# 开发说明书（面向后期开发者） Retrieval 1.0 By jzhao

**技术架构说明：**关于项目的基本概念及技术选型，请参看本人写的《鱼类检索系统项目书.docx》、《鱼类检索系统-进度讨论.pptx》、《鱼类检索系统-介绍.pptx》。

为了加快开发速度（因本人即将毕业的缘故），虽然之前使用GWT的作为前端开发技术完成了大部分MASTER端的页面，但是由于加入了CLUSTER端，免不了需要改动MASTER端的代码，由于GWT的开发速度远不如JSP来的快，所以当前版本使用JSP作为主要的前端技术，在某些通信量较少的比如删除节点的功能上才使用AJAX与服务端通信。当然，使用JSP就免不了客户端与服务端产生了耦合，如果后期需要将本项目集成到ESB中，还请使用AJAX技术，而服务端代码90%以上是可以复用的。现在项目的技术架构如下：

|  |  |
| --- | --- |
| 领域 | 所用技术 |
| 前端 | JSP + AJAX |
| 后端 | STRUTS2 + Spring + JDBCTemplate |
| 数据库 | MySQL |

**JSP的作用范围：**用以格式化输出，典型的如query\_node.jsp。虽然add\_\*.jsp也是jsp文件，但并不包含java代码，add\_\*.jsp主要都是用表单的形式提交录入的数据。客户端开发最难的部分也是add\_\*.jsp的部分（请看代码中的JavaScript部分），原因是录入的数据具有层级关系，比如添加一个物种节点，物种节点具有自定义字段，自定义字段的数量需要根据录入人员的需要可增可减，这就需要JavaScript在客户端进行一定的DOM操作，DOM操作使用JQuery框架。另外物种节点还可以添加属性，属性的和自定义字段一样需要控制做到可增可减，每条属性还可以有自定义字段，也要求可增可减，由于这些可增可减的字段数目是动态的且有层级关系（属性1的自定义字段1，属性1的自定义字段2，属性2的自定义字段1......），所以这些字段在服务端统一以JSON格式接收，这就需要使用JavaScript在客户端依序拼接出JSON字符串，拼接的逻辑稍显复杂，请后续开发人员注意。



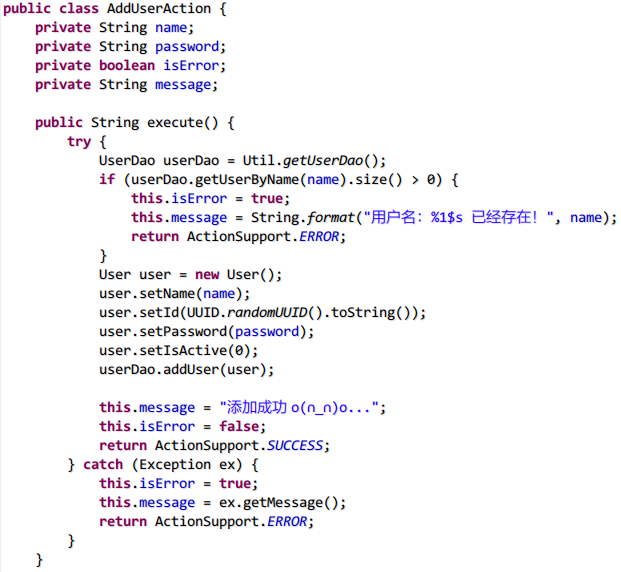
**这是一个典型的jsp文件用于排版输出数据的用途**

**AJAX的作用范围：**像删除一个节点这种只需要向服务端传送一个ID号的场景就不需要表单了，所以使用AJAX。AJAX通信也使用JQuery框架实现。



**典型使用AJAX的场景——删除XXX**

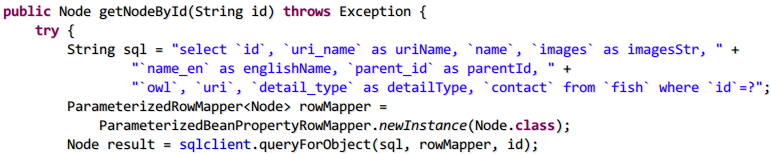
**STRUTS 2的作用范围：**最大的作用是替代了手工编写Servlet的工作，并且编写出的代码不仅不和STRUTS2耦合也不和Servlet耦合，非常方便做测试，另外STRUTS2对文件上传功能做了很好的包装，几乎已经做到了最简化。



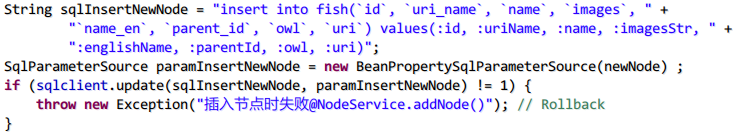
**使用STRUTS 2编写的用来替代Servlet功能的类，这是一个很普通的类（我是指没有继承什么特殊的接口或其它类，非常普通），它有一个execute方法（当然， 你也可以取其它的名字），STRUTS 2默认在接收到一个请求后会调用你编写的execute方法，而HTTP请求中的参数会根据变量名事先赋值到你编写的类中（具体细节看代码或视频），所以①你有了参数②你的方法在正确时间被调用了，所以你可以做该做的业务逻辑了。**

**SPRING：**简化类的实例化，更大的作用是为JDBCTemplate提供支持。

**JDBC TEMPLATE的作用范围：**最大的作用是对数据库的一次操作（增删改查+事务管理+连接池管理+ORM）可以浓缩为最短1行最长不过5~6行的代码，既简便有增加了可读性。



**典型的根据ID查找记录的例子，解读：完全自定义SQL语句，问号是占位符，在最后一行代码执行SQL语句时传入占位符的实际值。sql语句下面一行代码的作用是根据查询出来的列名自动匹配到Node类中的字段名，所以最后一行返回的不是什么RowSet，而是你希望得到的实体类。**



**典型的插入新记录的例子，解读：SQL语句自定义，冒号后跟单词的是占位符，在第二句代码中指定了由谁来填充这些占位符，填充的规则是占位符名和类中的字段名相匹配。**

# 推荐开发人员入门方法

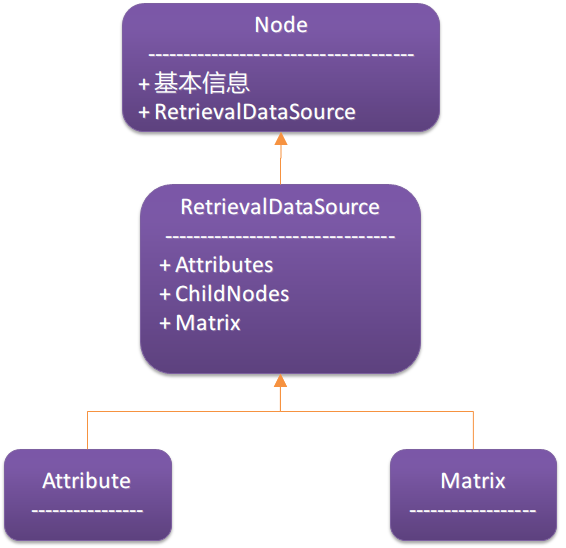
**首选视频：**JSP，STRUTS2，SPRING，JDBCTemplate在传智播客都有视频，推荐将播放器调至1.5x~2.0x速度观看。

**JQuery：**可以看官方的Getting Started。

|  |  |
| --- | --- |
| 视频网站 | 视频网址 |
| 传智播客 | http://www.itcast.cn/video |
| JQUERY Getting Started | http://docs.jquery.com/Tutorials:Getting\_Started\_with\_jQuery |

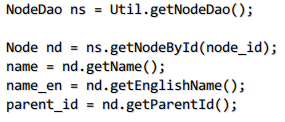
# 数据结构

**节点的抽象：**注意，阅读数据结构之前你必须先看完《Retrieval使用说明书》。节点中的数据抽象成详细信息（即基本信息）和检索信息（即牵涉到上下午的信息），基本信息有名称, 英文名称, uri, uri名称，自定义字段等等，都定义在Node类中，检索信息包括特征矩阵，属性，子结点编号，定义在RetrievalDataSource类中，Node对象包含一个RetrievalDataSource对象。



**Node类的结构**

所以假设现在有一个Node对象叫nd，操作其基本信息的代码可以为：

**

操作其检索信息的代码可以为：

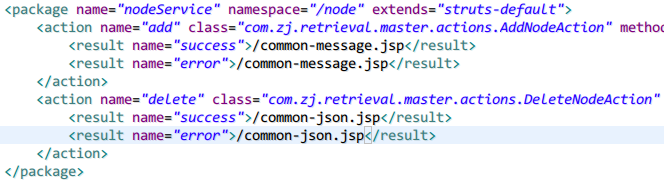




RetrievalDataSource类中还进一步定义了Matrix字段、Attributes字段和ChildNodes字段。Matrix顾名思义是保存特征矩阵的，Attributes字段保存了所有的属性，也就是检索过程中那些要问的问题，ChildNodes字段保存了当前节点下的所有子结点的信息（主要保存的是子结点在数据库中的id号）。Matrix、Attributes、ChildNodes加在一起提供了检索所必须的数据来源，所以取名“RetrievalDataSource”。

**关于三层架构：**本项目的三层架构比较模糊，举个例子，客户端是JSP，JSP上面或使用form或使用AJAX方式发起对服务端的请求，服务端通过STRUTS2框架将请求转到指定的XxxAction类中，在指定的action类中，取出客户端发来的各种参数（包括上传的文件），调用UserDao类的方法进行用户的权限验证，调用NodeDao进行节点的增删改查，在NodeDao和UserDao中的做法通常是拼接出SQL语句，然后调用JDBCTemplate的数据库操作方法获得数据库查询结果后返回。

**返回客户端的数据：**返回客户端的数据可以是普通网页型，也可以是JSON型，参看retrieval-master/src/struts.xml，可以看到不同的action（也就是用struts2代替Servlet的东东）各返回不同的类型，比如AddNodeAction，返回一个common-message.jsp，打开这个jsp你会发现其实是一个完整的html文件，里面有一个${message}占位符，在action中会指定占位符的值，最终返回客户端的是占位符被实际值替换后的html文件，例如显示“操作成功”或“添加成功”，这是一个完整的网页（有html标志）。又比如DeleteNodeAction，它接收来自AJAX类型的调用，在客户端实际等待服务端响应的是JavaScript代码，这时候就应该选择common-json.jsp类型，开打common-json.jsp你会发现其实是一串json字符串，中间穿插了一些占位符，这不是一个html文件（因为没有html标志），实际返回客户端的是占位符经过替换后的json字符串，JavaScript能够正确解析json字符串，它会取出其中的字段，通常这时候会在屏幕上弹出信息“删除成功”。

****

**struts.xml中的关于返回值的不同配置**

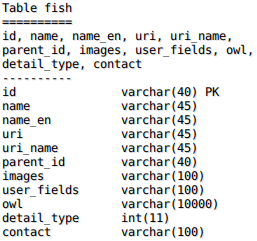
****

**典型的common-json.jsp中的内容**

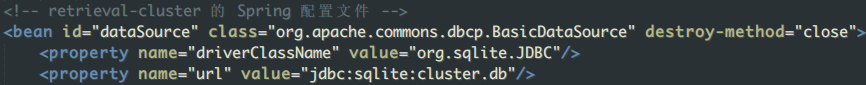
# 数据库

**MASTER端：**使用MySQL数据库，数据库的连接信息存放在retrieval-master/WebContent/WEB-INF/beans.xml中。

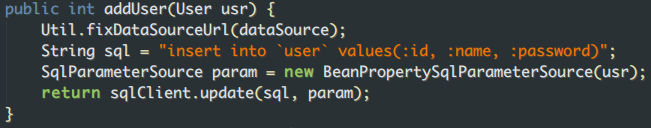
**MASTER端数据库表结构：**



**CLUSTER端：**CLUSTER端考虑到使用环境为不懂技术的研究学者或个人使用者，所以使用嵌入式数据库SQLite，数据库文件名为cluster.db，位于retrieval-cluster项目的WEB-INF目录中。数据库连接信息保存在retrieval-cluster/WebContent/WEB-INF/beans.xml中，这里需要额外指出的是，该配置文件中的url值是不完整的。由于嵌入式数据库实际是一个保存在WEB-INF下的文件，该文件的实际完整路径应该在运行时才能确定，所以我在\*Dao类中每个public方法的第一句都加上了获得cluster.db文件实际路径的方法。

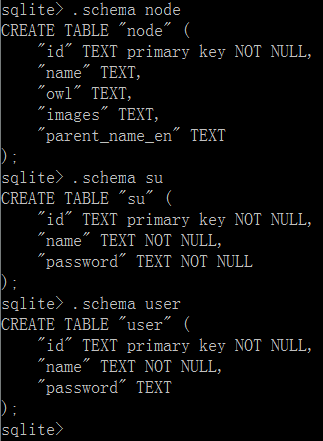


**url的值是不完整的**



**该方法的第一句代码用来确定嵌入式数据库文件的在文件系统中的实际位置，并赋予数据库连接对象，如此数据库才能正确连接。**

**CLUSTER端数据库表结构：**



**缺陷：**① 目前CLUSTER的部署使用TOMCAT，建议将来换一个容器，比如Jetty，争取将容器的大小控制在1M以内。

② CLUSTER端启动后，点击查看远程访问url时，显示的是当前浏览器地址栏里的域名，所以，如果输入的localhost，就会显示0.0.0.0，这显然是不对的，这就需要用户事先知道自己的IP地址，然后url中的localhost换成ip地址，然后再去点查看远程访问url，便能获得正确的结果，所以，需要开发人员做一个launcher，launcher的作用是启动服务器，获取本地ip地址，拼接出正确的url，然后帮助用户打开浏览器并转到正确的带实际ip地址的页面。

③ 界面没有美化，用户体验还有待加强。

④ master端和cluster端都缺了modify\_su.jsp，即缺少了修改管理员密码的页面。

⑤ 所有的数据库操作都没有开启事务，要求开启事务，根由于现有技术架构使用了JDBCTemplate框架，而JDBCTemplate框架自带事务机制，开启事务基本只需要一句代码。