IP Rotation for Python in Windows

The following description will enable IP rotation through the TOR network, which will provide you with a large number of different IP addresses and will hide your current IP address by using a multitude of proxies. More information about TOR can be found here: <https://www.torproject.org/>

1. Install the TOR expert bundle for windows.

<https://www.torproject.org/download/download.html.en>

1. Install Privoxy

<https://sourceforge.net/projects/ijbswa/>

1. Install stem for your python installation.

Anaconda command prompt: pip install stem

1. Determine your HashedControlPassword

Open command prompt and navigate to the tor directory and execute the following command with your desired password:

cd "Tor installation Directory"

tor.exe --hash-password "Your password"

1. In the tor directory (that contains the folder Data and Tor) add a file with the name “torrc” with no extension.

Open this file with a text editor and add the following lines of text, where the HashedControlPassword is the password you received in the previous step, without the “”.

ControlPort 9151

HashedControlPassword “HashedControlPassword”

1. Start Privoxy

Navigate to, Options, Edit Main Configuration.

A file will open, where you have to add the following text:

forward-socks5 / 127.0.0.1:9050 .

Then save the file.

1. Restart Privoxy and start tor.exe.

The tor console will mention:



1. Use the following script in python to create a Connection Manager.
2. Call cm.new\_identity() in python to change the IP address.
3. Print cm.new\_ip to check your current IP address.

Possible IP addresses are:

|  |  |
| --- | --- |
| 208.67.1.83 | 94.242.246.24 |
| 162.247.72.216 | 192.160.102.164 |
| 77.247.181.163 | 163.172.212.115 |
| 94.242.246.23 | 192.42.116.16 |
| 51.15.141.220 | 78.129.137.28 |
| 93.115.95.206 | 89.32.127.178 |
| 185.170.42.4 | 94.142.242.84 |
| 178.17.171.40 | 178.17.170.13 |
| 162.247.72.200 | 37.187.129.166 |
| 128.52.128.105 | 37.187.7.74 |
| 85.248.227.164 | 163.172.170.161 |
| 192.160.102.168 | 66.70.217.179 |
| 145.239.82.79 | 193.15.16.4 |
| 104.218.63.75 | 46.17.97.112 |
| 62.210.115.87 |  |

**Python script**

import urllib.request

from stem import Signal

from stem.control import Controller

class ConnectionManager:

def \_\_init\_\_(self):

self.new\_ip = "0.0.0.0"

self.old\_ip = "0.0.0.0"

self.new\_identity()

@classmethod

def \_get\_connection(self):

"""

TOR new connection

"""

with Controller.from\_port(port=9051) as controller:

controller.authenticate(password="1234")

controller.signal(Signal.NEWNYM)

controller.close()

@classmethod

def \_set\_url\_proxy(self):

"""

Request to URL through local proxy

"""

proxy\_support = urllib.request.ProxyHandler({"http": "127.0.0.1:8118"})

opener = urllib.request.build\_opener(proxy\_support)

urllib.request.install\_opener(opener)

@classmethod

def request(self, url):

"""

TOR communication through local proxy

:param url: web page to parser

:return: request

"""

try:

self.\_set\_url\_proxy()

request = urllib.request.Request(url, None, {

'User-Agent': "Mozilla/5.0 (X11; Linux x86\_64) "

"AppleWebKit/535.11 (KHTML, like Gecko) "

"Ubuntu/10.10 Chromium/17.0.963.65 "

"Chrome/17.0.963.65 Safari/535.11"})

request = urllib.request.urlopen(request)

return request

except urllib.request.HTTPError:

return e.message

def new\_identity(self):

"""

new connection with new IP

"""

# First Connection

if self.new\_ip == "0.0.0.0":

self.\_get\_connection()

self.new\_ip = self.request("http://icanhazip.com/").read()

else:

self.old\_ip = self.new\_ip

self.\_get\_connection()

self.new\_ip = self.request("http://icanhazip.com/").read()

seg = 0

# If we get the same ip, we'll wait 5 seconds to request a new IP

while self.old\_ip == self.new\_ip:

time.sleep(5)

seg += 5

print ("Waiting to obtain new IP: %s Seconds" % seg , end="\r")

self.new\_ip = self.request("http://icanhazip.com/").read()

cm = ConnectionManager()