，其中js文件名使用模块名命名。如果项目已经提供了公共的js脚本，则优先使用公共的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。

3.4.3 JavaScript脚本内部变量与参数

单词之间用下划线分隔开且全部小写，要求变量名要见名之意，代表所要标识的变量或者参数。

<form>表单name属性：

统一要以"form\_"开头，后面加该表单所需收集的信息的作用或动作，例如，form\_users\_update。

表单elements

表单element的名称以element所需收集的信息标识命名，若名称包含多个单词需用下划线分隔开，但必须所有单词都小写。

3.4.4 Cookie命名规范

命名方式：模块名\_存储信息名（多个单词用下划线分隔），所有单词全部大写，例如USER\_SORT\_TYPE。

3.4.5 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，否则需要加上模块名。

3.5 数据库命名规范

3.5.1 数据库表命名

命名方式：模块名\_存储信息名（多个单词用下划线分隔），且全部小写。表明尽量用全名，但表明应该限制在30个字符内，当表明过长时，可以用缩写来减少表明的长度，例如，users\_info。

3.5.2 数据库字段命名

命名方式：存储信息名，全部小写。字段名应该为有意义的单词或者单词的缩写，如果字段名由多个单词组成，则需在单词间用下划线分隔开；同样，字段名应该限制在30个字符之内，当字段名过长时可以用缩写来减少字符的长度。例如，users\_id。

3.5.3 索引命名

命名方式：IDX\_<table>\_<column>\_<column>，其中，table为建立索引的表名，column为建立索引的字段名（可以建立单个字段的唯一索引，也可以建立多个字段的复合索引）。同样，索引名应该限制在30个字符之内，当索引名过长时可以通过缩写来减少索引名的字符长度。

3.5.4 主键、外键命名

命名方式：PK\_<table>，其中，table为建立的主键所在的数据库的表名。唯一主键按照UK\_<table>\_<column>的方式命名，<column>为作为主键的字段名。外键按照FK\_<ppp>\_<ccc>\_<nn>的方式命名，其中，<ppp>为父表名，<ccc>为子表名，<nn>为序列号

3.5.5 Sql语句编写

1. 字符类型数据

SQL中的字符类型数据应该统一使用单引号。特别对纯数字的字符串，必须用单引号，否则，会导致内部转换而引起性能问题或者索引失效问题。利用trim(),lower()等函数格式化匹配条件。

1. 复杂sql

对于非常复杂的sql（特别是有多层嵌套，带子句或者相关查询的），应该先考虑是否是设计不当引起的。对于一些复杂SQL可以考虑使用程序实现。

1. 避免in子句

使用in或者not in子句时，特别是当子句中有多个值时，且查询数据表数据较多时，速度会明显下降。可以采用连接查询或者外连接查询来提高性能。

1. char与varchar2的使用

char在进行查询或者建立索引时要比varchar2效率高，但是varchar2在存储上要比char有优势，在使用时要注意区分。

1. 避免嵌套的Select子句

在进行查询时要尽量比避免多层嵌套的查询语句，以降低select查询语句的复杂性。

1. 避免使用Select\* 语句

如果不是要取出所有数据，尽量避免用\*来代替，应给出字段列表，注：不包含select count(\*)。

1. 避免不必要的排序

不必要的数据排序会大大降低系统性能。

1. Insert语句

使用insert语句时要给出要插入值的字段列表，这样即使更改了表结构，增加了字段也不会影响现有系统的运行。

1. 多表连接

做多表操作时，应该给每个表都取一个别名，每个表字段都应该标明其所属哪个表。

1. 参数的传递

Sql语句的编写，变量尽量使用“？”。

3.5.6 其他







所有默认为空的字段都应该设置为空字符串，而不能是NULL，对于字段能否为NULL应该在sql建表脚本中明确指明，不应该使用缺省值。当字段为NULL时，应该用NVL（）函数把可能为NULL的字段值转变为非NULL的默认值，再进行其他操作。

**第四章 文档规范**



4.1 使用工具



目前，编写文档使用的主要工具为Word，辅助使用工具有Visio，PowerPoint,Excel等。

4.2 风格要求

文档必须保证段落格式整齐，文字字体、颜色、大小统一。文档结构整齐划一。

具体的格式要求根据具体的文档相应改变。

4.3 其他

1)注意中英文标点符号，正文文档一般使用中文标点符号；

2)如果需要摘引html代码，需要先去除html格式，然后粘贴到文档中；

3)如果需要在文档中插图，需要使用visio，此时，需要建立visio文档，绘制完后粘贴到word中；

4)修改他人文档务必使用修订模式，以便保留被修改的内容。

**第五章 代码规范**



5.1 Java源代码规范

5.1.1 命名规范

在编程源代码时需要遵循如下规范：

1. 所有的标示符都只能用ASCII字母，数字或者下划线组成；
2. **类名是一个名词**，采用大小写混合的方式，每个单词的首字母大写，尽量使用完成单词，避免缩写词，除非为常用缩写词；

例如， UserInfo---

VO类：Users，真实实现类：UsersDAOImpl，代理操作类：UsersDAOProxy，工厂类：DAOFactory，Servlet类：UsersServlet，测试类UsersTest；

1. **接口**的大小写规则与类名相似；
2. **方法名**是一个动词或者动词词组，采用大小写混合的方式，第一个单词的首字母小写，其后的单词的首字母大写；
3. **变量名**采取驼峰式命名规则，第一个字母小写，任何中间单词的首字母大写，变量名应该顾名思义，易于记忆，变量名最好不要以下划线或美元符号开头；
4. **常量的声明**应该全部大写，每个单词之间用\_连接；
5. 一个唯一**包名**的前缀总是全部小写的ASCII字母,并且是一个顶级域名通常是com，gov，edu，net，mil等，包名的后续部分根据不同机构各自内部的命名规范而不尽相同。这类命名规范可能以特定目录名的组成来区分部门、项目、机器或注册名，例如，com.sun.eng；

**注**：VO包：vo，通用包：util，接口包：dao，实现包：impl，代理包：proxy，工厂包：factory，Servlet包：servlet，测试包：test；

5.1.2 编程规范

Java编程时需要遵循如下规范：

1. 提供对实例以及类变量的public或private访问控制，尽可能不适用默认值或者protected访问控制
2. 避免用一个对象访问一个类的静态变量或者方法，应该用类名代替
3. 避免在一个语句中对给多个变量赋相同的值
4. 用switch语句实现多分支
5. 如果使用JDBC，则考虑使用java.sql.PreparedStatement，而不是java.sql.Statement
6. 用于设置对象状态的方法前缀必须是set；用于检索一个bool类型对象状态的方法前缀必须是is，而用于检索其他方法的前缀必须是get
7. 程序中应该尽可能少使用数字或者字符，尽可能定义静态变量来说明数字或者字符的含义，程序中需要赋值或比较时，使用前面定义的静态变量。在循环控制中例外。

5.1.3 语句规范

1. 每行至多包含一条简单语句；
2. 在return 语句中，返回值不使用小括号”()”括起来，除非它们以某种方式使返回值更显见；
3. 在for语句的初始化或更新子句中使用逗号时，避免因使用三个以上变量而导致复杂度提高；
4. 当一个switch的一个case顺着往下执行时，如果没有break语句，通常应在应该有break语句的位置添加注释；

5.1.4 注释规范

注释应被用来给出代码的总括，并提供代码自身没有提供的附加信息；

注释分为两种：文档注释(/\*\*..\*/)、实现注释（//或者/\* \*/），其中，

文档注释是Java独有的，可以通过Javadoc工具转换成HTML文件；

实现注释又有如下规范：

1. 块注释：对文件、方法、类、数据结构、算法的简单功能及作者、时间等相关信息的描述等，通常至于文件的开始或者每个方法之前，也可以用于方法的内部。
2. 行注释：单行注释前应有一个空行，且单行注释应该显示在一行内，并与其后的代码具有一样的缩进层级。/\*\*/
3. 尾端注释：与要描述的代码位于同一行，/\*\*/
4. 行末注释：”//” 注释掉一整行或者一行中的一部分

注意，对于注释的其他规范说明为：

1. 注释尽可能使用’//’，对于所有的javadoc的注释则使用’/\*’
2. 所有的源文件都在开头有一个注释，列出作者和时间，用模板即可
3. 每个方法都必须添加注释（类的main方法除外）
4. 每个属性必须添加注释
5. 代码中至少包含15%的注释，JS代码应该超过30%的注释
6. 注释使用的语言（中/英文）

5.1.5 缩进排版规范

1. 避免一行的长度超过60或80个字符；
2. 换行规范：在一个逗号后面断开或者在一个操作符前面断开，辛的一行应该与航一行同一级别表达式的开头处对齐，

5.1.5 声明规范

1. 一行声明一个变量
2. 不要将不同类型变量的声明放在同一行
3. 只在代码块的开始处声明变量
4. 尽量在所有的变量的声明时进行初始化
5. 避免声明的局部变量覆盖上一级声明的变量，在一个程序中，声明的变量之间最好不要重名
6. 类和接口的声明：在方法名与其参数列表之前的左括号”(”间不要有空格，

“{”位于声明语句同行的末尾,“}”另一行起，与相应的声明语句对齐，

5.1.6 空白规范

1. 空行规范：

空行将逻辑相关的代码段分隔开，以提高可读性

使用两个空行：一个源文件的两个片段之间；类声明和接口声明之间

使用一个空行：两个方法之间；方法内的局部变量和方法的第一条语句之间；块注释或单行注释之前；一个方法内的两个逻辑段之间，用以提高可读性

1. 空格规范：

空格应该位于参数列表中逗号的后面；所有的二元运算符应该使用空格将之与操作数据分开，一元操作符与操作数之间不应该用空格隔开；for语句中的条件表达式应该被空格分开；



5.2 JSP/Html代码规范

5.2.1 JSP/html描述注释

Jsp/html页面顶部必须存在一个基本的描述注释，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

\*

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

--%>

5.2.2 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

5.2.3 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：在<title> </title>字段必须制定一个有意义的<title>，严禁出现“Untitled”或“未命名”之类的<title>。

1. 所有html标签使用小写
2. Html页面一般需要设置一个背景色
3. 其他
4. Html语法校验

所有的jsp/html页面都需要能够使用DreamWeaver正确打开（即html语言没有错误标记）

1. 注释

一般不使用html注释，除非是必须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。对没有必要的注释，在发行版本中必须删除。

1. 注意事项

Form属于域的maxlength，必须根据数据库字段的长度设置相应的maxlength,例如数据库类型如果为VARCHAR(64)，那么maxlength是32。

**第六章 项目目录规范**



6.1 Java源文件目录

Java源代码位于项目源文件src文件夹下的java子目录下，

该部分详细的目录结构如下表所示：

|  |  |
| --- | --- |
| **目录名** | **说明** |
| src | 源文件目录 |
| |\_java | Java源码 |
| |\_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 |



6.2 Web应用程序目录

|  |  |
| --- | --- |
| **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 |

6.3 数据库文件目录

数据库代码文件均位于本项目源文件目录下的database子目录下，

该部分详细的目录结构如下表所示：

|  |  |
| --- | --- |
| **目录名** | **说明** |
| src | 源文件目录 |
| |\_database | 数据库sql源码 |
|  |  |

6.4 资源文件目录

本项目中所有的图片、图标等资源文件应位于项目源文件目录下的resources子目录下。

该部分详细的目录结构如下表所示：

|  |  |
| --- | --- |
| **目录名** | **说明** |
| src | 源文件目录 |
| |\_resources | 资源文件根目录 |
|  |  |

**第七章 其他规范**

7.1项目打包部署规范

将已经完成的项目打包成war包时，如果数据库的IP未指明具体的规定时只需将其修改为默认值127.0.0.1即可。