## 一.安装所需要的库和部件

**安装Requests库**

pip install requests

**安装beautifulsoup4：**

pip install beautifulsoup4 或者python -m pip install beautifulsoup4

测试：打开cmd窗口输入python打开python shell，

输入：from bs4 import BeautifulSoup，没有出错说明安装成功

pip install builtwith

pip install python-whois

## install selenium

## 需要安装ChromeDriver

网址：<https://chromedriver.storage.googleapis.com/index.html>

需要根据Chrome浏览器的版本来安装，例如这里是是92

下载后把chromedriver.exe保存到一个路径，如f:\chromedriver,然后将这个路径添加到环境变量

测试：创建一个test.py内容如下

|  |
| --- |
| from selenium import webdriver  browser=webdriver.Chrome() |

运行这个文件，如果弹出一个空白浏览器窗口说明chromedriver配置没有问题

安装GeckoDriver ：对应selenium 3.141.0 firefox 76版本，geckodriver 0.26,不好用

网址：<https://github.com/mozilla/geckodriver/releases>

前提：需要安装好Firefox浏览器

**去上面的网址找到geckodriver的发行版本，下载下来**

**在f盘创建一个geckodriver目录，把geckodriver.exe拷贝进来，并且将这个目录添加到**

**环境变量**

**安装OperaDriver**

**网址：**[**https://github.com/operasoftware/operachromiumdriver/releases**](https://github.com/operasoftware/operachromiumdriver/releases)

**根据自己浏览器版本来安装**

**安装MicrosoftWebDriver.exe（下载下来的是msedgedriver.exe,需要修改名字，否则不会成功）需要根据浏览器版本来91版本对应91版本**

**网址：**[**https://msedgewebdriverstorage.z22.web.core.windows.net/?prefix=90.0.815.0/**](https://msedgewebdriverstorage.z22.web.core.windows.net/?prefix=90.0.815.0/)

**可以复制到python安装目录的Scripts目录下面或者设置环境变量**

**小技巧1：python获取调用python命令的路径，并且实现路径拼接**

|  |
| --- |
| root=os.getcwd()  filePath=os.path.join(root,"helloPygame")  iconfile=os.path.join(filePath,"flower.png")  **表示加载在调用python命令的目录下面的helloPygame子目录下面的一个叫做**  **Flower.png的图标文件** |

**小技巧2：获取py文件所在的目录并且加载该目录下面的一个图标文件**

|  |
| --- |
| path=sys.path[0]  iconfile=os.path.join(path,"flower.png")  pygame.display.set\_caption(iconfile)  icon = pygame.image.load(iconfile)  **sys.path[0]可以获取py脚本文件所在的目录，然后使用os.path.join方法拼接路径**  **最后加载文件** |

**Selenium最好和一个没有界面的浏览器一起使用，如PhantomJs**

**下载地址：**[**http://phantomjs.org/download.html**](http://phantomjs.org/download.html)

**测试代码**

|  |
| --- |
| from selenium import webdriver  wb = webdriver.PhantomJS()  wb.get('https://www.baidu.com')  print(wb.current\_url)  **注意，selenium对phantomjs的支持已经过时** |

**安装异步请求库aiohttp：**

**pip3 install aiohttp**

**安装aiohttp是同时会安装字符编码检测库cchardet**

**安装加速dns解析库aiodns**

**pip3 install aiodns**

**解析库安装：**

**lxml：用XPath的方式解析html和XML，效率高。**

**安装：pip3 install lxml**

**安装pyquery**

**pip3 install pyquery**

**安装tesserocr**

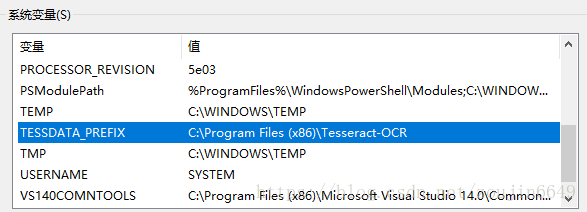
**需要先安装tesseract，需要下载**

**网址：**[**https://digi.bib.uni-mannheim.de/tesseract/**](https://digi.bib.uni-mannheim.de/tesseract/)

**选择： tesseract-ocr-setup-3.05.02-20180621.exe**

**下载后双击安装**

**完成后需要配置环境变量**



https://img-blog.csdn.net/201809140928301?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pvdWppbjY2NDk=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

**复制tessdata整个文件夹至Python安装目录下面，注意不是Scripts目录**

**可以安装tesserocr**

**注意有坑：如果直接使用pip3 install tesserocr会失败**

**需要使用whl文件来安装**

下载地址：<https://github.com/simonflueckiger/tesserocr-windows_build/releases/tag/tesserocr-v2.2.2>

**Python的版本不能超过3.7**

**先从上面的网址下载** [tesserocr-2.2.2-cp36-cp36m-win\_amd64.whl](https://github.com/simonflueckiger/tesserocr-windows_build/releases/download/tesserocr-v2.2.2/tesserocr-2.2.2-cp36-cp36m-win_amd64.whl)

然后使用：pip3 install tesserocr-2.2.2-cp36-cp36m-win\_amd64.whl安装

安装pillow

**pip3 install pillow**

**安装pymysql**

**pip3 install pymysql**

**安装pymongo**

**pip3 install pymongo**

**技巧：**

**Python批量安装第三方库**

**需要一个all.txt内容：**

|  |
| --- |
| **builtwith ==1.3.4**  **pymongo==3.12.0**  **PyMySQL==1.0.2**  **python-docx==0.8.11**  **python-whois==0.7.3**  **requests==2.26.0**  **selenium==3.141.0**  **beautifulsoup4==4.9.3**  **builtwith==1.3.4**  **certifi==2021.5.30**  **cffi==1.14.6**  **chardet==4.0.0**  **charset-normalizer==2.0.4**  **soupsieve==2.2.1**  **cssselect==1.1.0**  **pyquery==1.4.3** |

**然后打开cmd窗口，切换到python安装目录下面**

**输入目录pip install –r all.txt**

### 安装redis

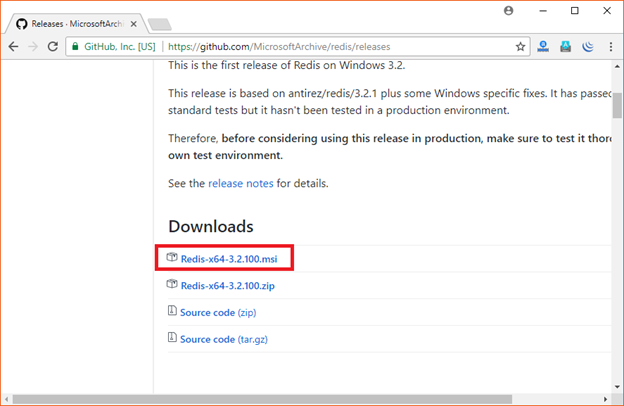
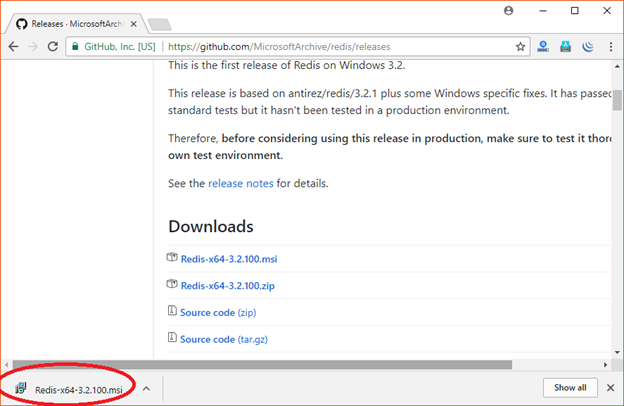
# 在 windows 上安装 Redis

Redis 官方不建议在 windows 下使用 Redis，所以官网没有 windows 版本可以下载。还好微软团队维护了开源的 windows 版本，虽然只有 3.2 版本，对于普通测试使用足够了。

## 安装包方式安装 Redis 服务

下载地址：<https://github.com/MicrosoftArchive/redis/releases>

点击下载：

您可以看到 Redis 现在已下载。

或者您也可以使用下面链接下载。

<https://github.com/rgl/redis/downloads>

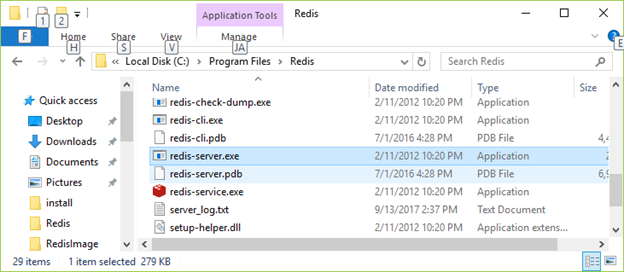
下载完成之后双击按着引导流程安装。



一路next    完成安装

## 启动 Redis

Redis 现在可以使用了。打开 Redis 程序目录：



文件介绍：

redis-server.exe：服务端程序，提供 redis 服务

redis-cli.exe: 客户端程序，通过它连接 redis 服务并进行操作

redis-check-dump.exe：RDB 文件修复工具

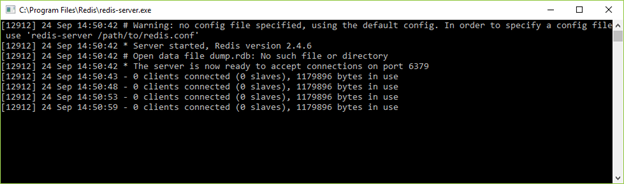
redis-check-aof.exe：AOF 文件修复工具

redis-benchmark.exe：性能测试工具，用以模拟同时由 N 个客户端发送 M 个 SETs/GETs 查询 (类似于 Apache 的 ab 工具)

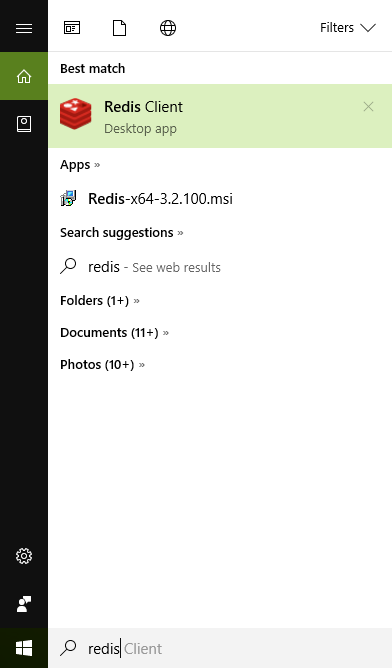
redis.windows.conf： 配置文件，将 redis 作为普通软件使用的配置，命令行关闭则 redis 关闭

redis.windows-service.conf：配置文件，将 redis 作为系统服务的配置

单击 redis-server.exe，启动 Redis 服务。

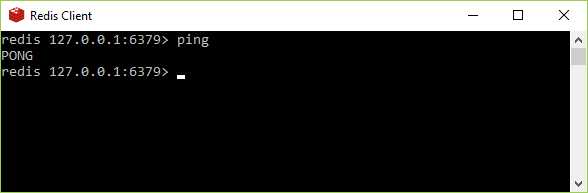


现在启动 Redis 客户端。

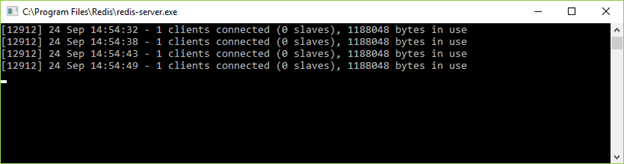
 

检查 Redis 是否已连接。

使用 PING 命令。

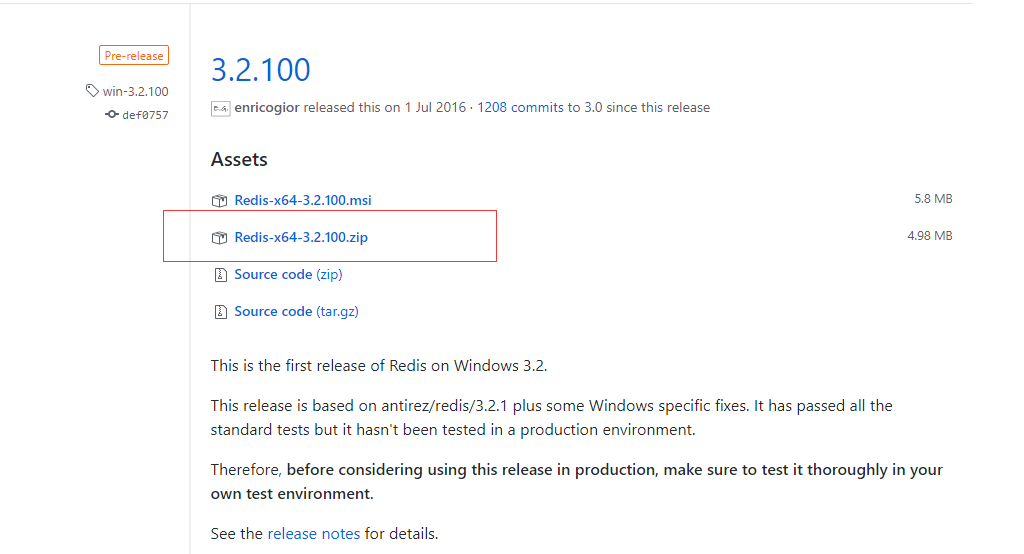


Redis 服务窗口也输出 1 个客户端已连接。



## 直接解压的方式安装 redis

首先下载 redis 安装包：<https://github.com/MSOpenTech/redis/releases>



解压安装包到相应文件夹，任何盘符都行，例如 E:\tools\redis-3.2.100。

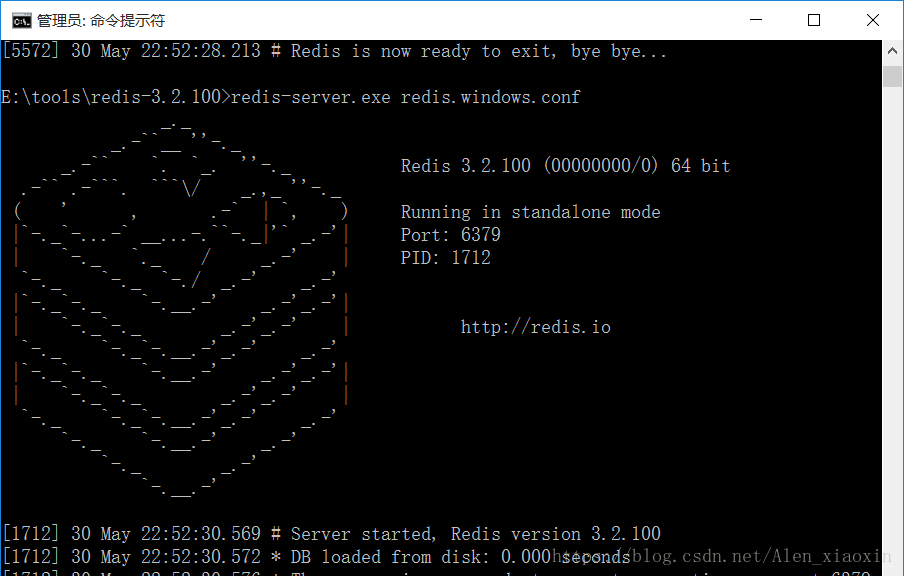
### 使用命令行启动 Redis 服务

运行 cmd，cd 进入对应目录 E:\tools\redis-3.2.100，执行：

redis-server.exe redis.windows.conf

\*注：可以把 redis 的路径加到系统的环境变量里，这样就省得再输路径了，后面的那个 redis.windows.conf 可以省略，如果省略，会启用默认的参数。

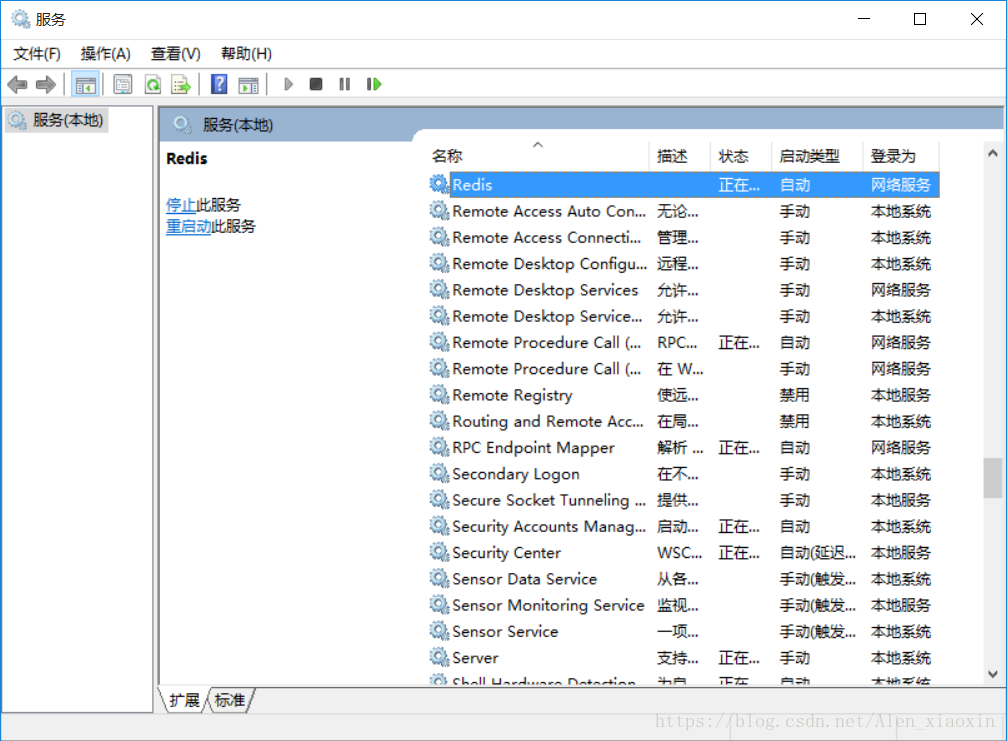
输入之后会显示如下：



### 安装 redis 到 windows 服务

redis-server --service-install redis.windows.conf

查看 windows 服务是否加入：



这时候先关闭打开的第一个 cmd 窗口，然后执行以下命令启动再次 redis：

redis-server --service-start

停止 redis 服务：

redis-server --service-stop

最后，测试一下 redis 是否能够正常使用：

切换到 redis 目录下：E:\tools\redis-3.2.100  下：

redis-cli.exe -h 127.0.0.1 -p 6379

### 安装redis-py

**pip3 install redis**

**验证：**

|  |
| --- |
| **C:\Users\Administrator.USER-20210609FU>python**  **Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32**  **Type "help", "copyright", "credits" or "license" for more information.**  **>>> import redis**  **>>> redis.VERSION**  **(3, 5, 3)** |

出现上面结果说明安装成功

安装RedisDump，前提是先安装Ruby

1.安装Ruby：

网址：<https://rubyinstaller.org/downloads/>

2.利用gem安装redisdump

gem install redis-dump

**安装flask**

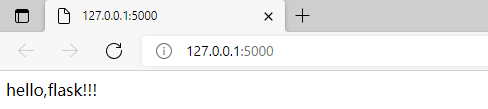
**pip3 install flask**

测试

编写一个flasktest.py文件，然后运行

|  |
| --- |
| from flask import Flask  app =Flask(\_\_name\_\_)  @app.route('/')  def hello():      return "hello,flask!!!"  if \_\_name\_\_=="\_\_main\_\_" :      app.run() |

可以打开一个浏览器窗口



说明成功

## 安装tornado

**pip3 install tornado**

**测试**

|  |
| --- |
| import tornado.ioloop  import tornado.web  class MainHandler(tornado.web.RequestHandler):      def get(self):          self.write("hello tornado app")  def make\_app():      return tornado.web.Application([          (r"/",MainHandler),      ])  if \_\_name\_\_=='\_\_main\_\_':     app =make\_app()     app.listen(8888)     tornado.ioloop.IOLoop.current().start() |

**我们会利用 Tomado+Redis 来搭建一个 ADSL 拨号代理池。**

# 爬取App相关的库

### 安装mitmproxy

#### 它有两个组件：mitmdump和mitmweb

注意：python3.6版本需要下面的版本，高于这个版本会报错

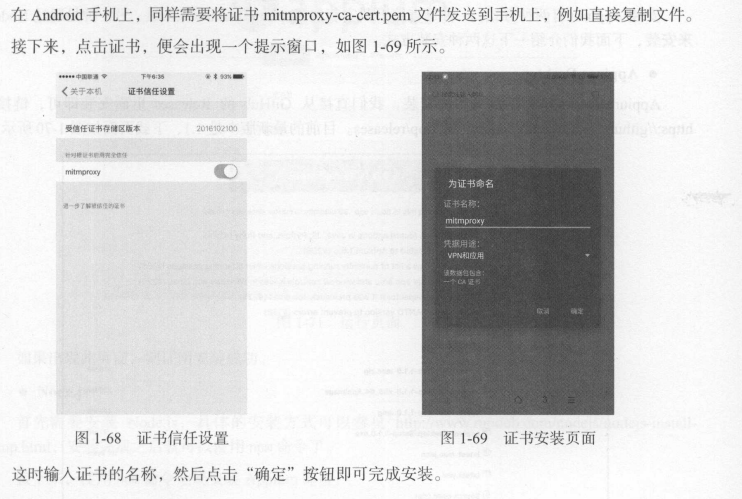
pip install mitmproxy==5.0.1

配置证书：

证书目录：**C:\Users\Administrator.USER-20210609FU\.mitmproxy**

**双击mitmproxy-ca.p12文件，**

**就会出现导人证书的引导页， 直接点击“下一步”按钮即可，出现选择证书默认选中了，直接next接下来需要选择证书的存储区域 这里点击第二个选项“将所有的证书都放入下列存储”，然后点击“浏览”按钮，选择证书存储位置为 “受信任的根证书颁发机构”，接着点击“确定”按钮，然后点击“下一步”按钮。会出现密码设置提示，不需要密码直接next，最后，如果有安全警告弹出，如图 1-63 所示，直接点击“是”按钮即可** 。



Appium 的安装  
Appium 是移动端的自动化测试工具，类似于前面所说的 Selenium ，利用它可以驱动 Android 、 iOS等设备完成向动化测试，比如模拟点击 、 滑动 、 输入等操作

1.网址：

GitHub: <https://github.com/appium/appium>

官方网站 ： <http://appium.io>

官方文档 ： <http://appium.io/introduction.html>

下载链接： <https://github.com/appium/appium-desktop/releases>

Python Client: <https://github.com/appium/python-client>

2.安装

a.下载安装包来安装

b.使用nodejs的npm来安装，前提安装了nodejs

格式：npm install -g appium

3.配置Android开发环境

安装 Android Studio 下载地址为 <https://developer.android.com/studio/index.html?hl=zh-cn>

双击安装，安装完后下载会下载Android SDK，配置完成后的信息

|  |
| --- |
| Preparing "Install Android SDK Platform-Tools (revision: 31.0.3)".  Downloading https://dl.google.com/android/repository/platform-tools\_r31.0.3-windows.zip  "Install Android SDK Platform-Tools (revision: 31.0.3)" ready.  Installing Android SDK Platform-Tools in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\platform-tools  "Install Android SDK Platform-Tools (revision: 31.0.3)" complete.  "Install Android SDK Platform-Tools (revision: 31.0.3)" finished.  Preparing "Install Android Emulator (revision: 30.8.4)".  Downloading https://dl.google.com/android/repository/emulator-windows\_x64-7600983.zip  "Install Android Emulator (revision: 30.8.4)" ready.  Installing Android Emulator in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\emulator  "Install Android Emulator (revision: 30.8.4)" complete.  "Install Android Emulator (revision: 30.8.4)" finished.  Preparing "Install SDK Patch Applier v4 (revision: 1)".  Downloading https://dl.google.com/android/repository/3534162-studio.sdk-patcher.zip  "Install SDK Patch Applier v4 (revision: 1)" ready.  Installing SDK Patch Applier v4 in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\patcher\v4  "Install SDK Patch Applier v4 (revision: 1)" complete.  "Install SDK Patch Applier v4 (revision: 1)" finished.  Preparing "Install Android SDK Tools (revision: 26.1.1)".  Downloading https://dl.google.com/android/repository/sdk-tools-windows-4333796.zip  "Install Android SDK Tools (revision: 26.1.1)" ready.  Installing Android SDK Tools in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\tools  "Install Android SDK Tools (revision: 26.1.1)" complete.  "Install Android SDK Tools (revision: 26.1.1)" finished.  Preparing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 7.6.5)".  Downloading https://dl.google.com/android/repository/extras/intel/haxm-windows\_v7\_6\_5.zip  "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 7.6.5)" ready.  Installing Intel x86 Emulator Accelerator (HAXM installer) in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\extras\intel\Hardware\_Accelerated\_Execution\_Manager  "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 7.6.5)" complete.  "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 7.6.5)" finished.  Preparing "Install Android SDK Build-Tools 31 (revision: 31.0.0)".  Downloading https://dl.google.com/android/repository/09489e417c0a266f2862ddd82b4ac29a1b7af55e.build-tools\_r31-windows.zip  "Install Android SDK Build-Tools 31 (revision: 31.0.0)" ready.  Installing Android SDK Build-Tools 31 in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\build-tools\31.0.0  "Install Android SDK Build-Tools 31 (revision: 31.0.0)" complete.  "Install Android SDK Build-Tools 31 (revision: 31.0.0)" finished.  Preparing "Install Android SDK Platform 31 (revision: 1)".  Downloading https://dl.google.com/android/repository/platform-31\_r01.zip  "Install Android SDK Platform 31 (revision: 1)" ready.  Installing Android SDK Platform 31 in C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\platforms\android-31  "Install Android SDK Platform 31 (revision: 1)" complete.  "Install Android SDK Platform 31 (revision: 1)" finished.  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\build-tools\31.0.0\package.xml  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\emulator\package.xml  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\extras\intel\Hardware\_Accelerated\_Execution\_Manager\package.xml  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\patcher\v4\package.xml  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\platform-tools\package.xml  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\platforms\android-31\package.xml  Parsing C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk\tools\package.xml  Android SDK is up to date.  Running Intel® HAXM installer  Intel HAXM installed successfully! |

设置ANDROID\_HOME 为 Android SDK 所在路

如：C:\Users\Administrator.USER-20210609FU\AppData\Local\Android\Sdk

然后再添加 SDK 文件夹下的tools和**platform-tools**文件夹到 PATH 中 。

|  |
| --- |
|  |

# 爬虫框架的安装

#### 1.pyspider，需要phantomJS支持，只支持32位的python6

网址：

|  |
| --- |
| 官方文档 : <http://docs.pyspider.org/>  PyPI: <https://pypi.python.org/pypi/pyspider>  GitHub: <https://github.corn/binux/pyspider>  官方教程: <http://docs.pyspider.org/en/latest/tutorial>  在线实例: <http://demo.pyspider.org> |

利用pip安装，这里有坑，

必须先安装pycurl

网址：<https://www.lfd.uci.edu/~gohlke/pythonlibs/>

按Ctrl+f弹出搜索框，输入pycurl，出现下载列表

下载python3.6对应的版本pycurl-7.43.0.4-cp36-cp36m-win32.whl

然后切换到whl文件的目录，打开一个cmd窗口：使用pip安装

pip3 install pycurl-7.43.0.4-cp36-cp36m-win32.whl

安装Werkzeug 0.16.0版

pip install Werkzeug==0.16.0

安装WsgiDAV 2.4.1版

pip install WsgiDAV ==2.4.1

现在可以安装pyspider

还有坑，不能是0.3.10

pip3 install pyspider==0.3.9

或者去这个地址下载源码：

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

选择v0.3.9版本，使用pip install XX.whl的方法安装

验证：在一个cmd窗口中输入：pyspider all

成功界面

|  |
| --- |
|  |

如果还有问题：参考“pyspider报错解决.docx”

安装scrapy

1 . 相关链接

官方网站: <https://scrapy.org>

官方文档: <https://docs.scrapy.org>

PyPI: <https://pypi.python.org/pypi/Scrapy>

GitHub: <https://github.com/scrapy/scrapy>

中文文档 ： <https://scrapy-chs.readthedocs.io>

2. Windows 下的安装参考如下方式来一步步安装 Scrapy 。  
·安装 lxml  
lxml 的安装过程请参见 1.3 . 1 节，在此不再赘述， 此库非常重要，请一定要安装成功 。  
·安装 pyOpenSSL  
在官方网站下载 wheel 文件（详见 https://pypi .python.org/pypi/pyOpenSSL#downloads ）

下载后利用 pip 安装即可：

pip3 install pyOpenSSL-17.2.0-py2 .py3-none-any.whl

安装 twisted  
到 http://www.lfd.uci.edu/~gohlke/pythonlibs/#twisted

下载 wheel 文件，利用 pip 安装 即可 。  
比如，对于 Python 3.6 版本 、 Windows32位系统，则当前最新版本为 Twisted-20.3.0-cp36-cp36m-win32.whl， 直接下载即可，

然后切换到whl文件路径运行命令：

pip install Twisted-20.3.0-cp36-cp36m-win32.whl

安装 PyWin32

从官方网站下载对应版本的安装包即可，链接为： <https://sourceforge.net/projects/pywin32/files/pywin32/Build%2022l/>

下载32位版本：pywin32-221.win32-py3.6.exe

双击安装，完成。

#### 安装 Scrapy

安装好了以上的依赖库后，安装 Scrapy 就非常简单了，这里依然使用 pip ，命令如下：

pip3 install Scrapy

等待命令结束，如果没有报错，就证明 Scrapy 已经安装好了 。

成功界面

|  |
| --- |
|  |

Scrapy-Splash 的安装

Scrapy-Splash 是一个 Scrapy 中支持 JavaScript 渲染的工具。

这个库原来Splash服务，这个服务使用docker安装

1 相关链接

GitHub: <https://github.com/scrapy-plugins/scrapy-splash>

PyPI: <https://pypi.python.org/pypi/scrapy-splash>

使用说明: <https://github.com/scrapy-plugins/scrapy-splash#configuration>

Splash 官方文档： <http://splash.readthedocs.io>

2 . 安装 Splash

2a.安装docker

网址：<http://mirrors.aliyun.com/docker-toolbox/windows/docker-toolbox/>

最好选择新的版本，如DockerToolbox-18.03.0-ce.exe

下载后双击安装程出现如下界面：

|  |
| --- |
|  |

点击next，注意注意这里一个大坑！不要勾选docker自带的Vbox！！！

|  |
| --- |
|  |

路next，完成

2b，安装Vbox，window版本5.2.44以上。

然后双击Docker Quickstart Terminal启动docker

成功后出现如下界面

|  |
| --- |
|  |

注意docker可以在cmd窗口中使用

|  |
| --- |
|  |

|  |
| --- |
|  |

2c 安装Splash

在docker命令行界面中输入：docker run -p 8050:8050 scrapinghub/splash

安装成功界面

|  |
| --- |
|  |

注意：需要重新启动docker修改一个端口如6050 重新启动splash

$ docker run -p 8050:8050 scrapinghub/splash

重新启动后的界面

|  |
| --- |
|  |

需要复制IP：这里是192.168.99.100

打开浏览器，在地址栏中输入：http:// 192.168.99.100:8050回车就会出现

Splash的UI界面

|  |
| --- |
|  |

### 3. Scrapy-Splash 的安装

成功安装 Sp lash 之后，接下来再来安装其 Python 库，命令如下：

pip3 install scrapy-splash

测试：打开python的shell窗口：

|  |
| --- |
| 说明安装成功 |

命令运行完毕后，就会成功安装好此库

### Scrapy-Redis 的安装

Scrapy-Redis 是 Scrapy 的分布式扩展模块，有了它，我们就可以方便地实现 Scrapy 分布式爬虫的搭建。

1 . 相关链接

GitHub: <https://github.com/rmax/scrapy-redis>

PyPI: <https://pypi.python.org/pypi/scrapy-redis>

官方文档： <http://scrapy-redis.readthedocs.io>

2 pip安装

这里推荐使用 pip 安装，命令如下：

pip3 install scrapy-redis

测试：

|  |
| --- |
| 说明安装成功 |

一些部署的库暂时不装

|  |
| --- |
|  |