Information Retrieval Project 1 Pre

src

main

java

servlets

searchs.java

```
package servlets;

import test1.SearchFiles;
import test2.fileeach;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import javax.servlet.RequestDispatcher;
import java.io.IOException;
import java.util.ArrayList;


public class searchs extends javax.servlet.http.HttpServlet {
```

```
    protected void doPost(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws
javax.servlet.ServletException, IOException {

    }

    protected void doGet(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws
javax.servlet.ServletException, IOException {
        String keyword = request.getParameter("keywords");
        String type = request.getParameter("type1");
        if (!keyword.matches("[a-zA-Z\\s]+") && !keyword.equals("")) {

 request.getRequestDispatcher("../jsp/Search_Error2.jsp").forward(request,
response);
        } else if (keyword.length() < 3) {

 request.getRequestDispatcher("../jsp/Search_Error1.jsp").forward(request,
response);
        }else{
            request.setAttribute("keyword1", keyword);
            request.setAttribute("type1", type);

 request.getRequestDispatcher("../jsp/Answer_Page_1.jsp").forward(request,
response);
        }
        System.out.println(keyword);
        System.out.println(type);
    }
}
```

search2.java

```
package servlets;

import test1.SearchFiles;
import test2.fileeach;
import test2.listlala;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
```

```
@WebServlet(name = "search2",urlPatterns = "/search2")
public class search2 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String keyword=request.getParameter("keyword");
        String page=request.getParameter("page");
        String type=request.getParameter("type");
        System.out.println(keyword);
        ArrayList<fileeach> lp3= SearchFiles.indexSearch(keyword,type);
        int yvshu=lp3.size()%5;
        int yv=0;
        if(yvshu>0)yv=1;
        int allpagenumber=lp3.size()/5+yv;
        int page2=Integer.valueOf(page).intValue();

        System.out.println("size"+lp3.size());
        System.out.println("yvshu"+yvshu);
        System.out.println("yv"+yv);
        System.out.println("page"+page2);
        System.out.println("allpage"+allpagenumber);

        if(page2<=allpagenumber&&page2>0){
            request.setAttribute("keyword1",keyword);
            request.setAttribute("page1",page);
            request.setAttribute("type1",type);
            listlala listAll=new listlala();
            listAll.setPage(page2);
            listAll.setAllPage(allpagenumber);
            listAll.setKeyword(keyword);
            if(page2<allpagenumber){
                ArrayList<fileeach> tem=new ArrayList<>();
                for(int ii=(page2-1)*5;ii<page2*5;ii++)
                {
                    tem.add(lp3.get(ii));
                }
                listAll.setContent(tem);
            }
            else {
                ArrayList<fileeach> tem=new ArrayList<>();
                for(int ii=(page2-1)*5;ii<lp3.size();ii++)
                {
                    tem.add(lp3.get(ii));
                }
```

```
                listAll.setContent(tem);


            }

            request.setAttribute("pagesize2",lp3.size());
            request.setAttribute("listAll2",listAll);

 request.getRequestDispatcher("/jsp/Answer_Page_2.jsp").forward(request,res
ponse);
        }
        else{
System.out.println("yvshu"+yvshu);
            System.out.println("yv"+yv);
            System.out.println("page"+page2);
            System.out.println("allpage"+allpagenumber);
            System.out.println("size"+lp3.size());
//request.getRequestDispatcher("../html/Members.html").forward(request,resp
onse);
        }
    }
}
```

test1

SearchFiles.java

```
package test1;

//import java.nio.file.Paths;

import java.io.*;
import java.nio.file.Paths;
import java.util.ArrayList;

import org.apache.lucene.analysis.Analyzer;
import org.apache.lucene.analysis.TokenStream;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.document.Document;
import org.apache.lucene.index.DirectoryReader;
import org.apache.lucene.index.IndexReader;
import org.apache.lucene.index.Term;
import org.apache.lucene.queryparser.classic.QueryParser;
import org.apache.lucene.search.*;
import org.apache.lucene.search.highlight.*;
```

```
import org.apache.lucene.store.Directory;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.util.Version;
import test2.fileeach;


public class SearchFiles {
    public static Version luceneVersion = Version.LATEST;

    public static ArrayList<fileeach> indexSearch(String keywords, String
type1) {
        //String res = "";

        ArrayList<fileeach> filelist = new ArrayList<fileeach>();
        try {

//            1、创建Directory
            FSDirectory directory =
FSDirectory.open(Paths.get("E:\\server\\apache-tomcat-
9.0.12\\webapps\\IR_Project1\\index"));//在硬盘上生成Directory
//            2、创建IndexReader
            IndexReader reader = DirectoryReader.open(directory);
//            3、根据IndexWriter创建IndexSearcher
            //System.out.println(reader.numDocs());
            IndexSearcher searcher = new IndexSearcher(reader);
//            4、创建搜索的query
//            创建parse用来确定搜索的内容，第二个参数表示搜索的域
            QueryParser parser = new QueryParser(type1, new
StandardAnalyzer());//content表示搜索的域或者说字段
            Analyzer analyzer1 = new StandardAnalyzer();
            Query query = parser.parse(keywords);//被搜索的内容
//            5、根据Searcher返回TopDocs
            TopDocs tds = searcher.search(query, 3000);//查询20条记录
//            6、根据TopDocs获取ScoreDoc
            ScoreDoc[] sds = tds.scoreDocs;
            int cou = 0;
            for (ScoreDoc sd : sds) {
                cou++;
                fileeach new1 = new fileeach();
                Document d = searcher.doc(sd.doc);
                // System.out.println("哈哈哈"+d.get("filePath"));
                String text1 = d.get(type1);
                SimpleHTMLFormatter simpleHTMLFormatter = new
SimpleHTMLFormatter("<span style=\"background-color: yellow\"><b>", "</b>
</span>");
                Highlighter highlighter = new
Highlighter(simpleHTMLFormatter, new QueryScorer(query));
                highlighter.setTextFragmenter(new SimpleFragmenter(500));
                if (text1 != null) {
```

```
                    TokenStream tokenStream = analyzer1.tokenStream(type1,
new StringReader(text1));

                    String highLightText =
highlighter.getBestFragment(tokenStream, text1);

                    // System.out.println(highLightText);
                    if (highLightText != null)
                        new1.setHighlight(highLightText);
                    else
                        new1.setHighlight("\n\n");
                }
                new1.setFilename(d.get("fileName"));
                new1.setFilepath(d.get("filePath"));
                new1.setAuthor(d.get("author"));
                new1.setTitle(d.get("title"));
                new1.setAffiliation(d.get("affiliation"));
                new1.setDate(d.get("date"));
                new1.setFulltext(type1);
                new1.setPage(d.get("page5"));
                filelist.add(new1);
            }
            //System.out.println(cou);
            reader.close();
            return filelist;

        } catch (Exception e) {
            e.printStackTrace();
            return filelist;
        }

    }

    public static String indexSearch2(String keywords, String returnfile,
String type1) {
        String res = "";
        try {

//            1、创建Directory
            FSDirectory directory =
FSDirectory.open(Paths.get("E:\\server\\apache-tomcat-
9.0.12\\webapps\\IR_Project1\\index"));//在硬盘上生成Directory
//            2、创建IndexReader
            IndexReader reader = DirectoryReader.open(directory);
//            3、根据IndexWriter创建IndexSearcher
            //System.out.println(reader.numDocs());
            IndexSearcher searcher = new IndexSearcher(reader);
//            4、创建搜索的query
//            创建parse用来确定搜索的内容，第二个参数表示搜索的域
```

```
            QueryParser parser = new QueryParser(type1, new
StandardAnalyzer());//content表示搜索的域或者说字段
            Query query = parser.parse(keywords);//被搜索的内容
//            5、根据Searcher返回TopDocs
            TopDocs tds = searcher.search(query, 18);//查询20条记录
//            6、根据TopDocs获取ScoreDoc
            ScoreDoc[] sds = tds.scoreDocs;
//            7、根据Searcher和ScoreDoc获取搜索到的document对象
            int cou = 0;
            for (ScoreDoc sd : sds) {
                cou++;
                Document d = searcher.doc(sd.doc);
//                    8、根据document对象获取查询的字段值
                res += d.get(returnfile);
            }
            reader.close();
            return res;

        } catch (Exception e) {
            e.printStackTrace();
            return res;
        }

    }

    public static String realPath(String path) {
        path = path.replace(".xml", "");
//        path = "/Users/alexsun/IdeaProjects/Information_Retrieval/data/"
+ path;
        return path;
    }

    public static void main(String[] args) throws IOException {
        ArrayList<fileeach> ooo = indexSearch("math", "fulltext");
        for (int i = 0; i < ooo.size(); i++) {

            System.out.println(ooo.get(i).getTitle());
            System.out.println(ooo.get(i).getAuthor());
            System.out.println(ooo.get(i).getHighlight());
            System.out.println("****************************");
        }
    }
}
```

test2

fileeach.java

```
package test2;
```

```java
public class fileeach {
    private String filename;
    private String filepath;
    private String title;
    private String author;
    private String date;
    private String affiliation;
    private String address;
    private String page;
    private String highlight;



    private String fulltext;
    public String getHighlight() {
        return highlight;
    }

    public void setHighlight(String highlight) {
        this.highlight = highlight;
    }
    public String getPage() {
        return page;
    }
    public void setPage(String page2) {
        this.page = page2;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    public String getAffiliation() {
        return affiliation;
    }
```

```
    public void setAffiliation(String affiliation) {
        this.affiliation = affiliation;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getFulltext() {
        return fulltext;
    }
    public void setFulltext(String fulltext) {
        this.fulltext = fulltext;
    }
    public String toString(){
        return "fulltext [title="+title+", authors="
                +author+",date="+ date+", affiliation="+affiliation+",
address="+address+" and fulltext="+fulltext+"]";
    }
    public String getFilename() {
        return filename;
    }
    public void setFilename(String filename) {
        this.filename = filename;
    }
    public String getFilepath() {
        return filepath;
    }
    public void setFilepath(String filepath) {
        this.filepath = filepath;
    }
}
```

listlala.java

```
package test2;

import java.util.ArrayList;

public class listlala {
    private String keyword;
    private  int page;
    private int allPage;
    private ArrayList<fileeach> content;

    public int getAllPage() {
        return allPage;
```

```
    }

    public void setAllPage(int allPage) {
        this.allPage = allPage;
    }

    public ArrayList<fileeach> getContent() {
        return content;
    }

    public void setContent(ArrayList<fileeach> content) {
        this.content = content;
    }

    public int getPage() {
        return page;
    }

    public void setPage(int page) {
        this.page = page;
    }

    public String getKeyword() {
        return keyword;
    }

    public void setKeyword(String keyword) {
        this.keyword = keyword;
    }
}
```

Grobid

main.py

```
#coding:utf-8
import os
import requests
from time import sleep
# filrname=r"某个文件，如 D:\download\tw-springer-050x.pdf"
# # 请求URL
# url="http://localhost:8070/api/processFulltextDocument"
# #构造请求数据
# params = dict(input=open('./datas/'+filename, 'rb'))
# # post给服务器，并得到返回
# tei = requests.post(url, files=params,timeout=300)
# # 保存文件
```

```
# fh = open(r"E:\py\using_grobid\to\%s.xml"%(filename), 'w', encoding="utf-8")
# fh.write(tei.text)
# fh.close()

def run(filename):
    try:
        params = dict(input=open('./datas/'+filename, 'rb'))
        tei = requests.post(url, files=params,timeout=300)
        fh = open(r"E:\py\using_grobid\to\%s.xml"%(filename), 'w', encoding="utf-8")
        fh.write(tei.text)
        fh.close()
    except UnicodeDecodeError as e:
        pass
if __name__ == '__main__':
    for root, dirs, files in os.walk(r"E:\py\using_grobid\datas"):
        datas = files
    for root,dirs,files in os.walk(r"E:\py\using_grobid\old"):
        old=files
    for file in datas:
        if file+".xml" in old:
            continue
        print("处理"+file)
        # sleep(10)
        run(file)
        fh = open(r"E:\py\using_grobid\old\%s.xml" % (file), 'w')
        fh.close()
```