

背景：

PySpider: 一个国人编写的强大的网络爬虫系统并带有强大的WebUI。采用Python语言编写，分布式架构，支持多种数据库后端，强大的WebUI支持脚本编辑器，任务监视器，项目管理器以及结果查看器。在线示例：

<http://demo.pyspider.org/>

官方文档：<http://docs.pyspider.org/en/latest/>

Github：<https://github.com/binux/pyspider>

本文爬虫代码 Github 地址：<https://github.com/zhisheng17/Python-Projects/blob/master/v2ex/V2EX.py>

更多精彩文章可以在微信公众号：猿blog 阅读到，欢迎关注。



说了这么多，我们还是来看正文吧！

前提：你已经安装好了Pyspider 和 MySQL-python（保存数据）

如果你还没安装的话，请看看我的前一篇文章，防止你也走弯路。

1. [Pyspider 框架学习时走过的一些坑](#)
2. [HTTP 599: SSL certificate problem: unable to get local issuer certificate](#)错误

我所遇到的一些错误：

```
pyspider > V2EX

{
  "process": {
    "callback": "on_start"
  },
  "project": "V2EX",
  "taskid": "data:,on_start",
  "url": "data:,on_start"
}

Traceback (most recent call last):
  File "C:\Python27\lib\site-packages\pyspider\processor\project_module.py", line 49, in build_module
    module = loader.load_module(project['name'])
  File "C:\Python27\lib\site-packages\pyspider\processor\project_module.py", line 207, in load_module
    six.exec_(code, mod, __dict__)
  File "C:\Python27\lib\site-packages\six.py", line 699, in exec_
    exec("""exec_code_in_globs_,_locs_""")
  File "<string>", line 1, in <module>
  File "<V2EX>", line 11, in <module>
ImportError: No module named MySQLdb

on_start

# created by 10412
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
# Created on 2016-10-20 20:12:00
# Project: V2EX

from pyspider.libs.base_handler import BaseHandler

import re
import random
import MySQLdb

class Handler(BaseHandler):
    crawl_config = {
        # ...
    }

    def __init__(self):
        self.db = MySQLdb.connect(
            host='localhost',
            port=3306,
            user='root',
            passwd='123456',
            charset='utf8')

    def add_question(self, url):
        try:
            cursor = self.db.cursor()
            sql = 'insert into question (url, comment_count) values ("%s", %s)' % (url, 'now()')
            print sql
            cursor.execute(sql)
            self.db.commit()
        except Exception, e:
            print e
            self.db.rollback()

@every(minutes=24 * 60)
```

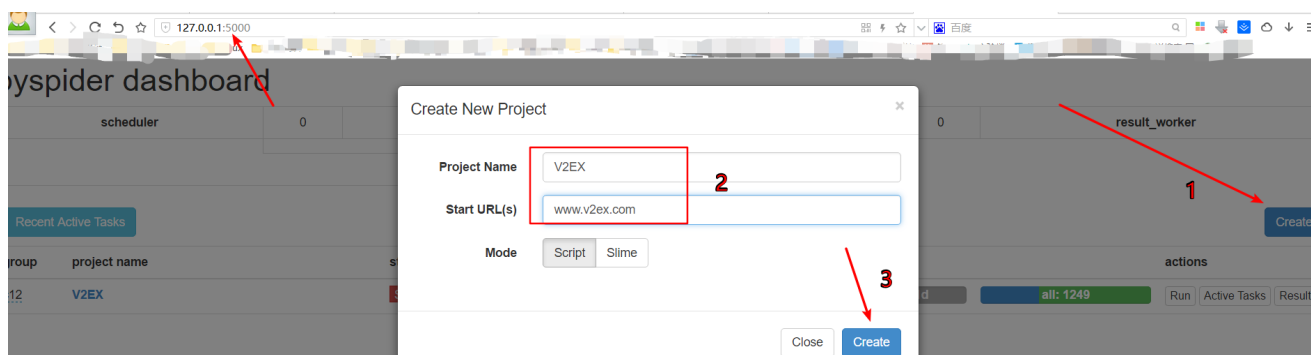
首先，本爬虫目标：使用 Pyspider 框架爬取 [V2EX](https://www.v2ex.com/) 网站的帖子中的问题和内容，然后将爬取的数据保存在本地。

V2EX 中大部分的帖子查看是不需要登录的，当然也有些帖子是需要登录后才能够查看的。（因为后来爬取的时候发现一直 error，查看具体原因后才知道是需要登录的才可以查看那些帖子的）所以我觉得没必要用到 Cookie，当然如果你非得要登录，那也很简单，简单地方法就是添加你登录后的 cookie 了。

我们在 <https://www.v2ex.com/> 扫了一遍，发现并没有一个列表能包含所有的帖子，只能退而求其次，通过抓取分类下的所有的标签列表页，来遍历所有的帖子：<https://www.v2ex.com/?tab=tech> 然后是 <https://www.v2ex.com/go/programmer> 最后每个帖子的详情地址是（举例）：<https://www.v2ex.com/t/314683#reply1>

创建一个项目

在 pyspider 的 dashboard 的右下角，点击 “Create” 按钮



替换 on_start 函数的 self.crawl 的 URL:

```
@every(minutes=24 * 60)
def on_start(self):
    self.crawl('https://www.v2ex.com/', callback=self.index_page, validate_cert=False)
```

- self.crawl 告诉 pypider 抓取指定页面，然后使用 callback 函数对结果进行解析。
- @every 修饰器，表示 on_start 每天会执行一次，这样就能抓到最新的帖子了。
- validate_cert=False 一定要这样，否则会报 HTTP 599: SSL certificate problem: unable to get local issuer certificate 错误

首页：

点击绿色的 run 执行，你会看到 follows 上面有一个红色的 1，切换到 follows 面板，点击绿色的播放按钮：

pypider > V2EX

```
{
  "process": {
    "callback": "on_start"
  },
  "project": "V2EX",
  "taskid": "data:,on_start",
  "url": "data:,on_start"
}
```

on_start

enable css selector helper

web

html

follows¹

messages



第二张截图一开始是出现这个问题了，解决办法看前面写的文章，后来问题就不再会出现了。

Tab 列表页：

```
pyspider > V2EX

{
  "fetch": {
    "validate_cert": false
  },
  "process": {
    "callback": "index_page"
  },
  "project": "V2EX",
  "schedule": {
    "age": 864000
  },
  "taskid": "627db7a08834b93df9c3e6ec4095bd15",
  "url": "https://www.v2ex.com/"
}
run

tab_page > https://www.v2ex.com/?tab=tech
tab_page > https://www.v2ex.com/?tab=creative
tab_page > https://www.v2ex.com/?tab=play
tab_page > https://www.v2ex.com/?tab=apple
tab_page > https://www.v2ex.com/?tab=jobs
tab_page > https://www.v2ex.com/?tab=deals
tab_page > https://www.v2ex.com/?tab=city
tab_page > https://www.v2ex.com/?tab=qna
tab_page > https://www.v2ex.com/?tab=hot
tab_page > https://www.v2ex.com/?tab=all
tab_page > https://www.v2ex.com/?tab=r2
```

在 tab 列表页 中，我们需要提取出所有的主题列表页 的 URL。你可能已经发现了，sample handler 已经提取了非常多的 URL

代码：

```
@config(age=10 * 24 * 60 * 60)
def index_page(self, response):
    for each in response.doc('a[href^="https://www.v2ex.com/?tab="]').items():
        self.crawl(each.attr.href, callback=self.tab_page, validate_cert=False)
```

- 由于帖子列表页和 tab列表页长的并不一样，在这里新建了一个 callback 为 self.tab_page
- @config(age=10 * 24 * 60 * 60) 在这表示我们认为 10 天内页面有效，不会再次进行更新抓取

Go列表页：

```
pyspider > V2EX
{
  "project": "V2EX",
  "schedule": {
    "age": 864000
  },
  "taskId": "5565041b5dcfbfc47a677d964d42c978",
  "url": "https://www.v2ex.com/?tab=tech"
}

board_page > https://www.v2ex.com/go/qna
board_page > https://www.v2ex.com/go/share
board_page > https://www.v2ex.com/go/jobs
board_page > https://www.v2ex.com/go/programmer
board_page > https://www.v2ex.com/go/macOS
board_page > https://www.v2ex.com/go/create
board_page > https://www.v2ex.com/go/python
board_page > https://www.v2ex.com/go/iphone
board_page > https://www.v2ex.com/go/android
board_page > https://www.v2ex.com/go/dev
board_page > https://www.v2ex.com/go/apple
board_page > https://www.v2ex.com/go/linux
board_page > https://www.v2ex.com/go/v2ex
board_page > https://www.v2ex.com/go/pixel
board_page > https://www.v2ex.com/go/stats
board_page > https://www.v2ex.com/go/mermaid
board_page > https://www.v2ex.com/go/mermaid

enable css selector helper web html follow 160 messages

# Project: V2EX
from pyspider.libs.base_handler import *

class Handler(BaseHandler):
    crawl_config = {
    }

    @every(minutes=24 * 60)
    def on_start(self):
        self.crawl('https://www.v2ex.com/', callback=self.index_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def index_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/?tab="]').items():
            self.crawl(each.attr.href, callback=self.tab_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def tab_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/go/"]').items():
            self.crawl(each.attr.href, callback=self.board_page, validate_cert=False)

    @config(priority=2)
    def board_page(self, response):
        return {
            'url': response.url,
            'title': response.doc('title').text(),
        }

    @config(priority=2)
    def detail_page(self, response):
        return {
            'url': response.url,
            'title': response.doc('title').text(),
        }
```

代码:

```
@config(age=10 * 24 * 60 * 60)
def tab_page(self, response):
    for each in response.doc('a[href^="https://www.v2ex.com/go/"]').items():
        self.crawl(each.attr.href, callback=self.board_page, validate_cert=False)
```

帖子详情页 (T):

```
pyspider > V2EX
{
  "fetch": {
    "validate_cert": false
  },
  "process": {
    "callback": "board_page"
  },
  "project": "V2EX",
  "schedule": {
    "age": 864000
  },
  "taskId": "955a8c6189c1d90674eeada29a206fa3",
  "url": "https://www.v2ex.com/go/programmer"
}

detail_page > https://www.v2ex.com/t/314133#reply13
detail_page > https://www.v2ex.com/t/314225#reply0
detail_page > https://www.v2ex.com/t/314074#reply71
detail_page > https://www.v2ex.com/t/314176#reply10
detail_page > https://www.v2ex.com/t/313968#reply140
detail_page > https://www.v2ex.com/t/314107#reply22
detail_page > https://www.v2ex.com/t/314156#reply18
detail_page > https://www.v2ex.com/t/314168#reply3
detail_page > https://www.v2ex.com/t/313979#reply29
detail_page > https://www.v2ex.com/t/314055#reply6
detail_page > https://www.v2ex.com/t/314193#reply0
detail_page > https://www.v2ex.com/t/314077#reply0
detail_page > https://www.v2ex.com/t/314096#reply0

enable css selector helper web html follow 20 messages

# created by 10412
# !/usr/bin/env python
# -*- encoding: utf-8 -*-
# Created on 2016-10-18 20:43:00
# Project: V2EX

from pyspider.libs.base_handler import *

class Handler(BaseHandler):
    crawl_config = {
    }

    @every(minutes=24 * 60)
    def on_start(self):
        self.crawl('https://www.v2ex.com/', callback=self.index_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def index_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/?tab="]').items():
            self.crawl(each.attr.href, callback=self.tab_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def tab_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/go/"]').items():
            self.crawl(each.attr.href, callback=self.board_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def board_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/t/"]').items():
            self.crawl(each.attr.href, callback=self.detail_page, validate_cert=False)

    @config(priority=2)
    def detail_page(self, response):
        title = response.doc('h1').text()
        content = response.doc('div.topic_content').html()
        return {
            'url': response.url,
            'title': title,
        }
```

你可以看到结果里面出现了一些reply的东西,对于这些我们是可以不需要的,我们可以去掉。

同时我们还需要让他自己实现自动翻页功能。

代码:

```
@config(age=10 * 24 * 60 * 60)
def board_page(self, response):
    for each in response.doc('a[href^="https://www.v2ex.com/t/"]').items():
        url = each.attr.href
        if url.find('#reply')>0:
            url = url[url.find('#')]
        self.crawl(url, callback=self.detail_page, validate_cert=False)
    for each in response.doc('a.page_normal').items():
        self.crawl(each.attr.href, callback=self.board_page, validate_cert=False) #实现自动翻页
功能
```

去掉后的运行截图：

不需要的东西去掉了

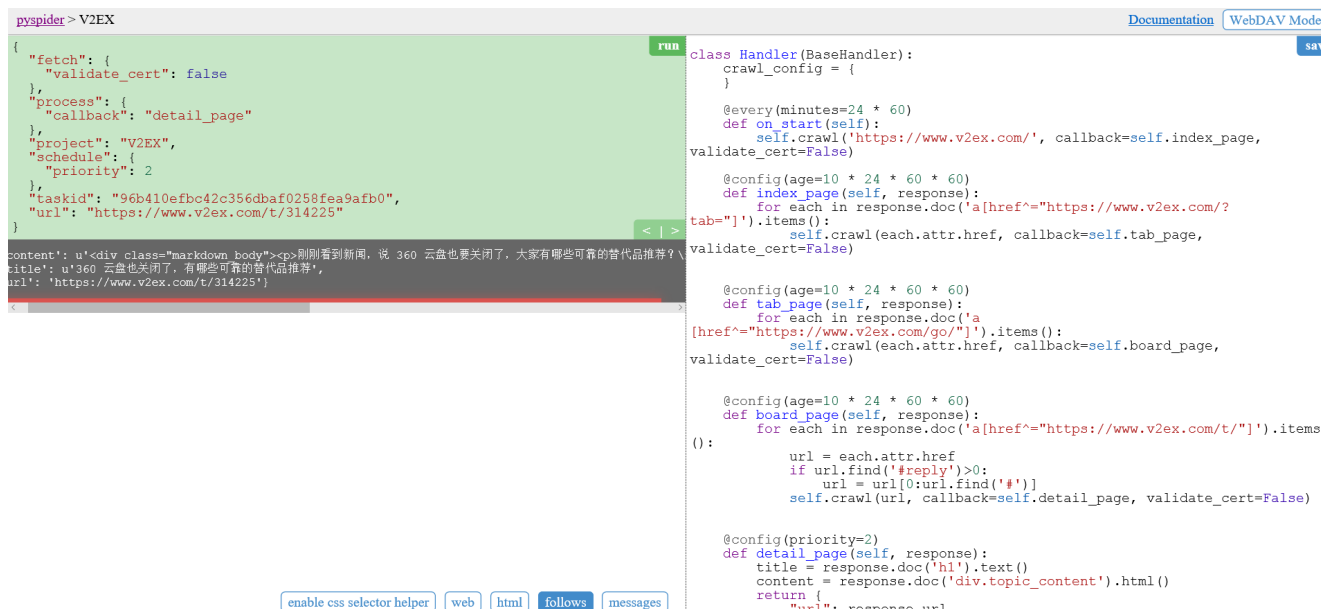
将后面的一些不需要的东西给去掉

实现自动翻页后的截图：

实现翻页功能

此时我们已经可以匹配了所有的帖子的 url 了。

点击每个帖子后面的按钮就可以查看帖子具体详情了。



代码:

```
@config(priority=2)
def detail_page(self, response):
    title = response.doc('h1').text()
    content = response.doc('div.topic_content').html().replace("'", '\\\'')
    self.add_question(title, content) #插入数据库
    return {
        "url": response.url,
        "title": title,
        "content": content,
    }
```

插入数据库的话, 需要我们在之前定义一个add_question函数。

```
#连接数据库
def __init__(self):
    self.db = MySQLdb.connect('localhost', 'root', 'root', 'wenda', charset='utf8')

    def add_question(self, title, content):
        try:
            cursor = self.db.cursor()
            sql = 'insert into question(title, content, user_id, created_date, comment_count)
values ("%s", "%s", %d, %s, 0)' % (title, content, random.randint(1, 10), 'now()'); #插入数据库的SQL语句

            print sql
            cursor.execute(sql)
            print cursor.lastrowid
            self.db.commit()
        except Exception, e:
            print e
            self.db.rollback()
```

查看爬虫运行结果:

pyspider dashboard

| | | | | | | |
|-----------|---|---------|-------|-----------|---|---------------|
| scheduler | 0 | fetcher | 0 | processor | 0 | result_worker |
| | | | 0 + 0 | | | |

Recent Active Tasks

Create

| group | project name | status | rate/burst | avg time | progress | actions |
|-------|--------------|---------|------------|----------|-----------------|------------------------|
| c12 | V2EX | RUNNING | 0.1/1.0 | | 5m1h1dall: 1249 | RunActive TasksResults |

1. 先debug下，再调成running。pyspider框架在windows下的bug
2. 设置跑的速度，建议不要跑的太快，否则很容易被发现是爬虫的，人家就会把你的IP给封掉的
3. 查看运行工作
4. 查看爬取下来的内容

| | | | |
|---------|---|----------------|---------------------|
| SUCCESS | V2EX > https://www.v2ex.com/go/photograph | 9 seconds ago | 272.0+39.00ms +30 |
| SUCCESS | V2EX > https://www.v2ex.com/?tab=tech | 19 seconds ago | 133.0+172.00ms +160 |
| SUCCESS | V2EX > https://www.v2ex.com/?tab=r2 | 29 seconds ago | 214.0+62.00ms +165 |
| SUCCESS | V2EX > https://www.v2ex.com/?tab=city | 39 seconds ago | 214.0+97.00ms +157 |
| SUCCESS | V2EX > https://www.v2ex.com/?tab=play | 49 seconds ago | 143.0+177.00ms +158 |
| SUCCESS | V2EX > https://www.v2ex.com/ | 1 minute ago | 383.0+118.00ms +11 |
| SUCCESS | V2EX > data:.on_start | 1 minute ago | 0.0+1.00ms +1 |
| SUCCESS | V2EX > data:.on_start | 1 minute ago | 0.0+14.00ms +1 |
| SUCCESS | V2EX > data:.on_get_info | 4 hours ago | |

| V2EX - Results | | | | JSON | URL-JSON | CSV |
|-----------------------------------|---|--|-------------------------------------|------|----------|-----|
| url | content | title | url | | | |
| https://www.v2ex.co m/t/304441 | 捣鼓了一会儿，发现除了电子取景器有些恼，其他都还不错，很小巧，手感很好。目前也只有两个镜头，一个 50 f1.4，一个 16-35 f4，后者估计里面也摔坏了，准备拿去修。 ... | "出去玩 D750 摔得支离破碎，惨不忍睹.....是时候换入门 A7 II 了吗？有没有用过的分享下体会" | "https://www.v2ex.co m/t/304441" | | | |
| https://www.v2ex.co m/t/283970 | "<div class=\\\\"markdown_body\\\\"><p>自己平时拍了几张照片，有的是觉得蛮有纪念意义，有些是觉得还蛮有意思的，然后将部分照片上传到了 Blog 作为存档。虽然我... | "自己拍摄的照片，怎么保护版权？" | "https://www.v2ex.co m/t/283970" | | | |
| https://www.v2ex.co m/t/250036 | "看了一些评测，小三元的成像质量（色散，形变，锐度等等）和做工还是比较好的，唯一考虑的就是价格，还有 200mm 的会不会不够用？ 70-300mm 的话，300mm 的感觉应该... | "想买一只长焦头打鸟，在犹豫小三元的 70-200mm f/4 还有比较便宜的 70-300mm f/4.5-5.6G" | "https://www.v2ex.co m/t/250036" | | | |
| https://www.v2ex.co m/t/300544 | "<a target=\\\\"_blank\\\\" href=\\\\"https://cycleuser.tuchong.com/followers/\\\\" rel=\\\\"nofollow\\..." | "摄影入门中，有同好的朋友来圈虫互粉可好？" | "https://www.v2ex.co m/t/300544" | | | |
| https://www.v2ex.co m/t/263980 | "刚入了 a5100 想出来练练手。顺便放松一下" | "魔都有什么可以练习拍照的地方" | "https://www.v2ex.co m/t/263980" | | | |
| https://www.v2ex.co m/t/284914 | "1.Paypal 绑了中行的 visa，但是每次付款都提示 " Unfortunately, your payment was declined "  <i>... | "在 ebay 上拍老相机遇到问题，大家帮忙看下" | "https://www.v2ex.co m/t/284914" | | | |
| https://www.v2ex.co m/t/297913 | "<div class=\\\\"markdown_body\\\\"><p>如题，手上还有个大疆的 osmo，不过默认智能拍 360 度的照片，不清楚头上和脚下怎么拍。</p></div>" | "如何用手机等常见工具拍出 360x180 的全景照片" | "https://www.v2ex.co m/t/297913" | | | |
| https://www.v2ex.co m/t/262324 | "<div class=\\\\"markdown_body\\\\"><p><a target=\\\\"_blank\\\\" rel=\\\\"nofollow\\\\" href=\\\\"https... | "Flickr 开始收费了" | "https://www.v2ex.co m/t/262324" | | | |
| https://www.v2ex.co m/t/261827 | "<div class=\\\\"markdown_body\\\\"><p>第一次买单反，还没来得及看摄影教程的新手。预算[6000,7000]。求各路牛人，给点这两台相机的优缺点。谢谢。</p>... | "纠结佳能 70D 和尼康 D7200" | "https://www.v2ex.co m/t/261827" | | | |
| https://www.v2ex.co m/t/289038 | "<div class=\\\\"markdown_body\\\\"><p>之前有用过一段时间的 [搜图壁纸]，但觉得图片不是很清晰在加上有些吃内存，所以就停用了。就像之前用的输入法是 [搜狗拼... | "大家有没有好用的 PC 端 [电脑壁纸] 软件，尽量是少占些内存的..." | "https://www.v2ex.co m/t/289038" | | | |
| https://www.v2ex.co m/t/240744 | "我相信有发现美得眼睛，和好的场合即使最原始 iPhone4 也能拍出好看的照片~大家怎么认为?" | "摄影不在于设备，在于灵感 and 场合。对吧？" | "https://www.v2ex.co m/t/240744" | | | |
| https://www.v2ex.co m/t/250438 | "没找到 \\\\"美\\\\"这个节点。就来摄影了。  视觉艺术作者，以前是绘画，雕塑，现在摄影也算吧。这是一类人。下面拿摄影代指这些视觉艺术。&#... | "对展示人的身体的摄影作品如何看？" | "https://www.v2ex.co m/t/250438" | | | |
| https://www.v2ex.co | "之前入手了个微单，想学学摄影，丰富下精神生活。可惜平时太懒，勉强把相机的说明书看完就出去各种乱拍。 " | "敲代码之余，拍拍照片陶冶下情操" | "https://www.v2ex.co" | | | |

然后再本地数据库GUI软件上查询下就可以看到数据已经保存到本地了。

自己需要用的话就可以导入出来了。

在开头我就告诉大家爬虫的代码了，如果详细的看看那个**project**，你就会找到我上传的爬取数据了。（仅供学习使用，切勿商用！）

当然你还会看到其他的爬虫代码的了，如果你觉得不错可以给个 **Star**，或者你也感兴趣的话，你可以**fork**我的项目，和我一起学习，这个项目长期更新下去。

最后：

代码：

```

# created by 10412
# !/usr/bin/env python
# -*- encoding: utf-8 -*-
# Created on 2016-10-20 20:43:00
# Project: V2EX

from pypider.libs.base_handler import *

import re
import random
import MySQLdb

class Handler(BaseHandler):
    crawl_config = {
    }

    def __init__(self):
        self.db = MySQLdb.connect('localhost', 'root', 'root', 'wenda', charset='utf8')

    def add_question(self, title, content):
        try:
            cursor = self.db.cursor()
            sql = 'insert into question(title, content, user_id, created_date, comment_count)
values ("%s", "%s", %d, %s, 0)' % (title, content, random.randint(1, 10), 'now()');
            print sql
            cursor.execute(sql)
            print cursor.lastrowid
            self.db.commit()
        except Exception, e:
            print e
            self.db.rollback()

    @every(minutes=24 * 60)
    def on_start(self):
        self.crawl('https://www.v2ex.com/', callback=self.index_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def index_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/?tab="]').items():
            self.crawl(each.attr.href, callback=self.tab_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def tab_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/go/"]').items():
            self.crawl(each.attr.href, callback=self.board_page, validate_cert=False)

    @config(age=10 * 24 * 60 * 60)
    def board_page(self, response):
        for each in response.doc('a[href^="https://www.v2ex.com/t/"]').items():
            url = each.attr.href

```

```
        if url.find('#reply')>0:
            url = url[url.find('#'):]
        self.crawl(url, callback=self.detail_page, validate_cert=False)
    for each in response.doc('a.page_normal').items():
        self.crawl(each.attr.href, callback=self.board_page, validate_cert=False)

@config(priority=2)
def detail_page(self, response):
    title = response.doc('h1').text()
    content = response.doc('div.topic_content').html().replace('"', '\\\\"')
    self.add_question(title, content) #插入数据库
    return {
        "url": response.url,
        "title": title,
        "content": content,
    }
```