

Scrapy 爬虫部署 -- Scrapyd

Scrapyd 是一个用来部署和运行 Scrapy 项目的应用，由 Scrapy 的开发者开发。其可以通过一个简单的 Json API 来部署（上传）或者控制你的项目。

Scrapyd 可以用来管理多个项目，并且每个项目还可以上传多个版本，不过只有最新的版本会被使用。

在安装并开启 Scrapyd 之后，它将会挂起一个服务来监听运行爬虫的请求，并且根据请求为每一个爬虫启用一个进程来运行。Scrapyd 同样支持同时运行多个进程，进程的数量由 max_proc 和 max_proc_per_cpu 选项来限制。

scrapyd 安装

使用 pip 命令安装

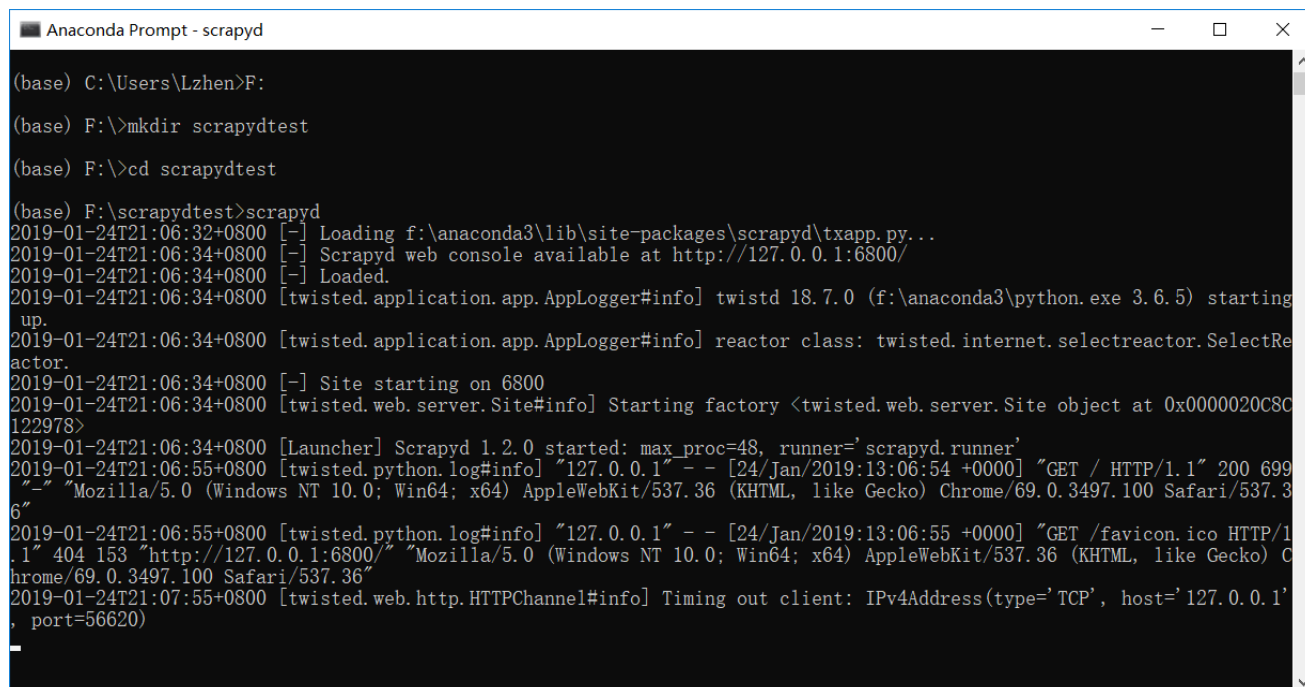
```
pip install scrapyd
```

安装完成后，需要创建一个文件夹，然后到新建的文件夹中使用命令开始 Scrapyd 服务：

开启scrapyd服务命令：

```
scrapyd
```

开后的效果图：

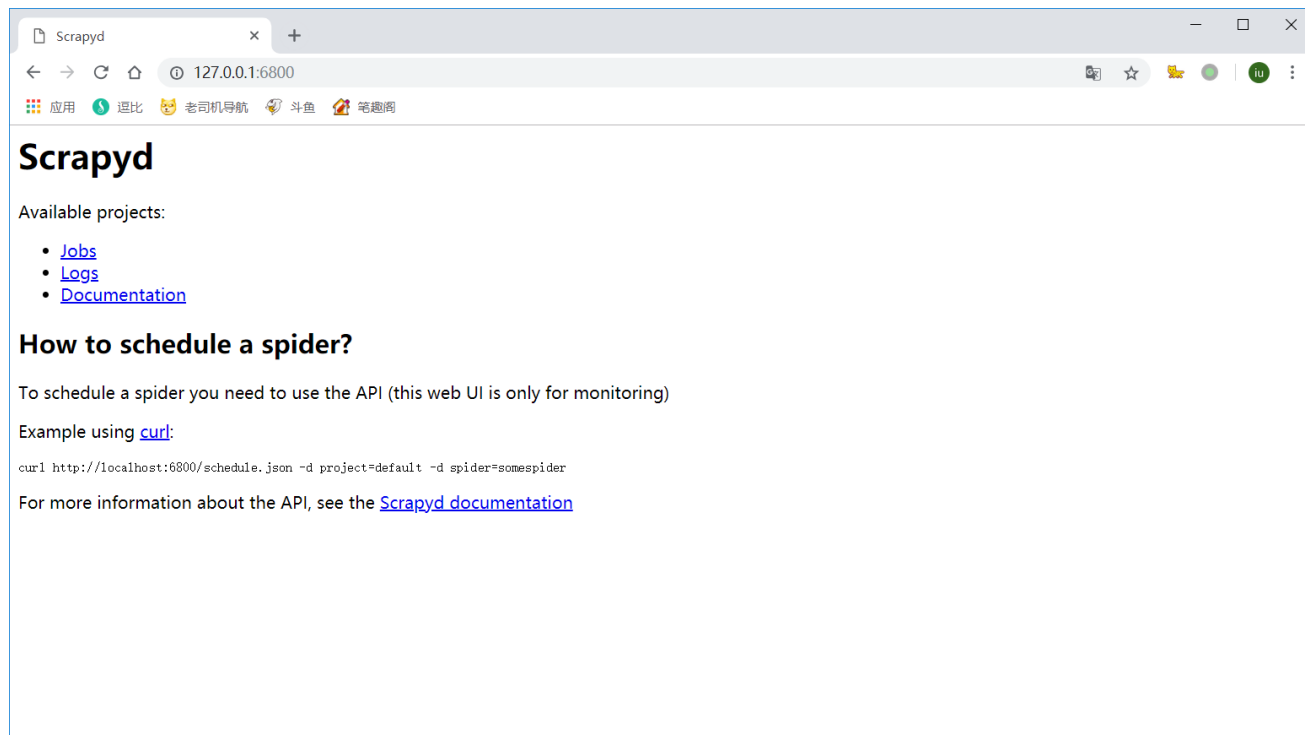


```
Anaconda Prompt - scrapyd
(base) C:\Users\Lzhen>F:
(base) F:\>mkdir scrapydtest
(base) F:\>cd scrapydtest
(base) F:\scrapydtest>scrapyd
2019-01-24T21:06:32+0800 [-] Loading f:\anaconda3\lib\site-packages\scrapyd\txapp.py...
2019-01-24T21:06:34+0800 [-] Scrapyd web console available at http://127.0.0.1:6800/
2019-01-24T21:06:34+0800 [-] Loaded.
2019-01-24T21:06:34+0800 [twisted.application.app.AppLogger#info] twisted 18.7.0 (f:\anaconda3\python.exe 3.6.5) starting
up.
2019-01-24T21:06:34+0800 [twisted.application.app.AppLogger#info] reactor class: twisted.internet.selectreactor.SelectRe
actor.
2019-01-24T21:06:34+0800 [-] Site starting on 6800
2019-01-24T21:06:34+0800 [twisted.web.server.Site#info] Starting factory <twisted.web.server.Site object at 0x0000020C8C
122978>
2019-01-24T21:06:34+0800 [Launcher] Scrapyd 1.2.0 started: max_proc=48, runner='scrapyd.runner'
2019-01-24T21:06:55+0800 [twisted.python.log#info] "127.0.0.1" - - [24/Jan/2019:13:06:54 +0000] "GET / HTTP/1.1" 200 699
 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.3
6"
2019-01-24T21:06:55+0800 [twisted.python.log#info] "127.0.0.1" - - [24/Jan/2019:13:06:55 +0000] "GET /favicon.ico HTTP/1
.1" 404 153 "http://127.0.0.1:6800/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/69.0.3497.100 Safari/537.36"
2019-01-24T21:07:55+0800 [twisted.web.http.HTTPChannel#info] Timing out client: IPv4Address(type='TCP', host='127.0.0.1'
, port=56620)
```

开启 scrapyd 服务后我们能使用浏览器访问我们刚刚开启的 scrapyd ,访问地址为：

127.0.0.1:6800

访问界面如下图：



能用浏览器打开这个界面，就说我们就成功开启了 `scrapy`，然后他一直监听 `6800` 端口。

scrapy 配置

- `http_port`

`Scrapy` 的API监听的端口，默认为 `6800`。

- `bind_address`

网页和json服务监听的IP地址，默认为 `127.0.0.1`。修改成 `0.0.0.0`，可以让局域网访问。

- `max_proc`

同时启动的最大Scrapy进程数，如果没有设置或者设置为 `0`，那么将会使用当前cpu可用的核数乘以 `max_proc_per_cpu` 的值。默认为 `0`。

- `max_proc_per_cpu`

每个cpu能同时启动的最大Scrapy进程数。默认为 `4`。

- `debug`

是否开启 `debug` 模式，默认为 `off`。开启之后，如果在调用Scrapy的 `Json API` 的时候出错，则会返回详细的 `traceback` 信息。

- `eggs_dir`

项目的 `eggs` 文件存储的目录。

- `dbs_dir`

项目存储数据库的目录，也包括爬虫队列。

- `logs_dir`

存储 `Scrapy` 日志的目录。如果不希望存储日志，那么需要设置成如下所示：

```
logs_dir =
```

- `items_dir`

存储 `items` 的目录，一般来说不需要设置这个选项，因为抓取下来的数据都会存到数据库中。如果设置这个选项，那么将会覆盖 `Scrapy` 的 `FEED_URL` 设置，将抓取下来的 `items` 保存到指定目录。

- `jobs_to_keep`

每个 `spider` 保留多少个完成的 `job`，默认为 `5`。这更多指的是 `item` 和 `log`。

- `finished_to_keep`

启动器中保留的已完成进程的数量，默认为 `100`。

- `poll_interval`

轮询队列的间隔，以秒为单位，默认值为 `5`，可以为浮点数。


- `runner`

用来启动子进程的启动器，可以自定义启动的模块。

- `node_name`

每个节点的节点名称，默认为 `${socket.gethostname()}`。

搜索 `scrapy.conf` 可以搜索出 `scrapy` 配置文件，如下图：

 `F:\Anaconda3\Lib\site-packages\scrapyd\default_scrapyd.conf` - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

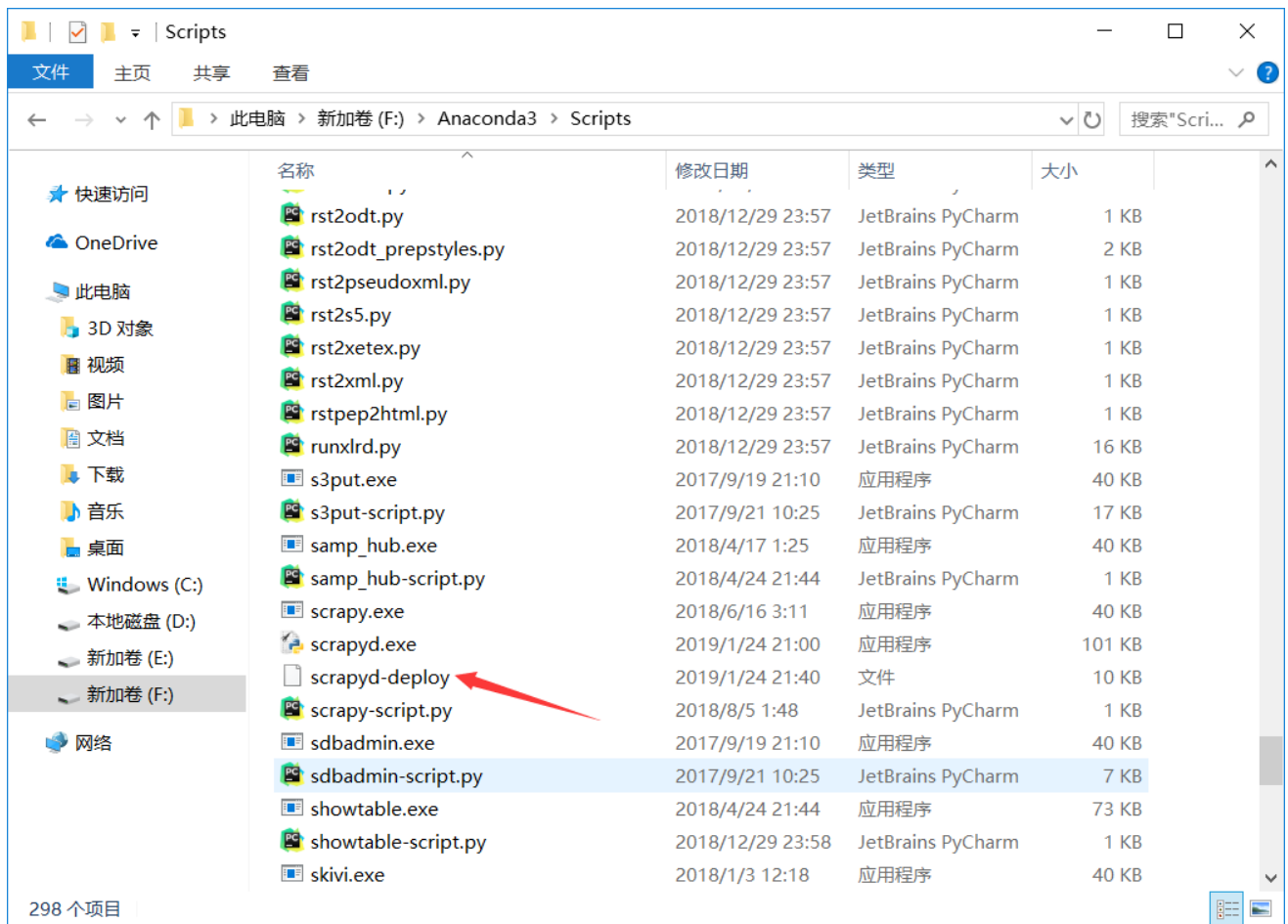
```
default_scrapyd.conf x
1 [scrapyd]
2 eggs_dir = eggs
3 logs_dir = logs
4 items_dir =
5 jobs_to_keep = 5
6 dbs_dir = dbs
7 max_proc = 0
8 max_proc_per_cpu = 4
9 finished_to_keep = 100
10 poll_interval = 5.0
11 bind_address = 127.0.0.1
12 http_port = 6800
13 debug = off
14 runner = scrapyd.runner
15 application = scrapyd.app.application
16 launcher = scrapyd.launcher.Launcher
17 webroot = scrapyd.website.Root
18
19 [services]
20 schedule.json = scrapyd.webservice.Schedule
21 cancel.json = scrapyd.webservice.Cancel
22 addversion.json = scrapyd.webservice.AddVersion
23 listprojects.json = scrapyd.webservice.ListProjects
24 listversions.json = scrapyd.webservice.ListVersions
25 listspiders.json = scrapyd.webservice.ListSpiders
26 delproject.json = scrapyd.webservice.DeleteProject
27 delversion.json = scrapyd.webservice.DeleteVersion
28 listjobs.json = scrapyd.webservice.ListJobs
29 daemonstatus.json = scrapyd.webservice.DaemonStatus
30
```

scrapyd 部署爬虫

在正式部署爬虫之前，我们还需要安装一个 `scrapyd-client`，命令如下：

```
pip install scrapyd_client
```

安装完成后，在Windows使用 `scrapyd-deploy` 命令启动 `scrapyd_client` 会出现scrapyd-deploy无后缀文件，这个scrapyd-deploy无后缀文件是启动文件，在Linux系统下可以运行，在windows下是不能运行的，所以我们需要编辑一下使其在windows可以运行

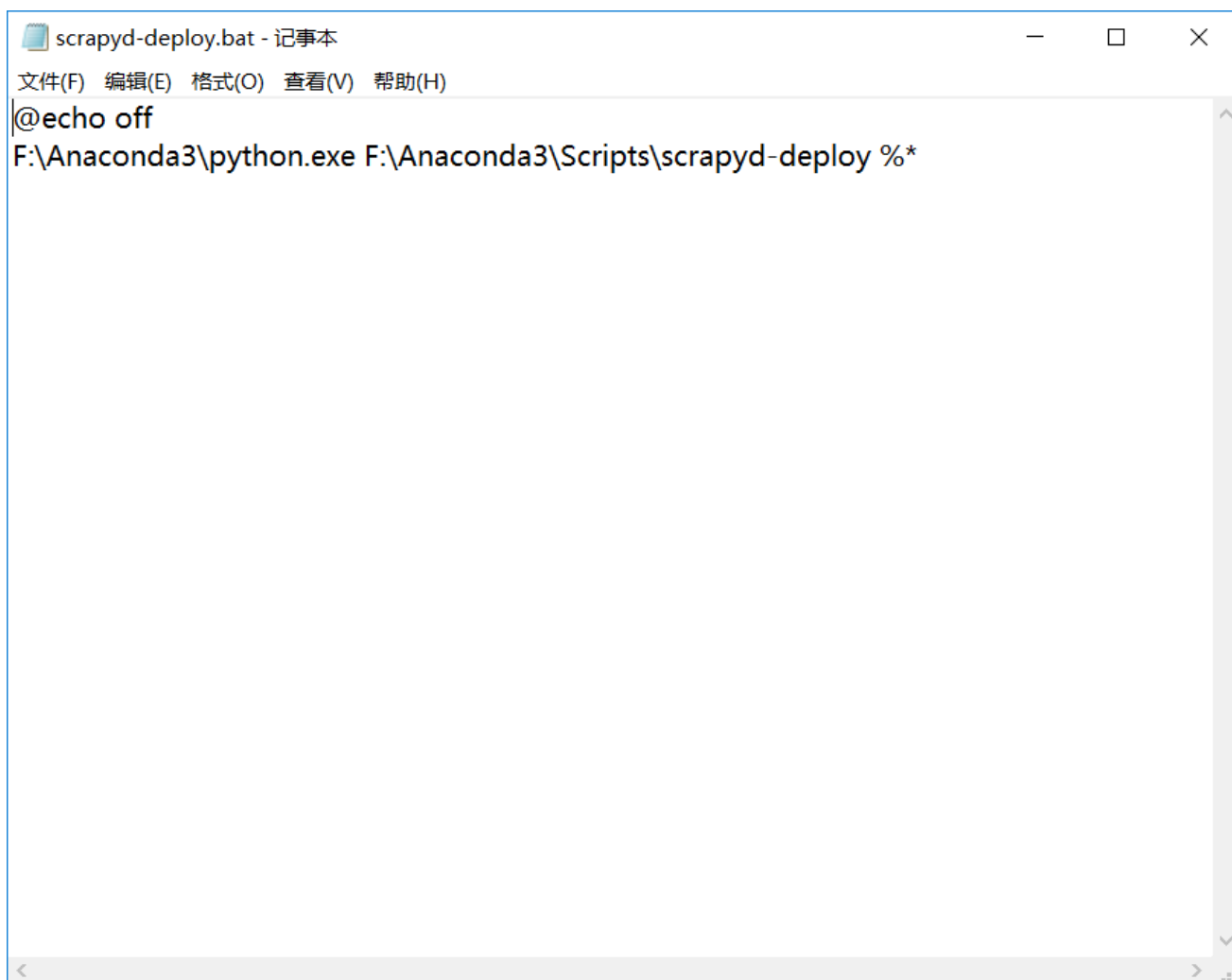


新建一个scrapyd-deploy.bat文件，右键选择编辑，输入以下配置，**注意：两个路径之间是空格**

```
@echo off
F:\Anaconda3\python.exe F:\Anaconda3\Scripts\scrapyd-deploy %*
```

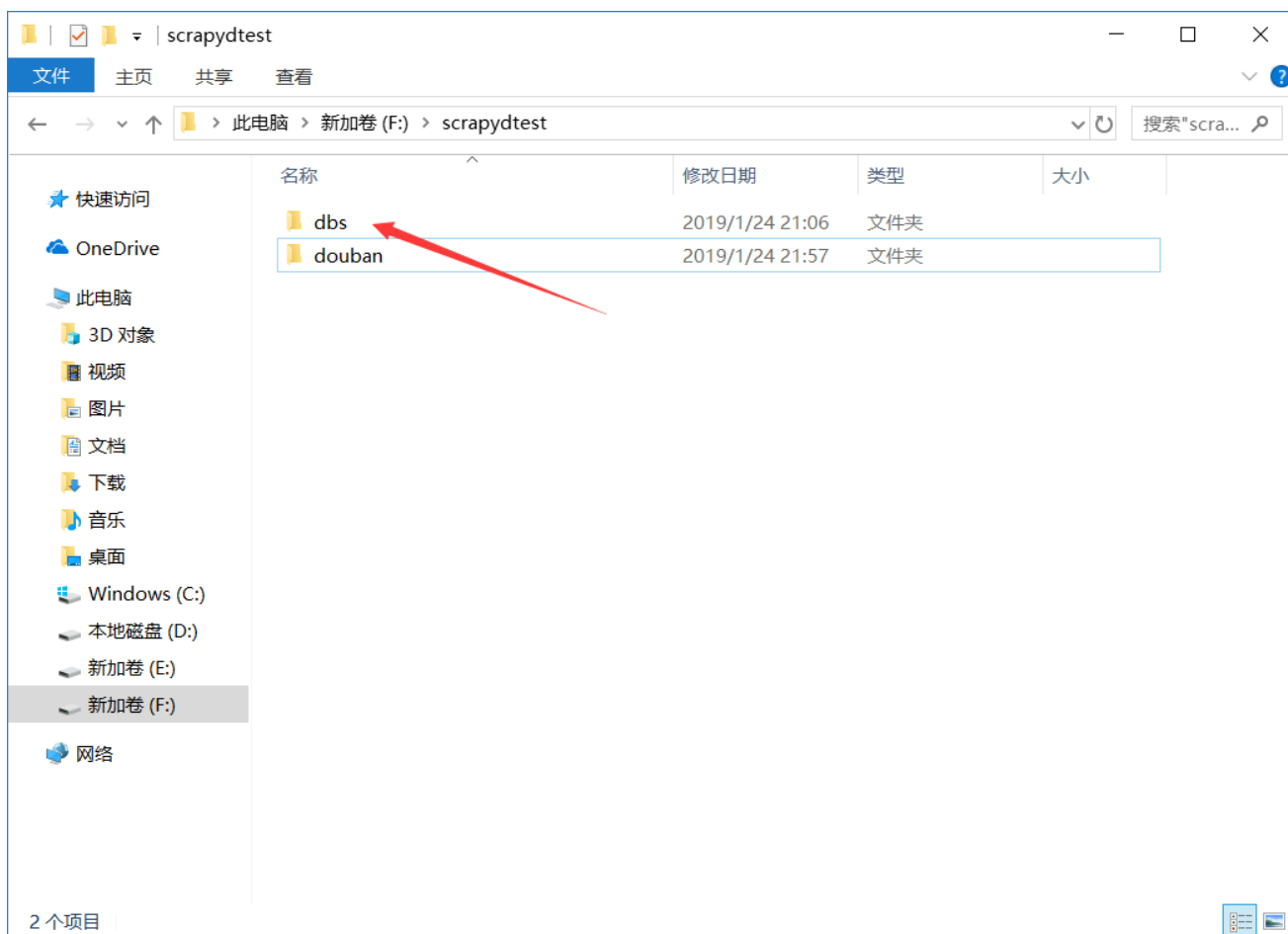
注解

- `F:\Anaconda3\python.exe` 是你当前python版本的解释器的路径
- `F:\Anaconda3\Scripts\scrapyd-deploy` 是 scrapyd-deploy 文件的路径



```
scrapydeploy.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
@echo off
F:\Anaconda3\python.exe F:\Anaconda3\Scripts\scrapydeploy %*
```

在刚刚我们创建的 `scrapytest` 文件夹放入之前编写的豆瓣爬虫。放入爬虫项目的时候，是不是当前文件夹多了一个空子文件夹 `db`，它是用来存放爬虫项目的数据文件。



然后我们进入 `scraoydtest` 中的 `douban` 的工程目录下，使用 `scrapyd-deploy` 命令打包爬虫，提示错误如下图：

```
Anaconda Prompt

(base) C:\Users\Lzhen>f:

(base) F:\>cd scrapytest

(base) F:\scrapytest>cd dbs

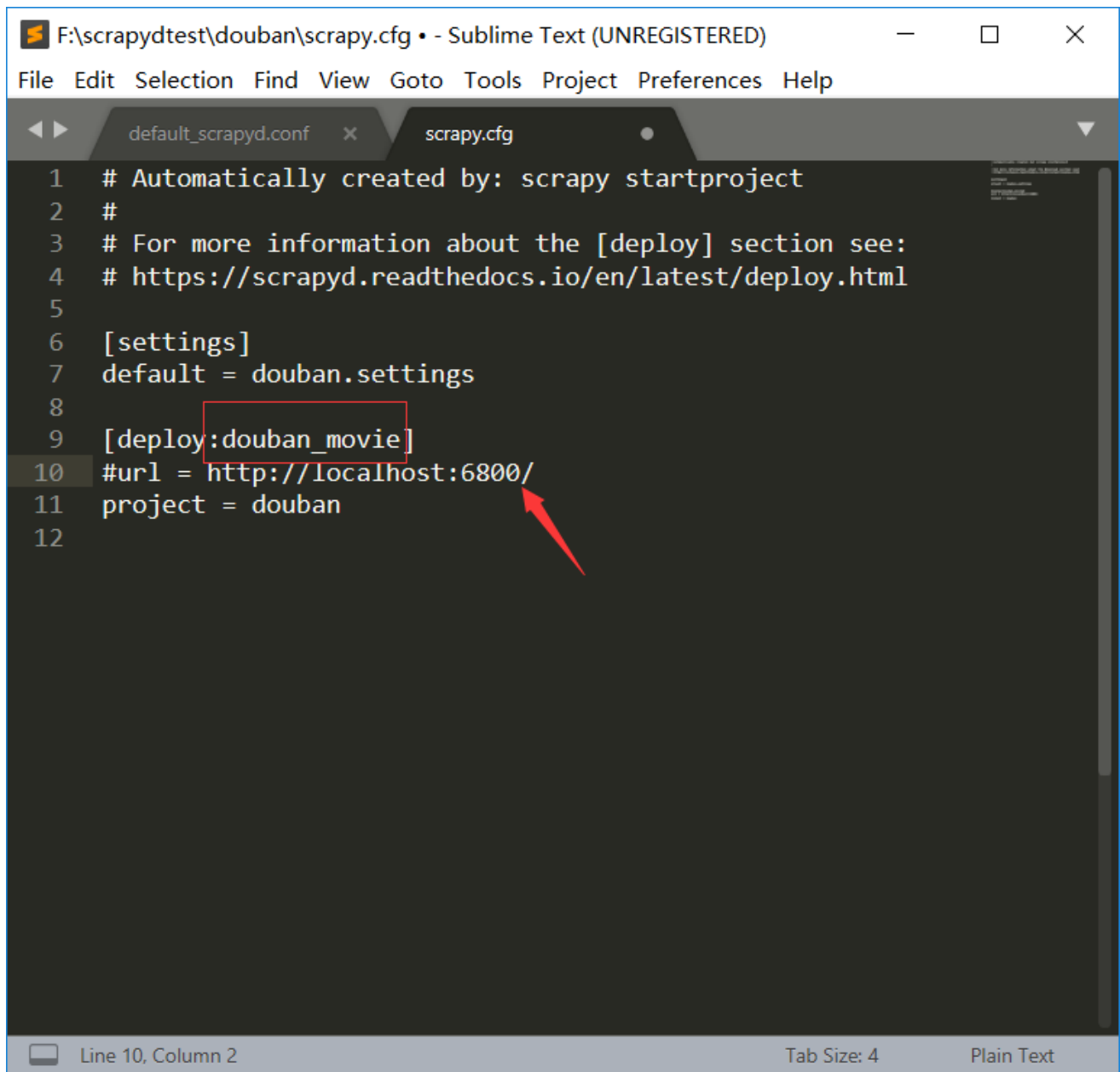
(base) F:\scrapytest\dbs>cd ..

(base) F:\scrapytest>cd douban

(base) F:\scrapytest\douban>scrapyd-deploy
Unknown target: default

(base) F:\scrapytest\douban>
```

提示我们没有找到刚刚放入的 `douban` 项目，我们需要修改的爬虫项目的 `scrapy.cfg`，这是 `Scrapy` 作者为 `Scrapyd` 预留的配置接口，配置如下图：



```
F:\scrapytest\douban\scrapy.cfg • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

default_scrapy.cfg x scrapy.cfg

1 # Automatically created by: scrapy startproject
2 #
3 # For more information about the [deploy] section see:
4 # https://scrapyd.readthedocs.io/en/latest/deploy.html
5
6 [settings]
7 default = douban.settings
8
9 [deploy:douban_movie]
10 #url = http://localhost:6800/
11 project = douban
12

Line 10, Column 2 Tab Size: 4 Plain Text
```

我们需要把 `url` 前面的注释去掉和加上红色框的 `:douban_movie`，保存即可。再打包之前使用命令 `scrapy-deploy -l`


```
Anaconda Prompt
(base) C:\Users\Lzhen>f:
(base) F:\>cd scrapydtest
(base) F:\scrapydtest>cd douban
(base) F:\scrapydtest\douban>scrapyd-deploy -l
douban_movie      http://localhost:6800/
(base) F:\scrapydtest\douban>
```

可以看到刚才新加上部署名称和url,在使用命令 `scrapy list` , 这个命令执行成功说明可以打包了, 如果没执行成功说明还有工作没完成。

注意

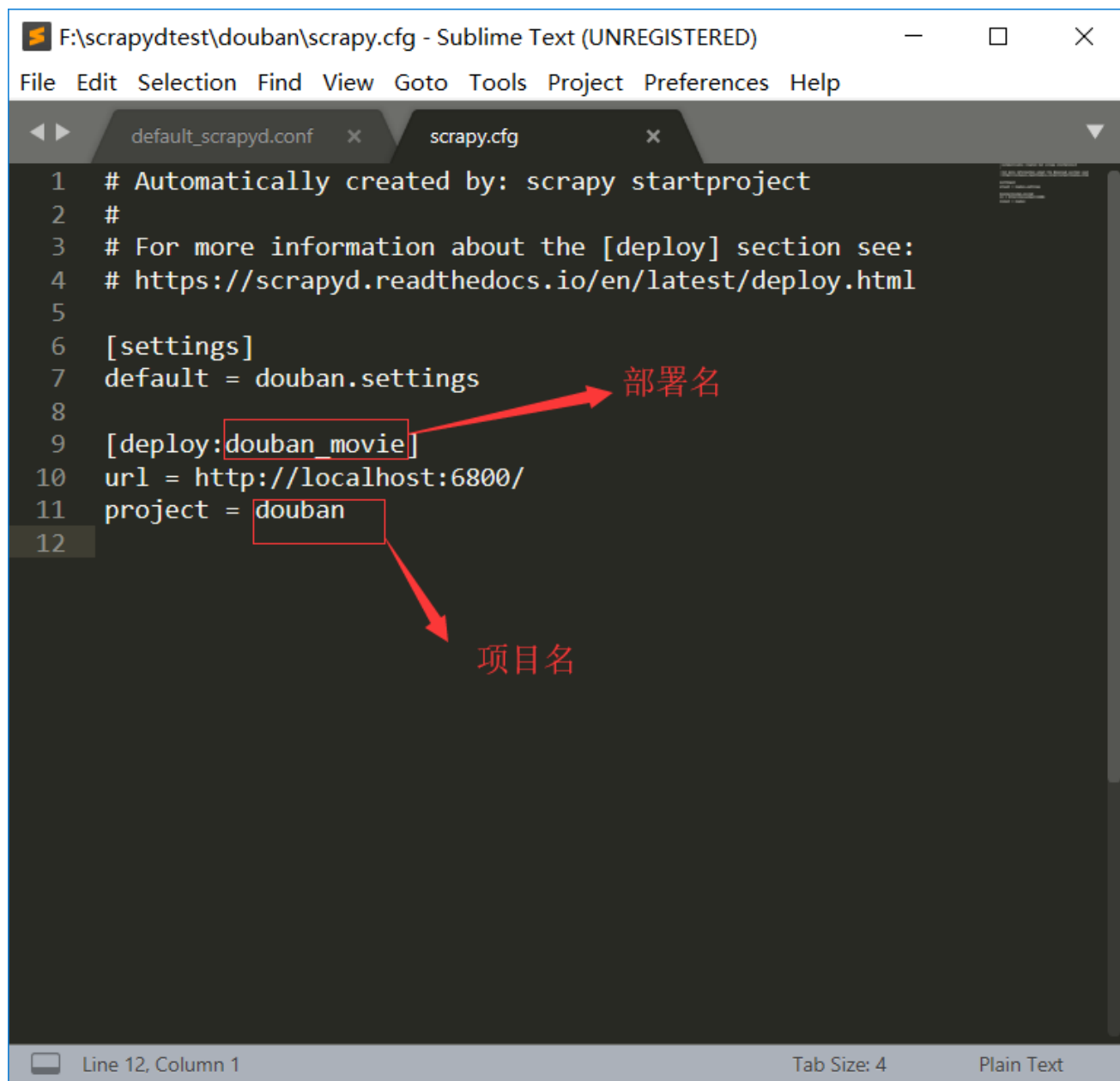
执行 `scrapy list` 命令的时候很有可能出现错误, 如果是python无法找到scrapy项目, 需要在scrapy项目里的 `settings.py` 配置文件里加上如下代码:

```
import os
import sys
BASE_DIR = os.path.dirname(os.path.abspath(os.path.dirname(__file__)))
sys.path.insert(0, os.path.join(BASE_DIR, "douban"))
```

执行成功如下:

```
Anaconda Prompt
(base) C:\Users\Lzhen>f:
(base) F:\>cd scrapydtest
(base) F:\scrapydtest>cd douban
(base) F:\scrapydtest\douban>scrapyd-deploy -l
douban_movie      http://localhost:6800/
(base) F:\scrapydtest\douban>scrapy list
doubanspider
(base) F:\scrapydtest\douban>
```

展示了将要被打包的爬虫列表，接下来可以使用命令 `scrapyd-deploy 部署名称 -p 项目名称` 将 `douban`，部署名和项目名在 `scrapy.cfg` 中，如下图：



```
F:\scrapydtest\douban\scrapy.cfg - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

default_scrapy.conf x scrapy.cfg x
1 # Automatically created by: scrapy startproject
2 #
3 # For more information about the [deploy] section see:
4 # https://scrapyd.readthedocs.io/en/latest/deploy.html
5
6 [settings]
7 default = douban.settings
8
9 [deploy:douban_movie]
10 url = http://localhost:6800/
11 project = douban
12
```

部署名

项目名

Line 12, Column 1 Tab Size: 4 Plain Text

我们的当前的项目打包命令为 `scrapyd-deploy douban_movie -p douban`

```
Anaconda Prompt

(base) C:\Users\Lzhen>f:

(base) F:\>cd scrapydtest

(base) F:\scrapydtest>cd douban

(base) F:\scrapydtest\douban>scrapydeploy -l
douban_movie      http://localhost:6800/

(base) F:\scrapydtest\douban>scrapy list
doubanspider

(base) F:\scrapydtest\douban>scrapydeploy douban_movie -p douban
Packing version 1548341791
Deploying to project "douban" in http://localhost:6800/addversion.json
Deploy failed: <urlopen error [WinError 10061] 由于目标计算机积极拒绝，无法连接。>

(base) F:\scrapydtest\douban>
```

这是需要开启 `scrapyd` 服务的，可能是你没有开启或者是需要重启一下 `scrapyd` 服务。

```
Anaconda Prompt

(base) C:\Users\Lzhen>f:

(base) F:\>cd scrapydtest

(base) F:\scrapydtest>cd douban

(base) F:\scrapydtest\douban>scrapydeploy -l
douban_movie      http://localhost:6800/

(base) F:\scrapydtest\douban>scrapy list
doubanspider

(base) F:\scrapydtest\douban>scrapydeploy douban_movie -p douban
Packing version 1548341791
Deploying to project "douban" in http://localhost:6800/addversion.json
Deploy failed: <urlopen error [WinError 10061] 由于目标计算机积极拒绝，无法连接。>

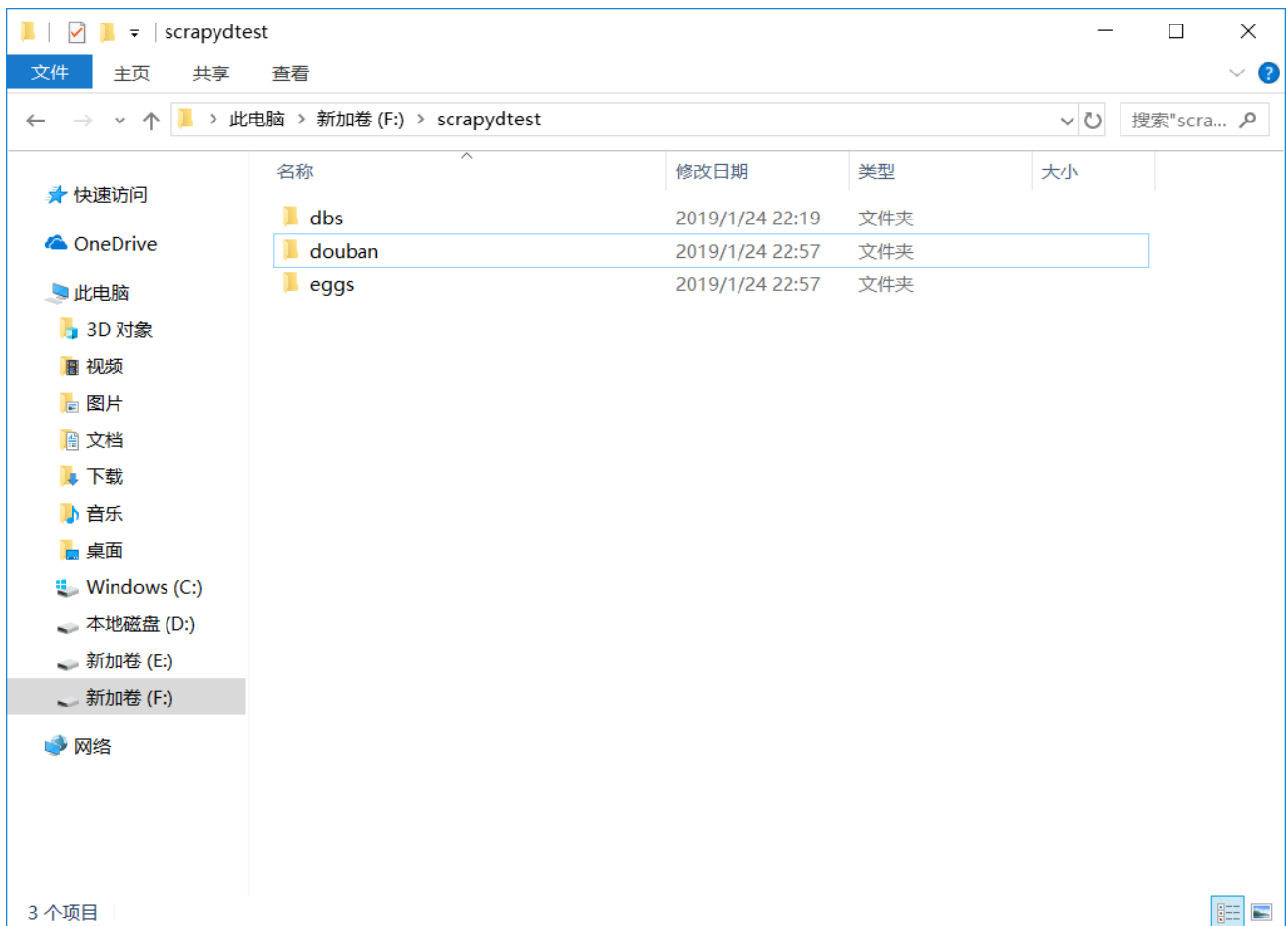
(base) F:\scrapydtest\douban>scrapydeploy douban_movie -p douban
Packing version 1548341839
Deploying to project "douban" in http://localhost:6800/addversion.json
Server response (200):
{"node_name": "DESKTOP-OVFSNP6", "status": "ok", "project": "douban", "version": "1548341839", "spiders": 1}

(base) F:\scrapydtest\douban>
```

开启了 `scrapyd` 服务后使用命令提示上面信息，笔者解释一哈啊：

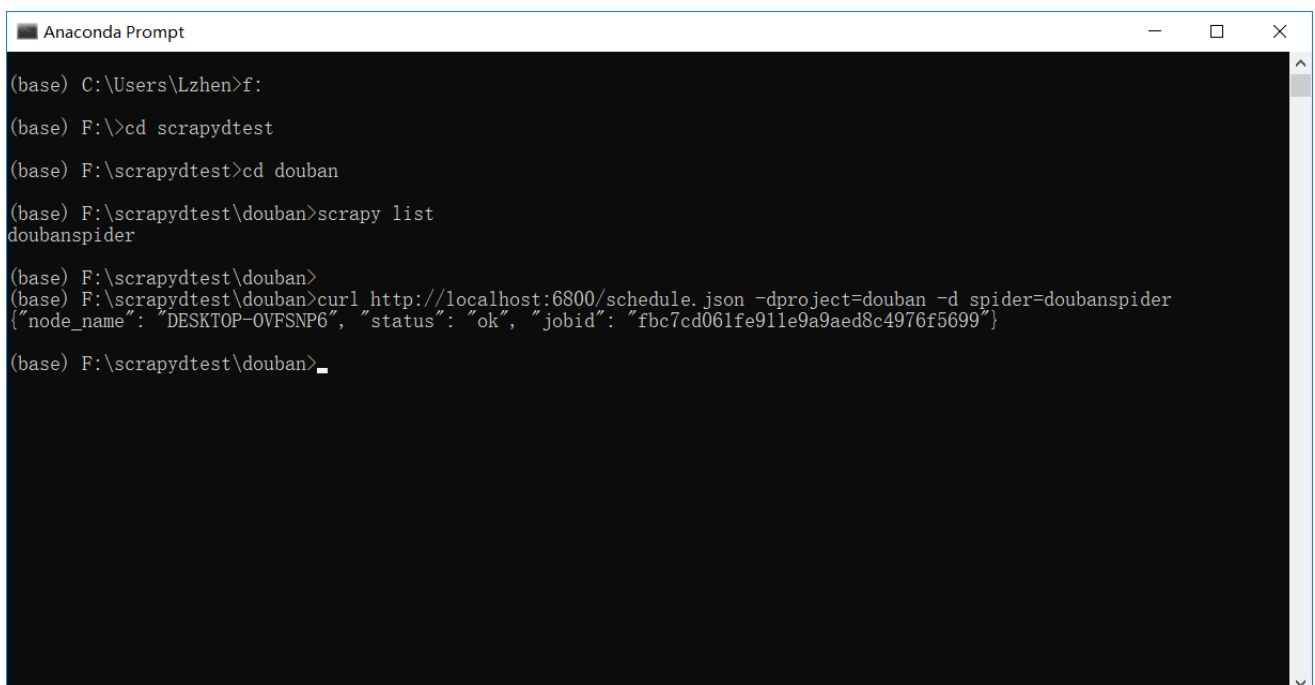
```
{"node_name": "DESKTOP-OVFSNP6", "status": "ok", "project": "douban", "version": "1548341839",
"spiders": 1}
# node_name 节点名
# status 项目状态
# project 项目名
# version 项目版本
# spiders 项目spiders/文件夹下的爬虫数
```

查看了一下我们的 `scrapytest` 文件夹，发现多了一个 `eggs` 夹，里面是有文件的，它是存储我们刚刚打包的爬虫文件的。

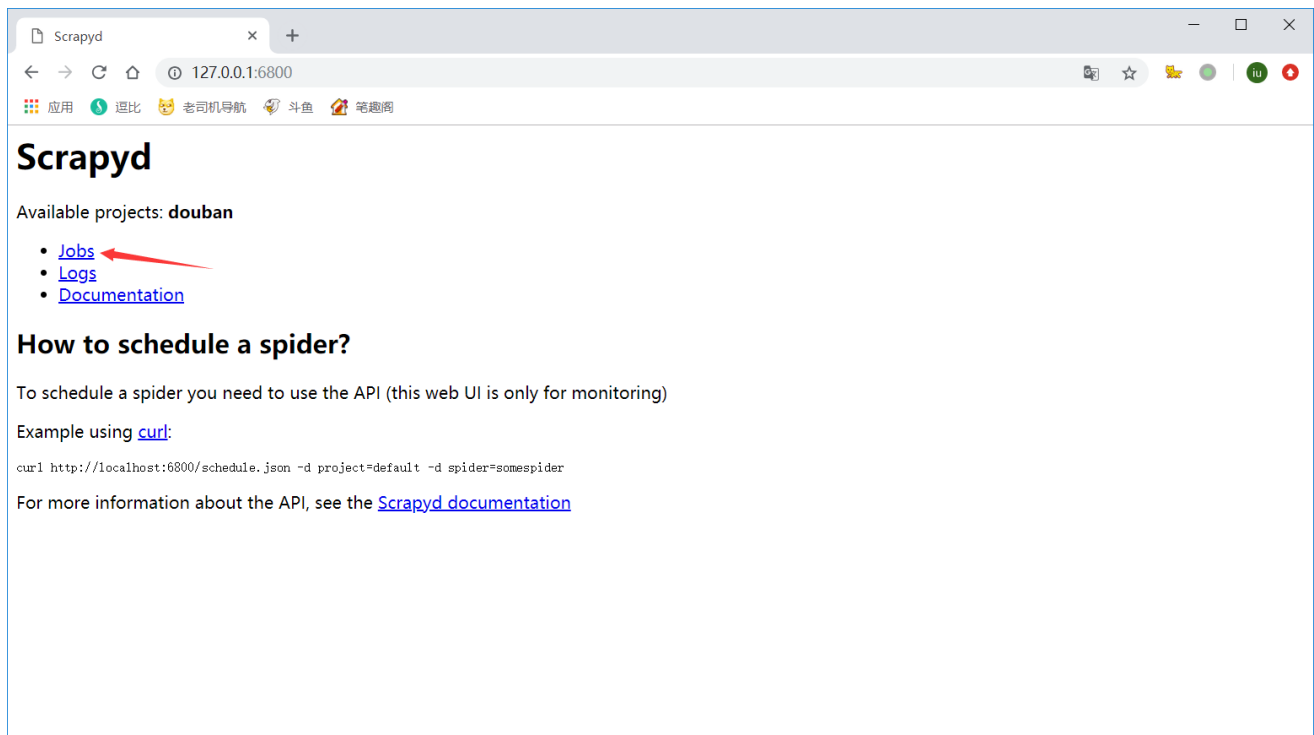


接下来可以使用命令启动爬虫了，命令如下：

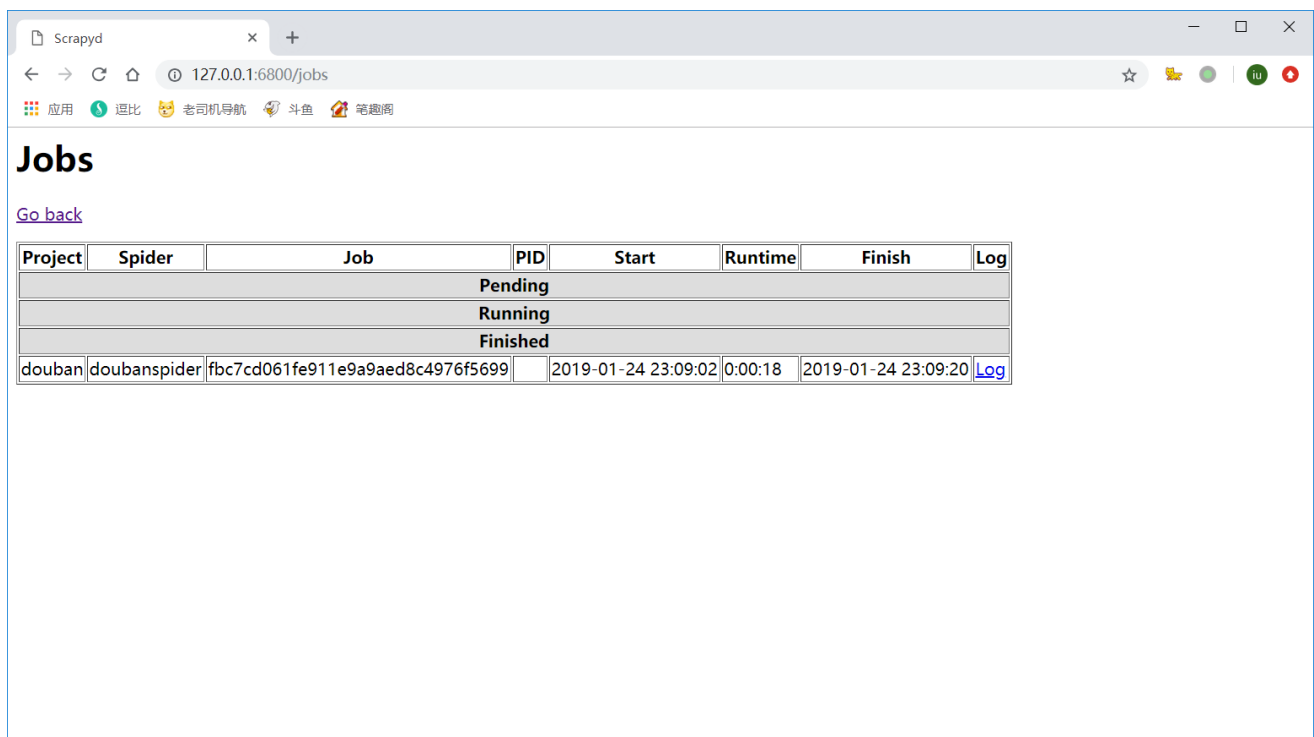
```
curl http://localhost:6800/schedule.json -dproject=项目名称 -d spider=爬虫名称
```



输入后，返回以上信息，说爬虫启动成功，在运行。此时可以使用浏览器访问 `127.0.0.1:6800` ,可以查看。



点击 `Jobs` ,就可以查看爬虫运行情况。



我们的爬去豆瓣电影top250的爬虫已经运行结束了，只运行了18秒。在这里可以方便查看各个爬虫运行状况和 `debug` 日志。