class urllib.request.**Request**(*url*[, *data*][, *headers*][, *origin\_req\_host*][, *unverifiable*])

This class is an abstraction of a URL request.

*url* should be a string containing a valid URL.

*data* may be a string specifying additional data to send to the server, or **None** if no such data is needed. Currently HTTP requests are the only ones that use *data*; the HTTP request will be a POST instead of a GET when the *data* parameter is provided. *data* should be a buffer in the standard *application/x-www-form-urlencoded* format. The[**urllib.parse.urlencode()**](https://docs.python.org/3.0/library/urllib.parse.html#urllib.parse.urlencode) function takes a mapping or sequence of 2-tuples and returns a string in this format.

*headers* should be a dictionary, and will be treated as if [**add\_header()**](https://docs.python.org/3.0/library/urllib.request.html#urllib.request.Request.add_header) was called with each key and value as arguments. This is often used to “spoof” the User-Agent header, which is used by a browser to identify itself – some HTTP servers only allow requests coming from common browsers as opposed to scripts. For example, Mozilla Firefox may identify itself as "Mozilla/5.0 (X11; U; Linux i686) Gecko/20071127 Firefox/2.0.0.11", while **urllib**‘s default user agent string is "Python-urllib/2.6" (on Python 2.6).

urllib.request.**urlopen**(*url*[, *data*][, *timeout*])

Open the URL *url*, which can be either a string or a [**Request**](https://docs.python.org/3.0/library/urllib.request.html#urllib.request.Request) object.

*data* may be a string specifying additional data to send to the server, or **None** if no such data is needed. Currently HTTP requests are the only ones that use *data*; the HTTP request will be a POST instead of a GET when the *data* parameter is provided. *data* should be a buffer in the standard *application/x-www-form-urlencoded* format. The[**urllib.parse.urlencode()**](https://docs.python.org/3.0/library/urllib.parse.html#urllib.parse.urlencode) function takes a mapping or sequence of 2-tuples and returns a string in this format.

response = urllib2.urlopen(request)

# check content type:

print response.info().getheader('Content-Type')

<https://docs.python.org/3.2/library/urllib.request.html#module-urllib.response>

**>>> import** **urllib.request**

**>>> import** **urllib.parse**

**>>>** data = urllib.parse.urlencode({'spam': 1, 'eggs': 2, 'bacon': 0})

**>>>** data = data.encode('utf-8')

**>>>** request = urllib.request.Request("http://requestb.in/xrbl82xr")

**>>>** *# adding charset parameter to the Content-Type header.*

**>>>** request.add\_header("Content-Type","application/x-www-form-urlencoded;charset=utf-8")

**>>>** f = urllib.request.urlopen(request, data)

**>>>** print(f.read().decode('utf-8'))

**>>> import** **urllib.request**

**>>> import** **urllib.parse**

**>>>** params = urllib.parse.urlencode({'spam': 1, 'eggs': 2, 'bacon': 0})

**>>>** f = urllib.request.urlopen("http://www.musi-cal.com/cgi-bin/query?%s" % params)

**>>>** print(f.read().decode('utf-8'))

url = 'https://www.google.com/search?q=python'

# now, with the below headers, we defined ourselves as a simpleton who is

# still using internet explorer.

headers = {}

headers['User-Agent'] = "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"

req = urllib.request.Request(url, headers = headers)

resp = urllib.request.urlopen(req)

respData = resp.read()

saveFile = open('withHeaders.txt','w')

saveFile.write(str(respData))

saveFile.close()

except Exception as e:

print(str(e))

<https://pythonprogramming.net/urllib-tutorial-python-3/>

<http://httpbin.org/>