

## 使用 Jsoup 抓取车标网各种类型相应车的信息

博客分类：

- [Jsoup](#)

【谷歌翻译，参考可以看官方原文】

**jsoup**: Java 的 HTML 解析器

jsoup 是与现实世界的 HTML 工作的 Java 库。它提供了用于提取和操作数据，使用最好的 DOM，

CSS 和 jquery 的方法很像，而且的 API 很方便。

jsoup 实现了 WHATWG 的 HTML5 规范，并解析 HTML 到同一个 DOM 现代浏览器做。

刮从一个 URL，文件或字符串解析 HTML

发现并提取数据，使用 DOM 遍历或 CSS 选择器 操纵 HTML 元素，属性和文本

对一个安全白名单干净的用户提交的内容，以防止 XSS 攻击 输出 HTML 整洁

jsoup 是专门用来对付 HTML 各品种在野外发现的;从原始和验证，无效标签汤; jsoup 将创建一个明智的解析树。

方法一、`//Document doc=Jsoup.parse(new URL(requestURL), 3000);`

方法二、`// Document doc=Jsoup.connect(requestURL).timeout(5000).get();`

方法三、`//Document doc =`

`Jsoup.connect(requestURL).timeout(3000).userAgent("Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)").get();`

方法四、`Document doc = Jsoup.connect(requestURL).timeout(3000).cookie("auth", "token").userAgent("Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)").get();`

方法五、`String homepage="www.google.com" ;`

`Jsoup.connect( homepage).userAgent("Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)").referrer("www.google.com").get()`

例：

获取维基百科网页，它解析为一个 DOM，并从在新闻栏目中选择头条新闻为元素（在线样品）的列表：

```
Document doc = Jsoup.connect("http://en.wikipedia.org/").get();
Elements newsHeadlines = doc.select("#mp-itn b a");//获取 id 为 mp-itn 下面 b 标签的 a 标签的元素
```

开源 jsoup 是在宽松的 MIT 许可证分发的一个开源项目。源代码可以在 [GitHub](#) 上。入门 [下载 jsoup](#)（版本 1.8.1） 阅读菜谱介绍 享受！ 开发和支持 如果您对如何使用 jsoup，或者有想法将来的发展有任何疑问，请通过邮件列表联系。


如果您发现任何问题，请检查重复之后提交的 bug。 状态 jsoup 是一般的发布。

你抓取得太狠了吧，速度快的话，对方网站服务器会不时有一会无响应，处理如下：

- 1.对方网站有多个 IP 的时候，自己写个分发类，轮流去每个 ip 取。
- 2.对于 1 个 IP 的时候，遇到这种情况，当前线程就自动暂停几秒钟，然后再重试，自动马上重试的话，也有问题。

建议 jsoup 和 httpclient 一起用，httpclient 去抓取信息，jsoup 做分析。上面 2 种处理，用 httpclient 都好解决的，jsoup 的特长在于分析，抓取是 httpclient 的特长。

connectTimeout 和 soTimeout 一般都设置 3 秒就好了，抓取么，用单例就好，多线程的话，更容易遇到 timeout。

Java 代码 

```
1. package ivyy.taobao.com.jsoup;

2. import ivyy.taobao.com.entity.CheBiao;

3. import java.net.URL;

4. import java.util.ArrayList;

5. import java.util.Iterator;

6. import java.util.List;

7.

8. import org.jsoup.Jsoup;

9. import org.jsoup.nodes.Document;
```

```
10. import org.jsoup.nodes.Element;

11. import org.jsoup.select.Elements;

12.

13. /**

14.  *@Date:2015-1-6

15.  *@Author:liangjilong

16.  *@Email:jilongliang@sina.com

17.  *@Version:1.0

18.  *@Description: 使用 Jsoup 抓取车标网各种相应车的信息

19.  */

20. public class JsoupCar {

21.     /**

22.      * @param args

23.      */

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

25.         /**

26.          * 国产，日本，德国，法国，意大利，英国，美国，韩国，其他

27.          */

28.         String [] countryNames={"guochan","riben","deguo","faguo","yidal

            i","yingguo","meiguo","hanguo","qita"};

29.

30.         //遍历获取太多信息估计会超时连接，可以单独一个一个的去设值抓取如
```

```

31.      //List<CheBiao> listsChe=getCheBiaoInfoByHtml("guochan");
32.      for(String countryName:countryNames){
33.          //System.out.println(countryName);
34.          List<CheBiao> listsChe=getCheBiaoInfoByHtml(countryName);
35.          //
36.          int count=1;
37.          for (Iterator iterator = listsChe.iterator(); iterator.hasNext
38.              ()); {
39.              CheBiao cheBiao = (CheBiao) iterator.next();
40.
41.              System.out.println("第"+count+"-----"+cheBiao.getDetailText());
42.
43.              //System.out.println("第"+count+"-----"+cheBiao.getConcise());
44.              count++;
45.          }
46.      }
47.
48.      /**

```


```
49.     * 根据相应的国家品牌的名称去获取车的信息
50.     * @param countryName
51.     * @return
52.     */
53.     public static List<CheBiao> getCheBiaoInfoByHtml(String countryName) throws Exception{
54.         List<CheBiao> listsChe=new ArrayList<CheBiao>();
55.         String url=getUrl(countryName);
56.         Document doc=Jsoup.parse(new URL(url), 3000);//方法一
57.         //Document doc=Jsoup.connect(url).get();//方法二
58.         if(doc!=null){
59.             //处理从页面的 class=expPicA 样式下面的 li 标签
60.             Elements liEls=doc.getElementsByAttributeValue("class", "expPicA").select("li");
61.             for(Element li:liEls){
62.                 CheBiao che=new CheBiao();
63.                 //从 i 标签的 class=iTit 的 a 标签拿出相应的信息内容出来
64.                 String carName=li.select("i[class=iTit]").select("a").text().trim();//获取汽车名称
65.                 String imgSmallSrc=li.select("i[class=iTxt]").select("img").attr("src");//获取汽车图片路径
66.                 String concise=li.select("i[class=iDes]").text().trim();//简要
```

```
67.
68.         String detailUrl=li.select("[class=iPic]").select("a").attr("href
           ");//获取汽车详情的 html 页面连接
69.         Document descDoc=Jsoup.parse(new URL(detailUrl), 300
           0);//方法一
70.         String imgBigSrc="",detailText="";
71.         if(descDoc!=null){
72.             Element article=descDoc.select("div[class=article]").get
               (0);
73.             detailText=article.html();//获取详情信息
74.             imgBigSrc=article.select("img").attr("src");//获取大图片
75.         }
76.
77.         che.setCarName(carName);
78.         che.setConcise(concise);
79.         che.setImgSmallSrc(imgSmallSrc);
80.         che.setDetailUrl(detailUrl);
81.         che.setImgBigSrc(imgBigSrc);
82.         che.setDetailText(detailText);
83.
84.         listsChe.add(che);
85.     }
```

```

86.         return listsChe;
87.     }else{
88.         //html="Network Connect Timeout";
89.     }
90.     return null;
91. }
92.
93.
94.  /**
95.   * 根据相应的国家品牌的名称请求相应的连接
96.   * @param countryName
97.   * @return
98.   */
99. public static String getUrl(String countryName){
100.     return "http://www.pcauto.com.cn/zt/chebiao/"+countryNam
        e;
101. }
102. }

```

Java 代码 

```

1. package ivyy.taobao.com.entity;
2.

```

```
3. import java.io.Serializable;

4.

5. /**

6.  *@Date:2015-1-6

7.  *@Author:liangjilong

8.  *@Email:jilongliang@sina.com

9.  *@Version:1.0

10. *@Description: 实体类

11. */

12. @SuppressWarnings("all")

13. public class CheBiao implements Serializable{

14.     private Integer id;//id 标识

15.     private String carName;//汽车名称

16.     private String concise;//简要说明

17.     private String imgSmallSrc;//汽车小图片路径

18.     private String imgBigSrc;//汽车大图片路径

19.     private String detailUrl;//汽车详情页面路径

20.     private String detailText;//汽车详情描述

21.

22.     /*****get/set*****/

23.     public String getDetailText() {

24.         return detailText;
```



```
25.     }
26.     public void setDetailText(String detailText) {
27.         this.detailText = detailText;
28.     }
29.     public Integer getId() {
30.         return id;
31.     }
32.     public void setId(Integer id) {
33.         this.id = id;
34.     }
35.     public String getCarName() {
36.         return carName;
37.     }
38.     public void setCarName(String carName) {
39.         this.carName = carName;
40.     }
41.     public String getConcise() {
42.         return concise;
43.     }
44.     public void setConcise(String concise) {
45.         this.concise = concise;
46.     }
```

```
47.  public String getImgSmallSrc() {
48.      return imgSmallSrc;
49.  }
50.  public void setImgSmallSrc(String imgSmallSrc) {
51.      this.imgSmallSrc = imgSmallSrc;
52.  }
53.  public String getImgBigSrc() {
54.      return imgBigSrc;
55.  }
56.  public void setImgBigSrc(String imgBigSrc) {
57.      this.imgBigSrc = imgBigSrc;
58.  }
59.  public String getDetailUrl() {
60.      return detailUrl;
61.  }
62.  public void setDetailUrl(String detailUrl) {
63.      this.detailUrl = detailUrl;
64.  }
65.
66. }
```