is it possible to get all possible urls?

Asked 11 years, 6 months ago Modified 8 years, 5 months ago Viewed 15k times

6

I am trying to write a function as follows:

```
def get_urls(*urls,restrictions=None):
    #here there should be some code that
    #iterates through the urls and create
    #a dictionary where the keys are the
    #respective urls and their values are
    #a list of the possible extentions. The
    #function should return that dictionary.
```

First, to explain. If I have a site: www.example.com, and it has **only** the following pages: www.example.com/faq, www.example.com/history, and www.example.com/page/2. This would be the application:

```
In[1]: site = 'http://example.com'
In[2]: get_urls(site)
Out[2]: {'http://example.com':['/faq','/history','/page/2']}
```

I have spent hours researching, and so far this seems impossible! So am I missing some module that can do this? Is there one that exists but not in python? If so, what language?

Now you are probably wondering why there is `restrictions=None`, well here is why:

I want to be able to add restrictions to what is an acceptable url. For example `restrictions='first'` could make it only do pages that exist with one `'/'`. Here is an example:

```
In[3]: get_urls(site,restrictions='first')
Out[3]: {'http://example.com':['/faq','/history']}
```

I don't need to keep explaining the ideas for restrictions, but you understand the need for it! Some sites, especially social networks, have some crazy add ons for ever picture and weeding those out is important while keeping the original page consisting of all the photos.

So yes, I have absolutely no code for this, but that is because I have no clue what to do! But I think I made myself clear about what I need to be able to do, so, **is this possible? If yes, how? if no, why not?**

EDIT:

So after some answers and comments, here is some more info. I want to be given a url, not necessarily a domain, and return a dictionary with the original url as the key and a list of all the the extensions of that url as the items. Here is an example with my previous `'example.com'`:

```
In[4]: site = 'http://example.com/page'
In[5]: get_urls(site)
Out[5]: {'http://example.com/page':['/2']}
```

The crawling examples and beautiful soup is great, but if there is some url that is not directly linked on any of the pages, then I can't find it. Yes, that generally is not a concern, but I would like to be able to!

python **function** **url**

Share

Improve this question

Follow

edited May 29, 2013 at 4:33

asked May 29, 2013 at 4:05

Ryan Saxe

17.8k ● 23 ● 84 ● 130

4 Unless the website has an easy to parse sitemap, you'll probably need to crawl it. – GWW May 29, 2013 at 4:07

1 How broad is "all possible"? All of the links on that page? All of the links in the entire website? What have you tried so far? – Blender May 29, 2013 at 4:08

@Blender: I have not tried anything really because I was unable to find a library for this. See I have no clue how to get all the urls on a given site, and I was wondering what module is capable of it. And with all possible is all, but there would be a `restrictions` part of the function so that it only does some. – Ryan Saxe May 29, 2013 at 4:12

1 It's not possible in general to get all URLs on a given site. Not because of some deficiency in Python, but because the HTTP protocol does not allow for it. – David Z May 29, 2013 at 5:10

so there is no way to access that? is there some article somewhere that explains why? – Ryan Saxe May 29, 2013 at 5:15

2 Answers

Sorted by: Highest score (default)

I'm interpreting your question as "Given a URL, *find* the set of URLs that exist "below" that URL." - if that's not correct, please update your question, it's not very clear.

17

It is not possible to discover the entire set of valid paths on a domain, your only option would be to literally iterate over every valid character, e.g. `/`, `/a`, `/b`, `/c`, ..., `/aa`, ... and visit each of these URLs to determine if the server returns a 200 or not. I hope it's obvious this is simply not feasible.

It is possible (though there are caveats, and the website owner may not like it / block you) to crawl a domain by visiting a predefined set of pages, scraping all the links out of the page, following those links in turn, and repeating. This is essentially what Google does. This will give you a set of "discover-able" paths on a domain, which will be more or less complete depending on how long you crawl for, and how vigorously you look for URLs in their pages. While more feasible, this will still be very slow, and will not give you "all" URLs.

What problem exactly are you trying to solve? Crawling whole websites is likely not the right way to go about it, perhaps if you explain a little more your ultimate goal, we can help identify a better course of action than what you're currently imagining.

The underlying issue is there isn't necessarily any clear meaning of an "extension" to a URL. If I run a website (whether my site lives at `http://example.com`, `http://subdomain.example.com`, or `http://example.com/page/` doesn't matter) I can trivially configure my server to respond successfully to any request you throw at it. It could be as simple as saying "every request to `http://example.com/page/.*` returns `Hello World.`" and all of a sudden I have an infinite number of valid pages. Web servers and URLs are similar, but fundamentally not the same as hard drives and files. Unlike a hard drive which holds a finite number of files, a website can say "yes that path exists!" to as many requests as it likes. This makes getting "all possible" URLs impossible.

Beyond that, webservers often don't *want* you to be able to find all valid pages - perhaps they're only accessible if you're logged in, or at certain times of day, or to requests coming from China - there's no requirement that a URL always exist, or that the webserver tell you it exists. I could very easily put my infinite-URL behavior below `http://example.com/secret/path/no/one/knows/about/.*` and you'd never know it existed unless I told you about it (or you manually crawled all possible URLs...).

So the long story short is: No, it is not possible to get all URLs, or even a subset of them, because there could theoretically be an infinite number of them, and you have no way of knowing if that is the case.

> if I can add restrictions, that will make it easier!

I understand why you think this, but unfortunately this is not actually true. Think about URLs like regular expressions. How many strings match the regular expression `.*`? An infinite number, right? How about `/path/.*`? Less? Or `/path/that/is/long/and/explicit/.*`? Counter intuitive though it may seem, there are actually no fewer URLs that match the last case than the first.

Now that said, my answer up to this point has been about the general case, since that's how you posed the question. If you clearly define and restrict the search space, or loosen the requirements of the question, you can get an answer. Suppose you instead said "Is it possible to get all URLs *that are listed on this page* and match my filter?" then the answer is yes, absolutely. And in some cases (such as Apache's Directory Listing behavior) this will *coincidentally* be the same as the answer to your original question. However there is no way to guarantee this is actually true - I could perfectly easily have a directory listing with secret, unlisted URLs that still match your pattern, and you wouldn't find them.

edited Jul 25, 2016 at 3:14

answered May 29, 2013 at 4:18

dimo414

48.7k ● 19 ● 163 ● 263

I added an edit to make it more clear! As far as the problem I am trying to solve, there is not one at this time, this is just something I want to be able to do – Ryan Saxe May 29, 2013 at 4:37

Whether you want the list of all URLs on a domain or just in a sub-path of a domain, the problem (and the problem-space, i.e. the number of possible URLs) doesn't get any easier, unfortunately. – dimo414 May 29, 2013 at 5:11

Right, but if I can add restrictions, that will make it easier! It would be rare to run something like this without restrictions because of what you pointed out. – Ryan Saxe May 29, 2013 at 5:15

@RyanSaxe, see my additional edits. I'm afraid the answer to the question you're posing is still "No". If you loosen the requirements (for instance, "get some" instead of "get all") the answer changes, but "get all" is not possible. – dimo414 May 29, 2013 at 5:32

This is exactly what I was looking for! Yes I know it's possible to go grab all links on a page and all links on those pages that fit a restriction with many libraries, but you gave a valid explanation as to why I cannot find those other urls and garentee that the pattern will halt! – Ryan Saxe May 29, 2013 at 6:00

0

This question has a good answer. Essentially, you are asking why crawlers are necessary as opposed to a list of all the directories. Wikipedia explains, "The basic premise is that some sites have a large number of dynamic pages that are only available through the use of forms and user entries."

Share

Improve this answer

Follow

edited May 23, 2017 at 12:34

Community Bot

1 ● 1

answered May 29, 2013 at 4:13

isaach1000

1,839 ● 1 ● 14 ● 19