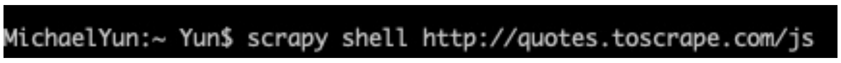
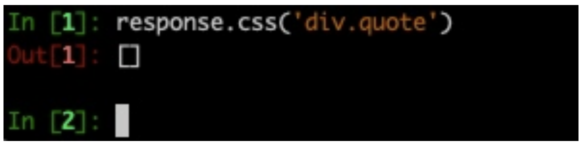
爬虫案例4(解决异步加载的问题，爬取动态页面)

1.我们先观察下这个网站



```
MichaelYun:~ Yun$ scrapy shell http://quotes.toscrape.com/js
```



```
In [1]: response.css('div.quote')
Out[1]: []

In [2]: 
```

查看源码，你会发现代码是通过js加载出来的

当然大部分的网页是通过异步ajax加载的

-------使用splash

https://www.cnblogs.com/cnkai/p/7403362.html

splash渲染引擎需要在Docker的环境下使用。上面这篇文章就是在windows下

假定环境都OK的情况下

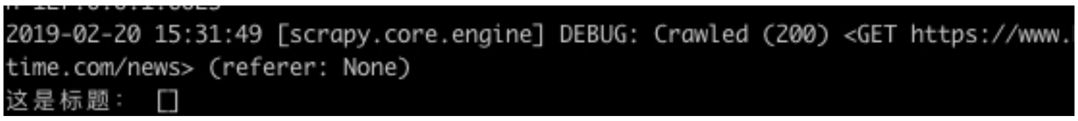
在测试一个示范

https://www.btime.com/news

新建一个淘宝的项目 (搜的是360新闻)

代码如下：

```python
class TnmSpider(scrapy.Spider):
    name = 'tnm'
    allowed_domains = ['s.taobao.com']
    start_urls = ['https://www.btime.com/news']

    def start_requests(self):
        for url in self.start_urls:
            yield scrapy.Request(url=url, callback=self.parse)

    def parse(self,response):
        titele = response.css('#main > div > div.layout-main > div.module-infoflow > div.j-content-list > div > a > h1::text').extract()
        print('这是标题：', titele)
```



```
2019-02-20 15:31:49 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.
time.com/news> (referer: None)
这是标题：  []
```

....现在来搞渲染后的

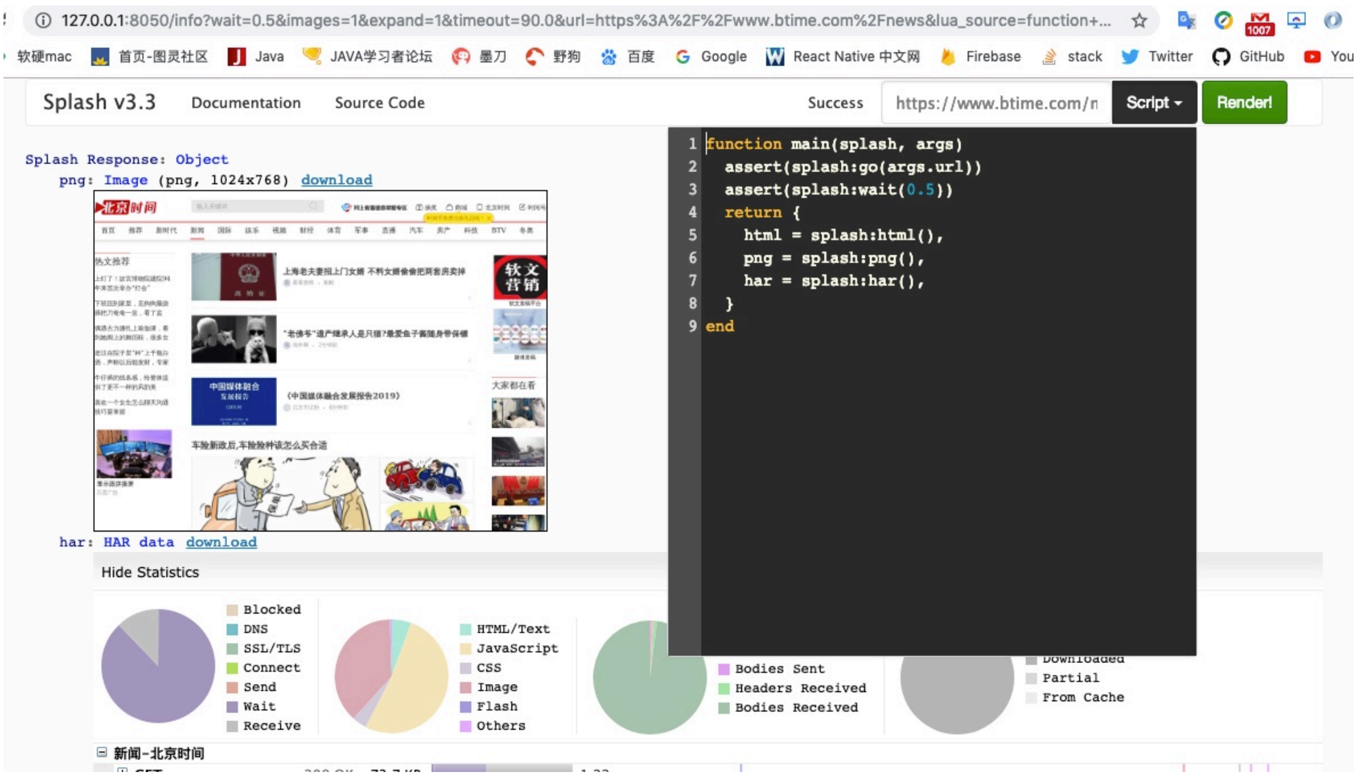
1.启动docker

2.安装splash `docker pull scrapinghub/splash`

3.启动splash docker run -p 8050:8050 scrapinghub/splash

停止直接ctrl+c

截图如下：



```
127.0.0.1:8050/info?wait=0.5&images=1&expand=1&timeout=90.0&url=https%3A%2F%2Fwww.btime.com%2Fnews&lua_source=function+...
```

```
Splash Response: Object
    png: Image (png, 1024x768) download
    har: HAR data download
```

```lua
function main(splash, args)
  assert(splash:go(args.url))
  assert(splash:wait(0.5))
  return {
    html = splash:html(),
    png = splash:png(),
    har = splash:har(),
  }
end
```

4.安装py的splash的库

pip3 install scrapy-splash

5.在项目中配置文件 setting.py

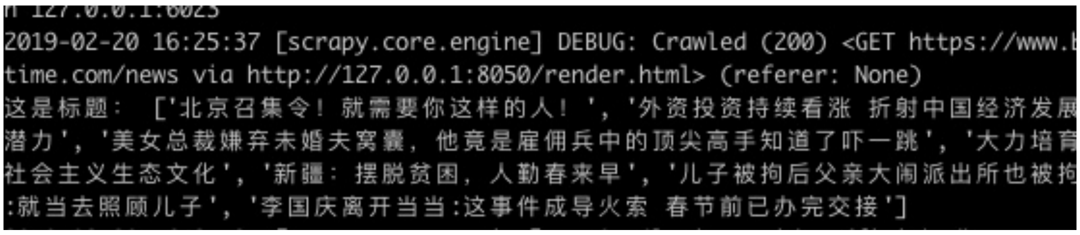
```python
# 渲染服务的url
SPLASH_URL = 'http://127.0.0.1:8050'

#下载器中间件
DOWNLOADER_MIDDLEWARES = {
    'scrapy_splash.SplashCookiesMiddleware': 723,
    'scrapy_splash.SplashMiddleware': 725,
    'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware': 810,
}
# 去重过滤器
DUPEFILTER_CLASS = 'scrapy_splash.SplashAwareDupeFilter'
# 使用Splash的Http缓存
HTTPCACHE_STORAGE = 'scrapy_splash.SplashAwareFSCacheStorage'
```

6.修改代码

```python
# -*- coding: utf-8 -*-
import scrapy
from scrapy_splash import SplashRequest

```

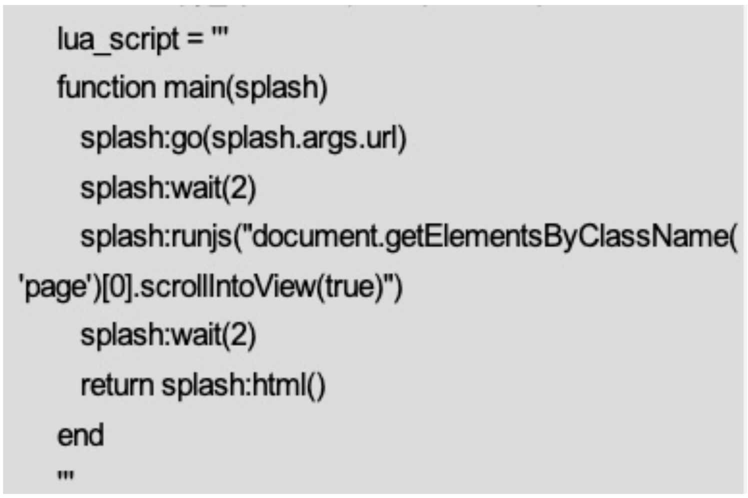
```python

class TnmSpider(scrapy.Spider):
    name = 'tnm'
    allowed_domains = ['s.taobao.com']
    start_urls = ['https://www.btime.com/news']

    def start_requests(self):
        for url in self.start_urls:
            #args 里面的是等待多久,endpoint 详细配置https://www.jianshu.com/p/2b04f5eb5785
            yield SplashRequest(url=url,callback=self.parse,args={'wait':1},endpoint='render.html')

    def parse(self,response):
        titele = response.css('#main > div > div.layout-main > div.module-infoflow > div.j-content-list > div > a > h1::text').extract()
        print('这是标题: ', titele)
```

注意:也就是后期的所有的request 都使用SplashRequest

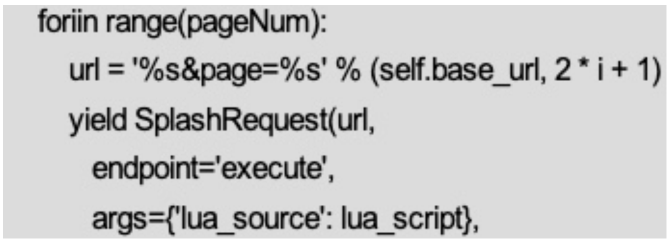


```
n 127.0.0.1:6023
2019-02-20 16:25:37 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.b
time.com/news via http://127.0.0.1:8050/render.html> (referer: None)
这是标题：  ['北京召集令！就需要你这样的人！', '外资投资持续看涨 折射中国经济发展
潜力', '美女总裁嫌弃未婚夫窝囊, 他竟是雇佣兵中的顶尖高手知道了吓一跳', '大力培育
社会主义生态文化', '新疆：摆脱贫困, 人勤春来早', '儿子被拘后父亲大闹派出所也被拘
:就当去照顾儿子', '李国庆离开当当:这事件成导火索 春节前已办完交接']
```

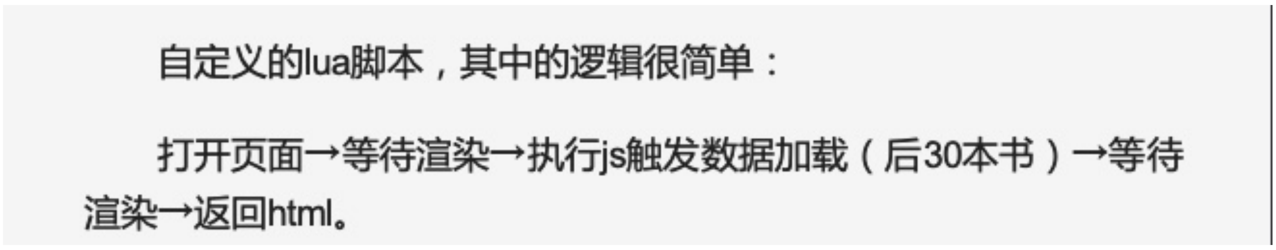
———还有一种情况



```
    lua_script = '''
    function main(splash)
      splash:go(splash.args.url)
      splash:wait(2)
      splash:runjs("document.getElementsByClassName(
'page')[0].scrollIntoView(true)")
      splash:wait(2)
      return splash:html()
    end
    '''
```



```
  foriin range(pageNum):
      url = '%s&page=%s' % (self.base_url, 2 * i + 1)
      yield SplashRequest(url,
        endpoint='execute',
        args={'lua_source': lua_script},
```



自定义的lua脚本，其中的逻辑很简单：

打开页面→等待渲染→执行js触发数据加载（后30本书）→等待渲染→返回html。