getsitemap

Release 0.1

capjamesg

Oct 13, 2022

CONTENTS

getsitemap is a simple Python library that retrieves all the URLs in the sitemaps associated with a website.

This library may be useful when building a web search crawler, an SEO validation tool, or a sitemap monitor.

You can download *getsitemap* using the following comamnd:

```
pip install getsitemap
```

See the documentation for *getsitemap* below.

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

TWO

GET ALL URLS IN A WEBSITE'S SITEMAPS

The *retrieve_sitemap_urls()* function returns all URLs found in a website's sitemaps.

This function:

1. Checks for *Sitemap* directives in a website's robots.txt file. All sitemap found are crawled recursively.
2. Checks for the presence of a sitemap.xml file. If one is found, it is crawled recursively.
3. Merges the results of all checks to return either a list of all URLs or a dictionary that maps each URL to the sitemap in which it was found.

getsitemap.**retrieve_sitemap_urls**(*root_page: str*, *as_flat_list: bool = True*, *allow_xml_inference: bool = True*, *thread_max: int = 20*, *dedupe_results: bool = True*) → Union[list, dict]

Find all of the URLs in every sitemap associated with a provided domain.

This function will take a bit of time to run depending on how many URLs are discovered.

Parameters

- **root_page** (*str*) – The root page of the domain to search for sitemaps.
- **as_flat_list** (*bool*) – Whether or not to return the URLs as a flat list.
- **allow_xml_inference** (*bool*) – Whether or not to infer that a URL ending in .xml is a sitemap.
- **thread_max** (*int*) – The maximum number of threads to use in sitemap retrieval requests.
- **dedupe_results** (*bool*) – Whether or not to remove duplicate URLs.

Returns
A list of URLs.

Return type
Union[list, dict]

Example:

```
import getsitemap

all_urls = getsitemap.retrieve_sitemap_urls("https://www.example.com")

print(all_urls) # ["https://www.example.com", "https://www.example.com/about", ...]
```

To get a list of all sitemaps in a website, you can append *.keys()* to the result of this function, as long as you specify *as_flat_list=False* in the command arguments.

Please note this function may take time to run if there are a lot of sitemaps to crawl. This is because a network request has to be made for each URL.

CHAPTER

THREE

GET ALL URLS IN A SINGLE SITEMAP

The *get_individual_sitemap()* function returns all URLs found in a single sitemap.

With the *recurse=True* argument, this function will also crawl all sitemaps found in the sitemap and do so recursively.

If *recurse=False*, this function will return only the list of URLs in the provided sitemap file. This will include sitemap files if you use this function on a sitemap index.

getsitemap.**get_individual_sitemap**(*root_url: str*, *thread_max: int = 20*, *dedupe_results: bool = True*, *allow_xml_inference: bool = True*, *recurse: bool = False*) → dict

Get all of the URLs associated with a single sitemap.

Parameters

- **root_url** (*str*) – The URL of the sitemap.
- **thread_max** (*int*) – The maximum number of threads to use in sitemap retrieval requests.
- **allow_xml_inference** (*bool*) – Whether or not to infer that a URL ending in .xml is a sitemap.
- **recurse** (*bool*) – Whether or not to recurse into other sitemaps.

Returns

A dictionary of URLs found in each discovered sitemap.

Return type

dict

Example:

```
import getsitemap

urls = getsitemap.get_individual_sitemap("https://jamesg.blog/sitemap.xml")

print(urls) # ["https://jamesg.blog/2020/09/01/my-experience-with-jekyll/", ...]
```

INDEX

G

R