Getting a raw, unparsed HTTP response

Are there any straightforward ways to make a HTTP request and get at the raw, unparsed response (specifically the headers)?

python http-headers http-request

edited Jun 14 '12 at 19:17
james.garriss
5,243 4 43 68

asked Jan 18 '12 at 22:18
Acorn
22k 11 67 126

What exactly are you hoping to do with it that you can't do with the more user-friendly ways of receiving the data? – Karl Knechtel Jan 19 '12 at 0:03

1 I want to see if the server is using `\n` instead of `\r\n` in its responses. – Acorn Jan 19 '12 at 0:06

1 Answer

Using the socket module directly:

```
import socket

CRLF = "\r\n"

request = [
    "GET / HTTP/1.1",
    "Host: www.example.com",
    "Connection: Close",
    "",
    "",
]

# Connect to the server
s = socket.socket()
s.connect(('www.example.com', 80))

# Send an HTTP request
s.send(CRLF.join(request))

# Get the response (in several parts, if necessary)
response = ''
buffer = s.recv(4096)
while buffer:
    response += buffer
    buffer = s.recv(4096)

# HTTP headers will be separated from the body by an empty line
header_data, _, body = response.partition(CRLF + CRLF)

print header_data
```

```
HTTP/1.0 302 Found
Location: http://www.iana.org/domains/example/
Server: BigIP
Connection: Keep-Alive
Content-Length: 0
```

edited Jan 19 '12 at 0:00
Jeremy Banks
43.9k 49 208 252

answered Jan 18 '12 at 22:29
Ian Clelland
21.1k 4 51 59

Wow, thanks for the fully working solution! Does this get hugely more complicated if you need to use HTTPS? – Acorn Jan 18 '12 at 22:58

Thanks, Jeremy -- I just logged in again to add the response splitting, and you beat me to it :)
– Ian Clelland Jan 18 '12 at 23:12

2 @Acorn: If you want to do SSL this way, you will need to impor the ssl module, and use an SSLSocket rather than a regular socket. I haven't used it myself, so there may be other differences. Sounds like a good topic for another SO question, though :) – Ian Clelland Jan 18 '12 at 23:17