案例2: 阳光热线问政平台

一、想一想

回顾案例1，发现花费了较多的功夫在处理寻找下一页的url地址或者内容的url地址，能不能简化这个过程?

- 思路点拨

1.从response中提取所有的a标签对应的url地址

2.自动构造request请求，发送给引擎

能不能再优化?

满足需求的url地址，才发给引擎，同时能指定callback函数解析响应

二、数据提取需求

网站首页: http://wz.sun0769.com/html/top/report.shtml

共需提取4个字段信息：编号、标题、url、投诉内容

对比列表页和详情页的页面结构，发现需要提取的数据在详情页也可以获取到

那么，可不可以只在列表页获取链接(包括详情页和下一页），而只在详情页做数据解析？可以，这种思想使用的是Scrapy框架中的CrawlSpider类爬虫

三、新建项目

创建阳光热线问政平台爬虫项目

```
$ scrapy startproject Sun
```

四、编写`items.py`

```
import scrapy


class SunItem(scrapy.Item):
    # define the fields for your item here like:
    number = scrapy.Field()      # 编号
    detail_url = scrapy.Field()  # 详情页url
    title = scrapy.Field()       # 标题
    context = scrapy.Field()     # 投诉内容
```

五、创建CrawlSpider爬虫

```
$ cd Sun  # 进入项目目录
$ scrapy genspider -t crawl sun 'sun.com'
```

> 上面的命令参数
>
> -t t代表template，即使用模板
>
> sun 代表爬虫名
>
> 'sun.com' 是起始的url

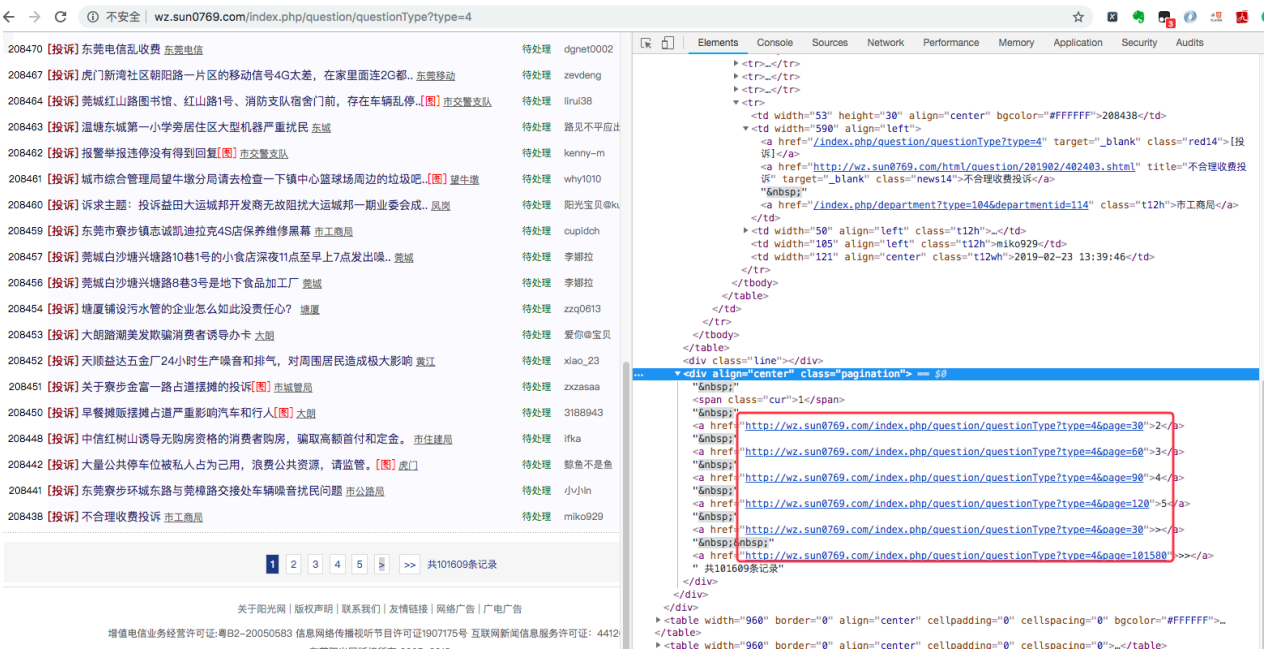
- 修改allowed_domains、start_urls

```
    allowed_domains = ['wz.sun0769.com']
    start_urls =
['http://wz.sun0769.com/index.php/question/questionType?type=4']
```

六、url提取规则

- 提取下一页url

[分析] 才页面上可以观察到列表url都包含路径`questionType`



立即导出，下一页url提取规则

```
        # 下一页url
        # http://wz.sun0769.com/index.php/question/questionType?type=4&page=30
        # http://wz.sun0769.com/index.php/question/questionType?type=4&page=60
        # http://wz.sun0769.com/index.php/question/questionType?type=4&page=90
        Rule(LinkExtractor(allow=r'questionType'), follow=True),
```

其中:

LinkExtractor为链接解析器，接收了2个参数

allow: 值的类型为正则表达式字符串

follow: 值的类型为布尔值

链接解析器的作用：从response中获取链接并解析出url地址，并判断是否满足allow设定的正则规则，如果满足就交给引擎，对该url发起请求，否则就过滤掉，follow参数是设定，请求该url获取到响应后，是否对响应继续做链接解析器过滤。

- 提取详情页url

先观察详情页url

总结得出详情页提取的url

```
# 详情页url
# http://wz.sun0769.com/html/question/201902/402402.shtml
# http://wz.sun0769.com/html/question/201902/402398.shtml
Rule(LinkExtractor(allow=r'http://wz.sun0769.com/html/question/\d+/\d+.shtml'),callback='parse_item', follow=False),
```

这里出现了链接解析器的第三个参数——callback，没错了，这就是指定解析数据的函数，但是注意了，其参数类型是字符串！！！！还需要注意的是这里的follow参数为False，即解析完数据后，直接将数据交给引擎，不必再在对当前页面做链接解析器过滤了。

七、解析详情页

编号

```
# 解析编号
item['number'] =
response.xpath('//tr/td[2]/span[2]/text()').extract_first().split(':')[
    -1].strip()
```

标题

```
# 解析标题
item['title'] =
response.xpath('/html/head/title/text()').extract_first().split('_')[0]
```

url

```
# 获取详情url
item["detail_url"] = response.url
```

投诉内容

注意，这里有坑了，投诉内容可能会包含图片，那么解析规则就会不同

```
# 获取是否有图片
img = response.xpath('//td[@class="txt16_3"]/div/img')
if img:
# 有图片时，解析投诉内容
text =
response.xpath('//div[@class="contentext"]/text()').extract()
else:
# 无图片时，解析投诉内容
text = response.xpath('//td[@class="txt16_3"]/text()').extract()
item["context"] = "".join([x.strip() for x in text if not
x.isspace()
```

爬虫部分`sun.py`完整代码

```
from scrapy.linkextractors import LinkExtractor
from scrapy.spiders import CrawlSpider, Rule

from Sun.items import SunItem


class SunSpider(CrawlSpider):
    name = 'sun'
    allowed_domains = ['wz.sun0769.com']
    start_urls =
['http://wz.sun0769.com/index.php/question/questionType?type=4']

    rules = (
        # 下一页url
        # http://wz.sun0769.com/index.php/question/questionType?
type=4&page=30
        # http://wz.sun0769.com/index.php/question/questionType?
type=4&page=60
        # http://wz.sun0769.com/index.php/question/questionType?
type=4&page=90
        Rule(LinkExtractor(allow=r'questionType'), follow=True),
        # 详情页url
```

```
        #
http://wz.sun0769.com/html/question/201902/402402.shtml
        #
http://wz.sun0769.com/html/question/201902/402398.shtml

 Rule(LinkExtractor(allow=r'http://wz.sun0769.com/html/question/
\d+/\d+.shtml'), callback='parse_item',
              follow=False),
    )

    def parse_item(self, response):
        # 创建item对象
        item = SunItem()
        # 解析编号
        item['number'] =
response.xpath('//tr/td[2]/span[2]/text()').extract_first().spli
t(':')[
            -1].strip()
        # 解析标题
        item['title'] =
response.xpath('/html/head/title/text()').extract_first().split(
'_')[0]
        # 获取详情url
        item["detail_url"] = response.url
        # 获取是否有图片
        img = response.xpath('//td[@class="txt16_3"]/div/img')
        if img:
            # 有图片时，解析投诉内容
            text =
response.xpath('//div[@class="contentext"]/text()').extract()
        else:
            # 无图片时，解析投诉内容
            text =
response.xpath('//td[@class="txt16_3"]/text()').extract()
        item["context"] = "".join([x.strip() for x in text if
not x.isspace() and x])
        print("---" * 50)
```

```
        print(item)
        print("---" * 50)
        yield item
```

八、数据持久化

数据存储到MongoDB，请同学们自行完成

九、启动爬虫

```
$ scrapy crawl sun
```