一、知网论文页面的分析

知网期刊论文的爬取，知网已经将各个期刊下的论文规整好了 http://navi.cnki.net/knavi/，所以期刊论文的爬取可以检索所有期刊下的所有论文从而爬取到所有的期刊论文。

首先我们从其中一个期刊入手，爬取该期刊下所有的论文的名字、摘要、作者等信息。

我们选择其中一个期刊 杭州电子科技大学学报 http://navi.cnki.net/knavi/JournalDetail?pcode=CJFD&pykm=HXDY 点击期刊不同年份的不同期可以看到会加载出这一期所有期刊的目录。

图 1-1

实际上当我们点击某个年份下的某一期时，会触发一个请求，请求会该期的所有论文目录，我们可以通过浏览器（例子中用的是 Chrome 浏览器）自带的源代码检查工具查看到请求的具体内容，Chrome 下按 F12 可以查看到网页的源代码和请求的包信息，Elements 可以查看网页的源代码，Network 可以查看请求和返回的包的内容（也可以通过各种抓包软件获取包的内容）。这里我们在开启抓包的情况下，请求图 1-1 的目录内容

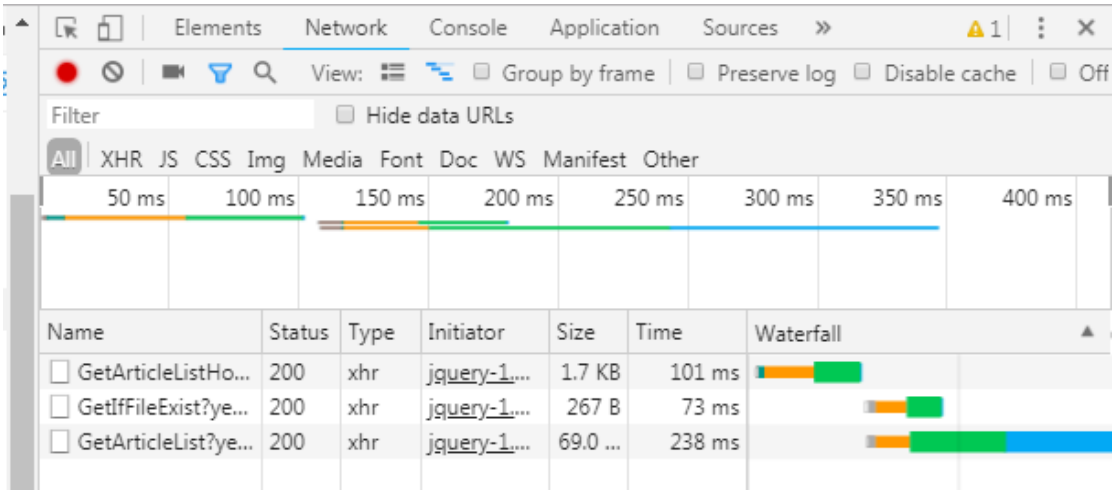


图 1-2

通过查看包的具体内容可以看到其实是向 http://navi.cnki.net/knavi/JournalDetail/GetArticleList?year=2018&issue=04&pykm=HXDY&pageIdx=0&pcode=CJFD 请求目录。其中请求的参数中的 year=2018，issue=04，pageIdx=0 就是我们请求的期刊的年份，第几期，和该目录下的第几页，pykm=HXDY 可以看做是期刊在知网中的标识，这该期刊在知网中的唯一标识，pcode=CJFD 在所有期刊 URL 中都是一样的，只需要在请求的时候加上这个参数就可以了，因此可以想到如果请求其他期刊下某个年份的某一期，只需要知道期刊的标识和年份等信息，比如我们想获得计算机学报（唯一标识 pageIdx=0）2018 年的第 6 期第一页的目录，我们请求的 url 应该为 http://navi.cnki.net/knavi/JournalDetail/GetArticleList?year=2018&issue=06&pykm=JSJX&pageIdx=0&pcode=CJFD 这样只要我们知道期刊在知网的唯一标识，就能获取各个期刊的每一期的目录页面了，知网的标识会在附件源代码的 resource 目录下提供。

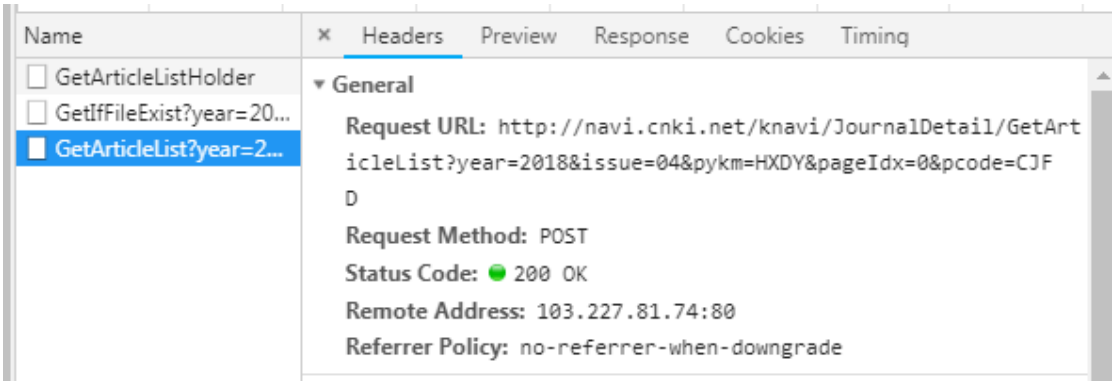


图 1-3

我们通过访问该网站可以看到返回的请求内容就是我们需要的目录信息，通过 html 的分析，发现论文的名字在 target = “_blank”的 a 标签下，该标签下还包含 href 的链接，由于知网论文页面是发生跳转的，所以该链接不能直接使用转到论文的具体页面，需要解析提取出有用的信息，下文会讲解

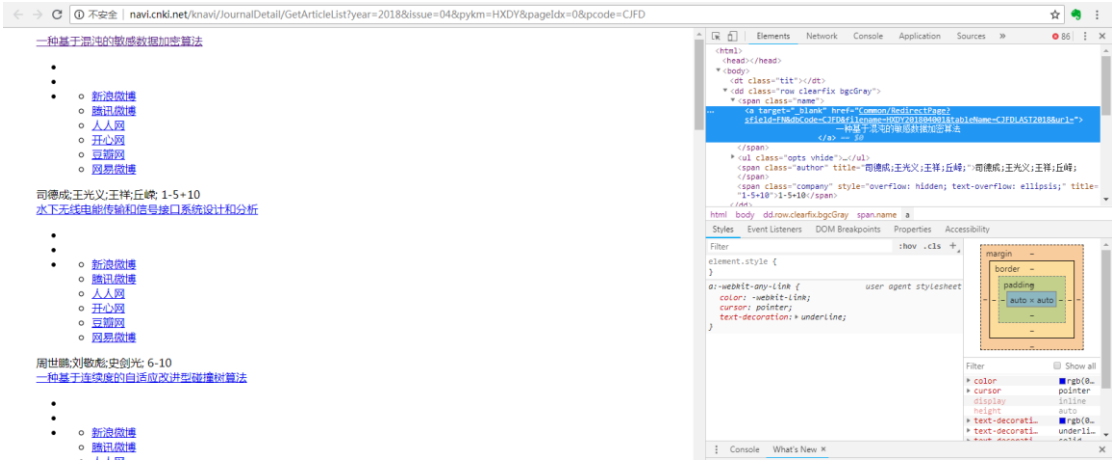


图 1-4

接下来我们需要分析该论文的具体页面，点击图 1-1 中对应的论文可以跳转到该论文相应的页面，以“一种基于混沌的敏感数据加密算法”为例子，跳转到 http://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&filename=HXDY201804001 的页面，仔细看这个网址，结合上面 href 的内容，是不是发现 dbcode=CJFD&filename=HXDY201804001&dbname=CJFDLAST2018 这串内容包含在 a 标签的 href 中，因此我们只需要提取目录页面中的所有这些信息就能拼接得到各个论文的 URL。在论文的具体页面下，就能提取出各个标签中的信息来得到论文的名字，作者，摘要等信息

图 1-5

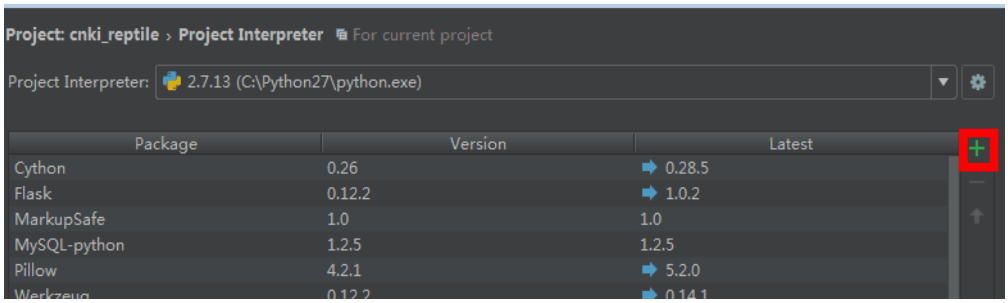
二、代码的实现

本次爬虫程序主要通过 python 编写，编写 python 的 IDE 为 pycharm,该软件可以通过在官网注册并进行学生认证得到免费的使用（https://www.jetbrains.com/pycharm/），当然也可以使用其他的 IDE，IDE 只是为了对于编写程序提供方便。

具体的 python 和 pycharm 的安装网上有很多教程，可以自行百度，这里也不多说了（http://www.runoob.com/python/python-install.html）。本文用的 python 版本为 2.7。

实现的代码放在附件当中，导入附件中的项目可以点击 open 选中 cnki_reptile 文件，项目中需要导入爬虫程序需要用到的包，要用到的包可以用 Pycharm 下载：

File->Default Settings->Default Project->Project Interpreter 中的+按钮，搜索 requests 安装该包，该包是用来采集网页的源代码的，同样的方法我们来安



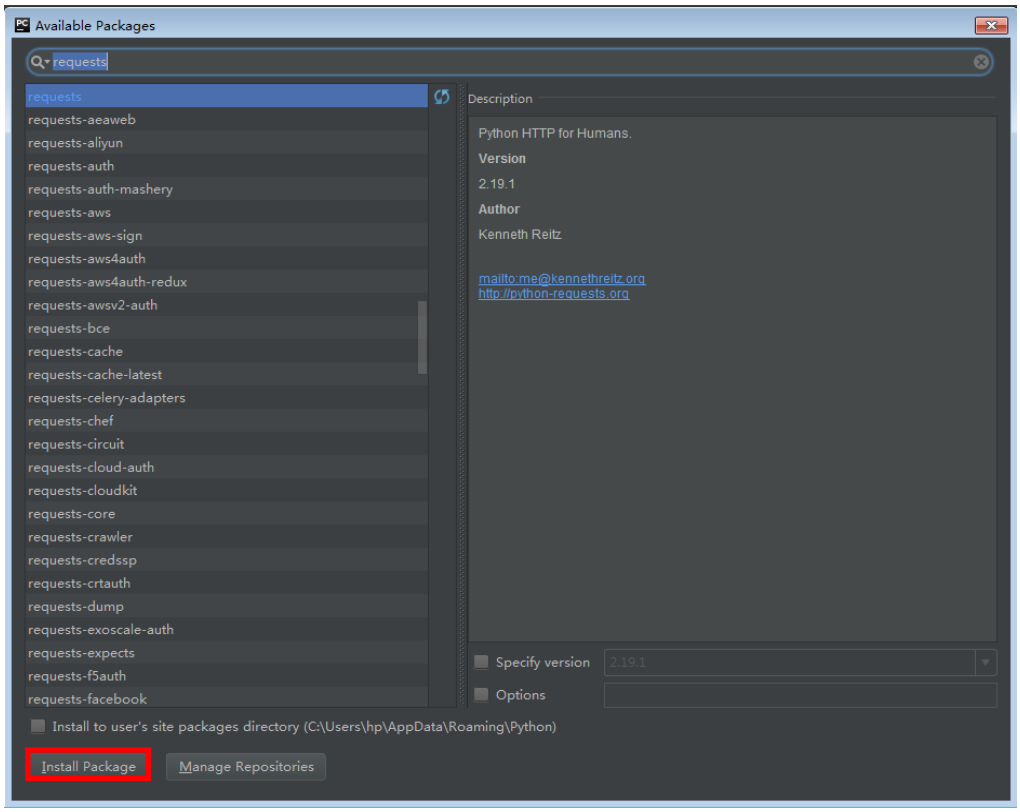
装 lxml。

图 1-1

图 1-2

Requests 包的作用可以向指定的 URL 发送请求，返回网页的源代码。lxml 可以将 html 代码形成类似树的结构，并且支持 xpath 方便我们快速寻找到需要的元素、标签的内容。

在查看代码的同时不明白可以结合百度，谷歌，实现的代码用的都是 pytho



n 的基本语法和 requests 和 lxml 的简单应用，可以在查看代码之前简单熟悉一下这两个库类的使用和 python 的基本语法。在搞明白代码的实现之后，最好能自己动手改动觉的代码中不合理的地方，或者自己重新实现一遍论文的爬取。Demo 中实现的只是一个时间点的期刊论文的爬取，后续可以结合 journalNameC

odes.txt 中的内容爬取该文件下所有的期刊中的论文。