Atitit 歌曲年份抓取的nlp ai项目

原理通过百度搜索，抓取第一页数据，正则数字，过滤年份。。

显示格式。。歌曲，年份，年份周围前后40字符，方便核对

通过百科抓取比较准确

红尘情歌

{

"中文名称":"红尘情歌",

"所属专辑":"梦中情人",

"歌曲时长":"04：05",

"发行时间":"2012年6月",

"歌曲原唱":"郑源，蒋姗倍",

"填\u00A0\u00A0\u00A0\u00A0词":"潘龙江",

"编\u00A0\u00A0\u00A0\u00A0曲":"潘龙江",

"音乐风格":"流行",

"歌曲语言":"普通话",

"版权所在":"孔雀唱片"

}

时间煮雨

{

"中文名称":"时间煮雨",

"所属专辑":"小时代电影原声带、小时代3：刺金时代电影原声带",

"歌曲时长":"4:07",

"发行时间":"2013年5月22日",

"歌曲原唱":"郁可唯",

"填\u00A0\u00A0\u00A0\u00A0词":"郭敬明，落落",

"谱\u00A0\u00A0\u00A0\u00A0曲":"武部聪志",

"编\u00A0\u00A0\u00A0\u00A0曲":"刘大江",

"音乐风格":"流行，抒情",

"MV导演":"郭敬明",

"歌曲语言":"普通话",

"介\u00A0\u00A0\u00A0\u00A0质":"数字",

"出\u00A0\u00A0\u00A0\u00A0版":"天娱"

}

未成年本兮

{}

## /bookmarksHtmlEverythingIndexPrj/src/apkg/SearchSongYear.java

**package** apkg;

**import** java.io.File;

**import** java.io.IOException;

**import** java.net.URLEncoder;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**import** java.util.function.Consumer;

**import** org.apache.commons.io.FileUtils;

**import** org.apache.http.client.ClientProtocolException;

**import** org.apache.http.client.methods.CloseableHttpResponse;

**import** org.apache.http.client.methods.HttpGet;

**import** org.apache.http.client.utils.HttpClientUtils;

**import** org.apache.http.client.utils.URLEncodedUtils;

**import** org.apache.http.impl.client.HttpClients;

**import** org.apache.http.util.EntityUtils;

**import** org.jsoup.Jsoup;

**import** org.jsoup.nodes.Document;

**import** org.jsoup.nodes.Element;

**import** org.jsoup.select.Elements;

**import** com.alibaba.fastjson.JSON;

**import** com.attilax.net.http.HttpClientUtilV3t55;

**import** com.google.common.collect.Lists;

**import** com.google.common.collect.Maps;

**import** data.html.html2txtConverter;

**public** **class** SearchSongYear {

**public** **static** **void** main(String[] args) **throws** Exception {

// honchen cinge

String songs = "红尘情歌,时间煮雨,未成年本兮";

// songs = "时间煮雨";

String[] sa = songs.split(",");

**for** (String song : sa) {

Map m = *searchByekae*(song);

// Map m=search(song);

System.***out***.println(JSON.*toJSONString*(m, **true**));

}

}

**private** **static** Map searchByekae(String song) **throws** Exception {

System.***out***.println(song);

String url = "http://baike.baidu.com/item/" + URLEncoder.~~encode~~(song);

String html2 = *infoHtml*(url);

FileUtils.*write*(**new** File("d:\\recy\\htmlbyekae\_firstitem\_.html"), html2);

Map info = *getInfo*(html2);

**return** info;

}

**private** **static** String infoHtml(String url) **throws** Exception, IOException {

String html = HttpClientUtilV3t55.*get*(url, "utf8");

FileUtils.*write*(**new** File("d:\\recy\\htmlbyekae.html"), html);

**try** {

Document doc = Jsoup.*parse*(html);

Element polysemantList = doc.select("ul.polysemantList-wrapper").get(0);

Element item = polysemantList.select("li a").get(0);

String link = item.attr("href");

String url2 = "http://baike.baidu.com" + link;

String html2 = HttpClientUtilV3t55.*get*(url2, "utf8");

**return** html2;

} **catch** (IndexOutOfBoundsException e) {

**return** html;

}

}

**private** **static** Map getInfo(String html) {

Document doc = Jsoup.*parse*(html);

Elements basicInfos = doc.select("dl.basicInfo-block");

Map m = Maps.*newLinkedHashMap*();

**for** (Element e : basicInfos) {

Elements dt = e.select("dt");

Elements dd = e.select("dd");

**for** (**int** n = 0; n < dt.size(); n++) {

m.put(dt.get(n).text(), dd.get(n).text());

}

}

**return** m;

}

**private** **static** Map search(String song) **throws** Exception {

// List<Map> li=Lists.newArrayList();

Map m = Maps.*newLinkedHashMap*();

*search*(song, **new** Consumer<List>() {

@Override

**public** **void** accept(List t) {

m.put(song, t);

}

});

**return** m;

}

**private** **static** **void** search(String song, Consumer<List> consumer) **throws** IOException, ClientProtocolException {

String url = "http://www.baidu.com/s?wd=" + URLEncoder.~~encode~~(song + " 百度百科");

// 执行get请求.

CloseableHttpResponse response = HttpClients.*createDefault*().execute(**new** HttpGet(url));

// 获取响应实体

String html = EntityUtils.*toString*(response.getEntity());

FileUtils.*write*(**new** File("d:\\recy\\htmltmp.html"), html);

// System.out.println(html);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*down is doc.select\r\n\r\n\r\n\r\n");

Document doc = Jsoup.*parse*(html);

Element content\_left = doc.getElementById("content\_left");

List<Map> li = Lists.*newArrayList*();

Elements masthead = content\_left.select("div.c-row");

**for** (Element e : masthead) {

Map m = Maps.*newLinkedHashMap*();

String text = e.text();

m.put("kw", song);

m.put("txt", text);

// System.out.println("\r\n" + text);

Set<String> tel = com.attilax.text.RegexUtil.*getYears*(text, 1972, 2020);

m.put("years", tel);

li.add(m);

// System.out.println(JSON.toJSONString(tel, true));

}

consumer.accept(li);

}

}