[Искатель Python продвинул четыре использования PySpider](https://cuiqingcai.com/2652.html)

[питон](https://cuiqingcai.com/category/technique/python)  [Цуй Цинкай](https://cuiqingcai.com/author/cqcre)  2 года назад (2016-03-25)  Просмотр 111854  [158 отзывов](https://cuiqingcai.com/2652.html#comments)

**Судя по ситуации**

PySpider - это инфраструктура искателя, которая, как мне кажется, очень удобна и мощна. Она поддерживает многопоточное сканирование, динамический синтаксический анализ JS и предоставляет такие функции, как операционный интерфейс, повтор ошибки, тайминг и т. Д., Что очень удобно.

Эта статья понимает процесс работы PySpider, создавая у меня интересный проект PySpider.

**Вербовка лошадей**

Пожалуйста, обратитесь к этому разделу для конкретного процесса установки.

[устанавливать](http://cuiqingcai.com/2443.html)

Ну, после того, как вы его установили, сделайте это со мной.

**лебедь**

Статья, которую я написал ранее

[Захватите фотографии Taobao MM](http://cuiqingcai.com/1001.html)

Из-за пересмотра веб-страницы URL, требуемый во время процесса обхода, требует динамического разбора JS, поэтому ранее используемый urllib2 больше не может использоваться. Здесь мы используем PySpider для его повторного внедрения.

Итак, теперь нам нужно захватить личную информацию и фотографии Taobao MM и сохранить их локально.

**Судя по ситуации**

Обход целевого веб-сайта: <https://mm.taobao.com/json/request_top_list.htm?page=1> . После того, как вы его откроете, вы можете увидеть список многих MMC Taobao.

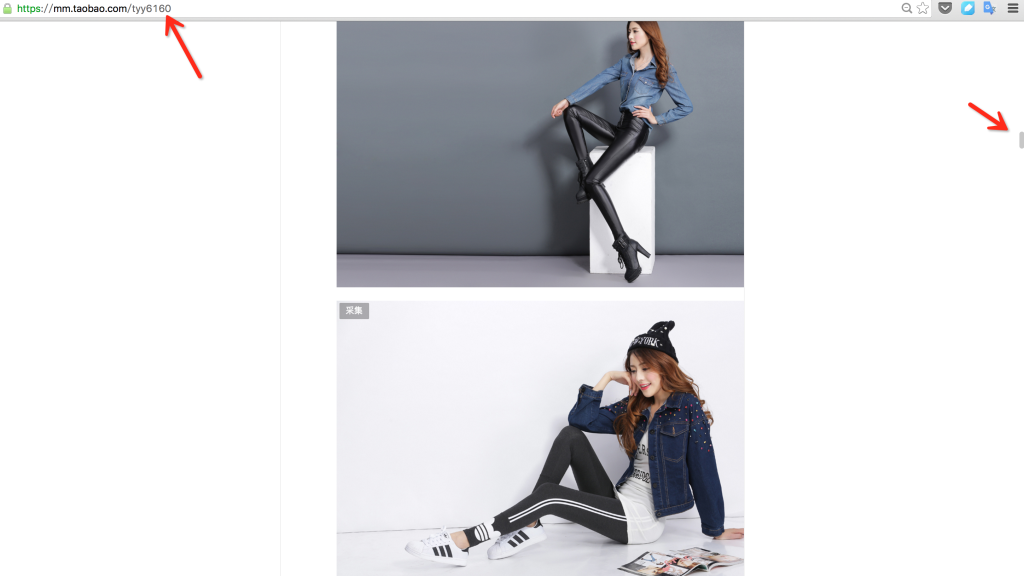
Сколько стоит список?

[Https://mm.taobao.com/json/request\_top\_list.htm?page=10000](https://mm.taobao.com/json/request_top_list.htm?page=10000) , стр. 10000, посмотрите, сколько вы хотите. Я ничего не знаю.

Случайно нажмите на имя ММ, чтобы увидеть ее основную информацию.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160326-4@2x.png)

Вы можете видеть, что на картинке есть персональное имя домена, мы копируем его в браузер, чтобы открыть. [mm.taobao.com/tyy6160](https://mm.taobao.com/tyy6160)

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160326-5@2x.png)

Ну, перетащив вниз, много MM-фотографий здесь, что вы знаете, нам нужно сохранить их фотографии и личную информацию.

**PS Обратите внимание на индикатор выполнения на картинке! Угадайте, сколько картинок ~**

**Меч**

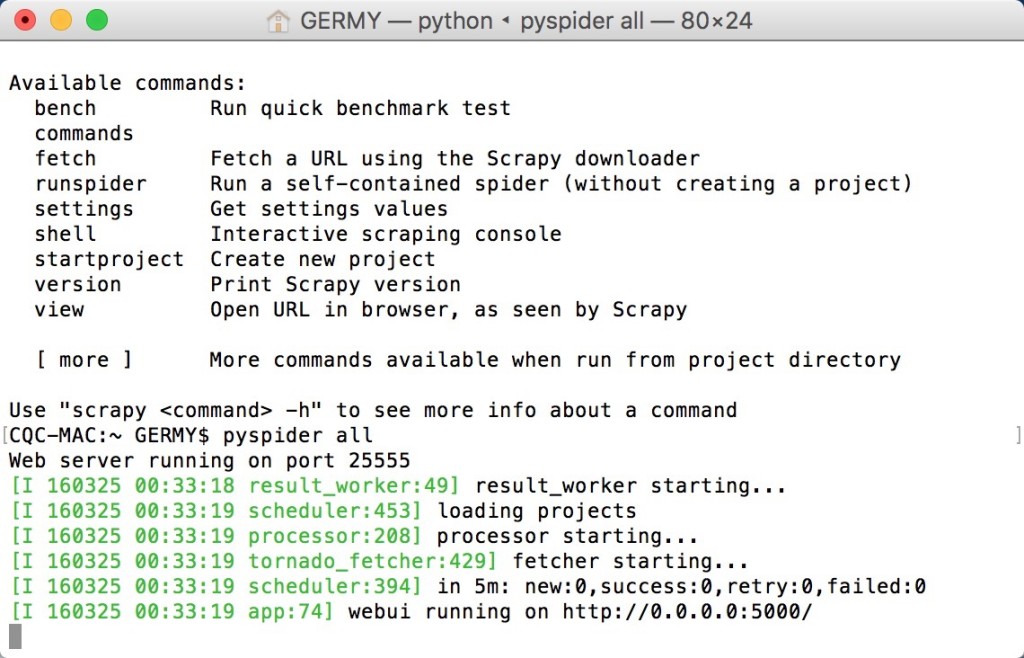
После успешной установки следуйте за мной шаг за шагом, чтобы завершить сканирование сайта, вы поймете, как использовать PySpider.

Выполнить из командной строки



|  |  |
| --- | --- |
| 1 | pyspider all |

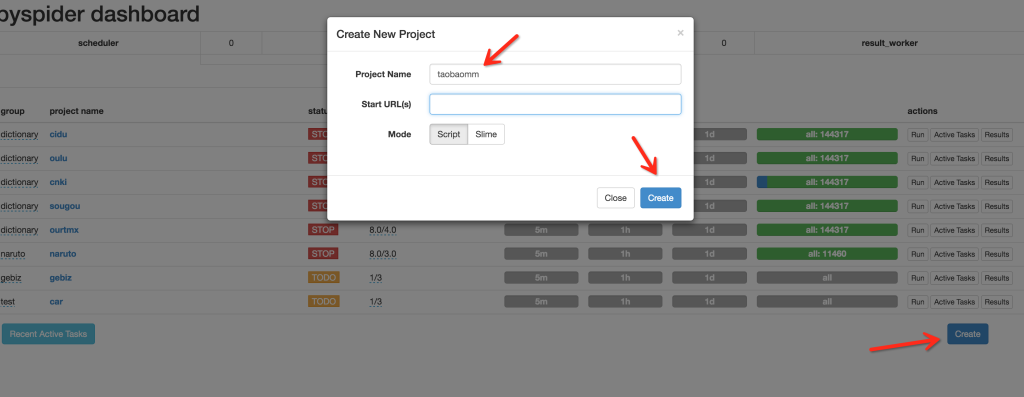
Смысл этой команды заключается в запуске pyspider и запуске всех его компонентов.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/E6632A0A-9067-4B97-93A2-5DEF23FB4CD8.jpg)

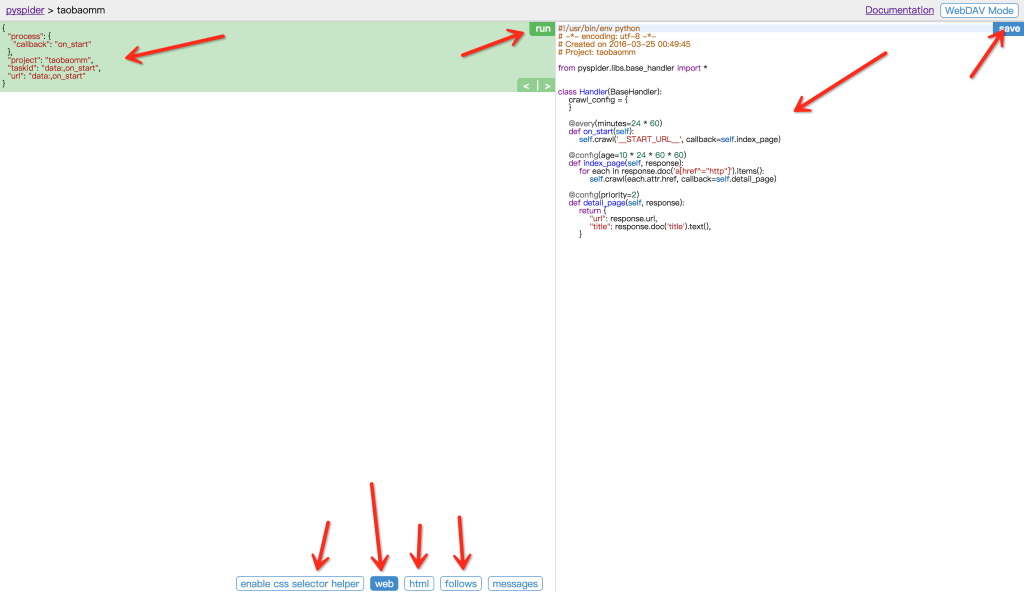
Можно узнать, что программа запускается нормально и работает на этом порту 5000.

**При прикосновении волос**

Затем введите в браузере  [http: // localhost: 5000](http://localhost:5000/) , вы увидите основной интерфейс PySpider, нажмите «Создать» в нижнем правом углу, назовите его taobaomm, конечно, вы можете свободно выбирать имя, продолжать нажимать «Создать».

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-0@2x.png)

Таким образом, мы перейдем на страницу операции сканирования.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-1@2x.png)

Вся страница разделена на два столбца, левая сторона - область предварительного просмотра страницы обхода, а правая сторона - область написания кода. Блок поясняется ниже:

Зеленая область слева: переменная JSON, соответствующая этому запросу. В PySpider каждый запрос имеет соответствующую переменную JSON, включая функцию обратного вызова, имя метода, ссылку запроса, данные запроса и т. Д.

Запустите в правом верхнем углу зеленой зоны: нажмите кнопку запуска в правом верхнем углу, чтобы выполнить запрос. Запрашиваемый результат может появиться в белом поле слева.

Включите помощник css selector слева: после захвата страницы нажмите эту кнопку, чтобы легко получить селектор CSS для элемента на странице.

Левый веб: предварительный просмотр захваченной страницы.

Left html: Возьмите HTML-код страницы.

Слева слева: Если в текущем методе обхода создается новый запрос обхода, следующий запрос появится в следующем порядке.

Сообщения слева: некоторая информация выводится во время процесса обхода.

Правая область кода: вы можете написать код в правой области и нажать кнопку «Сохранить» в правом верхнем углу для сохранения.

Правильный режим WebDAV: включите режим отладки и увеличьте левую сторону для удобства просмотра и отладки.

**Победа в погоне**

Еще URL-адрес предыдущего раздела, <https://mm.taobao.com/json/request_top_list.htm?page=1> , где параметр страницы представляет номер страницы. Поэтому мы временно захватываем первые 30 страниц. Номер страницы можно настроить по желанию в конце.

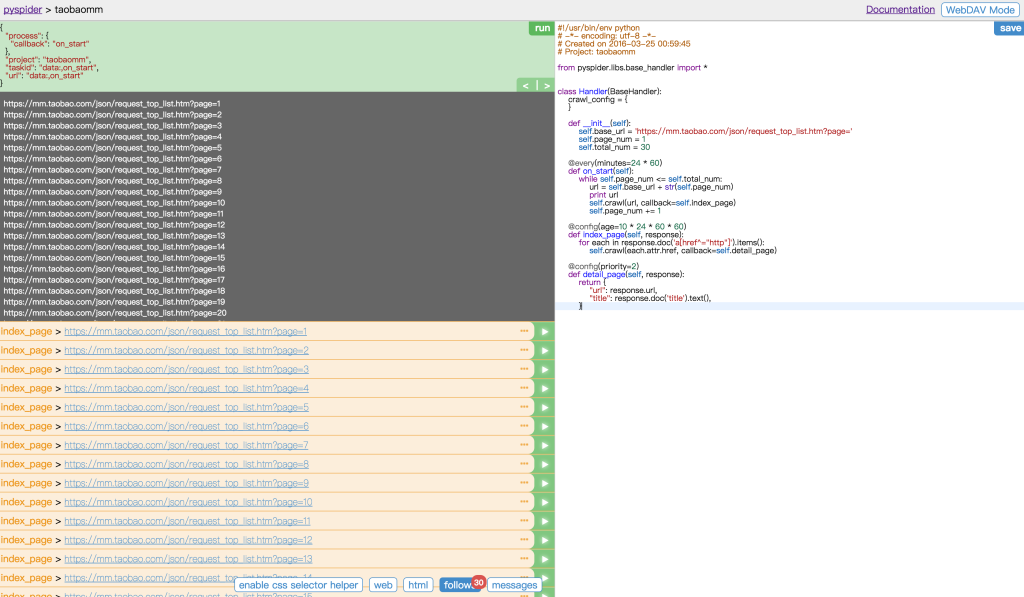
Сначала мы определяем базовый адрес, а затем определяем номер страницы и общий номер страницы для обхода.

Python



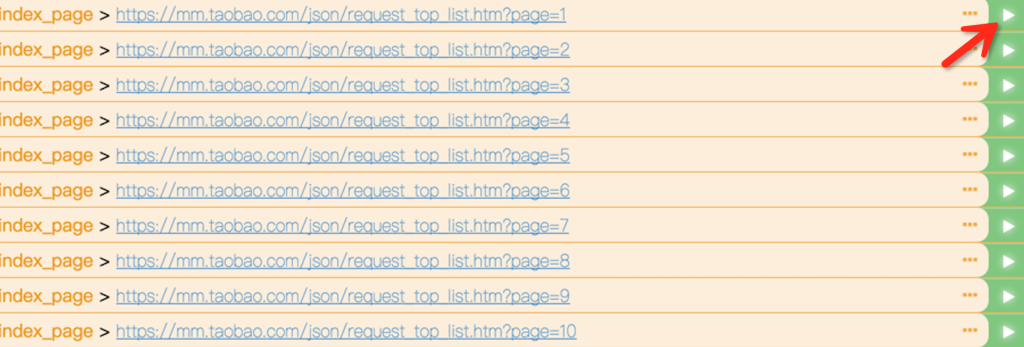
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | from pyspider.libs.base\_handler import \*      class Handler(BaseHandler):      crawl\_config = {      }        def \_\_init\_\_(self):          self.base\_url = 'https://mm.taobao.com/json/request\_top\_list.htm?page='          self.page\_num = 1          self.total\_num = 30        @every(minutes=24 \* 60)      def on\_start(self):          while self.page\_num <= self.total\_num:              url = self.base\_url + str(self.page\_num)              print url              self.crawl(url, callback=self.index\_page)              self.page\_num += 1        @config(age=10 \* 24 \* 60 \* 60)      def index\_page(self, response):          for each in response.doc('a[href^="http"]').items():              self.crawl(each.attr.href, callback=self.detail\_page)        @config(priority=2)      def detail\_page(self, response):          return {              "url": response.url,              "title": response.doc('title').text(),          } |

Нажмите «Сохранить», чтобы сохранить код, затем нажмите «Запустить» слева, чтобы запустить код.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-2@2x.png)

После запуска мы найдем номер 30 в следующем, указав, что у нас есть 30 новых запросов, нажмите, чтобы просмотреть все списки сканирования. Кроме того, на консоли также есть вывод, на котором распечатываются все URL-адреса, подлежащие обходу.

Затем мы нажимаем на любую из зеленых стрелок слева, чтобы продолжить сканирование этой страницы. Например, нажмите на первый URL-адрес для обхода этого URL-адреса.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-3@2x.png)

После щелчка, а затем просмотра веб-страницы ниже вы можете просмотреть страницу в режиме реального времени. Эта страница была просканирована нами и перезвана обратно к функции index\_page для обработки. В настоящее время функция index\_page еще не обработана, поэтому мы продолжаем строить все запросы ссылок.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-4@2x.png)

Хорошо, что мы будем делать дальше? Конечно, я пошел в ММ, чтобы перейти на личную страницу, чтобы ползать.

**В разгаре**

Перейдите в список MM, затем перейдите на страницу сведений о MM и измените метод index\_page.



|  |  |
| --- | --- |
| 1  2  3 | def index\_page(self, response):      for each in response.doc('.lady-name').items():          self.crawl(each.attr.href, callback=self.detail\_page) |

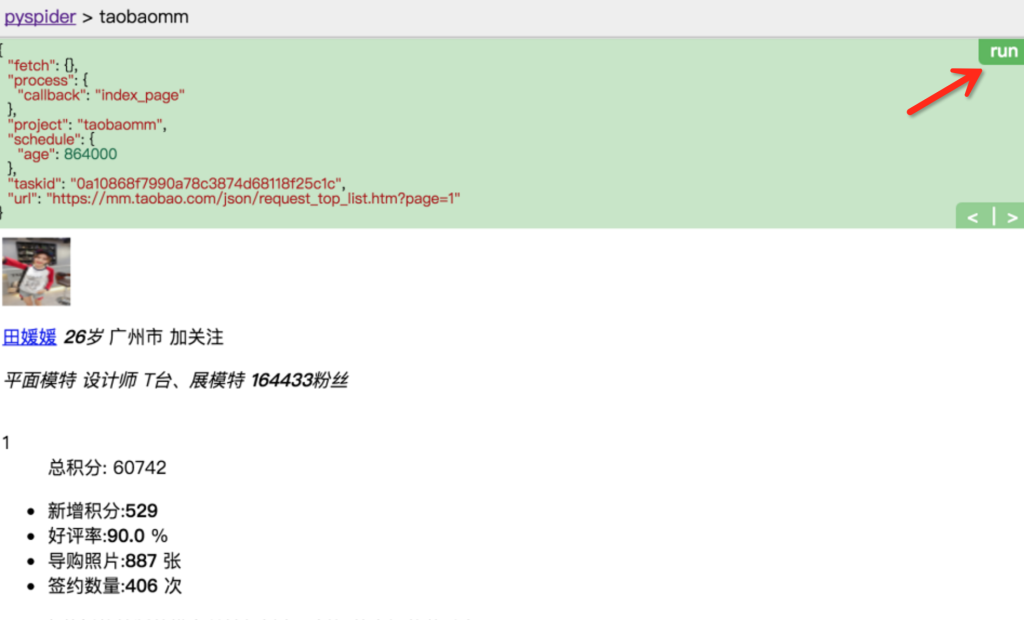
Ответ - это только просматриваемая страница списка. Ответ на самом деле эквивалентен html-коду страницы списка. Используя функцию doc, вызывается PyQuery, селектор CSS используется для получения ссылки каждого MM, а затем новый запрос повторно инициируется.

Например, здесь мы получаем each.attr.href может быть  [mm.taobao.com/self/model\_card.htm?user\_id=687471686](http://mm.taobao.com/self/model_card.htm?user_id=687471686) , здесь продолжают вызывать метод обхода, представители продолжают получать детали этой ссылке.



|  |  |
| --- | --- |
| 1 | self.crawl(each.attr.href, callback=self.detail\_page) |

Тогда функция обратного вызова является подробной\_страницей, а результат обхода передается как переменная ответа. Detail\_page получает эту переменную и продолжает анализ ниже.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-7@2x.png)

Итак, давайте продолжим нажимать кнопку запуска, чтобы начать сканирование следующей страницы. Полученный результат - это.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-5@2x.png)

О, некоторые страницы не загружены, почему?

В предыдущей статье говорилось, эта страница является довольно особенным, правой части страницы с помощью JS-генерируемый рендеринга, в то время как обычные страницы сканирования, которые не могут быть оказаны после JS, который может быть в беде.

К счастью, однако, PySpider обеспечивает механизм динамического разбора JS.

Дружественный совет: некоторые друзья могут не знать PhantomJS, вы можете обратиться к

[Динамический анализ рептилий JS](http://cuiqingcai.com/2599.html)

Поскольку мы установили PhantomJS спереди, на этот раз настала очередь. В начале запуска PySpider pyspider allкоманда была использована для запуска всех компонентов PySpider, включая PhantomJS.

Итак, как нам изменить код? Очень просто.

Python



|  |  |
| --- | --- |
| 1  2  3 | def index\_page(self, response):      for each in response.doc('.lady-name').items():          self.crawl(each.attr.href, callback=self.detail\_page, fetch\_type='js') |

Просто добавьте fetch\_type = 'js', нажмите зеленую стрелку назад и запустите его снова.

Мы можем видеть, что страница была успешно загружена нами, она не может быть более красивой!

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160325-9@2x.png)

Смотрите следующие доменные имена кодировок, все, что нам нужно ММ картина там, так что мы должны продолжать сканировать страницу.

**Победа в поле зрения**

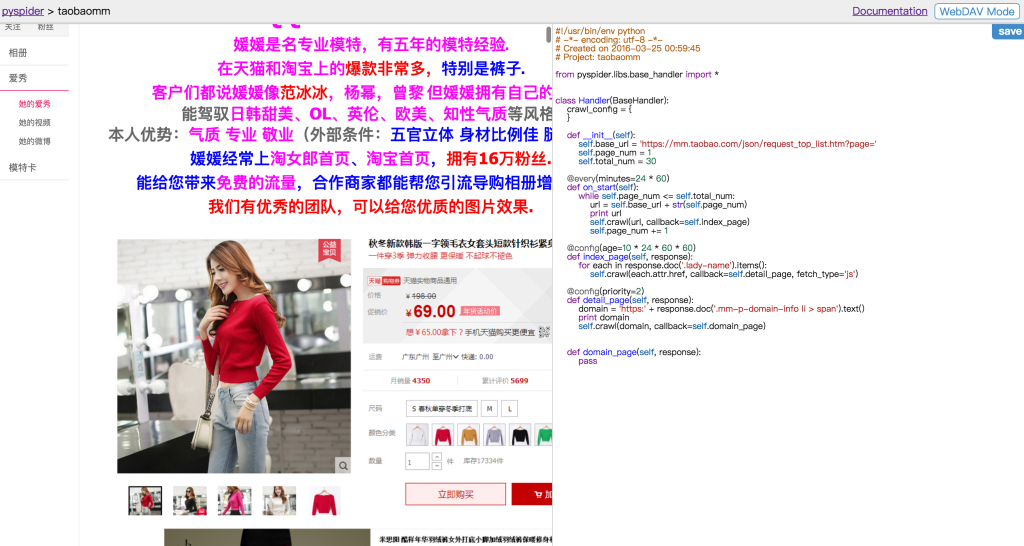
Итак, продолжайте изменять метод detail\_page, а затем добавьте метод domain\_page для обработки индивидуального имени домена для каждого MM.

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | def detail\_page(self, response):      domain = 'https:' + response.doc('.mm-p-domain-info li > span').text()      print domain      self.crawl(domain, callback=self.domain\_page)    def domain\_page(self, response):      pass |

Хорошо, продолжайте снова запускать, просматривайте страницу и, наконец, мы видим все образы MM.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160326-0@2x.png)

Ну, знаешь!

**Только должен ветер**

Хорошо, фотографии есть, тогда мы скажем их тайно ~

Улучшите код domain\_page, чтобы сохранить профиль и пройти метод сохранения изображения.

Здесь у PySpider есть функция, в которой все запросы сохраняются в очереди с дедупликацией и автоматическими механизмами повторной попытки. Поэтому наше лучшее решение - написать запрос для каждого изображения в качестве запроса, а затем записать его с файлом после успешного завершения, что позволит избежать проблемы с неполной загрузкой изображения.

Я использовал для написания процесса загрузки изображений и создания папок в предыдущей статье. Я не буду вдаваться в подробности здесь. Я напишу класс инструмента напрямую, и позже будет полный код.

Python



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | import os    class Deal:      def \_\_init\_\_(self):          self.path = DIR\_PATH          if not self.path.endswith('/'):              self.path = self.path + '/'          if not os.path.exists(self.path):              os.makedirs(self.path)        def mkDir(self, path):          path = path.strip()          dir\_path = self.path + path          exists = os.path.exists(dir\_path)          if not exists:              os.makedirs(dir\_path)              return dir\_path          else:              return dir\_path        def saveImg(self, content, path):          f = open(path, 'wb')          f.write(content)          f.close()        def saveBrief(self, content, dir\_path, name):          file\_name = dir\_path + "/" + name + ".txt"          f = open(file\_name, "w+")          f.write(content.encode('utf-8'))        def getExtension(self, url):          extension = url.split('.')[-1]          return extension |

Существует четыре метода.

mkDir: создать папку для создания папки, соответствующей имени MM.

saveBrief: Сохраните введение и сохраните текст MM.

saveImg: передать бинарный поток изображения и сохранить путь для сохранения изображения.

getExtension: Получить имя суффикса ссылки, полученной из URL-адреса изображения.

Затем в domain\_page для достижения следующего



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | def domain\_page(self, response):          name = response.doc('.mm-p-model-info-left-top dd > a').text()          dir\_path = self.deal.mkDir(name)          brief = response.doc('.mm-aixiu-content').text()          if dir\_path:              imgs = response.doc('.mm-aixiu-content img').items()              count = 1              self.deal.saveBrief(brief, dir\_path, name)              for img in imgs:                  url = img.attr.src                  if url:                      extension = self.deal.getExtension(url)                      file\_name = name + str(count) + '.' + extension                      count += 1                      self.crawl(img.attr.src, callback=self.save\_img,                                 save={'dir\_path': dir\_path, 'file\_name': file\_name})        def save\_img(self, response):          content = response.content          dir\_path = response.save['dir\_path']          file\_name = response.save['file\_name']          file\_path = dir\_path + '/' + file\_name          self.deal.saveImg(content, file\_path) |

Вышеуказанный метод сначала получает весь текст страницы, а затем вызывает метод saveBrief для хранения вставки.

Затем повторите все изображения MM и получите имя суффикса по ссылке, а также имя MM и автоинкремент, чтобы объединить новое имя файла, и вызовите метод saveImg, чтобы сохранить изображение.

**консуммировать**

Хорошо, основные вещи написаны.

Далее. Продолжайте улучшать код. Первая версия завершена.

**Функция 1-й версии: согласно подпапке имени Taobao MM, сохраните введение текста MM txt и все красивые изображения на локальном.**

Настраиваемые элементы:

* PAGE\_START: номер стартовой страницы списка
* PAGE\_END: ​​введите номер конечной страницы
* DIR\_PATH: путь сохранения ресурса

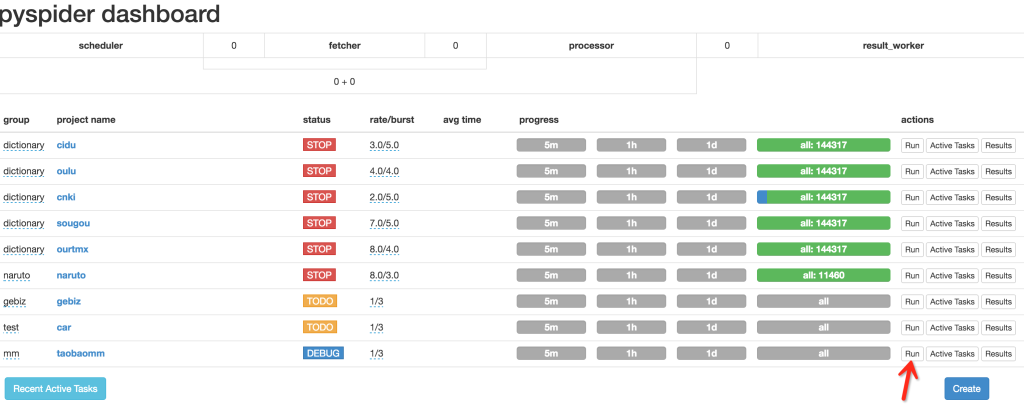
Python



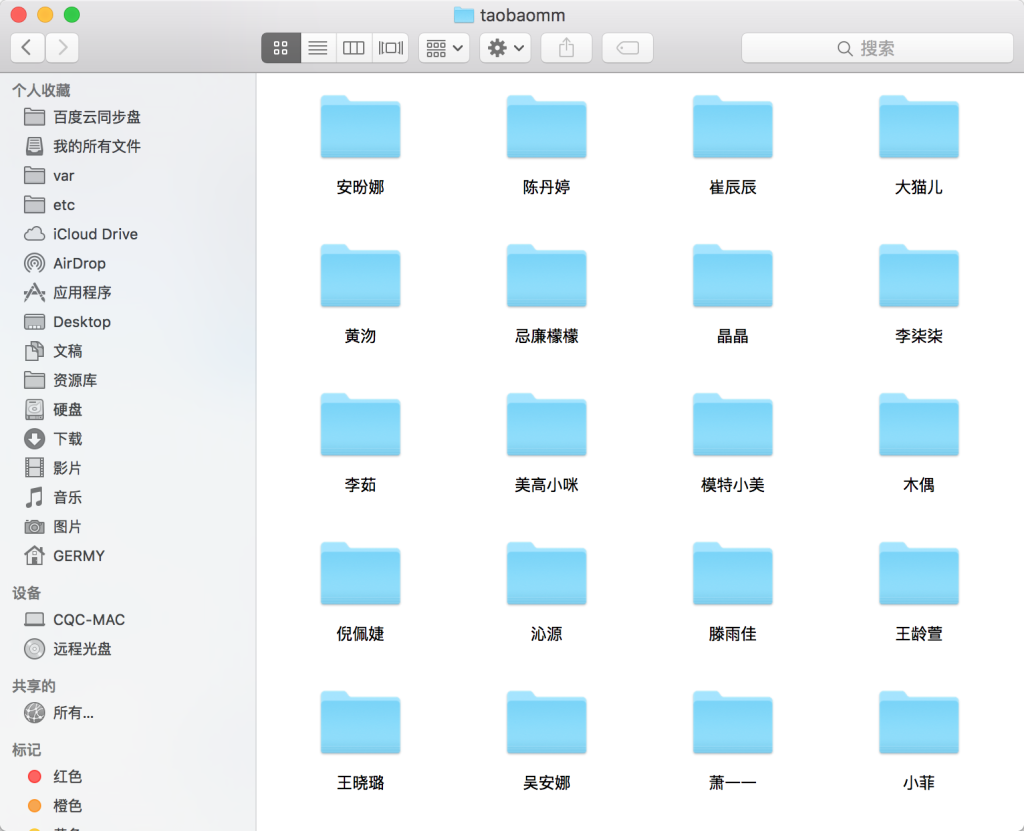
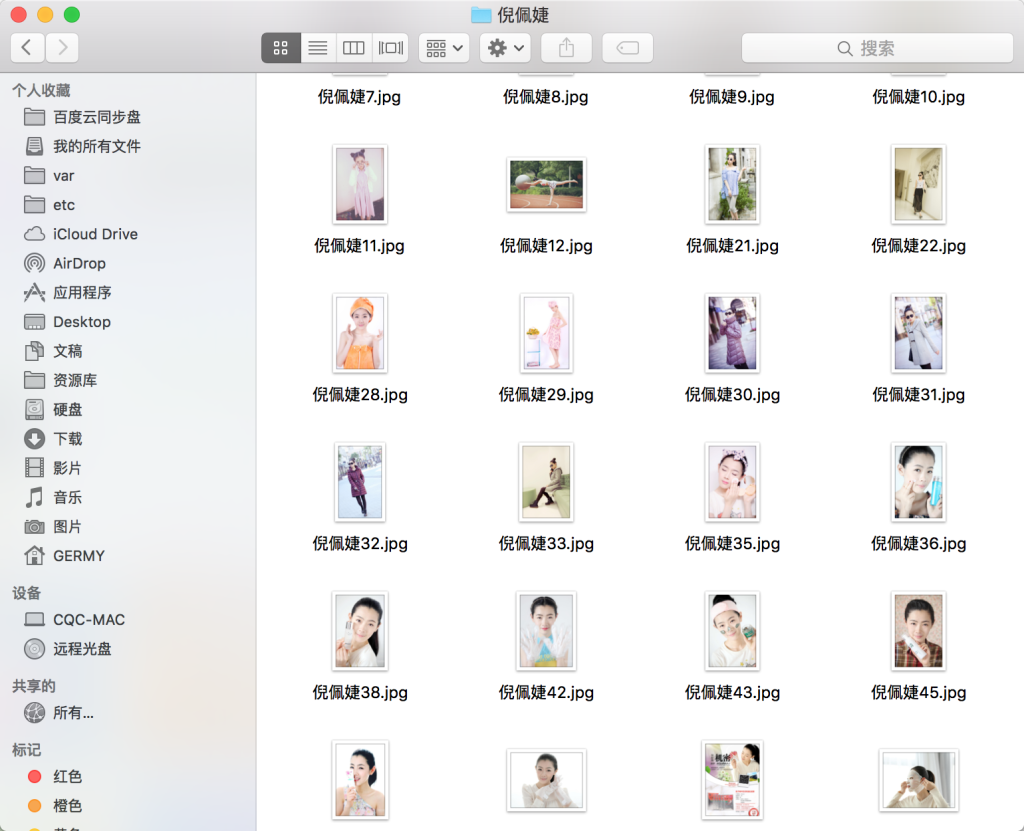
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 | #!/usr/bin/env python  # -\*- encoding: utf-8 -\*-  # Created on 2016-03-25 00:59:45  # Project: taobaomm    from pyspider.libs.base\_handler import \*    PAGE\_START = 1  PAGE\_END = 30  DIR\_PATH = '/var/py/mm'      class Handler(BaseHandler):      crawl\_config = {      }        def \_\_init\_\_(self):          self.base\_url = 'https://mm.taobao.com/json/request\_top\_list.htm?page='          self.page\_num = PAGE\_START          self.total\_num = PAGE\_END          self.deal = Deal()        def on\_start(self):          while self.page\_num <= self.total\_num:              url = self.base\_url + str(self.page\_num)              self.crawl(url, callback=self.index\_page)              self.page\_num += 1        def index\_page(self, response):          for each in response.doc('.lady-name').items():              self.crawl(each.attr.href, callback=self.detail\_page, fetch\_type='js')        def detail\_page(self, response):          domain = response.doc('.mm-p-domain-info li > span').text()          if domain:              page\_url = 'https:' + domain              self.crawl(page\_url, callback=self.domain\_page)        def domain\_page(self, response):          name = response.doc('.mm-p-model-info-left-top dd > a').text()          dir\_path = self.deal.mkDir(name)          brief = response.doc('.mm-aixiu-content').text()          if dir\_path:              imgs = response.doc('.mm-aixiu-content img').items()              count = 1              self.deal.saveBrief(brief, dir\_path, name)              for img in imgs:                  url = img.attr.src                  if url:                      extension = self.deal.getExtension(url)                      file\_name = name + str(count) + '.' + extension                      count += 1                      self.crawl(img.attr.src, callback=self.save\_img,                                 save={'dir\_path': dir\_path, 'file\_name': file\_name})        def save\_img(self, response):          content = response.content          dir\_path = response.save['dir\_path']          file\_name = response.save['file\_name']          file\_path = dir\_path + '/' + file\_name          self.deal.saveImg(content, file\_path)      import os    class Deal:      def \_\_init\_\_(self):          self.path = DIR\_PATH          if not self.path.endswith('/'):              self.path = self.path + '/'          if not os.path.exists(self.path):              os.makedirs(self.path)        def mkDir(self, path):          path = path.strip()          dir\_path = self.path + path          exists = os.path.exists(dir\_path)          if not exists:              os.makedirs(dir\_path)              return dir\_path          else:              return dir\_path        def saveImg(self, content, path):          f = open(path, 'wb')          f.write(content)          f.close()        def saveBrief(self, content, dir\_path, name):          file\_name = dir\_path + "/" + name + ".txt"          f = open(file\_name, "w+")          f.write(content.encode('utf-8'))        def getExtension(self, url):          extension = url.split('.')[-1]          return extension |

Вставьте в свой PySpider и запустите его ~

Есть некоторые точки знания, я сделаю подробное резюме использования позже. Сначала вы можете понять код.

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160326-1@2x.png)

После сохранения нажмите на пробег ниже, вы обнаружите, что огромное количество изображений MM вылилось в ваш компьютер ~

[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160326-2@2x.png)[](http://qiniu.cuiqingcai.com/wp-content/uploads/2016/03/QQ20160326-3@2x.png)

Нужно объяснить? Мне нужно не объяснять!

**Код проекта**

[TaobaoMM - GitHub](https://github.com/cqcre/TaobaoMM)

**Шан Фан Бао Цзянь**

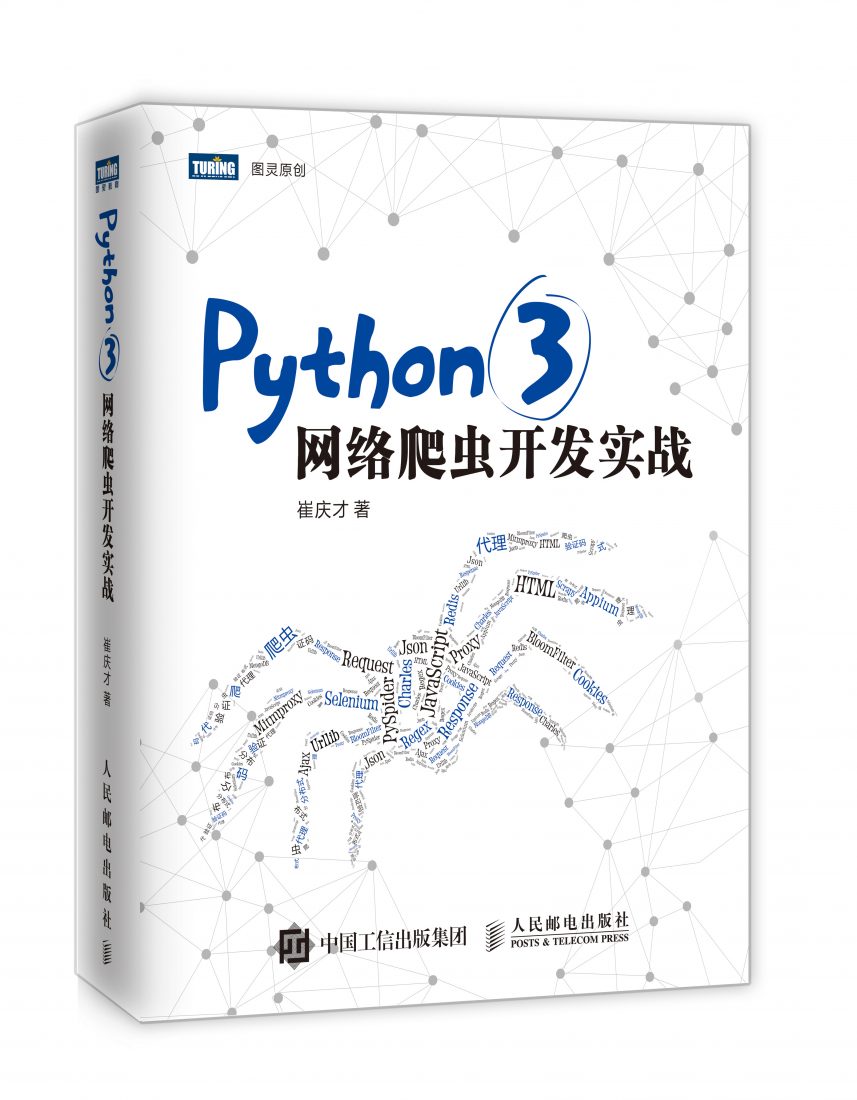
Если вы хотите узнать больше о PySpider, вы можете проверить официальную документацию.

[Официальный документ](http://docs.pyspider.org/en/latest/Quickstart/)

Перепечатано, пожалуйста, укажите: [quiet](https://cuiqingcai.com/) » [Prawon crawler расширен четыре использования PySpider](https://cuiqingcai.com/2652.html)

[Подобно ( 231 )](javascript:;)илиПоделиться ( 0 )

Хотите узнать больше о рептилиях? «Python 3 для веб-робота по поиску» эта книга может быть более подходящей для вас ~

[](https://item.jd.com/12333540.html)

[Узнать больше](https://cuiqingcai.com/6009.html)или[купить сейчас](https://item.jd.com/12333540.html)

Мой личный общественный номер WeChat

Код сканирования или поиск: Attack Coder



Ударный кодер

Микро-канал общедоступный номер подметать внимание

Хотите подружиться?

Приходите на атаку Кодера 瞧瞧



Ударный кодер

Номер группы QQ 99350970[сразу же присоединился](http://shang.qq.com/wpa/qunwpa?idkey=31ed8a7d15339353ad090379bb9243118f90220324a7a0f4b453bce52dcd02b6)

Есть ли слишком много воды для атакующего Кодера?

Вот чисто техническая территория



Агрессивный кодер

Номер группы QQ 627725766[сразу же присоединился](https://shang.qq.com/wpa/qunwpa?idkey=6096ee3de52a5a948a9fe45178ed9e9ba9084b909815c5f7775b85f4734fdda1)

Ваша поддержка - самая большая мотивация для блоггеров. Если вам нравится моя статья и чувствуете, что моя статья вам полезна, пожалуйста, нажмите ниже.

[Python](https://cuiqingcai.com/tag/python)[Рептилии](https://cuiqingcai.com/tag/%e7%88%ac%e8%99%ab)

[Пистолет-](https://cuiqingcai.com/2636.html)[манипулятор](https://cuiqingcai.com/2703.html)[Python шесть Использование](https://cuiqingcai.com/2636.html)[PyQuery PySpider HTTP 599: решение проблемы с ошибкой SSL-сертификата](https://cuiqingcai.com/2703.html)

* [[](https://cuiqingcai.com/6217.html)  
  Приходите узнать, как визуально контролировать ваш искатель](https://cuiqingcai.com/6217.html)
* [[](https://cuiqingcai.com/6214.html)  
  Введение в Elasticsearch и его внедрение в стыковку с Python](https://cuiqingcai.com/6214.html)
* [[](https://cuiqingcai.com/6182.html)  
  Предварительное изучение пакета Python glom](https://cuiqingcai.com/6182.html)
* [Придет третья волна деятельности по подаче книг «Python 3 Web Crawler Development»!](https://cuiqingcai.com/6173.html)
* [Приходите узнать, как визуально контролировать ваш искатель](https://cuiqingcai.com/6217.html)
* [Введение в Elasticsearch и его внедрение в стыковку с Python](https://cuiqingcai.com/6214.html)
* [Предварительное изучение пакета Python glom](https://cuiqingcai.com/6182.html)
* [Придет третья волна деятельности по подаче книг «Python 3 Web Crawler Development»!](https://cuiqingcai.com/6173.html)
* [Введение в использование асинхронных сопрограмм в Python](https://cuiqingcai.com/6160.html)
* [Приходит вторая волна Python3 Web Crawler Development Campaign!](https://cuiqingcai.com/6110.html)
* [Основное использование модуля протоколирования в Python](https://cuiqingcai.com/6080.html)
* [«Python 3 работает с поисковым роботом»!](https://cuiqingcai.com/6009.html)

Начало формы



Отправить комментарий



 Отправить комментарий

 выражение

Конец формы

 (158) небольшой партнер в Тукао

Blogger, я хотел бы спросить, как решить проблему автоматического поворота страницы js

MUM 2018-06-19 9:05 [Ответить](https://cuiqingcai.com/2652.html?replytocom=9813#respond)

1

0o鹏o0 2017-09-08 15:03 [ответить](https://cuiqingcai.com/2652.html?replytocom=4392#respond)

Привет, я смотрю ваш учебник! Я хотел бы спросить вас, как используется response.doc ()? Можешь мне сказать? Является ли вход один за другим тегом, который может быть помещен в оператор? Благодарю!

Qq874342339 2017-09-04 21:15 [ответить](https://cuiqingcai.com/2652.html?replytocom=4384#respond)

[Е 170825 19:04:35 base\_handler: 203] 501 Ошибка сервера Traceback (самый последний вызов последнего): Файл /usr/lib/python2.7/site-packages/pyspider/libs/base\_handler.py, строка 196, в run\_task результат = self.\_run\_task (задача, ответ) Файл /usr/lib/python2.7/site-packages/pyspider/libs/base\_handler.py, строка 175, в \_run\_task response.raise\_for\_status () Файл / USR / Lib / python2. 7 / сайт-пакеты / pyspider / LIBS / response.py, строка 184, в raise\_for\_status поднять ошибка HTTP\_ERROR HTTPError: 501 Ошибка сервера Что такое призрак?

666666 2017-08-25 19:05 [ответить](https://cuiqingcai.com/2652.html?replytocom=4369#respond)

Фигура сообщает об этой ошибке, это вопрос разрешения, как мне ее решить?

Nikola Strensu 2017-07-13 14:59 [Ответить](https://cuiqingcai.com/2652.html?replytocom=4239#respond)

да

Пустой план города 2017-07-14 22:42 [ответить](https://cuiqingcai.com/2652.html?replytocom=4244#respond)

Если вы используете UNIX (LINUX), измените путь к вашему домашнему каталогу, по умолчанию должны быть права на запись. / Пользователи / имя пользователя / ... или (linux) / home / username / ...

Пустой план города 2017-07-14 22:45 [ответить](https://cuiqingcai.com/2652.html?replytocom=4243#respond)

Спасибо, я уже это решил. Я некоторое время смотрел блог блоггера и многому помогал.

Nikola Strensu 2017-07-20 10:06 [Ответить](https://cuiqingcai.com/2652.html?replytocom=4258#respond)

Благодарим за предоставленный код, 2017-06-14, python2.7 32bit, прокси-тест кода выполним, python35, у 64-битного, похоже, проблемы

Осенний ветер 梧 2017-06-14 10:29 [ответить](https://cuiqingcai.com/2652.html?replytocom=4168#respond)

Также возможно изменить python35 на 32 бита.

Yun sugar no1 2017-07-28 17:06 [ответить](https://cuiqingcai.com/2652.html?replytocom=4298#respond)

\ Programdata \ anaconda3 \ Lib \ сайт-пакеты \ Торнадо \ concurrent.py, строка 237, в результате raise\_exc\_info (self.\_exc\_info) Файл & Lt; строка & GT;, строка 3, в raise\_exc\_info Исключение: HTTP 599: Операция истекло после 120000 миллисекунд С 1944467 из 5242880 байт получено {Это сетевая проблема? ? ? ? ? ?

Solo157 Sohu 2017-06-02 11:08 [Ответить](https://cuiqingcai.com/2652.html?replytocom=4568#respond)

Я просто столкнулся с этой проблемой, потому что файл, открытый в функции saveBrief (), неверен. Клип, который сканирует вниз, является двоичным, поэтому файл должен быть открыт в двоичном режиме. Я изменил f = open (имя\_файла, 'w +') на f = open (имя\_файла, 'wb') в функции saveBrief ().

Anonymous 2017-05-20 17:48 [ответить](https://cuiqingcai.com/2652.html?replytocom=4096#respond)

Марк, посмотри на него позже, спасибо за обмен

杜蕾 2017-04-11 22:22 [ответить](https://cuiqingcai.com/2652.html?replytocom=3901#respond)

Traceback (самый последний вызов последнего): Файл "D: /Softwareconcerned/Python/Spider\_test/TaobaoMM-master/spider/spider.py", строка 97, в MMspider.on\_start () Файл «D: / Softwareconcerned / Python / Spider\_test / TaobaoMM-мастер / паук / spider.py "строка 26, в on\_start self.crawl (URL обратного вызова, = self.index\_page, validate\_cert = False) Файл" C: softwarePythonlibsite-packagespyspider-0.3.9-py2.7.eggpyspiderlibsbase\_handler. ру "строка 388, в обратном ползать self.\_crawl (URL, \*\* kwargs) Файл" C: softwarePythonlibsite-packagespyspider-0.3.9-py2.7.eggpyspiderlibsbase\_handler.py», строка 332, в \_crawl если cache\_key не в себе .\_follows\_keys: AttributeError: объект «обработчик» не имеет атрибута «\_follows\_keys». Что это за призрак?

[Ядро](http://t.qq.com/hurenjian1120) грецкого[ореха](http://t.qq.com/hurenjian1120) 2017-04-08 20:39[ответить](https://cuiqingcai.com/2652.html?replytocom=3858#respond)

Blogger, почему index\_pag для каждого в response.doc ('a [href ^ = "http"]'). Items (): быть замененным для каждого в response.doc ('. Lady-name'). Items ( ): Я не могу подняться с предыдущей. , Что такое стресс?

[FHE](http://t.qq.com/duanzhaoxia123) 2017-04-06 21:50[Ответить](https://cuiqingcai.com/2652.html?replytocom=3846" \l "respond)

10 000 v часть A кусок высокой четкости внутри страны. Япония и Южная Корея www.288D.pW

[Какао](http://www.kaixin001.com/home/?uid=175175150) 2017-01-12 15:15[ответить](https://cuiqingcai.com/2652.html?replytocom=3159#respond)

Картинку нельзя загрузить, так как она перейдет на страницу входа в Taobao.

[Peng Yuanjie](http://t.qq.com/pyjacm1991) 2017-01-04 17:55[ответить](https://cuiqingcai.com/2652.html?replytocom=3130#respond)

Blogger, я хотел бы задать вопрос, то есть когда я использую urllib или urllib2 для обхода кода веб-страницы, я обнаружил, что часть кода веб-страницы персонализированного имени домена пуста. Это то, что вы сказали динамическим генератором JS? Но когда я смотрю на код HTML, эта часть находится только в div, а не в скрипте, как определить, какой сайт полезен для какой технологии, чтобы скрыть ключевую часть кода?

Xiong Bao 2016-12-28 13:57 [ответить](https://cuiqingcai.com/2652.html?replytocom=3079#respond)

Это действительно генерируется js, но это видно в браузере. Браузер является результатом js-рендеринга. Это должно зайти в сетевой запрос консоли браузера

[Cui Qingcai](http://www.baidu.com/p/%E5%B4%94%E5%BA%86%E6%89%8D) 2017-01-01 15:49[ответить](https://cuiqingcai.com/2652.html?replytocom=3107#respond)

Taobao необходимо войти в систему. ,

[Kang Kangkang занят приятелем](http://weibo.com/5368701784) 2016-12-09 21:40[ответить](https://cuiqingcai.com/2652.html?replytocom=2954#respond)