Tiki Crawl

A crawler tool for checking links or gathering content from websites.

This is the alpha version of a crawler tool for checking links or gathering content from websites. It relies on the Crawler library from Spatie. Kudo to those guys.

Introduction

Tiki-Crawl is a powerful tool that allows you to crawl and index external websites or content from remote sources. It enables you to create a local index of content from external sites, which can be useful for various purposes such as aggregating content, building a search index, or archiving information.

With Tiki-Crawl, you can specify a list of URLs or feeds from external website or remote sources that you want to index, and TikiWiki will periodically crawl these sources to fetch and index the content. This process is often referred to as web scraping and web crawling

What is Tiki-Crawl?

Tiki-Crawl is a web scraping and web crawling feature that enables users to fetch and index content from external websites or remote data sources. It automates the process of gathering information from these sources and can stores it.

Requirements

To use Tiki-Crawl effectively, ensure that you have this version installed (but it may very well be that it works on older versions):

- PHP: Tiki-crawl requires `php version 8.2.6` or higher.
- Node: Tiki-crawl requires `node version 18.16.0` or higher.

Installation

First you need to clone the tool from https://gitlab.com/tikiwiki/tiki-crawl.

This piece of code has been tested with `php 8.2.6` and `node v18.16.0`, make sure you have those version installed (but it may very well be that it works on older versions, code is not that complicated).

```
composer update npm install
```

Usage

First you need to configure your options. You can override any options from `config/config.default.php` by creating a `config/config.php` file where you will be adding new values that will override the one in `config/config.default.php`, for example:

```
<?php $options['timeout'] = 600; $options['js'] = true;
```

- **Note**: You don't have to change values from `config/config.default.php`.

Then you can launch the crawling with the command `./bin/crawl` following with the link of the website that you need to crawl. like the bellow command:

```
./bin/crawl https://books.toscrape.com
```

- **Note**: `toscrape.com` is useful for testing crawler, you can replace it with the link of any website that you need to crawl

Config options

This are configuration options that you can override in you new `config/config.php` file

Name	Type	Default	Description
timeout	integer	60	the maximum time that should be waited before getting a response for any page crawled
cli	boolean	true	show a progress dotted line while launching it in console
log	boolean	true	keeps logs in `logs/` directory (one for access one for errors)
store	boolean	false	store crawled pages in `collected-data/` directory
max_size	integer	2	when `store/` is enabled, the maximum size of stored documents, if you want to get pdfs you should up
limit	?integer	5	number of pages to crawl, set to `false` for unlimited crawls
js	boolean	false	use headless chrome with puppeteer (takes much longer, and requires to have puppeteer installed)
internal	boolean	true	only crawl urls that are on the same host
concurrency	integer	10	number of concurrent requests to make
max_depth	?integer	false	if you want to only get the immediate links on the page you are crawl, set `1`, you can decide how deep you want to crawl from initial page
delay	?integer	false	add a delay in milliseconds between requests
skip_urls	?array	false	if you stumble upon problematic urls that make the crawler crash, you can skip them by listing them in this array
allow_redirect	boolean	false	wether or not follow 30x redirects
ignore_robots	boolean	false	bypass instructions contained in `robots.txt` file
allow_nofollow	boolean	false	bypass the `rel="nofollow"` directive in links
user_agent	string	TikiCrawl	the user-agent header sent with http requests

The jQuery Sortable Tables feature must be activated for the sort feature to work.

Those options will be passed to spatie, you can learn more about those at https://github.com/spatie/crawler/

Roadmap

- handle the case where headless browser fails miserably and crawler lib don't catch it
- improve crawling on sites by using the canonical url that may be declared inside the html
- add an option to try gathering or checking http status on images (for finding missing ones)
- publish as a composer package
- add a feature in Tiki to make use of it

Notes

- It relies on the Crawler library from Spatie (Over 7 million downloads).
- It starts off as a tool just for developers, and will eventually be integrated in Tiki, so accessible to power users.

This was used to gather data for an upcoming AI Chatbot we are working on (more news on this later, along with some code).

Related Link

https://gitlab.com/tikiwiki/tiki-crawl