前言：前段时间由于需求，用Fiddler各种抓包，各种分析，主要是测试请求时添加cookie、添加请求头、添加url， 虽然前面有一键转换cookie/headers的工具（可以看我前面博客），但还是有些烦琐。 想着能不能将Fiddler抓到的请求一键转换成Python代码，但是在网上找了一下，无果（有知道的同学可以评论留言，十分感谢）， 于是决定自己写一个能将Fiddler一键转换成Python代码的Python代码。

话不多说，上代码（如果觉得有更好的优化方案请联系我修改）：

#!/usr/bin/python

# -\*- coding: UTF-8 -\*-

str\_filename = "2.txt"

# save\_name = "re\_test.py"

save\_name = str\_filename.replace("txt", "py")

class FidToPy():

def \_\_init\_\_(self, str\_name, sa\_name):

self.str\_filename = str\_name

self.save\_name = sa\_name

self.text = ""

self.url\_list = []

self.headers = {}

self.cookies = {}

self.data = {}

def get\_url(self):

infos = self.text.split("\n")[0]

self.url\_list = [infos.split(" ")[0], infos.split(" ")[1]]

def get\_headers(self):

infos = self.text.split("\n")[1:]

info = ""

for i in infos:

if "Cookie: " in i:

break

info += i + "\n"

headers = info.split("\n")

while "" in headers:

headers.remove("")

for i in headers:

if ": " not in i:

break

self.headers[i.split(": ")[0]] = i.split(": ")[1]

def get\_cookies(self):

infos = self.text.split("\n")[1:]

cookies\_flag = 0

for i in infos:

if "Cookie: " in i:

self.cookies = i.replace("Cookie: ", "")

print(self.cookies )

cookies\_flag = 1

break

if cookies\_flag==1:

self.cookies = {i.split("=")[0]: i.split("=")[1] for i in self.cookies.split("; ")}

def get\_data(self):

try:

infos = self.text.split("\n")

for i in range(2, len(infos)):

if infos[i - 1] == "" and "HTTP" in infos[i + 1]:

self.data = infos[i]

break

self.data = {i.split("=")[0]: i.split("=")[1] for i in self.data.split("&")}

except:

pass

def get\_req(self):

info\_beg = "#!/usr/bin/python\n# -\*- coding: UTF-8 -\*-\nimport requests\n\n"

info\_url = "url = \'{}\'\n".format(self.url\_list[1])

info\_headers = "headers = {}\n".format(self.headers)

info\_cookies = "cookies = {}\n".format(self.cookies)

info\_data = "data = {}\n\n".format(self.data)

if "GET" in self.url\_list[0]:

info\_req = "html = requests.get(url, headers=headers, verify=False, cookies=cookies)\n"

else:

info\_req = "html = requests.post(url, headers=headers, verify=False, cookies=cookies, data=data)\n"

info\_end = "print(len(html.text))\nprint(html.text)\n"

text = info\_beg + info\_url + info\_headers + info\_cookies + info\_data + info\_req + info\_end

with open(save\_name, "w+", encoding="utf8") as p:

p.write(text)

print("转化成功！！")

print(save\_name, "文件保存!")

def read\_infos(self):

with open(self.str\_filename, "r+", encoding="utf-8") as p:

old\_line = ""

for line in p:

if old\_line == b"\n" and line.encode() == b"\n":

break

old\_line = line.encode()

self.text += old\_line.decode()

# print("self.text:", self.text)

def start(self):

self.read\_infos()

self.get\_url()

self.get\_headers()

self.get\_cookies()

self.get\_data()

print("self.url\_list:", self.url\_list)

print("self.headers:", self.headers)

print("self.cookies:", self.cookies)

print("self.data:", self.data)

self.get\_req()

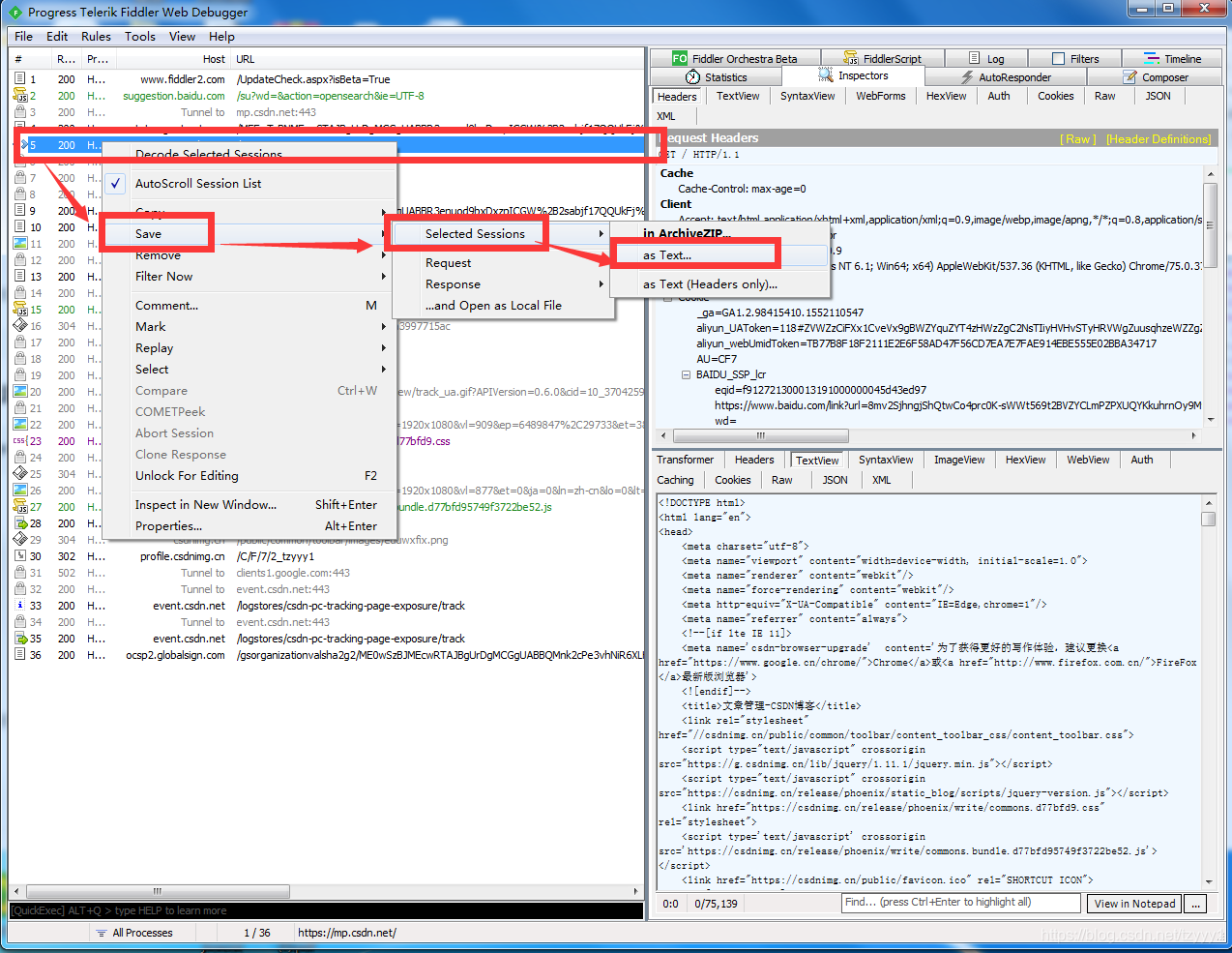
if \_\_name\_\_ == '\_\_main\_\_':

f = FidToPy(str\_filename, save\_name)

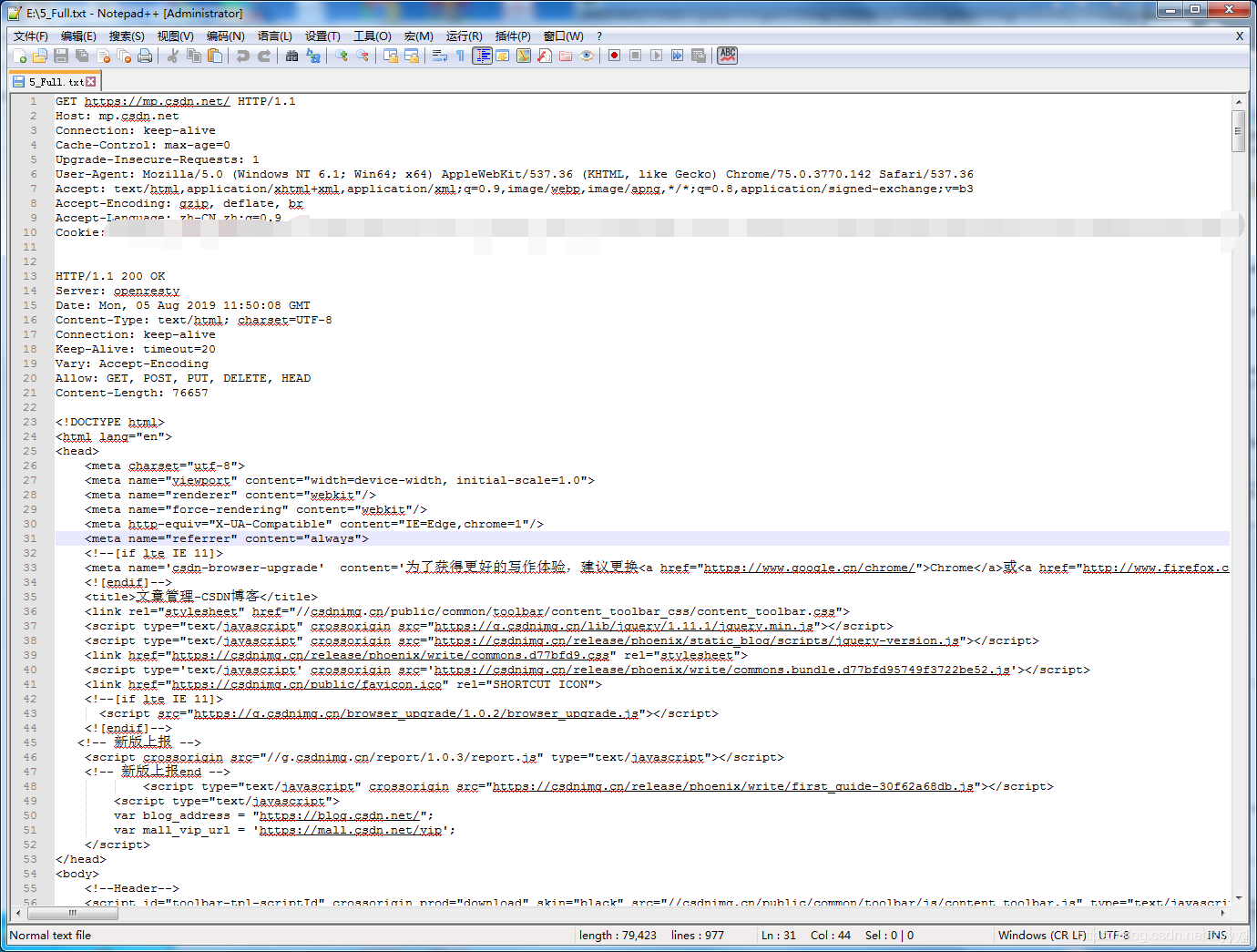
f.start()

使用教程：

首先从Fiddler选中你要转换的请求 -> Save -> Selected Session -> as Text（也可以选择as Text（Header only）...），如图所示：



这样，我们就得到了包含了这个请求的txt文本：

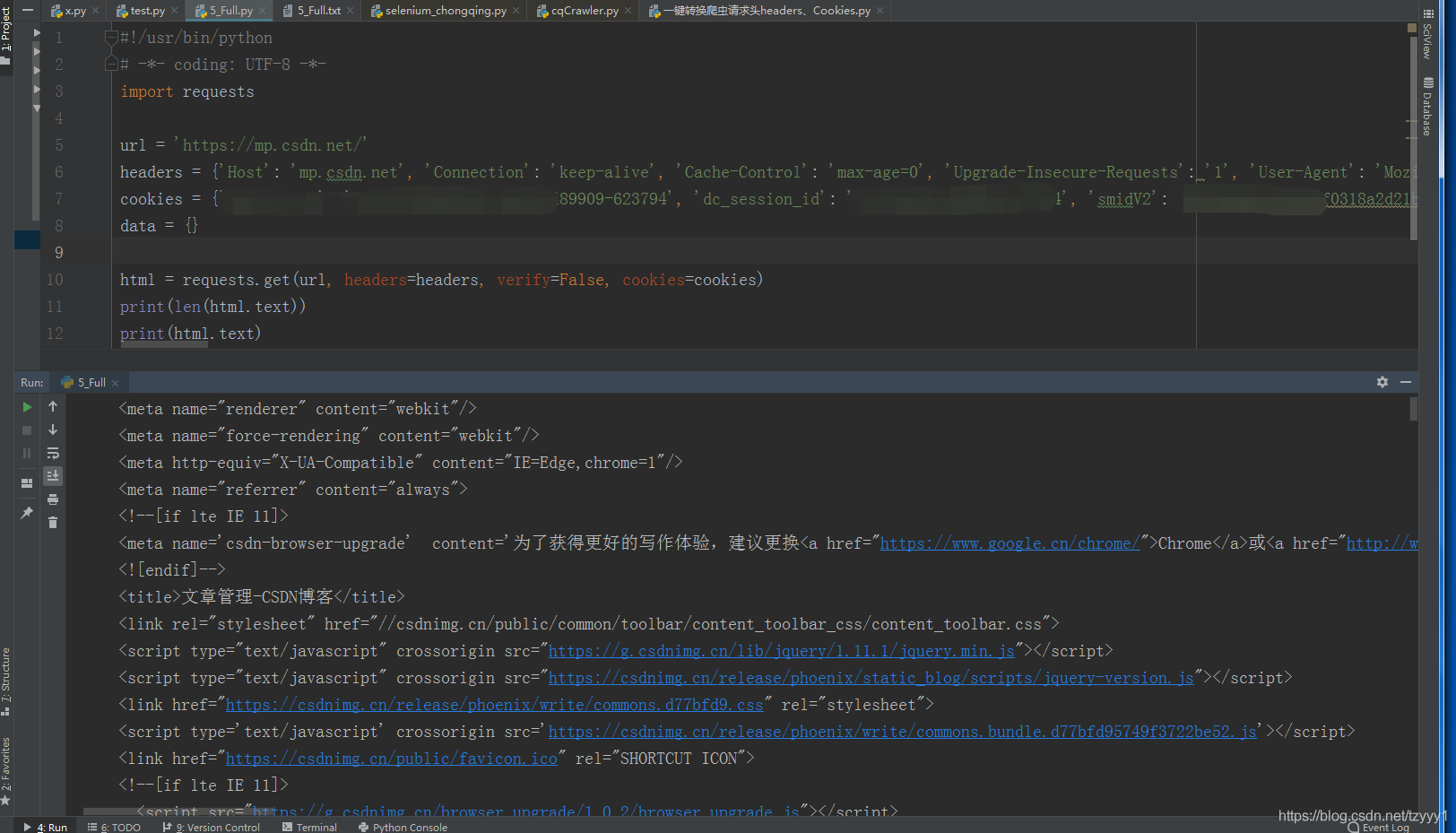


把这个文件移动到和前面Python代码同一个文件夹，然后修改Python中第四行的，改成你请求信息txt文件的名字，如：

str\_filename = "15\_Full.txt"

右键运行，就可以得到一个同名的.py文件，这个就是我们要的Python请求代码了，so simply！！

效果图:



注：如果是App抓包，post请求中data可能需要加一个解码步骤

原创不易，转载请标明出处。