U5NMSURAQ: true that

U5NMSURAQ: but I'd guess the most useful stuff is already there

U4BMZ90T0: Does anyone know if there is any possible way to figure out the proper URL for an API call? I have an issue with an API im trying to work with that basically one of the URL parameters can be changed server side and the guy who maintains the server might as well be AWOL. So basically is there any to see the endpoints available if I make a request to the parent endpoint of the endpoint I don't know? Example -

`<http://crucible.server.com/{unknown-endpoint-name}>`

I'd imagine if there is no WADL or some sort of blueprint for a given server than probably not

U2UP91YLE: anyone comfortable with marshmallow?, trying to remove missing data after deserialization, i don't want `{'first_field': <marshmallow.missing>, 'second_field': u'some value'}`

U6AFDVDA9: hello, question related to something i asked yesterday.

i have a small web app that allows users to search instagram for two hashtags. on the first page they enter their hashtags, on the second page they get the results and a button to process more results. clicking this button send the user to a URL with the next page token and the hash tags (required for processing next results) stored in the url. like this

`<a href="http://10.51.50.10:42066/results/J0HWW-lwQAAAF0HWW-jjgAAAFoAKAA%253D%253D/%5B%27lol%27%2C%20%27dog%2

```
my problem is i keep getting this error and i'm not sure what it means
werkzeug.routing.BuildError: Could not build url for endpoint 'results' with values ['tags']. Did you forget to specify values
['token']?
10.51.50.1 - - [19/Jul/2017 18:36:54] "GET
/results/J0HWW-lwQAAAF0HWW-jjgAAAFoAKAA%253D%253D/%5B%27lol%27%2C%20%27dog%27%5D?
HTTP/1.1" 500 -
INFO:werkzeug:10.51.50.1 - - [19/Jul/2017 18:36:54] "GET
/results/J0HWW-lwQAAAF0HWW-jjgAAAFoAKAA%253D%253D/%5B%27lol%27%2C%20%27dog%27%5D?
HTTP/1.1" 500 -
this is the route
@app.route('/results/', methods=['GET', 'POST'], defaults={'token': None, 'tags': None})
@app.route('/results/<token&gt;/&lt;tags&gt;', methods=['GET', 'POST'])
def results(token, tags):
  if request.method == 'POST':
     tags = request.form['text'].replace(' ', '').split(',')
     data = fetch_data(tags=tags, next_page_token=token)
     processed_data, token = process_data(data=data, tags=tags)
  if request.method == 'GET':
     <a href="http://logging.info|logging.info">("TOKEN: {}".format(token))</a>
     <a href="http://logging.info|logging.info">("TAGS: {}".format(tags))</a>
     data = fetch_data(tags=tags, next_page_token=token)
     processed_data, token = process_data(data=data, tags=tags)
     return render template('results.html', data=processed data[:1], token=token, tags=tags)
  return render_template('results.html', data=processed_data[:1], token=token, tags=tags)
and this is the template
```

<!DOCTYPE html>

<html>

<body style="background-color:powderblue;">

<h1>{{ data }}</h1>

```
<form action="{{ url_for('next', token=token, tags=tags) }}"&gt;
  <input type="submit" value="More results plz" /&gt;
```

```
</form&gt;
</body&gt;
</html&gt;
U6AFDVDA9: any thoughts?
U5NMSURAQ: ```127.0.0.1 - - [19/Jul/2017 21:43:01] "GET
/results/J0HWW-lwQAAAF0HWW-jjgAAAFoAKAA%253D%253D/%5B%27lol%27%2C%20%27dog%27%5D?
HTTP/1.1" 200 -
U5NMSURAQ: ```from flask import Flask
app = Flask('app')
app.config['DEBUG'] = True
@app.route('/results/', methods=['GET', 'POST'], defaults={'token': None, 'tags': None})
@app.route('/results/<token&gt;/&lt;tags&gt;', methods=['GET', 'POST'])
def results(token, tags):
  return 'OK'
U5NMSURAQ: Strange indeed.
U6AFDVDA9: one thought i just had
U6AFDVDA9: my token and tag vars come through as ""
INFO:root:TOKEN: J0HWW-IwQAAAF0HWW-jjgAAAFoAKAA%3D%3D
INFO:root:TAGS: ['lol', 'dog']
U6AFDVDA9: could the tags coming through as a python data structure as a string have somethign to do with it?
U6AFDVDA9: i've been banging my head on this for hours. and you saying strange indeed gave me some validation
that is helping me to not go crazy haha
U5NMSURAQ: wait, what do you mean coming through as a python data structure?
U6AFDVDA9: it's getting placed in the url like this `['lol', 'dog']` but encoded
U6AFDVDA9: \hat\%5B\%27\ol\%27\%2C\%20\%27\dog\%27\%5D?\hat\
U5NMSURAQ: Works either way:confused:
U6AFDVDA9: hmmmmmmm
U653RFLBD: `J0HWW-IwQAAAF0HWW-jjgAAAFoAKAA%253D%253D` !=
`J0HWW-lwQAAAF0HWW-jjgAAAFoAKAA%3D%3D`. % sign in debug token got url encoded to %25
U6AFDVDA9:?
U4BMZ90T0: If I'm trying to parse some xml should I be using something like `beautifulsoup`?
U2BS4M1RV: I've not used beautifulsoup, but if it is just xml there are other xml libraries
U5NMSURAQ: why not lxml: slightly smiling face:
U5NMSURAQ: it's got `xml` in its name
U2BS4M1RV: I believe that is the one I liked.
U2BS4M1RV: It can parse xml without you having to know anything about the content.
U5NMSURAQ: and beautifulsoup only has `soup`...
U5NMSURAQ: ah. and 'beautiful'...
U29163YQH: be sure to use 'defusedxml' if you are parsing unknown xml
U4BMZ90T0: Yeah I saw there was some possibility of malicious xml with certain libraries unable to deal with it
U4BMZ90T0: Why does xml seem so much more annoying than json... ugh
U29163YQH: because it is?:smile:
U6AFDVDA9: oh wow <@U653RFLBD> i see what you are saying. it's not failing at the flask level, it's failing at the
instagram fetch level because the token is wrong
U2BS4M1RV: Yeah, I would much prefer json.
U4BMZ90T0: Woo 'from lxml import etree' can't find reference to etree: smile:
U5NMSURAQ: `import lxml.etree as etree`
```

U5NMSURAQ: `tree = etree.fromstring(binary data)`

U4BMZ90T0: `no module named etree`

U5NMSURAQ::open mouth:

U4BMZ90T0 : gonna stick with just xml.etree lol U4BMZ90T0 : pip installed and everything

U4BMZ90T0: odd

U5NMSURAQ : enjoy slow parsing :slightly_smiling_face:

U5NMSURAQ : but yeah, pretty strange U2BS4M1RV : Slow and frustrating.

U2BS4M1RV: lxml should really be in the standard library.

U5NMSURAQ: well, `xml` already is and they share an interface (I think)

U42LCHFD2: all you have to do is ```import requestsimport lxml

from lxml import html

def xpathScraper(url,tag):

page = requests.get(url)
tree = html.fromstring(page.content)
web_content = tree.xpath(tag)
return web_content```

U4BMZ90T0: Yeah I think that the Etree class is shared

U4L4AH15L: Why no `BeautifulSoup`? it supports `lxml` as well

U5NMSURAQ : BeautifulSoup's interface is bizarre

U5NMSURAQ: like really strange

U5NMSURAQ: And it's an abstraction over lxml either way

U42LCHFD2: im not a fan of bsoup

U42LCHFD2: and everything i have ever tried/wanted to do i was able to solve with lxml in one way or another

U2BS4M1RV: I just installed lxml and I noticed I get an import error as well. "

Traceback (most recent call last):

File "<input>", line 1, in <module>

File "C:\Users\AB14925\Desktop\Viewfinity\PyCharm Community Edition

2017.1.3\helpers\pydev_pydev_bundle\pydev_import_hook.py", line 21, in do_import

module = self._system_import(name, *args, **kwargs)

ImportError: DLL load failed: The specified procedure could not be found.

U5NMSURAQ: Ixml has cssselect, which is a godsend