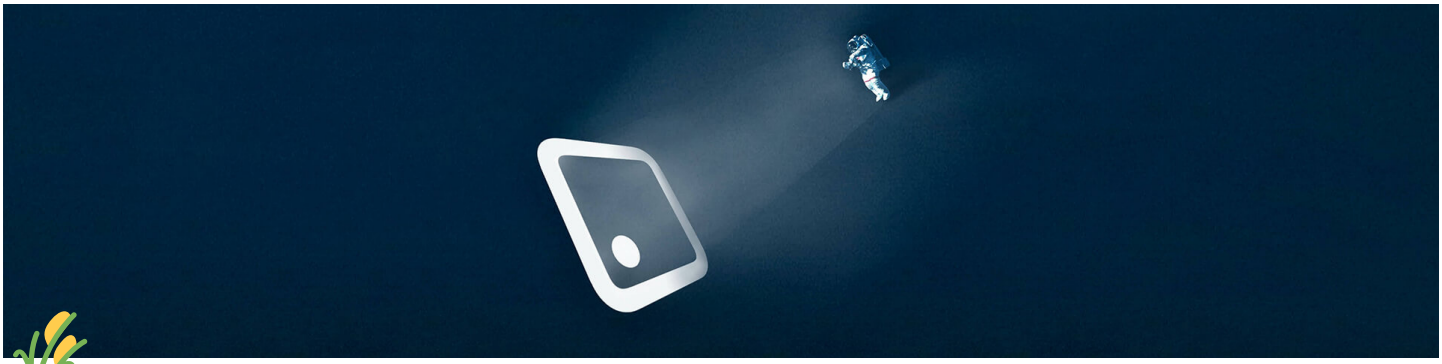

网络知识

爬虫技术

爬虫不要滥用，否则可能会触犯法律！

爬取代码追求以下几点：

- 准确性：可以爬取到，并且数据通过处理正好是自己需要的。
- 稳定性：可以全部爬到而不崩溃。
- 效率：爬取速度快。

跳过异常

如果使用java循环发起大量请求时，可以跳过部分请求失败的地址：

Java

```java
List<String> urlList = new ArrayList<>();
for(String url : urlList){
    try{
        http.get(url);
      }catch(Exception e){
        continue;
    }
}
```

当然也可以将异常的地址存下，待下次再重新获取：

Java

```java
List<String> urlList = new ArrayList<>();
List<String> errList = new ArrayList<>();
for(String url : urlList){
    try{
        http.get(url);
        ...
    }catch(Exception e){
        errList.add(url);
        continue;
    }
}
```

可以使用递归一次性解决：

Java

```java
public static void toGet(List<String> urlList){
    // 无地址
    if(urlList.size() == 0){
        return urlList;
    }
    // 请求并处理
    List<String> errList = new ArrayList<>();
    for(String url : urlList){
        try{
            http.get(url);
            ...
        }catch(Exception e){
            errList.add(url);
```

```
            continue;
        }
    }
    // 递归调用
    toGet(errList);
}
```

注意：此法只是不断重试失败地址，如果地址本身就无法访问，那么就需要进行相应改造。

HTML 解析

Jsoup

jsoup 可以通过 css 选择器来解析 dom 树。

引入

maven 依赖：

```XML
<dependency>
    <groupId>org.jsoup</groupId>
    <artifactId>jsoup</artifactId>
    <version>1.11.3</version>
</dependency>
```

document 解析

获取 dom 树：

```Java
// 从html文本加载dom树
Document document = Jsoup.parse("<html>...</html>");

// 从远程地址获取dom树
Document document = Jsoup.connect(url).get();
```

查询节点集：

```Java
// 使用css选择器进行查询
Elements elements = document.select("#name .value");
```

查询节点：

```Java
// 节点集直接取出
Element element = elements.get(0);

// 使用Dom语法直接获取节点
Element element = document.getElementById("user");
```

获取节点值：

```Java
// 获取html文本
String html = element.html();

// 获取标签名
String tag = element.tagName();

// 获取属性值
String attr = element.attr("href");

// 获取节点文本
String text = element.text();
```

修改节点值：

```Java
// 设置html文本
Element element = element1.html("<div>test</div>");

```

```java
// 设置属性值
Element element = element1.attr("key", "value");

// 设置标签文本值
Element element = element1.text("test");
```

操作节点：

```java
element.append("<div>test</div>");
```

使用示例

解析远程 dom 示例:

```java
public static void jsonDom(String url) throws IOException {
        //Jsoup.parse("");
    Document document = Jsoup.connect(url).get();
    // 获取全部
    Elements elements = document.select(".name .value");
    for (Element element : elements) {
        Elements select = element.select("td");
        String code = select.get(0).text();
        String type = select.get(1).text();
        String name = select.get(2).text();
        System.out.println(code + " | " + type + " | " + name);
    }
}
```

HtmlParser