

# 爬虫案例1(校花网)

校花网址:<http://www.xiaohuar.com/hua/>

校花简介信息比如:<http://www.xiaohuar.com/p-1-2016.html>

一.利用scrapy shell 分析页面:

分析主页中每个标题,链接和下一页

```
In [5]: le
Out[5]: <scrapy.linkextractors.lxmlhtml.LxmlLinkExtractor at 0x109cb5048>

In [6]: le = LinkExtractor(restrict_css='div.item_t > div.title > span > a')

In [7]: le
Out[7]: <scrapy.linkextractors.lxmlhtml.LxmlLinkExtractor at 0x109d55198>

In [8]: le.extract_links(response)
Out[8]:
[Link(url='http://www.xiaohuar.com/p-1-2016.html', text='北京交通大学校花高雅',
fragment='', nofollow=False),
 Link(url='http://www.xiaohuar.com/p-1-2015.html', text='吉林大学珠海学院校花余文丽',
fragment='', nofollow=False),
 Link(url='http://www.xiaohuar.com/p-1-2010.html', text='聊城大学校花飞儿',
fragment='', nofollow=False),
 Link(url='http://www.xiaohuar.com/p-1-2009.html', text='陕西科技大学校花左宸怡',
fragment='', nofollow=False),
 Link(url='http://www.xiaohuar.com/p-1-2007.html', text='韶关市田家炳中学校花邓雯',
fragment='', nofollow=False),
 Link(url='http://www.xiaohuar.com/p-1-2005.html', text='鄞州职业高级中学校花翁川美',
fragment='', nofollow=False),
 Link(url='http://www.xiaohuar.com/p-1-2004.html', text='河北医科大学校花孙佳萌']
```

```
In [14]: sel.css('a::attr(href)').extract_first()
Out[14]: 'http://www.xiaohuar.com/p-1-2016.html'

In [15]: sel.css('a::text').extract_first()
Out[15]: '北京交通大学校花高雅'

In [16]: sel = response.css('div.item_t > div.title > span > a')

In [17]: sel.extract_first()
Out[17]: '<a href="http://www.xiaohuar.com/p-1-2016.html" target="_blank">北京交通大学校花高雅</a>'
```

还有下一页

```
In [21]: sel.css('a::attr(href)').extract_first()
Out[21]: 'http://www.xiaohuar.com/list-1-1.html'

In [22]: sel = response.css('div.page_num a:nth-last-child(2)')
```

```
In [23]: sel = response.css('div.page_num a:nth-last-child(2)')
In [24]: sel.css('a::text').extract_first()
Out[24]: '下一页'
```

注意:要满足前提有下一页

分析详情页面: 详细信息分别为 姓名 学校 空间地址

```
In [23]: sel
Out[23]: '北京交通大学'

In [24]: sel = response.css('div.infodiv table tr:nth-child(1) td:nth-child(2)::text').extract_first()

In [25]: sel
Out[25]: '高雅'

In [26]: sel = response.css('div.infodiv table tr:nth-child(7) td:nth-child(2) a::attr(href)').extract_first()

In [27]: sel
Out[27]: 'http://www.xiaohuar.com/s-1-2016.html'

In [28]:
```

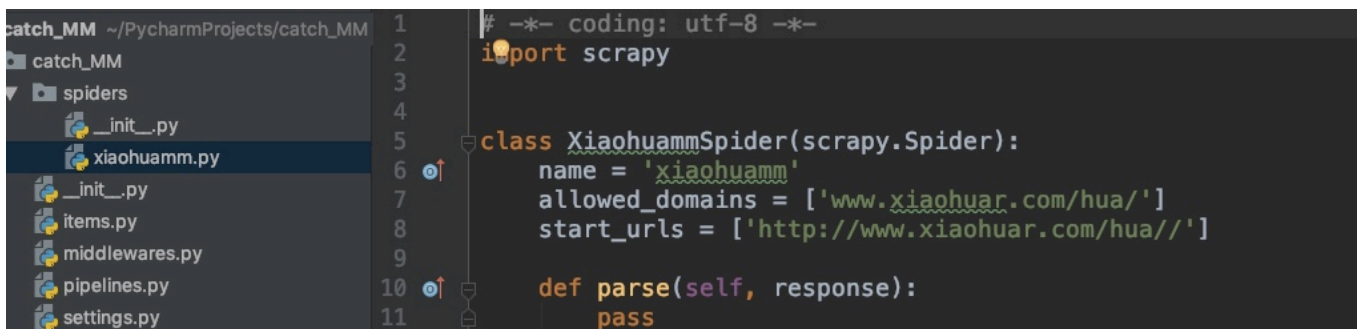
-----分析完毕

## 1.构建 catchMM

```
MichaelYun:~ Yun$ cd PycharmProjects/
MichaelYun:PycharmProjects Yun$ ls
OPP          example          studyPY          toscrape_book
MichaelYun:PycharmProjects Yun$ scrapy startproject catch_MM
New Scrapy project 'catch_MM', using template directory '/usr/local/lib/python3.7/site-packages/scrapy/templates/project', created in:
/Users/Yun/PycharmProjects/catch_MM

You can start your first spider with:
cd catch_MM
scrapy genspider example example.com
MichaelYun:PycharmProjects Yun$ scrapy genspider xiaohuamm www.xiaohuar.com/hua/
Created spider 'xiaohuamm' using template 'basic'
MichaelYun:PycharmProjects Yun$
```

自动构建了基础类



```
1  # -*- coding: utf-8 -*-
2  import scrapy
3
4
5  class XiaohuammSpider(scrapy.Spider):
6      name = 'xiaohuamm'
7      allowed_domains = ['www.xiaohuar.com/hua/']
8      start_urls = ['http://www.xiaohuar.com/hua//']
9
10     def parse(self, response):
11         pass
```

## 2.在item中定义好实体类

```
import scrapy

class CatchMmItem(scrapy.Item):
    title = scrapy.Field() # 校花的标题
    mm_name = scrapy.Field() # 校花的名字
    mm_school = scrapy.Field() # 校花所属学校
    mm_zone = scrapy.Field() # 校花的空间
```

### 3.实现具体的逻辑步骤

```
1 class XiaohuammSpider(scrapy.Spider):
2     name = 'xiaohuamm'
3     # 注意这里这里用于设置允许被访问的网页地址，一般填充一级域名
4     allowed_domains = ['www.xiaohuar.com']
5     start_urls = ['http://www.xiaohuar.com/hua//']
6
7     # 解析校花的列表页面
8     def parse(self, response):
9         # 提取书籍页面的链接列表
10        le = LinkExtractor(restrict_css='div.item_t > div.title > span >a')
11        # 循环列表,每循环一次就解析一次页面
12        for link in le.extract_links(response):
13            yield scrapy.Request(link.url,callback=self.parse_mm)
14        # 循环做完后,提取下一页连接
15        le = LinkExtractor(restrict_css='div.page_num a:nth-last-child(2)')
16        links = le.extract_links(response)
17        if links[0].text == '下一页':
18            next_url = links[0].url
19            yield scrapy.Request(next_url,callback=self.parse)
20
21
22
23
24    # 解析校花的详细信息页面
25    def parse_mm(self, response):
26        # 声明实体
27        mm = CatchMmItem()
28        # 填充数据
29        mm['title'] = response.css('div.div_h1 > h1::text').extract_first()
30        mm['mm_name'] = response.css('div.infodiv table tr:nth-child(1) td:nth-
child(2)::text').extract_first()
31        mm['mm_school'] = response.css('div.infodiv table tr:nth-child(5) td:nth-
child(2)::text').extract_first()
32        mm['mm_zone'] = response.css('div.infodiv table tr:nth-child(7) td:nth-child(2)
a::attr(href)').extract_first()
33        yield mm
```

最后:检查是否可以

```
MichaelYun:~ Yun$ cd PycharmProjects/
MichaelYun:PycharmProjects Yun$ ls
OPP          catch_MM      example      studyPY      toscrape_book
MichaelYun:PycharmProjects Yun$ cd catch_MM/
MichaelYun:catch_MM Yun$ scrapy crawl xiaohuamm -o MMmsg.csv
```

执行 并 输出到 MMmsg.csv

注意：robots.txt 是遵循 Robot协议 的一个文件，它保存在网站的服务器中，它的作用是，告诉搜索引擎爬虫，本网站哪些目录下的网页 不希望 你进行爬取收录。在Scrapy启动后，会在第一时间访问网站的 robots.txt 文件，然后决定该网站的爬取范围。

当然，我们并不是在做搜索引擎，而且在某些情况下我们想要获取的内容恰恰是被 robots.txt 所禁止访问的。所以，某些时候，我们就要将此配置项设置为 False ，拒绝遵守 Robot协议 ！

可选项

```
#指定各项排列顺序
FEED_EXPORT_FIELDS = ['title', 'mm_name', 'mm_school', 'mm_zone']
```