

编程规范

1. 遵循Java编码规范。
2. 一个编程约定：返回类型为List的方法，若没有数据，则返回一个size为0的空list，而不是返回null。其实，在Hibernate中，Session.find(sql)返回的List就是这样的。
3. 对有必要的Java代码一定要进行注解，如果代码已非常明了，就不用注解了。
4. Javadoc中@param、@throw等，能写的一定要写。句子以“.”结尾，短语无需“.”结尾。
5. 命名要尽量简短明了，单词能不用简写就不用简写，但对于常用简写的一定要简写。
6. 文件夹命名，类似于Java中包的命名，要用小写，且尽量简短，能用缩写或截取尽量用缩写或截取。
7. 文件命名，HTML、CSS、images、JS等文件名暂定为小写，若由多个单词组成，则用中划线“-”分割。这是参考ExtJS文件命名、maven site生成的站点的文件得出的结论。JSP文件命名规则同Java类命名。
8. 开发文档等文档的名字使用Java类的命名方式。
9. 代码中，首字母拼起来的简写当成一个单词来看，该小写要小写，但在描述性文字中，要大写，比如数据库连接池，在代码中写为dbcp，但在代码注释中，要写为DBCP。
Java程序中变量或参数的命名尽量不要包含变量或参数的类型，因为变量或参数的类型以后有可能会发生变化。
10. 类中方法名如果必须动词和名词一起使用，则尽量动在前名在后。dao类的名称约定以Dao结尾，而非DAO。
11. 注意项目的文件目录和结构，要好好考虑，最后稳定下来。
12. Java类文件不用tab键，只用空格，缩进4个空格。
eclipse设置：Window/Preferences/Java/Code Style/Formatter，点击“Edit”，选项卡切换到“Indentation”，Tab only改为Space only，Indentation size和Tab size改为4。
13. Java类源文件代码的字符宽度不超过80。
右侧界线的eclipse设置：Window/Preferences/General/Editors-/Text Editors，Print margin column修改为80。
14. 在写产品时，须遵守上面规定，但写项目时，在尽量遵守上面规定的同时，还要考虑到项目的特殊性，比如代码中有较长的sql，考虑程序的可读性，每行可以超过规定的字符个数。
基于代码的可读性、行云流水是第一位的考虑，代码宽度放到100。另外，还有两个因素：
1>. 我们用Java写的这种类型的项目，变量的名字相对C语言写的基础性项目，还是比较长的；
2>. 我们现在的电脑显示器都是宽屏了，所以列数可以加大些。
尽量遵循这它，但也不应拘泥于它，少数情况下，比如就差几个字母代码就搞一段落了，也可超100。
最后，程序中的Javadoc或注释更不应该拘泥于它，可更加宽恕些。
15. Java类中方法名如果类似于queryRecordsByCardNo(String cardNo)，则可以简写为queryRecords(String cardNo)，ByCardNo含义可以在Javadoc中通过主说明以及@param等进行说明。但是，如果方法名和参数一样，这就必须加By...了。另外，在DAO方法中，类似getByGroupId(...)的Id是不必的，写成getByGroup()即可，因为@param会具体说明的。
16. 在DAO类的方法中，参数名可以使用诸如id、name、no等，因为一个DAO一个类，所以这样不会有歧义，但在Service类中，参数名最好使用诸如productId、productName、cardNo等，这样会避免歧义。
17. 使用英语书写Javadoc，应使用最合适的自然语言来说明，不论是量词、单数、复数等。
另外，@param p1和@return后一般是一个词或短语，小写开头。@throws后形如：SQLException if...。这三种，最后不以句号结尾。若后面还有附加说明，则说明前加句号隔开。
如果返回值是复数（类型为List、Array等），则在主注释说明中要加“列表”、“数组”等。
18. 一般的增、删、改、查操作一个表一个DAO，但在报表的DAO，由于涉及三个或三个以上表的关联查询，则可以酌情按业务分配到一个新建的的报表DAO。
19. 数据库约定，见《DBConvention》。
20. CSS书写时，由外部属性向本身属性，然后再向内部属性写。
21. 一个包可大可小，可根据系统的大小和复杂性来决定是否要划定为一个包。
22. CSS中的class和id暂定使用小写，若由多个单词组成，则用中划线“-”分割。另外，命名和注释规范参见yiqusoft-site中index.htm和carousel.css的相关做法。

架构

1. 从目前的发展趋势来看，像mis和erp之类的软件，采用js、ajax等异步请求响应的方式比使用JSP、Struts等传统的页面刷新的方式在客户体验(偏cs体验)、软件运行速度(取决于客户机的性能)、服务器端负载(客户机负责显示数据，控制显示格式等，这就将服务器解放出来了，服务器端只负责处理数据)等方面具有优势，所以作系统的时候可以使用js、jquery、extjs等来做。但做网站的话可以主要采用JSP、Struts等传统的页面刷新方式，在比较适合的地方使用js、jquery等，是非常好的。
2. 写框架时，当遇到难题时，可参考别人的框架，我写这个项目时，就参考了Mybatis。
写dbutil时参考Mybatis项目的几个类：
org.apache.ibatis.session.defaults.DefaultSqlSession相当于我的Session类。org.apache.ibatis.executor.BaseExecutor和org.apache.ibatis.executor.SimpleExecutor extends BaseExecutor被DefaultSqlSession引用（使用），完成操作查询、更新等操作。

- org.apache.ibatis.transaction.jdbc.JdbcTransaction被BaseExecutor引用（使用），完成数据库的提交和回滚等操作。
3. 在有service和dao层的框架中使用我的org.projects.sample.commons.dbutils.Session时，要：service层的类中的方法应该负责SessionFactory.openSession()和session.commit()以及session.close()，在其中创建若干dao对象，将session作为参数传给dao类的构造方法，调用多个dao对象的方法，完成一次事务操作。
 4. 对于一般的三层架构：表示层-service层-dao层，创建database connection和conn.commit以及conn.close应该位于service类的方法中，一般情况下，一个dao类对应一个业务实体类，而service负责业务逻辑控制，一个service方法可能调用若干个dao，操作多个业务实体，这时，service方法中应该创建dao实体，将conn作为参数传入dao类的构造方法，调用多个dao对象的方法来完成一次事务操作。
 5. 暂定：service中的类大致按模块划分(特殊的话自己定)，一个模块一个类，action采用同样办法。JSP也大致按模块划分(特殊的话自己定)，一模块一目录，里面放JSP文件。
 6. 过滤器主要使用在解决中文乱码和权限控制方面，另外官方还有若干个过滤器的例子，如压缩响应数据流过滤器，在examples/WEB-INF/classes/compressionFilters中，还有examples/WEB-INF/classes/filters中的有几个，其中，RequestDumperFilter可以打印Request中的值，作调试用，这些类根据需要的可以拿过来用。
 7. 有些页面比如用户注册页面是一个页面(register.jsp)两个动作，一个是进入动作registerInput，一个是提交动作registerSubmit，这时要注意需要定义两个动作。
 8. 静态类和单态类，见《StaticAndSingletonClass》。
 9. 每个Servlet类完成操作一个抽象实体的功能，在Servlet类中增加若干方法处理对该抽象实体的不同的操作，如UserServlet中有add、delete、modify、view、list、query、bindToRole等方法。
 10. 在一个有bean、dao、service、servlet的一般的web应用项目中，对若干子模块的一个，如security子模块，包的结构是security.bean还是bean.security，目前决定security.bean，然后service及后面是这样约定。但web等前台展示是否可以考虑再做一个子项目。
 11. 如果界面采用JSP（相对JS），对网页布局：偏网站项目的菜单在上部，一般不分级或最多两级，偏系统软件的项目在左部，分级且可能较多级，个人工具菜单在右上角。
 12. DAO方法命名的get和query，无参的用get，根据唯一键(如id、name)获得唯一实体的也用get，其它可用query。
 13. Java的实体包名（dao，service，servlet等）和实体类名（对应数据库的实体的表），约定不用复数。
 14. 对于同表对应的实体类，目前根据经验以及分析、考虑，暂作如下处理：
实体类的属性同表中字段一一对应（关联字段也一样），不增加关联对象的属性，若有需要增加非对应表中字段的临时属性，则要加上注释。
 15. service层类在也可以像DAO那样同表一一对应，但不像DAO那样严格遵守这个原则，因为对DAO来讲，那是标准，对它来说标准不是这，而是从表示层看，从逻辑上讲，某些方法是否应该归到同一个类中。
另外，对于方法中参数变量的命名，最好包含实体名，如用roleId代替id等。
还有，Service类应该当作Manager类来写，而不能当作一个实体类来写。
 16. service层类的方法，如果有参数可用实体类，也可用实体类的主键id，那么从面向对象和程序设计的规范性的角度来讲，应该用实体类，且从程序通用性来讲，一个实体类不一定要有主键id的，还有就是程序本身的可读性，所以也应该如此，但从灵活、方便、够用即可的角度来讲，可以用类的主键id。
目前决定：够用即可，可用主键id，另外考虑到调用的方便性，也可以用实体，可灵活些。
 17. 原来想着在权限控制中，将URL权限定义为ServletPath+Operation，因为这符合Permission=Resource+Operation这个模型。但现在感觉有以下问题：1>. URL都定义为ServletPath+Operation，显得比较强硬，不好，因为在有些情况下，这种拆分实在不靠谱的，可能并没有什么Operation。2>. 对Web框架有要求，处理URL权限时须按这个模式来，这样不好。
 18. DAO中，一般情况下：小关联可直接处理，例如取销售单时可通过SQL关联得到操作人、销售人员名称；大关联，最多处理两级，多级的可放到Service中，调用DAO的get(id)处理。
 19. 层中哪一部分来校验和处理不合法的数据，在产品级别或工具类的API实现代码中，大部分都是需要处理所有不合法数据的，但在一般的应用系统中，在考虑到安全调用的前提下，可各层处理自己的不合法数据，不需要每层都处理所有的不合法数据，可假设调用方传来的参数数据是经过它筛选的，是安全的，可以不考虑不合法情况。
但这些一定要保证在安全调用的前提下。比如，登录页面，为了防止恶意用户编程提交，除了在客户端要做用户名和密码的非空校验外，服务器端也要做非空校验。
 20. 使用jQuery吗？包括core和ui。
从简单角度讲：不使用。仅专注于该做的事情，即展示simplest-dbutils和simplest-web的如何使用为目的。从项目的使用性角度讲，应该使用core，最低限度的使用ui，可用到：datepicker控件，ajax请求，类似于邮件格式的校验等。
目前考虑：使用datepicker控件。所以，暂时使用jQuery UI的Download builder只在Widgets选择Datepicker和Dialog，Download builder会根据依赖关系自动选择上UI Core和Interactions的其它的。然后UI的Theme选择Smotherness，跟目前的页面风格比较搭，比较简单。

使用工具

1. 使用DBCP，需要导入commons-dbcp-1.4.jar，它又依赖于commons-pool-1.5.5.jar，它们可以从官网下载的commons-dbcp-1.4-bin.tar.gz和commons-pool-1.5.5-bin.tar.gz压缩包中获得。
2. 使用JSTL，需要导入的包为jstl.jar和standard.jar，它们可以从官网下载的jakarta-taglibs-standard-current.tar.gz包中获得。
3. 日志系统欲使用slf4j和logback，见<http://www.slf4j.org>和<http://logback.qos.ch>，另外，关于日志系统的选择，有篇文章还行<http://yanboy.iteye.com/blog/204436>。
4. 使用Javadoc，有几篇文章不错，见“[studynote/java/resource/java-doc相关文章](#)”，其中英文的那个可重点看一下。实现类的方法的java doc中一

般用@see标注该方法的描述说明请参见接口类对应的方法。

参考项目

1. 参考项目iBATIS_JPetStore-4.0.5中我的笔记中关于值得学习的点。

.知识点

1. 参考JDK API, Statement的executeQuery()返回的ResultSet对象永远不可能为null。
2. 对于PreparedStatement的setXXX(), 若该列的值为null, 则可以写为setNull(1, JdbcType)或setXXX(1, null)或setObject(1, null), 据我了解, 有的驱动程序这三种方法都支持, 有的则部分支持, 这个要注意首次使用数据库驱动或更换数据库驱动时要进行认真的测试。
3. 异常, 见《Exception》。
4. 解析xml, dom4j功能比较强大, 但需要引入包, 而dom在jdk中已经包含了。由于我的项目是simples, 所以用dom。
5. 编写递归函数的基本技巧: 自己调用自己; 函数中有条件语句, 一个分支自己调自己, 一个分支什么都不做, 返回空。
6. 这两天(今天2013年7月6日)跟同事讨论组织结构等有如下心得:
 - 1>. 一般的, 一种设计考虑只能解决一种问题, 不要指望一种设计可以搞定所有问题, 所以, 每种设计考虑有自己的优势, 当然, 也有自己的劣势。虽然可以多个设计综合使用来解决不同方面的问题, 但这也引入复杂性。这时, 关键是要分析, 哪些问题是我们要首先解决的, 哪些次之, 哪些不重要或可以通过其他简单方式简单处理, 根据问题的重要性排个序, 做个列表, 然后据此提出综合解决方案。
 - 2>. 在进行软件设计考虑时, 对于该软件的使用, 可能会有一些要求, 或者是在思想上, 或者是在原则上, 需要使用者遵循某种约定, 不能花乱的滥用, 否则会导致软件在使用时, 概念等会变的越来越混乱。这些需要遵循的约定在设计文档和软件成品说明书里都应该有体现。
7. CSS控制布局的高度和宽度:
 - 1>. 一般情况下, 高度使用绝对值, 而非百分比, 宽度情况各占一半吧。但当要控制高度、宽度全屏或占百分比时, 应使用百分比。
 - 2>. 一般来说, 如果最外层布局的高度或宽度使用百分比(非绝对值), 但里层又要用绝对值, 则可以采用div嵌套实现。外层百分比, 里层绝对值, 通过嵌套来实现这种转化。
 - 3>. 2>中讨论的, 当内层采用绝对值时, 会有剩下一部分的空隙, 这是个弊端。因此, 在设计页面时, 掌握着百分比和绝对值的特点, 然后从全局角度考虑。
 - 4>. 最后考虑, 高度和宽度都采用绝对值, 这样控制最简单, 现在大部分显示器是宽屏, 16:9或者16:10的, 而这里再取一个1280 * 800(16:10)的主流中最小分辨率作为标准来定义宽度和高度。
8. 网站整体色彩搭配设计: 主题部分大部分为白色, 外框和主题部分的组件颜色可谓活泼的颜色如橙色、嫩绿等。

.项目管理

1. 有一个技巧不错: 跟大家商量, 然后约定在在某一天, 将所有的程序提交, 进行联调。这种具体、明确、硬性的目标对激励大家按时完成任务是很有用的。每个人可能会觉得, 如果自己没有按时完成, 拖大家后腿的话, 会很丢人, 所以也会积极配合。