ELWG, EI,HUST

CrawerAPI 爬虫

文档指南

修订历史

版本	修订时间	修订人	修订原因
Rev 1.0	2014-1-16	Ivsunshine	创建文档，发布
Rev 1.1	2014-1-17	Ivsunshine	修改 HttpClient 方法

Ivsunshine

2014/1/16 Thursday

目录

概要

CrawerAPI 是为爬取网页内容（尤其是新浪微博）的用户粉丝关系、用户个人信息、用户个人的微博内容而开发的一套 Java API，使用该套 API，用户可以方便的使用 API 定义的接口和方法获取到自己想要的内容。

本 API 以普通网页内容爬取为基准，偏重在新浪微博的数据爬取上。对于普通网页的爬取，一般以 Html 字符串的方式返回，新浪微博的数据一般以数据集的方式返回，用户可以通过程序中的“正则表达式”或 XQuery 工具类实现对网页内容的解析，提取关键内容，供自己的项目或研究用。

此外，本 API 还提供了常见的各种工具类的使用，比如文件读写、数据流转换、调试、数据库操作（目前为 MySQL，今后会扩展 Mongo）等常见的操作类，使本 API 的使用者在最短的时间之内拿到需求的数据而不用关心具体的实现。

包简介

本 API 的包的结构如下所示：

CrawerAPI 包结构
-com.elwg.crawer
-com.elwg.crawer.test
-com.elwg.crawer.weibo
-com.elwg.crawer.weibo.xquery
-com.elwg.tools
-com.elwg.util
-com.elwg.util.weibo

下面对每一个包的结构进行重点说明。

1. 爬取 HttpClient 生成包

   com.elwg.crawer，主要包含 HttpClientUtil 类，该类通过调用不同的函数获取到 HttpClient 的集合，通过该 HttpClient 集合，从而对数据爬取做铺垫。

2. 本 API 测试包

   com.elwg.crawer.test，主要包含 TestHttpClientUtil 类，该类主要测试 HttpClient 的获取是否正确以及与新浪微博相关的数据是否正确。

3. 新浪微博数据爬取实体包

   com.elwg.crawer.weibo，主要包含与新浪微博紧密相关的各种实体和执行

类，包含新浪微博登陆加密、微博用户对象实体、微博消息对象实体、微博粉丝爬取执行类、微博个人用户信息爬取执行类、微博消息爬取执行类。

4. 新浪微博微博内容 XQuery 解析包

   com.elwg.crawer.weibo.xquery，主要包含通过 XQuery 的方式解析新浪微博消息的各种方法和接口，如果惯用正则表达式，可以不使用该 XQuery 方式，推荐使用正则表达式。

5. 本 API 工具包

   com.elwg.tools，主要包含常用的工具，譬如文件操作、字符串操作、网页内容获取、Html 转 XML 等工具。

6. 本 API 常量包

   com.elwg.util，主要包含在公共爬取过程中使用到的常变量，譬如传递的需要获取的 HttpClient 类型（新浪微博和普通网页不同，登陆与非登陆不同）、网页格式编码、当前时间、错误原因和处理方式等。

7. 新浪微博常量包

   com.elwg.util.weibo，主要包含与新浪微博数据获取紧密相关的部分，比如默认的四个微博公众爬取账号、微博粉丝、微博个人信息正则表达式集、微博时间统一化等。

类简介

类简介可以详见，Java Doc。

重点类的介绍如下。

1. com.elwg.crawer. HttpClientUtil

   通过传递 HttpClient 类型、HttpClient 生成数量、微博账户密码集生成对应的 HttpClient 集，详细说明如下。

   当调用 getNormalHttpClient 时，为普通的 HttpClient 集；当调用 getWeiboHttpClient 时，为新浪微博的 HttpClient 集。输入参数 count 为需要获取到的 HttpClient 集的数量，为限制使用，最少为 1 个，最大为 4 个。当传递的 HttpClient 类型为 1（微博数据爬取）时，使用 String[][]{{“usernames”},{”pwds”}}传递进微博爬取所使用的账号和密码，如果用户没有写入，则会默认使用本 API 自带的 4 对账号和密码（由于可能存在多人使用导致账号同时并发或者被封的情况，推荐用户使用自己申请的爬取账号）。

   通过 HttpClient 集可以同时生成多个可以独立运行的 HttpClient，当

爬取的任务量过重的时候，可以进行任务分派，极大的节省了时间的开支。

2. com.elwg.crawer.test. TestHttpClientUtil

   该类主要有 testFansCrawer (int count, String accounts[][])测试粉丝爬取是否正常、testUserInfoCrawer (String accounts[][])测试用户个人信息是否爬取正常、testWeiboStatusCrawer (String accounts[][])测试个人微博消息是否爬取正常、testGetWebUrlContent (String url)测试使用普通 HttpClient 获取网页内容是否显示正常。

3. com.elwg.crawer.weibo.WeiboStatusEntity

   该类通过构造函数 WeiboStatusEntity(HttpClient client)生成对象，对象调用 sendWeiboStatusRequest(String userId, String startTime, String endTime, boolean isShowLog)将用户 userId，想要爬取的微博数据段、是否显示调试信息等输入即可得到 ArrayList<WeiboStatus>的列表。

4. com.elwg.crawer.weibo.WeiboUserFansEntity

   该类和上类的调用方式相同，通过构造函数 WeiboUserFansEntity(HttpClient client) 生成执行类对象，调用 sendFansListRequest(String userId, int fansCount, boolean isShowLog, String patternStr)输入待爬取用户的 userId、想要爬取的数目、是否显示调试信息以及匹配的正则表达式即可得到粉丝的集合。当正则表达式输入出错，API 会自动告诉用户没有获取到数据，并给出解决办法，同时本 API 自带了 2014-1-16 日的粉丝提取正则表达式，只需要填入 null 参数即可调用默认。

5. com.elwg.crawer.weibo.WeiboUserInfoEntity

   该类和上两个类属于同一类型的执行类，在该类中，重点在于正则表达式集的构造。因为对于用户个人信息来说，信息并不是一次性在一个数据段呈现的，因此需要“逐级”爬取信息，目前需要爬取的级数为 7。给定输入参数 null 即可调用 2014-1-16 日的微博个人用户信息提取的正则表达式。

示例

1. 获取普通网页 HttpClient 爬取集

   使用 com.elwg.crawer. HttpClientUtil 的 getNormalHttpClient (int count) 方法，传入（3）即可获取到 3 个 HttpClient 的集合。

使用 API 获取 http://blog.csdn.net/ivsunshine/article/details/7312282 数据

```
HttpClientUtil util = new HttpClientUtil();
String url = “http://blog.csdn.net/ivsunshine/article/details/7312282”;
System.out.println(util.getWebUrlContent(url, Constant.utf8CharSet));
```

2. 获取新浪微博授权的 HttpClient 爬取集

使用 com.elwg.crawer. HttpClientUtil 的 getWeiboHttpClient (int count, String [][]accounts)方法，传入（3, null）或（3, String[][]{{},{}}）即可获取到 3 个 HttpClient 的集合。

使用 API 获取 3 个新浪微博授权的 HttpClient 集合

```
HttpClientUtil util = new HttpClientUtil();
ArrayList<HttpClient> clients = util.getWeiboHttpClient(3, null);
```

获取到授权后的 HttpClient 集合后，即可将大量的爬取任务分别按照规则指派给对应的 HttpClient 类，快速完成爬取任务。

3. 通过 2 获取指定用户粉丝集

以单 HttpClient 为例。

使用 API 获取 1000349667 的用户的前 20 个粉丝集合

```
HttpClientUtil util = new HttpClientUtil();
ArrayList<HttpClient> clients = util.getWeiboHttpClient(1, null);
HttpClient client = clients.get(0);
WeiboUserFansEntity entity = new WeiboUserFansEntity(client);
ArrayList<WeiboUserInfo> userInfo = entity.sendFansListRequest("1000349667", 20, true, null);
```

4. 通过 2 获取指定用户微博消息集

以单 HttpClient 为例。

使用 API 获取 1851127221 的用户 2013-12-31 00:00 到现在的微博集合

```
HttpClientUtil util = new HttpClientUtil();
ArrayList<HttpClient> clients = util.getWeiboHttpClient(1, null);
HttpClient client = clients.get(0);
WeiboStatusEntity entity = new WeiboStatusEntity(client);
ArrayList<WeiboStatus> status = entity.sendWeiboStatusRequest("1851127221", "2013-12-31 00:00", null, true);
```

5. 通过 2 获取指定用户微博个人信息

以单 HttpClient 为例。

使用 API 获取 1000349667 的用户的微博个人详细信息

```
ArrayList<HttpClient> clients = util.getWeiboHttpClient(1, null);
HttpClient client = clients.get(0);
```

```
WeiboUserInfoEntity entity = new WeiboUserInfoEntity(client);
WeiboUserInfo info = entity.sendInfoRequest("1000349667", true, null);
System.out.println(info.getSex());
```

6. 工具类使用（以 2014-1-16 日功能为例）

   2014-1-16 日工具集合如下：文件操作（读入写出等）、获取页面 Html 内容、流读取、将 DOM 保存到文件、将 Html 保存到 XML、Unicode 字符检测与转换和转义；预备添加数据库保存操作等。

常见问题及解决办法

在本 API 中，基本上可能出现的问题，都在 ErrorUtils（com.elwg.util）类中定义了，并且在出现相应的问题的时候使用控制台进行了打印。

如果还有任何问题，可以加入 QQ 群，339163230。

说明

该 API 为 2014-1-16 日写毕，此时新浪模拟登陆、数据正则提取、XQuery 模板提取均可用，但由于新浪微博的登陆机制的更替和页面的频繁更新导致的新问题，可能使 API 获取不到新的数据，此时建议开启调试模式（即将每一个执行类的参数 isShowLog 置 1，按照 API 打印的提示下载“正则表达式测试工具”，下载地址为：http://pan.baidu.com/s/1ntjqyTz）以期编写新的正则表达式，从而获取网页结构变更后的数据。

后记

-Rev1.0

本爬取 API 前前后后总共花了大概 10 天左右的事情，期间由于考试等原因有所暂停，总体逻辑还算清晰，但是其中仍然有一些不完美的地方，比如在爬取用户微博数据的时候要去输入起止时间（至少是起始时间，并且按照 2014-1-16 10:47 的格式输入），当格式不对（包括字符串长度不对）的时候，程序不予处理，因为在时间匹配方面，有太多很自由的输入，程序很难做到全部识别。此外，没有提供查询最近多少条的微博消息的接口，不过可以将起始时间稍微设置久一点，终止时间设置为 null（null 为现在的时间）也可以获取到内容。此外，还有初始

化 HttpClient 的爬取集的时候，每次都需要为新浪微博的账号密码集赋值（当然如果是普通的 HttpClient 为 null 即可），但是总会让人觉得很奇怪，这个接口需要改变，写成独立的接口，用户理解也更加方便一些。

-Rev1.1

通过仔细的考虑，我认为应该将普通登陆的 HttpClient 和微博专用的登陆 HttpClient 分开进行处理，因此拆开成了 getNormalHttpClient 和 getWeiboHttpClient，但是又有一个新的问题，对于微博登陆来说，getWeiboHttpClient 是有意义的，但是对于普通登陆呢？getNormalHttpClient 返回普通的 HttpClient 貌似没有任何的意义，普通的页面更多关注的是页面的结果和结构，这个是下一步需要修改的地方。