**solr全文搜索**

需求分析：

在网页中通过关键词搜索，列出相关问题，并在关键词上显示出高亮。(目前实现的是问题标题和内容匹配，后面可扩展回答等。



实现概述

Solr单机服务器window环境下搭建：

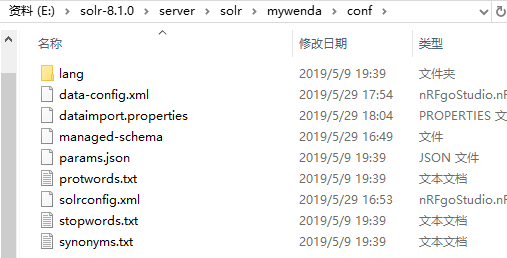
1安装包下载安装：<http://archive.apache.org/dist/lucene/solr/8.1.1/> Windows环境下选择zip下载，直接解压到d盘上即可。

具体过程参考：<https://blog.csdn.net/u011052863/article/details/80281941> ，里面的端口一般选择8983。

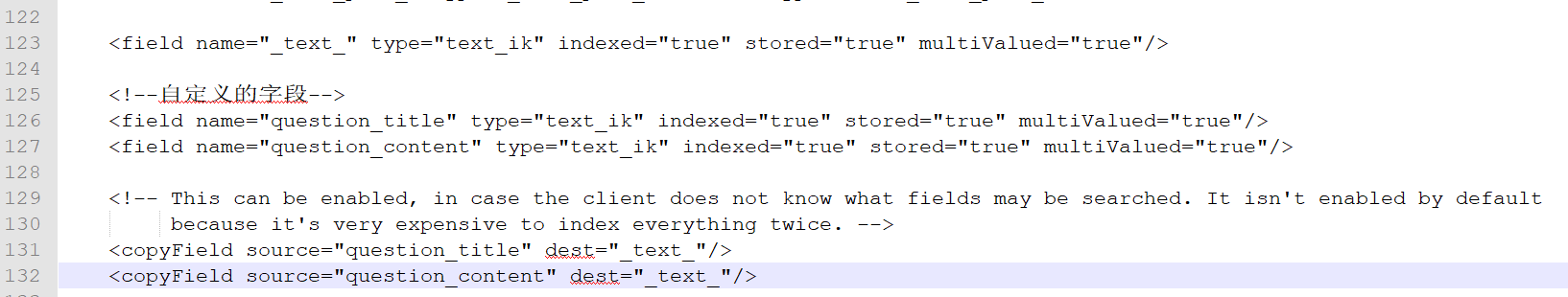
2 配置中文分词器：这里部分可以参考上面文章。

IK-analyze下载：<https://search.maven.org/search?q=com.github.magese> ，同时下载：mysql-connector-java-5.1.47包（注意版本太高可能报错，这里选择的是5.1版本可以使用，选择8.0.16出现错误） <https://search.maven.org/search?q=mysql-connector> ，将下载的两个包放入E:\solr-8.1.0\server\solr-webapp\webapp\WEB-INF\lib路径下。

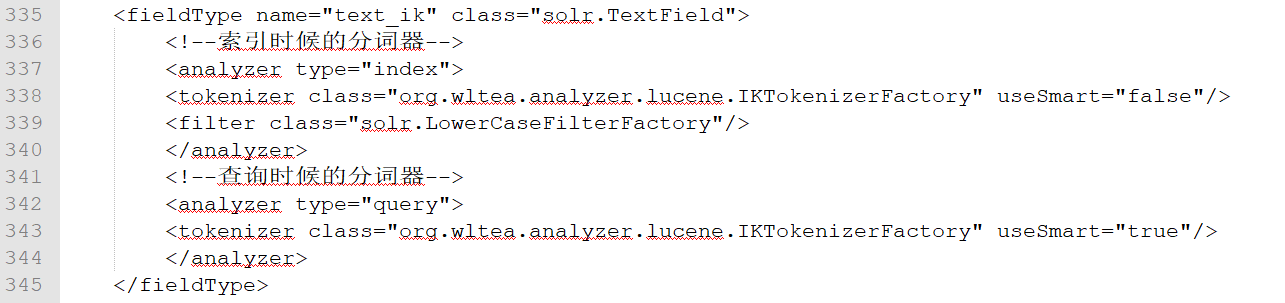
配置：在自己新建的core路径下打开配置文件 managed-schema文件



找到\_text\_的字段配置的位置进行下面改动，首先将\_text\_的type改成text\_ik(这个属性类型下文要加入的中文分词器，如果没改会使用默认分词器)，并将stored改为true。接下来自定义两个字段question\_content和question\_title。最后配置<copyField>标签将这两个字段复制给\_text\_字段。（因为当进行solr搜索的时候如果不指定字段搜索根据solrconfig.xml文件中的df属性定义，其会直接认为找的是\_text\_,这样的话我们自己定义的question相关字段就不会参与搜索了，因此进行copyField操作之后不指定字段搜索的情况下就也能对气门自定义的两个字段进行搜索，参考文章：<https://www.iteblog.com/archives/2399.html> ）



在这配置文件中找到对filetype定义的部分，加入自己的配置（中文分词器），这里的name与上面对应起来。最后重启solr，然后再控制台<http://localhost:8983>中进行分词测试。

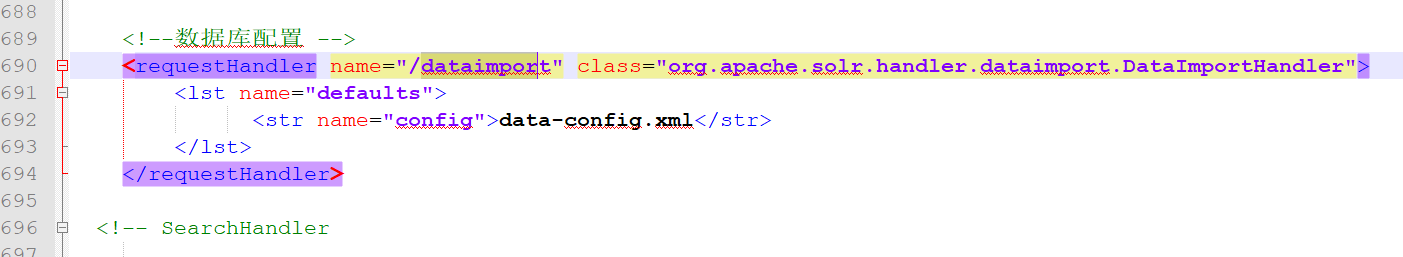


3、从mysql中将数据导入。参考：<https://blog.csdn.net/lhc0512/article/details/82315354>

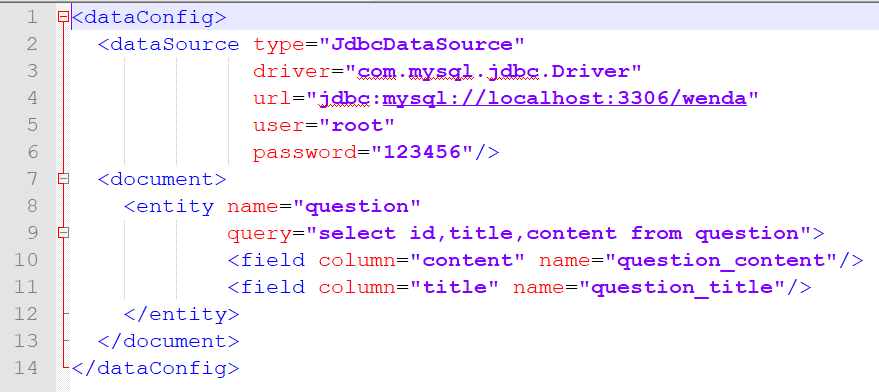
首先将solr安装路径E:\solr-8.1.0\dist下的两个包拷贝出来（如下）



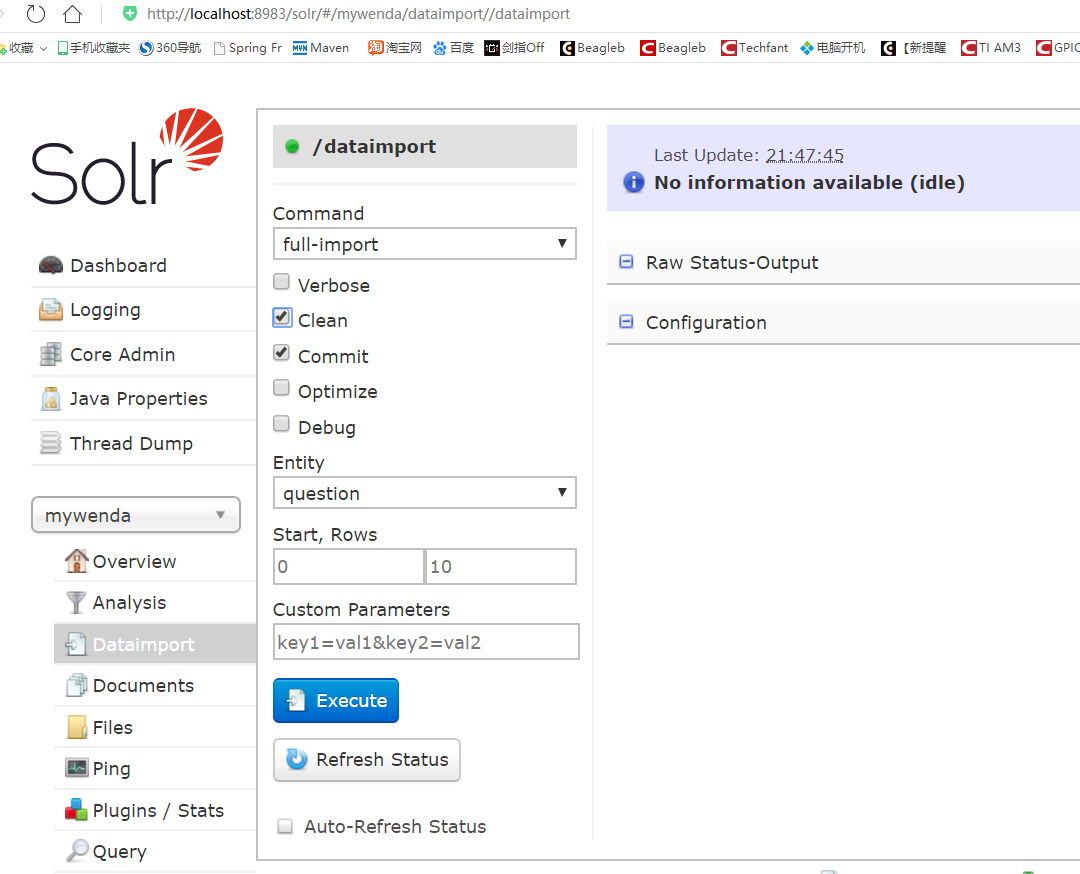
放入E:\solr-8.1.0\server\solr-webapp\webapp\WEB-INF\lib路径下。然后在上述的core路径：E:\solr-8.1.0\server\solr\mywenda\conf下找到solrconfig.xml文件打开进行数据库配置。先找到其配置相关handler的部分加入我们自定义内容。



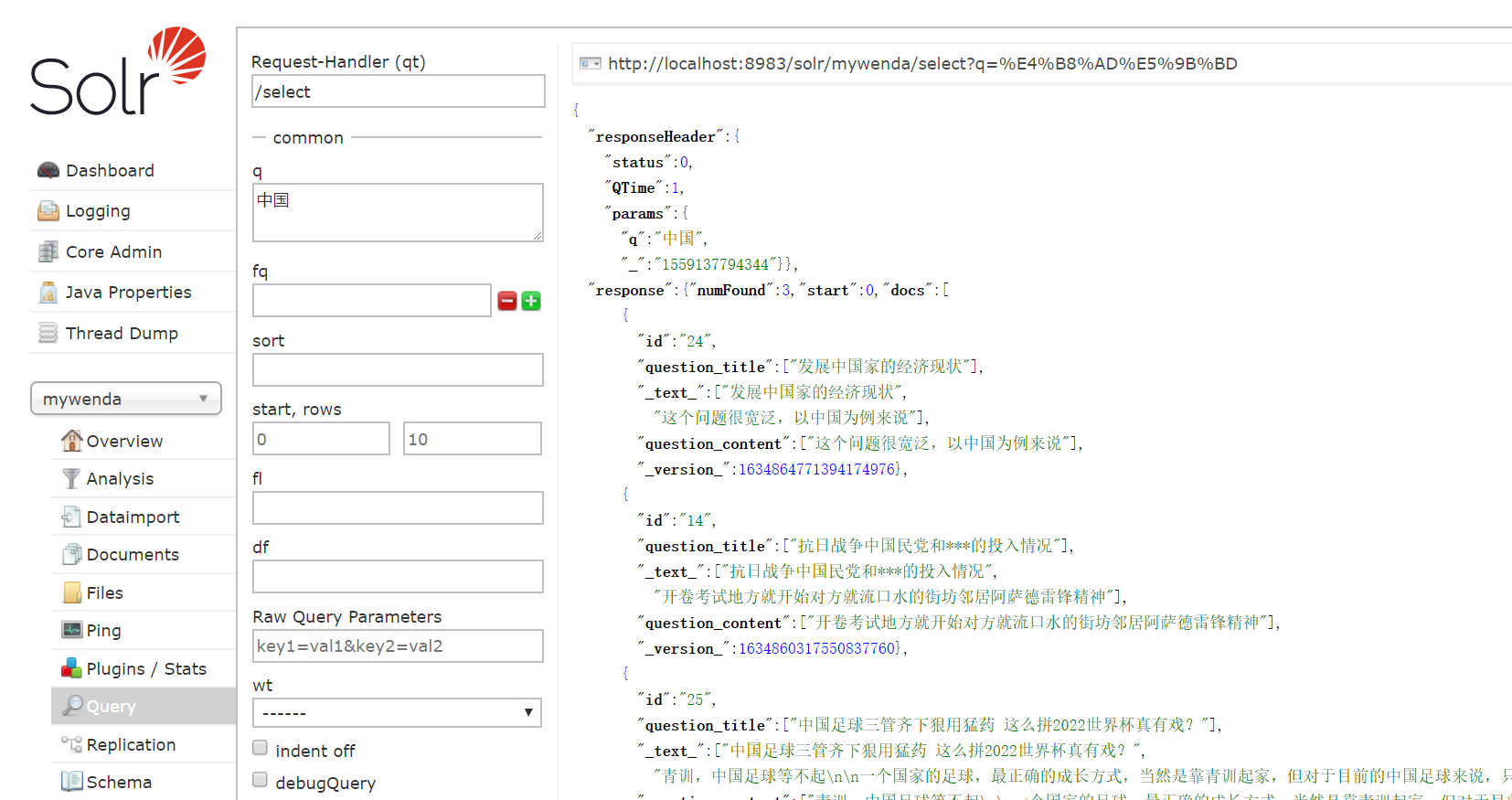
可以看到里面的配置文件data-config.xml由我们来写，主要配置和数据库的连接读取等。在E:\solr-8.1.0\server\solr\mywenda\conf路径下新建data-config.xml文件。需要注意的是file标签的name属性为我们之前新增的question相关字段。



接下来打开控制台：如下选择进行数据导入（注意rows直接使用默认10，改了可能会出错）。



数据导入后就可以在query下进行模拟搜索，环境搭建完成。



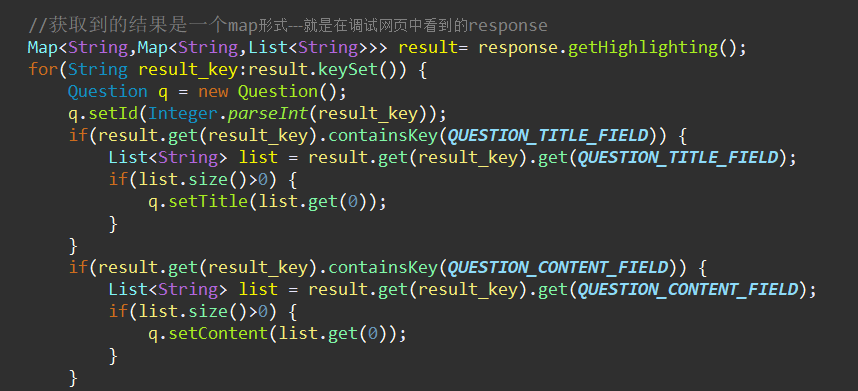
代码实现：

Controller层：

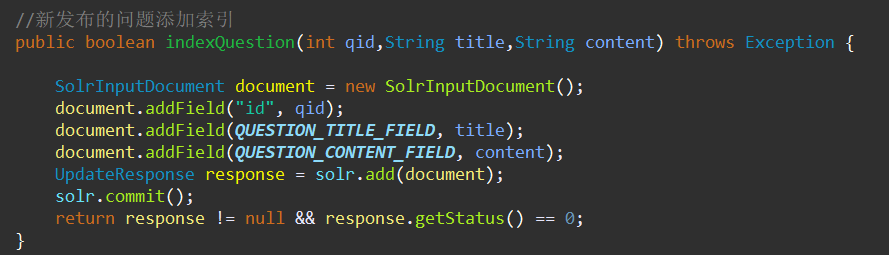
SearchController:（url：search）接收前端传来的关键词,首先调用service层的searchServieImpl将匹配到的quesiton拿到。注意这个question没有完整的对象信息，因为底层操作的是拿到有高亮的字段，（如：一个question的title中有匹配到关键词而content中没有关键词，因为查的是全部所有这个信息会被查到，但是高亮的返回只有title，content中是没有内容的，但是id是无论如何都会有的），所以为了给前端一个有完整信息的对象，先利用得到的id从数据库中找出原始对象，然后对比searchSerivce中传来的question,如果title或者content中是有内容的就将这个内容赋值给数据库中查到的对象（主要目的是为了得到有高亮的信息），最终将question和user以及关注数捆绑给view层进行渲染。

QuestionController:在这一层需要在原来的基础上添加内容。当用户发表了新的问题通过产生异步事件将tietle和content的内容给solr创建索引。

Service层：SearchServiceImpl主要用solrJ来和solr通信，首先是利用solrJ和solr建立连接并进行相关参数的设置，可以参考官网的入门步骤：<https://lucene.apache.org/solr/guide/6_6/using-solrj.html> ，这里获取的是其高亮部分，其返回类型是map<string,Map<String,List<string>>>，因此进行解析



最终返回给上层的是List<Question>(注意这里的question中信息不完整，只是为了方便携带高亮数据)。第二个功能是想索引库进行添加，跟着api来做就行。



Handler层：当有新问题发布的时候异步调用questionHandler,将id,title,content信息通过调用SearchServiceImpl传给solr。