How to handle response encoding from urllib.request.urlopen()

I'm trying to search a webpage using regular expressions, but I'm getting the following error:

> TypeError: can't use a string pattern on a bytes-like object

I understand why, urllib.request.urlopen() returns a bytestream and so, at least I'm guessing, re doesn't know the encoding to use. What am I supposed to do in this situation? Is there a way to specify the encoding method in a urlrequest maybe or will I need to re-encode the string myself? If so what am I looking to do, I assume I should read the encoding from the header info or the encoding type if specified in the html and then re-encode it to that?

python regex encoding urllib

edited Feb 13 '11 at 2:08
Tim Cooper
79k 20 153 174

asked Feb 13 '11 at 2:05
kryptobs2000
619 2 8 24

5 Answers

You just need to decode the response, using the `Content-Type` header typically the last value.
There is an example given in the tutorial too.

```
output = response.decode('utf-8')
```

answered Feb 13 '11 at 2:09
Senthil Kumaran
20.3k 8 47 77

Thanks, that's what I needed. – kryptobs2000 Feb 13 '11 at 2:12

5 What if the charset is not utf-8? Would it be a better idea to somehow determine it from the response instead of hard-coding this assumption? – Elias Zamaria Jun 23 '14 at 17:56

As for me, the solution is as following (python3):

```
resource = urllib.request.urlopen(an_url)
content =  resource.read().decode(resource.headers.get_content_charset())
```

answered Oct 3 '13 at 9:54
Ivan Klass
2,163 9 17

4 Looks like the best answer but what if the server doesn't send the charset info? – rvighne Jul 16 '14 at 18:05

If the server doesn't send charset info your best bet at that point is to guess. – Iguananaut Aug 6 '14 at 16:30

6 @rvighne: if the server doesn't pass `charset` in `Content-Type` header then there are complex rules to figure out the character encoding e.g., it may be specified inside html document: `<meta charset="utf-8">` . – J.F. Sebastian Oct 22 '14 at 4:38

```
urllib.urlopen(url).headers.getheader('Content-Type')
```

Will output something like this:

```
text/html; charset=utf-8
```

edited Dec 1 '11 at 17:08

Brian Deragon
2,064 9 37

answered Dec 1 '11 at 16:48

wynemo
856 1 7 7

I had the same issues for the last two days. I finally have a solution. I'm using the `info()` method of the object returned by `urlopen()` :

```
req=urllib.request.urlopen(URL)
charset=req.info().get_content_charset()
content=req.read().decode(charset)
```

edited Nov 17 '15 at 12:50

Glenn
5,764 2 17 39

answered Nov 17 '15 at 12:41

pytohs
21 4

---

after you make a request `req = urllib.request.urlopen(...)` you have to read the request by calling `html_string = req.read()` that will give you the string response that you can then parse the way you want.

answered Feb 13 '11 at 2:09

Jesse Cohen
2,968 12 22

1 I do, that's how I get it, but it returns a bytesteam, b'<HTML>...'. – kryptobs2000 Feb 13 '11 at 2:10

i see, then you can use `.decode()` as @Senthil pointed out or you can use urllib2 which should handle this transparently to you. – Jesse Cohen Feb 13 '11 at 2:13
