**数博科技**

数据组-爬虫技术文档

*中国裁判文书网*

|  |  |  |
| --- | --- | --- |
| 2019-05-23 | 1.0 | 陈子昂 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

中国裁判文书网

先理一下总体的反反爬流程

通过爬虫访问中国裁判文书，请求时记得携带固定UA。然后我们会得到一堆加密后的JS代码，通过执行这段JS代码，我们可以获得一个新的URL，通过访问此URL，我们可以得到目标网页，且此网页是携带Cookie的，这Cookie里面有我们需要的参数。

通过此参数，我们可以计算得到后序的POST请求参数，然后携带Cookie，发起一个POST请求，参数即为解析出来的参数。

返回JSON数据，我们可以从中取出两个重要参数，进而得到详情页的URL

中国裁判文书网爬取大致思路：

1. 发起附带”User-Agent”的请求，去访问所需页面，此时会返回得到一段加密的JavaScript代码。此段代码主要作用就是重定向URL，通过解析JS代码，得到重定向所需要的动态URL与参数，手动构造URL，发起附带”User-Agent” 和”Cookie”的请求，去访问新的URL
2. 重定向之后，返回的Cookie中会附带一个”vjkl5”的参数，可通过正则等方法取出具体的值，此数据是加密了的，需要继续解析JS，通过此数据得到POST请求的参数”vl5x”, ”number”, ”guid”这三个参数，进行后序的请求，请求连接为

”http://wenshu.court.gov.cn/List/ListContent”

1. 通过POST请求得到一段JSON数据，里面有两个所需参数

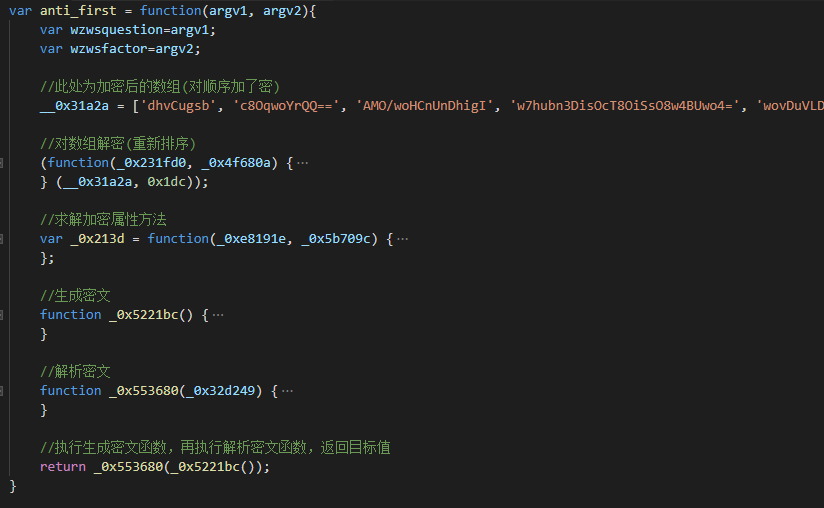
”RunEval”, ”文书ID”

从第一个值是唯一的，文书ID则对应着每篇文章，继续解析JS，其中第一个参数，是为第二个参数准备环境的，而从第二个参数中，我们可以拿到一个”DocID”，这个值就是详情页面所需要的了，详情页面连接为”http://wenshu.court.gov.cn/content/content”

1. 具体的文章内容是异步加载的，如果需要，可以直接访问相应的连接，上面所有访问都需要带上第一步所得Cookie

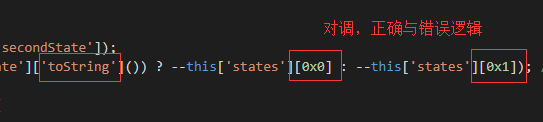
”http://wenshu.court.gov.cn/CreateContentJS/CreateContentJS.aspx”

1、anti\_first



注意事项：

1.1 求解加密属性方法，这里”toString”在本地环境会报错，需要对调源码中正确与错误的处理逻辑



1.2 返回的JS代码中，包含动态URL和一些参数，这些参数需要用于解密，有”wzwsquestion” 和 “wzwsfactor”，这两参数可用正则取出

以下代码都是需要从网页中慢慢扣的，执行过程中解决相应问题即可，如某些函数未定义，则找出其定义函数，copy过来即可



调用方法：

>>>import execjs

>>>with open(“zgcpwsw.js”, “r”) as f\_r:

>>> js\_dode = f\_r.read()

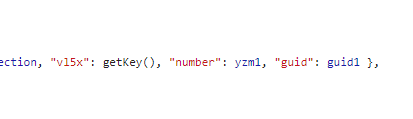
>>> anti\_first = execjs.compile(js\_code)

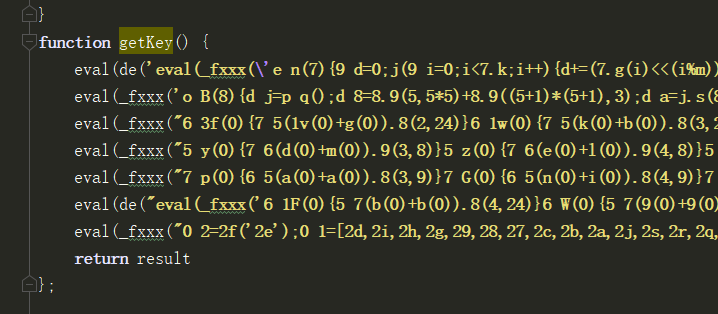
>>>wzwschallange = anti\_first.call(“anti\_first”, argv1, argv2)

其中argv1是”wzwsquestion”而argv2是“wzwsfactor”

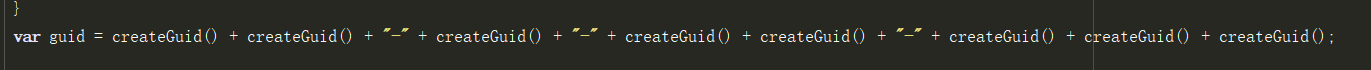
通过第一步构建出的动态URL后，访问得到了带有vjkl5参数的cookie，我们获取此参数后，接着就是要计算vl5x、guid、number这三个重要的参数。

其中这三个参数，分别由对应的函数或变量取得，我们需要详情看看是如何获取的



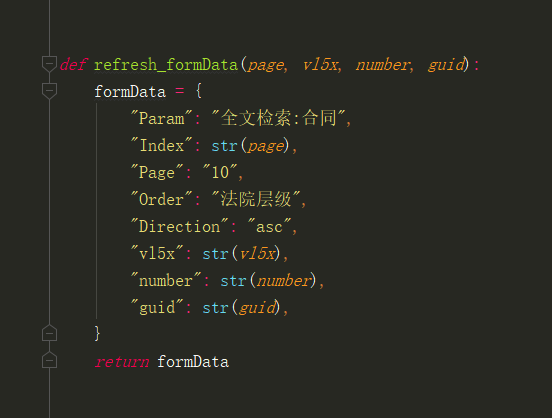


首先是vl5x是通过这么一段得到的，直接执行getKey即可



guid则是一段随机生成的，直接模拟随机生成即可

number则是Math.random函数随机生成的数字，name，我们就可以获取到这三个参数了

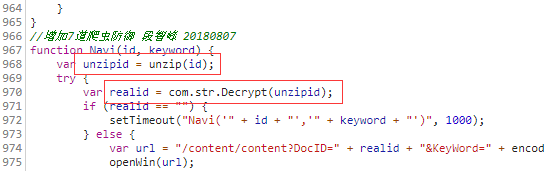


构建请求参数，对<http://wenshu.court.gov.cn/List/ListContent> 列表页发请请求

最后就是计算DocID了，该参数是通过我们获取到的列表页数据返回值计算出来的。从中获取【RunEval】参数和【文书ID】参数

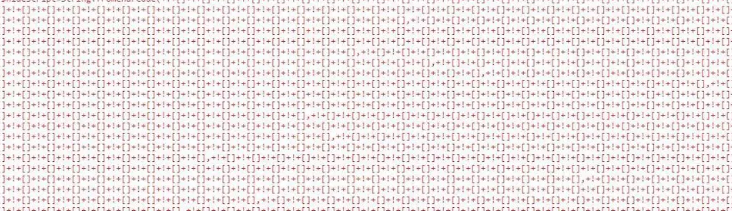


从图中我们可以看出，是通过”javascript:Navi”方法，计算一段加密数据，实现跳转的



最重要的就是上面这来年各个流程，其中unzip和com.str.Decrypt是公用的方法，我们可以通过copy文件实现。但是这里的unzip有一个问题，就是我们获取到的结果居然是完全无法阅读的代码，因为这里是一段匿名函数，我们可以不执行，通过eval构造函数，然后再通过toString函数转化为字符串，我们就可以通过正则取出来了。

当我们获取到了RunEval参数后，通过unzip函数，计算得到一堆不可思议的代码



该函数的实际应该是个匿名函数，也就是我们可以不执行，直接得到可视化的代码

function anonymous() {

setTimeout('com.str.\_KEY="3884522671f7407b88e0c682a17630a8";',8000\*Math.random());

}

这段代码里面就是给com.str\_KEY进行赋值，该值需要用于其他计算

com.str.Decrypt(unzip(“文书ID放到这，进行下一步解密”));

通过最后一步，我们可以得到最终的DocID

我们就可以获得目标详情页的url了