Java实验报告

徐云凯 1713667

目录

[Java实验报告 1](#_Toc22672197)

[1. 问题描述 2](#_Toc22672198)

[2. 实验环境介绍 2](#_Toc22672199)

[3. 实现思路 2](#_Toc22672200)

[4. 具体实现 2](#_Toc22672201)

[4.1. 建立索引 2](#_Toc22672202)

[4.1.0. LuceneIndex类的成员变量与成员方法 2](#_Toc22672203)

[4.1.1. main()函数，程序入口点 3](#_Toc22672204)

[4.1.2. LuceneIndex构造函数 4](#_Toc22672205)

[4.1.3. index方法 4](#_Toc22672206)

[4.1.4. analyze方法 5](#_Toc22672207)

[4.1.5. close方法 5](#_Toc22672208)

[4.1.6. 日志与运行效率 6](#_Toc22672209)

[4.2. 在线查询 6](#_Toc22672210)

[3.2.1. index.jsp 6](#_Toc22672211)

[3.2.2. search.jsp 8](#_Toc22672212)

[5. 结果展示 10](#_Toc22672213)

#### 问题描述

实现邮件检索系统，被检索数据为安然公司150位用户50万封电子邮件，语言为英文。搜索引擎采用Apache的开源全文检索引擎Lucene。检索时按照收件人、发件人、标题、内容等进行邮件检索。并实现交互系统。

#### 实验环境介绍

开发使用的IDE为IDEA，Java版本为11.0.2。Lucene版本8.2.0。Log4j版本2.12.1。web界面使用jsp动态网页，运行在Tomcat服务器中，Tomcat版本为7.0.92。

#### 实现思路

邮件索引系统中的索引建立为一次性操作，一次索引即可反复使用。所以将该项目分为索引建立与查询两个部分并分别实现。其中索引建立部分使用Console界面实现。查询部分使用web界面实现，运行时只需要在服务器端运行索引程序，或者直接将已经生成的索引放置在指定文件夹下即可。

#### 具体实现

###### 建立索引

该子项目全部代码运行于LuceneIndex类中，main函数入口点负责初始化类，构造函数完成构建索引器、分析器与索引写入对象等操作后，由主函数调用LuceneIndex类的index()方法开始运行索引，index方法会递归调用自身完成对待索引项目的遍历索引，期间有完善的错误处理与日志机制保证在较大数据集上运行时不会中途退出。最后由主函数调用close()方法关闭索引写入对象结束索引过程。

* + 1. LuceneIndex类的成员变量与成员方法

logger 对象用于进行日志写入与信息输出。

writer对象为该程序中所有文件索引的共用索引写入对象。

numIndexTotal用于统计索引的文件总数。

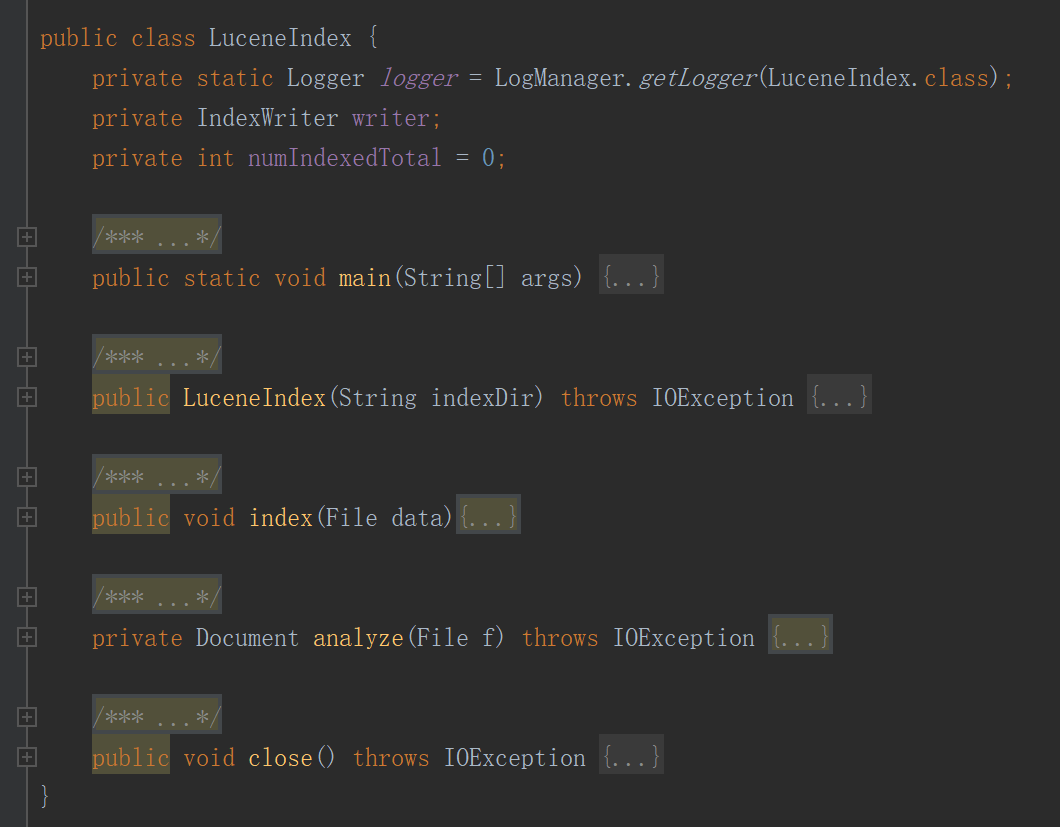
main函数为入口点，负责启动与关闭索引，并计时。

LuceneIndex用于初始化索引器，分析器等对象。

index方法用于递归调用以递归遍历所有目录下的可读取文件进行索引。

analyze方法为文件解析器，解析文件结构分割进入field进行索引。

close方法用于关闭writer。



* + 1. main()函数，程序入口点

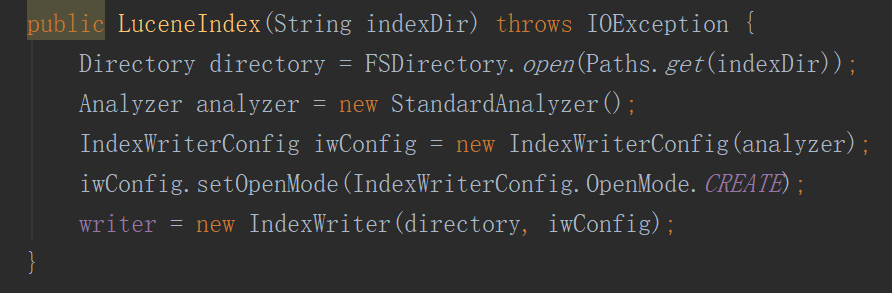
在索引开始与结束时计时，并在最后利用logger输出耗时与索引文件总量。

在所有可能出现异常的位置使用try-catch语句环绕并处理对应异常，利用logger进行输出。



* + 1. LuceneIndex构造函数

创建一个磁盘索引器FSDirectory，一个标准分析器并使用CREATE模式新建索引。这里不使用内存索引器的原因的内存索引器虽然更加高效，但是在Lucene 8.2.0版本中被弃用，且即便不使用内存索引器，索引也可以在半小时左右完成。



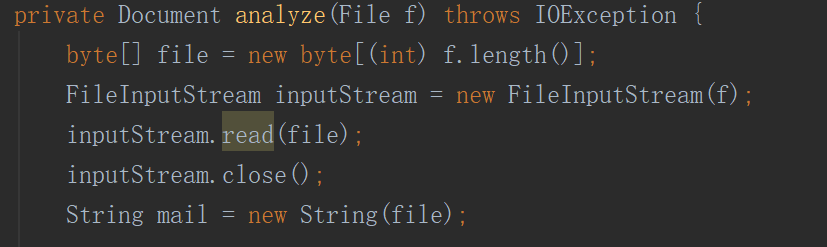
* + 1. index方法

递归遍历完成索引。

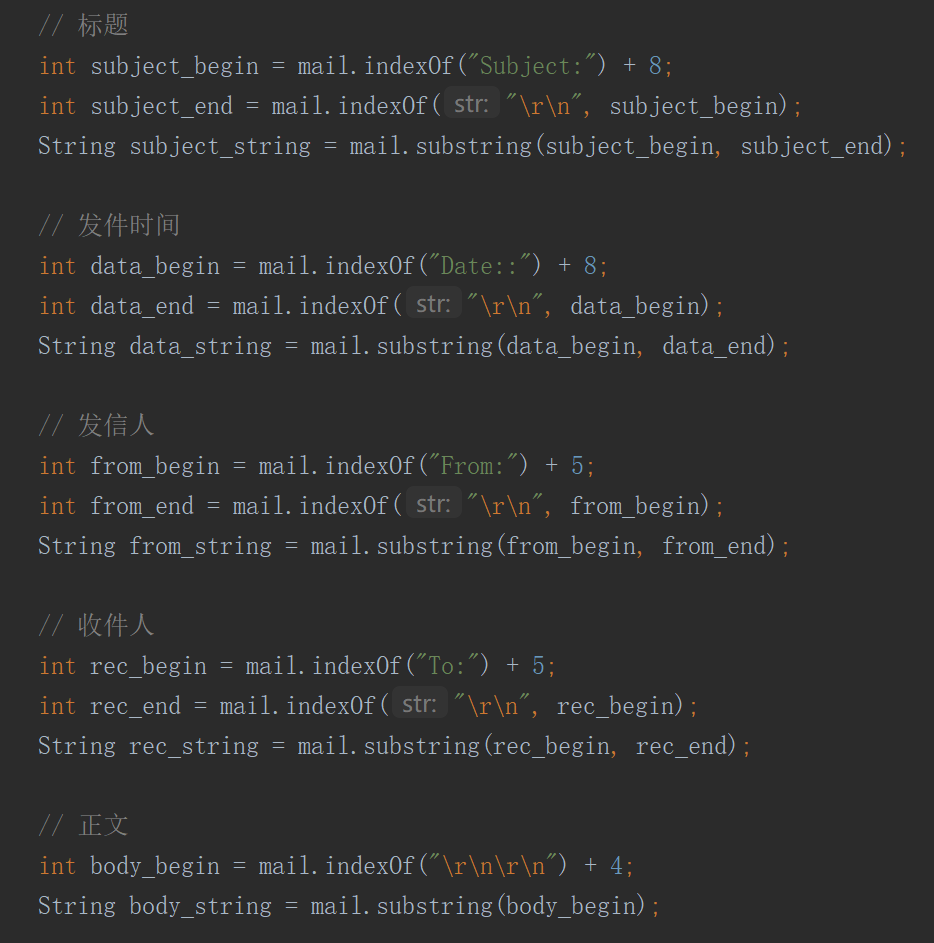


* + 1. analyze方法

该方法会尝试解析文本中的标题，发件时间，发件人，收件人，正文，文件名与文件路径等信息并添加到字段中进行索引。



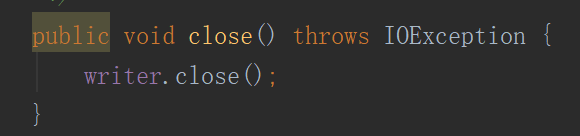
利用FileInputStream进行文件读取，因为是英文，不考虑文件编码问题。



使用subString方法进行截取，通过识别关键字进行截取。

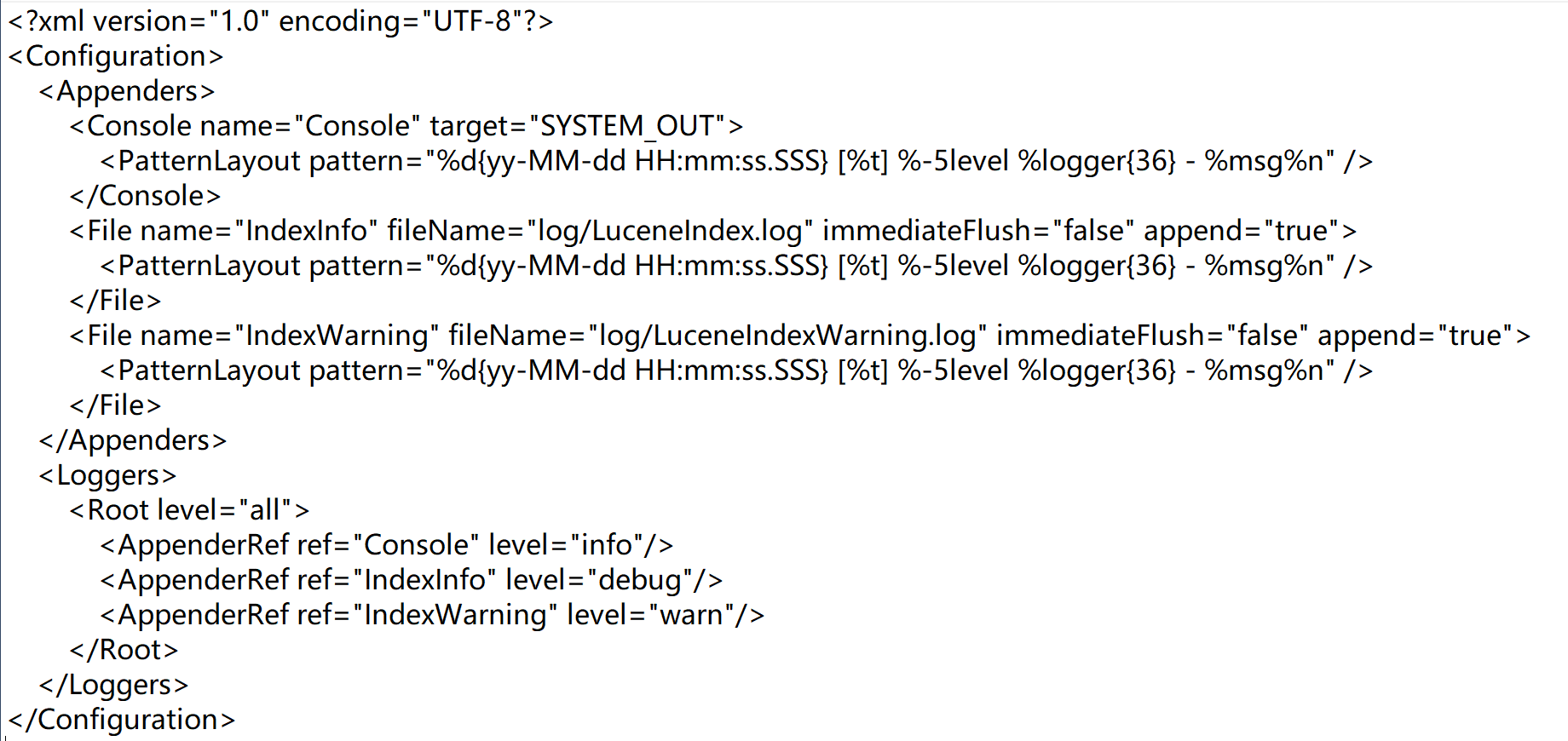
最后使用doc.add()将字符串添加到对应的字段中。

* + 1. close方法



关闭写入对象。

* + 1. 日志与运行效率



Log4j被配置为将日志输出为三部分，info级别以上的被输出到控制台。Debug级别以上的输出到IndexInfo.log文件中，Warn级别以上的单独输出到IndexWarning.log中。

经过实际索引运行，索引全部517k封邮件耗时约半小时，期间cpu占用低于5%，磁盘占用低于1%。该程序的性能瓶颈为磁盘存取的状态切换开销。通过多线程无法获得优化，但是可以通过减少磁盘写索引次数获得一定程度的优化。

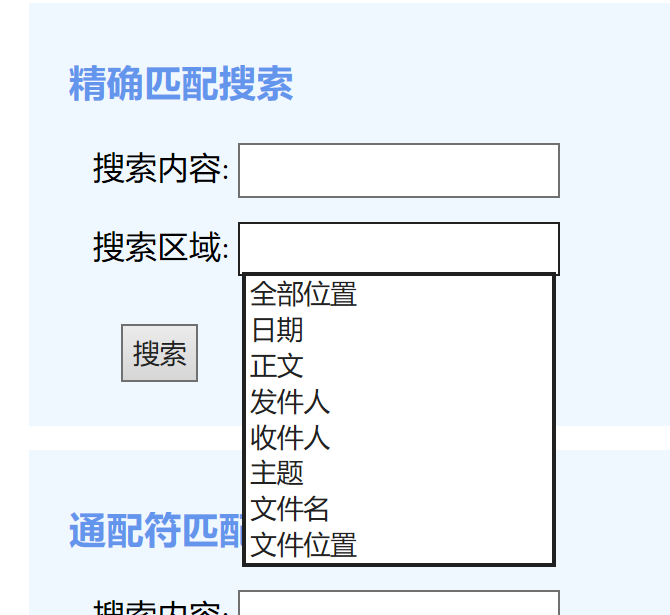
###### 在线查询

查询网页由两部分组成，index.jsp和search.jsp。

index.jsp负责处理前端显示并与用户交互，index.jsp通过get方法把查询参数传输给search.jsp，search.jsp负责处理并完成查询。

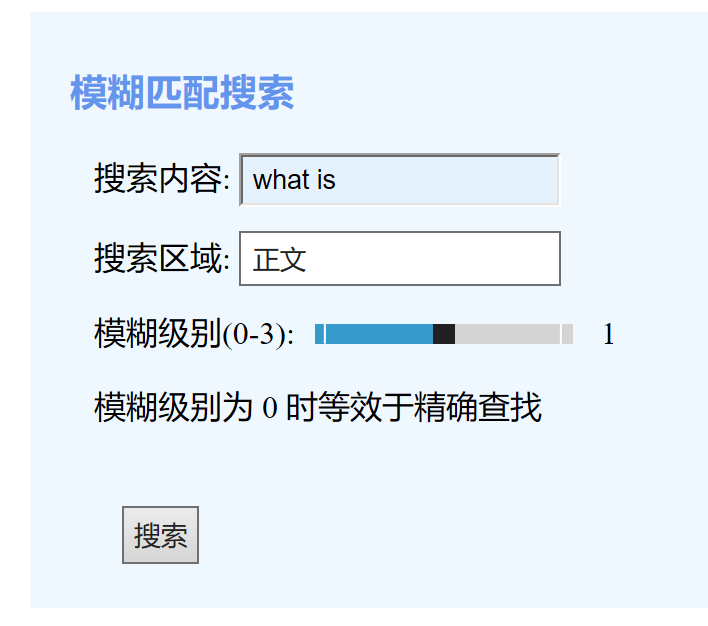
3.2.1. index.jsp

该页面构成主要基于html5，并通过css与javascript美化了页面。界面支持4中检索方式，精确匹配、通配符匹配、前缀匹配与模糊匹配搜索。前三种在传参时只发送搜索内容和字段信息并附带搜索类型。最后一种额外附加一个模糊级别信息。

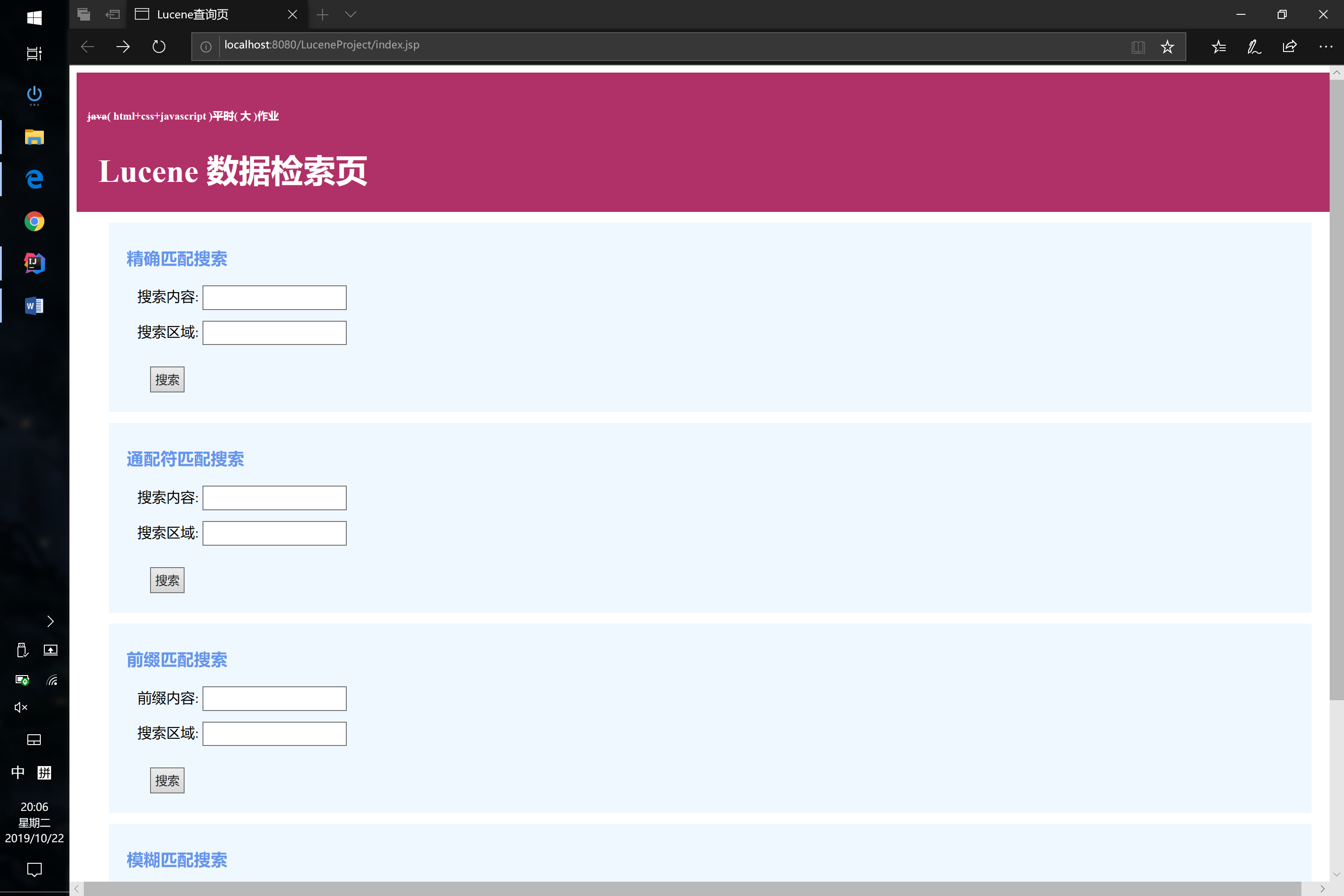


界面中，搜索字段部分提供了中文的下拉列表方便输入。

另外，在模糊输入部分提供了更加用户友好的界面，使用一个可以滑动的滑块提供模糊级别的输入，并且利用html5特性进行实时显示。



**利用form的oninput属性实现滑块值实时显示**



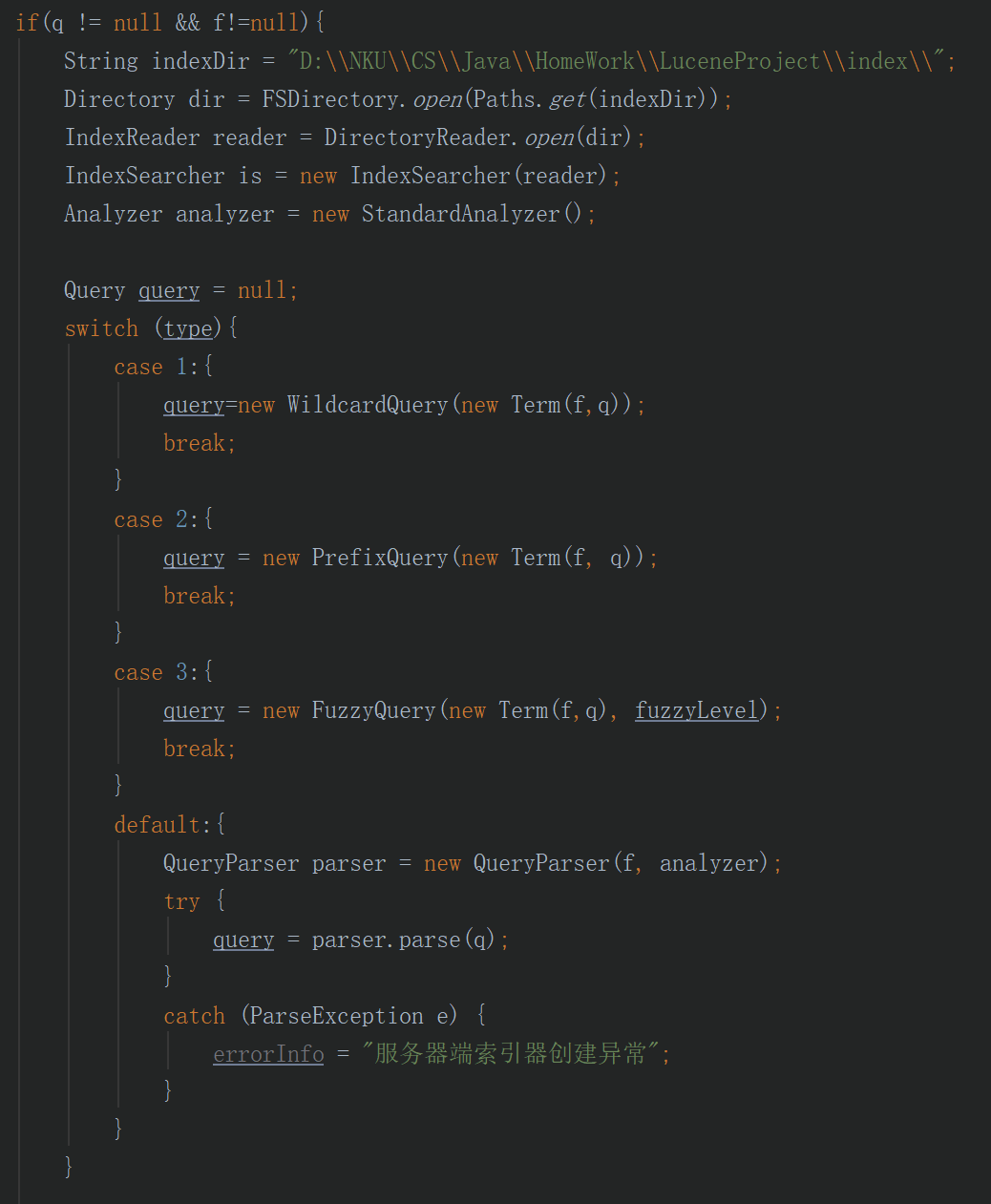
**输入页面整体效果**

3.2.2. search.jsp

首先进行参数解析，同时完成异常参数检测并进行处理。



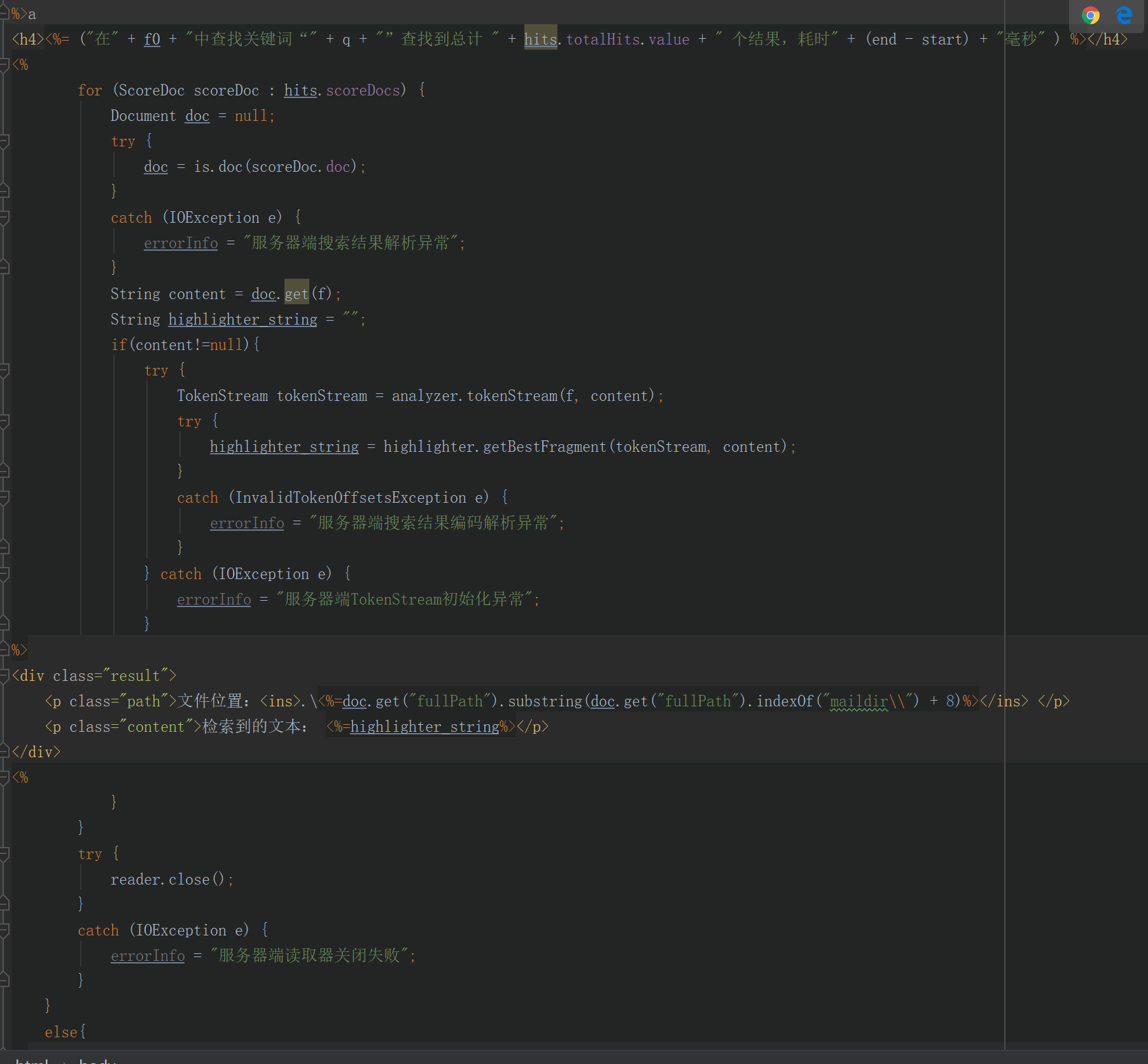
在将get方法传递的参数进行解码和检测异常之后，当参数无误则进入服务器后台开始搜索。

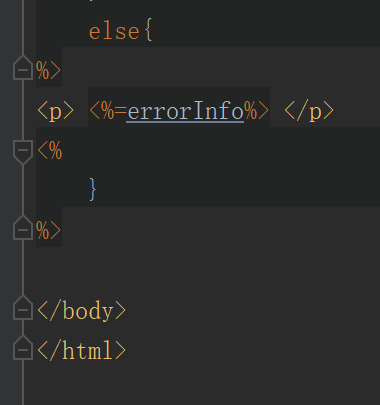


初始化各个搜索对象，并根据传入的搜索类型实例化不同的query对象，以实现不同的查找方式。



使用search进行搜索，将结果利用highlighter类提供的方法进行优化处理后输出。





当检测出参数有异常时，则输出对应的错误信息，跳过服务器端的后台搜索部分。

#### 结果展示

