

파이썬 네트워크 프로그래밍 - httplib, urllib2

Xero

박준혁 (한국디지털미디어고등학교 2 학년)

2012-03-22

wnsgurzxc@nate.com

목차

- 0. 잡담
- 1. httplib
 - 1-1 httplib 란?
 - 1-2 httplib 를 이용한 GET 형식
 - 1-2.1 헤더가 없을 때
 - 1-2.2 헤더가 있을 때
 - 1-3 httplib 를 이용한 POST 형식

- 1-3.1 헤더가 없을 때
 - 1-3.2 헤더가 있을 때
- 2. urllib2
 - 2-1 urllib2 란?
 - 2-2 urllib2 를 이용한 GET 형식
 - 2-2.1 헤더가 없을 때
 - 2-2.2 헤더가 있을 때
 - 2-3 urllib2 를 이용한 POST 형식
 - 2-3.1 헤더가 없을 때
 - 2-3.2 헤더가 있을 때
- 3. httpplib, urllib2 비교
- 4. 후기

0. 잡담

파이썬을 배운지 근 1년이 다 되어 간다. 고등학생이 되면서 컴퓨터 공부를 시작하고 보안 공부를 하다가 몇 달 후 주 언어를 파이썬으로 잡고 파이썬을 공부하기 시작했다. PythonChallenge을 풀며 파이썬을 공부하다가 네트워크 프로그래밍에서 막혀 파이썬 공부를 잠시 멈췄었다. 모든 정보가 그렇겠지만 딱 파이썬 네트워크 프로그래밍에 대해 정리된 자료가 없고 막막했기에 잠시 공부를 멈췄었다. 아무래도 초심자에게 네트워크 프로그래밍을 독학하기에는 진입장벽이 높았나 보다. 그러다 이번에 시간이 나서 파이썬 네트워크 프로그래밍을 다시 공부하고 문서를 쓰기로 했다. 이 문서는 내가 파이썬 네트워크 프로그래밍을 하면서 알게 된 httpplib, urllib2를 다룬 문서이다. 둘 중 한 가지로 프로그래밍을 해도 되지만 파이썬 네트워크 프로그래밍을 새롭게 공부하게 되는 다른 누군가를 위해 이 문서를 쓴다. 이 문서를 읽고 httpplib와 urllib2를 자유자재로 사용하며 네트워크 프로그래밍을 빠르게 익히면 좋겠다.

1. httplib

1-1 httplib 란?

HTTP 와 HTTPS protocol 을 이용하는 라이브러리로써 GET, POST 형식으로 Connection 이 가능하고, parameter 와 header 조작이 가능하다.

자세한 설명은 다음을 참조하면 된다.

<http://docs.python.org/library/httplib.html>

1-2 httplib 를 이용한 GET 형식

1-2.1 헤더가 없을 때

<http://webgame.wowhacker.com/level1.php> 사이트를 이용하겠다.

위의 사이트에 들어가면 다음과 같이 개발자의 실수로 소스코드를 볼 수 있다고 한다.

?key=

Can not open a key file

Hint : 개발자의 실수로 소스코드를 볼수있다. php파일의 소스코드를 웹에서 보기 위한 방법

<http://webgame.wowhacker.com/level1.phps> 에 들어가면 다음과 같이 소스가 보인다.

```

<?
echo "?key=";
echo "$_GET[key]";
echo "<br>";
    if($_GET[key]=="wowhacker_hardware"){
        include "../key/key1";
    } else echo "Can not open a key file";
?>
<br>
<br>
Hint : 개발자의 실수로 소스코드를 볼수있다. php파일의 소스코드를 웹에서 보기 위한 방법<br>
<br>

```

GET 형식으로 key 변수의 값을 받아서 그 값이 wowhacker_hardware 이면 정답을 출력한다.

위의 경우 urllib 을 이용해 파이썬으로 코딩하면 다음과 같다.

```

>>> import urllib, urllib
>>> con=urllib.HTTPConnection('webgame.wowhacker.com')
>>> param=urllib.urlencode({'key':'wowhacker_hardware'})
>>> con.request('GET','/level1.php?'+param)
>>> response=con.getresponse()
>>> res=response.read()
>>> print res
?key=wowhacker_hardware<br>WEBGAME KEY : i love tiger mac
<br>
<br>
Hint : 개발자의 실수로 소스코드를 볼수있다. php파일의 소스코드를 웹에서 보기 위한 방법<br>
<br>

```

소스의 순서를 해석하면 다음과 같다.

1. con 에 HTTPConnection 함수를 이용해 webgame.wowhacker.com 과 연결시킨다.
2. GET 형식으로 key 변수에 값을 받았으니 param 변수에 key 는 wowhacker_hardware 라고 사전 형식으로 urllib 의 urlencode 함수를 이용해 url 인코딩하여 넣어준다.
3. request 를 이용해 GET 형식으로 /level1.php?+param 사이트에 요청을 보낸다.
(http://webgame.wowhacker.com/level1.php?key=wowhacker_hardware 가 됨)
4. response 변수에 위에서 요청보냈던 결과를 받아온다.
5. res 변수에 response 를 읽는다.
6. res 변수를 출력한다. (소스 출력)

위는 헤더가 없는 GET 형식이었으므로 헤더를 추가하지도, POST 형식처럼 값을 넘겨주지도 않고 단지 GET 형식으로 값을 넘기므로 request 함수의 모양이 무척 간단하다.

다음이 request 함수 부분의 원 모양이다.

con.request('GET','주소?'+값)

1-2.2 헤더가 있을 때

헤더가 있는 GET 형식은 <http://suffad.netne.net/challenge/26/> 사이트를 이용하기로 했다.

위 사이트에 접속하면 다음과 같이 폼 하나와 1 ~ 200 이라는 글자가 있다.

pw :

아무 숫자나 입력하면 다음과 같이 wrong! 이 출력된다.

pw :

위를 파이썬으로 코딩하면 다음과 같이 된다.

```
>>> import urllib, http, re
>>> con=httplib.HTTPConnection('suffad.netne.net')
>>> cookie='userid=1c4c62efd0e1db054a48312709bc6476; PHPSESSID=5646b8435cf4033b0c194fabe48bbca3'
>>> header={'Cookie':cookie}
>>> for i in range(1,200):
    con.request('GET','/challenge/26/?pw='+str(i),headers=header)
    response=con.getresponse()
    res=response.read()
    if(re.findall('wrong',res)):
        continue
    else:
        print i
        print res

79
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
<html lang="en">
<head>
  <title>Prob 26</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
  Password is Wh4t_TH3_Hell<br><br>      <form method='get'>
    pw : <input type="text" name="pw" style="width:93;">
    <input type="submit" value="Go">
  </form>
</body>
</html>
```

소스의 순서는 다음과 같다.

1. con 에 HTTPConnection 함수를 이용해 suffad.netne.net 과 연결시킨다.
2. 로그인을 해야 하는 페이지이므로 로그인했을때의 쿠키를 긁어와 사전 형식으로 header 변수에 넣는다.

3. for 문으로 1 부터 200 까지 반복시킨다.

3-1 request 를 이용해 GET 형식으로 /challenge/26/?pw+str(i) 사이트에 헤더와 함께 요청을 보낸다.

(<http://suffad.netne.net/challenge/26/?pw=i> 변수의 값 이 됨)

3-2 response 변수에 위에서 요청보냈던 결과를 받아온다.

3-3 res 변수에 response 를 읽는다.

3-4 정규식을 이용해 답이 아닌지 검사한다.

(wrong 이 들어있으면 답이 아님)

3-5 wrong 이 들어있지 않으면 response 를 출력한다. (소스 출력)

GET 형식으로 헤더를 포함해 웹 사이트에 요청시키는 소스였다.

위의 경우는 로그인에 필요한 사이트에 쿠키의 세션을 헤더에 포함해 요청시킬 때 쓰인다.

다음은 request 함수 부분의 원 모양이다.

con.request('GET','주소?'+값, 헤더)

1-3 httplib 를 이용한 POST 형식

1-3.1 헤더가 없을 때

1-3.2 헤더가 있을 때

5. 후기

이번 기회로 인해 한글 인코딩 구조에 대해 자세히 알게 되었다.

UTF-8 인코딩 문제 때문에 엄청 많은 시간을 소비하여 공부하였다.

많은 시간을 들여 파고든 만큼 더더욱 기억에 남을 것이다.

혹시나 나처럼 한글 인코딩 구조 때문에 시간 허비하는 사람들이 있을까 해서 문서를 쓴다.

이 문서를 봄으로써 한글 인코딩 구조를 파악하는 시간을 줄이고 좀 더 쉽게 이해했으면 한다.