

CSDN (http://www.csdn.net)



喜欢 03crawler02 爬取贴吧排名, 制作图片集



原创 2015年10月02日 20:48:03



收藏

422

0

0



评论 接下来一个应用是在贴吧看到的, 一位吧友发了一个关于前三百名的吧友的排名的头像, 不过 在前几天, 没事的时候, 突然看见这个帖子漂了上来, 然后 思考这个问题的思路 后来 使用我的”爬虫框架”, 实现了这个功能



分享 程序输入: 贴吧的名称, 开始的排名, 获取多少个排名

keyWord startIdx offset

主要有四个脚本:

1. 抓取排名页面中 “keyWord吧”, 中吧友的人数, 然后根据每一页显示的人数, 计算总共的页数, 然后根据startIdx, offset 计算需要抓取的页数, 拼接对应排名页面的url传递给下一个脚本
2. 抓取给定的url的页面, 中所有的人的数据, 排名, 吧友名字, 吧友的主页, 吧友的经验值, 其中最重要的是url, 将当前页面所有的吧友的主页url传入下一个脚本
3. 抓取吧友主页的用户头像的连接, 并下载该图片
4. 将收集的吧友图片展示出来, 主要是使用了swings的JPanel



970655147 (http://blog.cs...

+ 关注

(http://blog.csdn.net/u011039332)

码云

0

原创

粉丝

喜欢

164

8

0

(https://giit utm_sour

他的最新文章

更多文章 (http://blog.csdn.net/u011039332)

05 战狼观后感

(/u011039332/article/details/77826165)

25 最近遇到的一些问题汇总

(/u011039332/article/details/77816865)

24 碰到的一个 idea的奇怪的编码问题

(/u011039332/article/details/77073688)

04 关于热血篮球的一个观点

(/u011039332/article/details/75948898)



返回顶部

编辑推荐

最热专栏



Python爬取百度贴吧的图片 (/u011001...

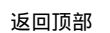
• 深入浅出 •

Docker内部原理 & 网络配置

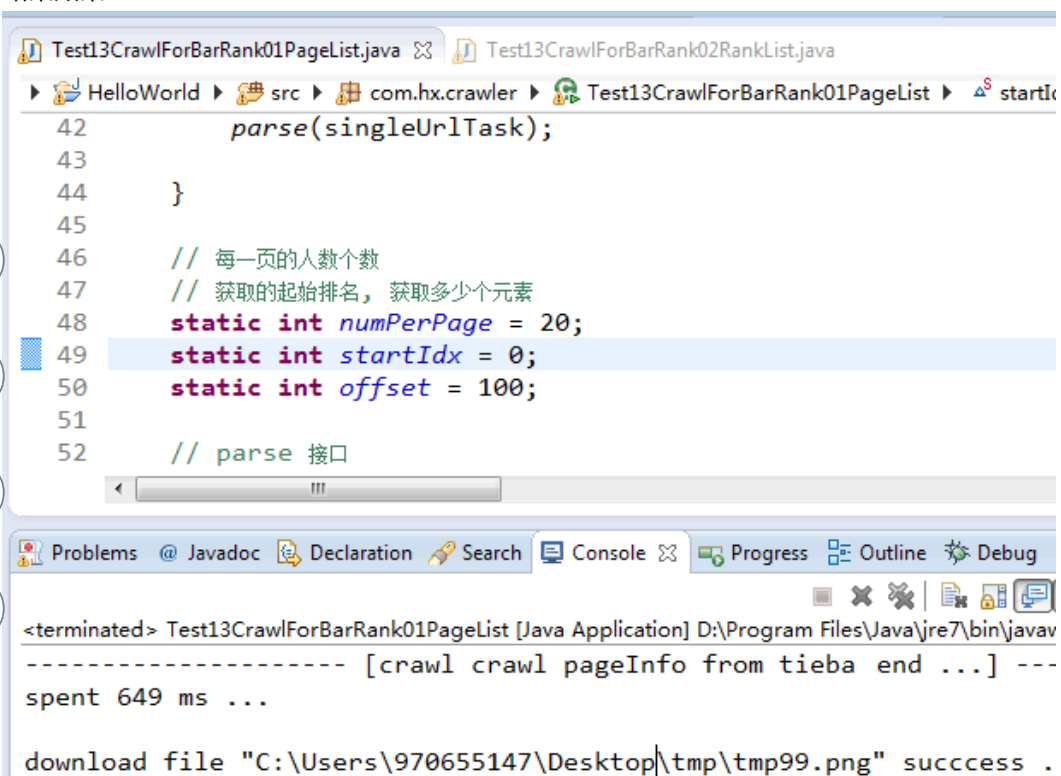
开课时间: 2017.09.21 20:00

log9)

```
utm_source=blog9)
```



结果效果：



The screenshot shows an IDE with two tabs: `Test13CrawlForBarRank01PageList.java` and `Test13CrawlForBarRank02RankList.java`. The active tab is `Test13CrawlForBarRank01PageList.java`, which contains the following code:

```
42     parse(singleUrlTask);
43
44 }
45
46 // 每一页的人数个数
47 // 获取的起始排名, 获取多少个元素
48 static int numPerPage = 20;
49 static int startIdx = 0;
50 static int offset = 100;
51
52 // parse 接口
```

The console output at the bottom shows the following messages:

```
<terminated> Test13CrawlForBarRank01PageList [Java Application] D:\Program Files\Java\jre7\bin\javaw
----- [crawl crawl pageInfo from tieba end ...] ---
spent 649 ms ...

download file "C:\Users\970655147\Desktop\tmp\tmp99.png" success .
```





代码：

脚本1



喜欢



收藏



评论



分享



返回顶部



喜欢



收藏



评论



分享

```
1  /**
2   * file name : Test11CrawlForDouyu01GameList.java
3   * created at : 9:32:38 PM Sep 5, 2015
4   * created by 970655147
5   */
6
7  package com.hx.crawler;
8
9  public class Test13CrawlForBarRank01PageList {
10
11     // 测试站点 : http://tieba.baidu.com/f/like/furank?kw=java&pn=1
12     // 获取贴吧排名信息
13     public static void main(String []args) throws Exception {
14
15         String url = "http://tieba.baidu.com/f/like/furank?kw=%s&pn=%s";
16         Map<String, Object> params = new HashMap<>();
17         params.put(Tools.TASK, "pageList");
18         params.put(Tools.SITE, "tieba");
19         // params.put(Tools.KEY_WORD, "94%C9%E4%B5%F1");
20         params.put(Tools.KEY_WORD, "JAVA");
21         params.put(Tools.PAGE_NO, "1");
22
23         ScriptParameter singleUrlTask = Tools.newSingleTask(HtmlCrawler.newInstance(), url, params);
24
25         parse(singleUrlTask);
26
27     }
28
29     // 每一页的人数个数
30     // 获取的起始排名, 获取多少个元素
31     static int numPerPage = 20;
32     static int startIdx = 0;
33     static int offset = 100;
34
35     // parse 接口
36     // 1. 获取页数
37     // 2. 将分页信息传递给下一个脚本
```



返回顶部



喜欢



收藏



评论



分享

```
38 public static void parse(ScriptParameter scriptParameter) throws Exception {
39     String nextStage = "com.hx.crawler.Test13CrawlForBarRank02RankList";
40
41     SingleUrlTask singleUrlTask = (SingleUrlTask) scriptParameter;
42     Crawler crawler = singleUrlTask.getCrawler();
43     String url = String.format(singleUrlTask.getUrl(), singleUrlTask.getParam().get(Tools.KEY_WORD), singleUrl
44
45     String xpath = "{\\name\\":\\"pageList\\",\\"xpath\\":\\"/html/body/div[@class='d_panel']\\"}\\",\\"values\\":[{\\name
46     xpath = Tools.getRealXPathByXPathObj(xpath);
47
48     // -----1-----
49     StringBuilder sb = new StringBuilder();
50     long start = System.currentTimeMillis();
51     Tools.appendCRLF(sb, "URL : " + url, true);
52     Tools.appendCRLF(sb, "----- [" + Tools.getTaskName(singleUrlTask) + "start ...] -----
53     Log.log(sb.toString());
54
55
56     CrawlerConfig config = new CrawlerConfig();
57     config.setTimeout(30 * 1000);
58     Page page = crawler.getPage(url, config);
59     String html = page.getContent();
60
61     // Log.log(html);
62     // Tools.save(html, Tools.getTmpPath(7));
63     // Tools.getPreparedDoc(url, html, Tools.getTmpPath(7));
64     JSONArray fetchedData = Tools.getResultByXPath(html, url, xpath);
65
66
67     long spent = System.currentTimeMillis() - start;
68     Tools.appendCRLF(sb, fetchedData.toString(), true);
69     Tools.appendCRLF(sb, "----- [crawl" + Tools.getTaskName(singleUrlTask) + "end ...] -----
70     Tools.appendCRLF(sb, "spent " + spent + " ms ...");
71     Log.log(sb.toString());
72
73     // -----2-----
74     if(! fetchedData.isEmpty()) {
```



返回顶部



喜欢



收藏



评论



分享

```
75     JSONArray categories = fetchedData.getJSONObject(0).getJSONArray("pageList");
76     if(categories.size() > 0) {
77         int memberNo = Tools.dealPageNum(categories.getJSONObject(0).getString("memberNo"));
78         int pageSum = Tools.calcPageNums(memberNo, numPerPage);
79         int startPage = Tools.calcPageNums(startIdx, numPerPage) - 1;
80         if(startPage > pageSum) {
81             Log.log("startIdx is not good ...");
82             return ;
83         }
84         int endPage = Math.min(Tools.calcPageNums(startIdx+offset, numPerPage), pageSum);
85
86         JSONObject nextStageParamsTmp = JSONObject.fromObject(singleUrlTask.getParam() );
87         nextStageParamsTmp.put(Tools.TASK, "rankList");
88         for(int i=startPage; i<endPage; i++) {
89             JSONObject nextStageParams = JSONObject.fromObject(nextStageParamsTmp);
90             nextStageParams.put(Tools.URL, singleUrlTask.getUrl() );
91             nextStageParams.put(Tools.PAGE_NO, i+1);
92
93             //         Log.log(nextStageParams);
94             Tools.parse(nextStage, nextStageParams.getString(Tools.URL), nextStageParams, Tools.PARSE_MET
95         }
96     }
97 }
98
99 }
100
101 }
```

脚本2



返回顶部



喜欢



收藏



评论



分享

```
1  /**
2   * file name : Test11CrawlForDouyu01GameList.java
3   * created at : 9:32:38 PM Sep 5, 2015
4   * created by 970655147
5   */
6
7  package com.hx.crawler;
8
9  public class Test13CrawlForBarRank02RankList {
10
11     // 测试站点 : http://tieba.baidu.com/f/like/furank?kw=java&pn=1
12     // 获取贴吧排名信息
13     public static void main(String []args) throws Exception {
14
15         String url = "http://tieba.baidu.com/f/like/furank?kw=%s&pn=%s";
16         Map<String, Object> params = new HashMap<>();
17         params.put(Tools.TASK, "rankList");
18         params.put(Tools.SITE, "tieba");
19         params.put(Tools.KEY_WORD, "java");
20         params.put(Tools.PAGE_NO, "1");
21
22         ScriptParameter singleUrlTask = Tools.newSingleTask(HtmlCrawler.newInstance(), url, params);
23
24         parse(singleUrlTask);
25
26     }
27
28     // parse 接口
29     // 1. 获取目录
30     // 2. 将目录信息传递给下一个脚本
31     public static void parse(ScriptParameter scriptParameter) throws Exception {
32         String nextStage = "com.hx.crawler.Test13CrawlForBarRank03PageInfo";
33
34         SingleUrlTask singleUrlTask = (SingleUrlTask) scriptParameter;
35         Crawler crawler = singleUrlTask.getCrawler();
36         String url = String.format(singleUrlTask.getUrl(), singleUrlTask.getParam().get(Tools.KEY_WORD), singleUrlTask.getPageNo());
37     }
```



返回顶部



喜欢



收藏



评论



分享

```

38 String xpath = "{\"name\":\"rankList\",\"xpath\":\"/html/body//tr[@class='drl_list_item']\",\"values\":{\"na
39 xpath = Tools.getRealXPathByXPathObj(xpath);
40
41 // -----1-----
42 StringBuilder sb = new StringBuilder();
43 long start = System.currentTimeMillis();
44 Tools.appendCRLF(sb, "URL : " + url, true);
45 Tools.appendCRLF(sb, "----- [" + Tools.getTaskName(singleUrlTask) + "start ...] -----
46 Log.log(sb.toString());
47
48
49 CrawlerConfig config = new CrawlerConfig();
50 config.setTimeout(30 * 1000);
51 Page page = crawler.getPage(url, config);
52 String html = page.getContent();
53
54 // Log.log(html);
55 // Tools.save(html, Tools.getTmpPath(7));
56 // Tools.getPreparedDoc(url, html, Tools.getTmpPath(7));
57 JSONArray fetchedData = Tools.getResultByXPath(html, url, xpath);
58
59
60 long spent = System.currentTimeMillis() - start;
61 Tools.appendCRLF(sb, fetchedData.toString(), true);
62 Tools.appendCRLF(sb, "----- [crawl" + Tools.getTaskName(singleUrlTask) + "end ...] -----
63 Tools.appendCRLF(sb, "spent " + spent + " ms ...");
64 Log.log(sb.toString());
65
66 // -----2-----
67 if(! fetchedData.isEmpty()) {
68     JSONArray categories = fetchedData.getJSONObject(0).getJSONArray("rankList");
69     JSONObject nextStageParamsTmp = JSONObject.fromObject(singleUrlTask.getParam());
70     nextStageParamsTmp.put(Tools.TASK, "pageInfo");
71     for(int i=0; i<categories.size(); i++) {
72         JSONObject game = categories.getJSONObject(i);
73         JSONObject nextStageParams = JSONObject.fromObject(nextStageParamsTmp);
74         nextStageParams.put(Tools.NAME, game.getString("name"));

```



返回顶部

```
75         nextStageParams.put(Tools.URL, game.getString("url"));
76
77     //     Log.log(nextStageParams);
78     Tools.parse(nextStage, nextStageParams.getString(Tools.URL), nextStageParams, Tools.PARSE_METHO
79     }
80 }
81
82 }
83
84 }
```



喜欢



收藏



评论



分享

脚本3



返回顶部



喜欢



收藏



评论



分享

```
1  /**
2   * file name : Test11CrawlForDouyu01GameList.java
3   * created at : 9:32:38 PM Sep 5, 2015
4   * created by 970655147
5   */
6
7  package com.hx.crawler;
8
9  public class Test13CrawlForBarRank03PageInfo {
10
11     // 测试站点 : http://tieba.baidu.com/home/main/?un=%C1%F5%B9%FA%B7%A211&fr=furank
12     // 获取贴吧排名信息
13     public static void main(String []args) throws Exception {
14
15         String url = "http://tieba.baidu.com/home/main/?un=%C1%F5%B9%FA%B7%A211&fr=furank";
16         Map<String, Object> params = new HashMap<>();
17         params.put(Tools.TASK, "pageInfo");
18         params.put(Tools.SITE, "tieba");
19         params.put(Tools.KEY_WORD, "java");
20         params.put(Tools.PAGE_NO, "1");
21
22         ScriptParameter singleUrlTask = Tools.newSingleTask(HtmlCrawler.newInstance(), url, params);
23
24         parse(singleUrlTask);
25
26     }
27
28     // parse 接口
29     // 1. 获取目录
30     // 2. 将目录信息传递给下一个脚本
31     public static void parse(ScriptParameter scriptParameter) throws Exception {
32         SingleUrlTask singleUrlTask = (SingleUrlTask) scriptParameter;
33         Crawler crawler = singleUrlTask.getCrawler();
34         String url = singleUrlTask.getUrl();
35
36         String xpath = "{\\"name\\":\\"infoList\\",\\"xpath\\":\\"/html/body\\",\\"values\\":[{\\"name\\":\\"img\\",\\"xpath\\":\\"./
37         xpath = Tools.getRealXPathByXPathObj(xpath);
```



返回顶部



喜欢

收藏

评论

分享

```
38
39 // -----1-----
40 StringBuilder sb = new StringBuilder();
41 long start = System.currentTimeMillis();
42 Tools.appendCRLF(sb, "URL : " + url, true);
43 Tools.appendCRLF(sb, "----- [" + Tools.getTaskName(singleUrlTask) + "start ...] -----");
44 Log.log(sb.toString());
45
46
47 CrawlerConfig config = new CrawlerConfig();
48 config.setTimeout(30 * 1000);
49 Page page = crawler.getPage(url, config);
50 String html = page.getContent();
51
52 // Log.log(html);
53 // Tools.save(html, Tools.getTmpPath(7));
54 // Tools.getPreparedDoc(url, html, Tools.getTmpPath(7));
55 JSONArray fetchedData = Tools.getResultByXPath(html, url, xpath);
56
57
58 long spent = System.currentTimeMillis() - start;
59 Tools.appendCRLF(sb, fetchedData.toString(), true);
60 Tools.appendCRLF(sb, "----- [crawl" + Tools.getTaskName(singleUrlTask) + "end ...] -----");
61 Tools.appendCRLF(sb, "spent " + spent + " ms ...");
62 Log.log(sb.toString());
63
64 // -----2-----
65 if(! fetchedData.isEmpty()) {
66     JSONArray categories = fetchedData.getJSONObject(0).getJSONArray("infoList");
67     String imgUrl = categories.getJSONObject(0).getString("img");
68     try {
69         Tools.downloadFrom(imgUrl, Tools.getNextTmpPath(Tools.PNG) );
70     } catch (IOException e) {
71         Log.err("error in page : " + imgUrl);
72     }
73 }
74
```



返回顶部

```
75  }  
76  
77  }
```

脚本4



喜欢



收藏



评论



分享


返回顶部



喜欢



收藏



评论



分享

```
1  /**
2   * file name : Test11CrawlForDouyu01GameList.java
3   * created at : 9:32:38 PM Sep 5, 2015
4   * created by 970655147
5   */
6
7  package com.hx.crawler;
8
9  public class Test13CrawlForBarRank04DrawIt {
10
11     // 绘制图片
12     public static void main(String []args) throws Exception {
13
14         Frame frame = new Frame();
15
16         for(int i=0; i<1; i++) {
17             frame.saveImgs(i * GRID_SUM);
18         }
19     }
20 }
21
22 // 常量
23 // 窗口的宽高, 行数, 列数, 显示的元素中枢
24 // 每一个元素的宽高, 结果的开始, 结果的索引[递增]
25 static int FRAMEWIDTH = 720;
26 static int FRAMEHEIGHT = 720;
27 static int ROW = 10;
28 static int COL = 10;
29 static int GRID_SUM = ROW * COL;
30 static int PIXES_PER_ROW = (FRAMEHEIGHT - 30) / ROW;
31 static int PIXES_PER_COL = (FRAMEWIDTH - 10) / COL;
32 static int SUM_START = 1000;
33 static int SUM_IDX = 0;
34 // 有些吧友的图片并没有110      例如 : 无名指七年 80x80
35 // static int IMG_HEIGHT = 110;
36 // static int IMG_WIDTH = 110;
37
```



返回顶部



喜欢



收藏



评论



分享

```
38 // 窗口
39 static class Frame extends JFrame {
40     MyPanel panel;
41
42     // 初始化
43     public Frame() {
44         panel = new MyPanel();    add(panel);
45         this.setVisible(true);
46         this.setBounds(100, 100, FRAMEWIDTH, FRAMEHEIGHT);
47         this.setTitle("DrawImg");
48         this.setDefaultCloseOperation(EXIT_ON_CLOSE);
49         this.setResizable(false);
50     }
51
52     // 绘制图片
53     public void saveImgs(int idx) {
54         // 绘制图片
55         Image img = this.createImage(this.getWidth(), this.getHeight() );
56         this.panel.initImgs(idx);
57         this.panel.paint(img.getGraphics() );
58         img.flush();
59         try {
60             ImageIO.write((RenderedImage)img, "png", new File(Tools.getTmpPath(SUM_START + (SUM_IDX ++),
61             Log.log("save img success ...");
62         } catch (IOException e) {
63             e.printStackTrace();
64         }
65     }
66 }
67
68 // 主面板
69 static class MyPanel extends JPanel {
70     // 所有需要展示的图片
71     static Image[] imgs;
72
73     // 初始化, 获取所有的需要展示的图片
74     static {
```



返回顶部



喜欢



收藏



评论



分享

```
75     imgs = new Image[GRID_SUM];
76     initImgs(0);
77 }
78
79 // 绘制所有的图片
80 @Override
81 public void paint(Graphics g) {
82     int idx = 0;
83     for(int col=0; col<COL; col++) {
84         for(int row=0; row<ROW; row++) {
85             g.drawImage(imgs[idx], getRowIdx(row), getColIdx(col), getRowIdx(row+1), getColIdx(col+1), 0, 0, i
86             idx ++;
87         }
88     }
89 }
90
91 // 获取第row行的相对于panel顶部
92 private int getRowIdx(int row) {
93     return row * PIXES_PER_COL;
94 }
95 private int getColIdx(int col) {
96     return col * PIXES_PER_ROW;
97 }
98
99 // 初始化图片
100 private static void initImgs(int idx) {
101     int picNum = GRID_SUM;
102     String suffix = Tools.PNG;
103
104     try {
105         for(int i=0; i<picNum; i++) {
106             imgs[i] = ImageIO.read(new File(Tools.getTmpPath(idx, suffix)));
107             idx ++;
108         }
109         // swallow it
110     } catch (IOException e) {
111         // e.printStackTrace();
112     }
```



返回顶部

```
112         Log.err(Tools.getTmpPath(idx, suffix) + " is not an valid png picture ..");
113     }
114 }
115
116 }
117
118 }
```



喜欢



收藏



评论

版权声明：本文为博主原创文章，未经博主允许不得转载。

举报



分享

标签：爬虫 (<http://so.csdn.net/so/search/s.do?q=爬虫&t=blog>) / 贴吧 (<http://so.csdn.net/so/search/s.do?q=贴吧&t=blog>) /





相关文章推荐

03crawler02 爬取贴吧排名, 制作图片集 (</u011039332/article/details/48862839>)



返回顶部

接下来一个应用是在贴吧看到的, 一位吧友发了一个关于前三百名的吧友的排名的头像, 不过 在前几天, 没事的时候, 突然看见这个帖子漂了上来, 然后 思考这个问题的思路 后来 使用我的"爬虫框架", 实...


 u011039332 (<http://blog.csdn.net/u011039332>) 2015-10-02 20:48  422

用Python 爬虫爬取贴吧图片 (/u012705410/article/details/47685417)



之前一直在看机器学习, 遇到了一些需要爬取数据的内容, 于是稍微看了看Python爬虫, 在此适当做一个记录。我也没有深入研究爬虫, 大部分均是参考了网上的资源。先推荐两个Python爬虫的教程, 网址分别是h...



u012705410 (<http://blog.csdn.net/u012705410>) 2015-08-15 20:34  4604

收藏



评论





精选: 深入理解 Docker 内部原理及网络配置 (http://edu.csdn.net/huiyiCourse/detail/563?utm_source=blog10)



网络绝对是任何系统的核心, 对于容器而言也是如此。Docker 作为目前最火的轻量级容器技术, 有很多令人称道的功能, 如 Docker 的镜像管理。然而, Docker的网络一直以来都比较薄弱, 所以我们有必要深入了解Docker的网络知识, 以满足更高的网络需求。

用Python 爬虫爬取贴吧图片 (/fu_shuwu/article/details/53241672)

<http://blog.csdn.net/u012705410/article/details/47685417> 之前一直在看机器学习, 遇到了一些需要爬取数据的内容, 于是稍微看了看Pytho...



 fu_shuwu (http://blog.csdn.net/fu_shuwu) 2016-11-20 16:41  1056

python:爬取百度贴吧图片 (/xuezhangjun0121/article/details/73719059)

coding:utf-8 import requests import urllib2 import urllib导入lxml库的 etree类, 用来将html文件转换为 HTML DOM, 方便使用 X...



返回顶部



 xuezhangjun0121 (<http://blog.csdn.net/xuezhangjun0121>) 2017-06-25 17:18  82

Python爬取百度贴吧的图片 (/u011001723/article/details/42980411)

Python是一个弱类型的动态语言 下面是我的第一个简单的爬虫脚本程序 #coding=gbk #导入re和urllib两个库 import re import urllib #定义一个有...



喜欢

 u011001723 (<http://blog.csdn.net/u011001723>) 2015-01-21 21:50  778



收藏

python爬取百度贴吧图片 (/codingmyself/article/details/52692948)



评论

由于我比较喜欢图片吧上面他们发的图片，但是一个帖子里都有几百几千张图片，所以就想做一个爬虫脚本。用的是urllib2这种方法去获取页面，注意！！这里只能获取静态页面内容，要想获取动态页面内容的话要...



分享

 CodingMyself (<http://blog.csdn.net/CodingMyself>) 2016-09-28 19:00  79

爬取百度贴吧照片 (/daqinjun/article/details/72140745)

```
"""# -- coding: utf-8 -- from bs4 import BeautifulSoup import urllib2 import urllib import recl...
```




daqinjun (<http://blog.csdn.net/daqinjun>) 2017-05-15 14:09  59

Python爬虫入门——爬取贴吧图片 (/z49434574/article/details/51552088)



最近忽然想听一首老歌，“I believe”于是到网上去搜，把几乎所有的版本的MV都看了一遍（也是够无聊的），最喜欢的还是最初版的《我的野蛮女友》电影主题曲的哪个版本，想起女神全智贤，心血来潮，于...




z49434574 (<http://blog.csdn.net/z49434574>) 2016-06-01 00:40  2648




python爬虫小项目：爬取百度贴吧图片 (/funj_/article/details/53054094)

前一段时间，通过对爬虫的学习，完成了爬取百度贴吧帖子和爬取糗事百科段子。看着自己爬取下来的文字，心里也是满满的自豪感！再次打开贴吧，在浏览时发现贴吧也是包含着大量图片，既然爬取下了贴吧文字，那么为何...


 funj_ (http://blog.csdn.net/funj_) 2016-11-06 13:42  911




第一个爬虫(对百度贴吧图片的爬取) (/thewindkee/article/details/50600704)

 喜欢 一直想写爬虫，也想学python，最近看了慕课网爬虫的视频，就自己试着写了个爬虫。单线程爬取，效率很慢，谢谢百度没有封IP 哈哈。我是win7 64位，python3.4,还用到了urllib...

 收藏  thewindkee (<http://blog.csdn.net/thewindkee>) 2016-01-28 14:56  1855



Python爬虫__爬取贴吧图片和文本 (/qq_24421591/article/details/52596076)

 评论 1. 爬取图片1.1 前言这是一个李清照吧<http://tieba.baidu.com/p/3825973883> 里面有楼主上传的书法作品，每一楼的格式大致是这样，文本加上书法图片：...

 分享  qq_24421591 (http://blog.csdn.net/qq_24421591) 2016-09-21 15:50  1766

Python爬去百度贴吧图片 (/mtbaby/article/details/70209729)

描述：用Python爬去百度贴吧图片并保存到本地。 本人刚学爬虫还不是很熟练，其中难点在于正则表达式的理解；#-*- coding:utf-8-*- import urllib import re ...

 MTbaby (<http://blog.csdn.net/MTbaby>) 2017-04-17 12:43  555

python爬取百度贴吧 (/almost_mr/article/details/53546315)

import sys reload(sys) sys.setdefaultencoding('utf-8') import urllib2 import re#----- 处理页面上的...



almost_Mr (http://blog.csdn.net/almost_Mr) 2016-12-09 22:46 177

贴吧爬虫 (/dai_jing/article/details/52459279)

写了一个简单的贴吧爬虫，送上代码：python版本：2.7 #coding:utf-8 __author__ = 'zhengjinwei' import sys reload(sys...



dai_jing (http://blog.csdn.net/dai_jing) 2016-09-07 14:32 330

喜欢



爬虫学习笔记一、python3.4爬虫爬取百度贴吧图片 (/lcyong_/article/details/53264411)

收藏使用python爬取百度贴吧图片



LCYong_ (http://blog.csdn.net/LCYong_) 2016-11-21 21:11 233

评论



爬取贴吧上的图片到本地 (/duxu24/article/details/51879961)

分享

获取整个页面数据#coding=utf-8 import urllibdef getHtml(url): page = urllib.urlopen(url) html = page....



duxu24 (<http://blog.csdn.net/duxu24>) 2016-07-11 13:09 1777

Python3爬虫入门之贴吧图片批量获取 (/niliushanls/article/details/51305197)

因为百度贴吧的网页结构较为简单，且无需登录即可读取帖子内容，所以批量获取贴吧图片非常适合用来作为Python爬虫入门的练手项目。本文所用的Python版本为Python3,用到的主要模块如下：...



NiliushaNLS (<http://blog.csdn.net/NiliushaNLS>) 2016-05-03 16:05 505

利用爬虫抓取百度贴吧的图片 (/wisedoge/article/details/51100290)

返回顶部

「近年来，百度贴吧已经成为一些同学生活中必不可少的一部分，利用空闲时间刷刷贴吧、看看帖子也成为一些人的日常。同学们经常会遇到这样一个问题，有一个图片帖，帖子中有大量自己喜欢的图片（如<http://ti...>



WiseDoge (<http://blog.csdn.net/WiseDoge>) 2016-04-08 22:11 601

httpClient抓取百度贴吧图片 (/zcgdsdu/article/details/24723275)



最近在研究httpClient

喜欢



u010122002 (<http://blog.csdn.net/u010122002>) 2014-04-29 17:06 818



收藏



强大的Python 获取贴吧图片 (/liuyuehui110/article/details/72872530)



评论

没想到python是如此强大，令人着迷，以前看见图片总是一张一张复制粘贴，现在好了，学会python就可以用程序将一张张图片，保存下来。今天逛贴吧看见好多美图，可是图片有点多，不想一张一张地复...



分享

liuyuehui110 (<http://blog.csdn.net/liuyuehui110>) 2017-06-05 23:20 134