GPT CRAWLER

GPT Crawler Project Documentation

1. Overview

GPT Crawler is a web scraping tool designed to extract data from websites and feed it to OpenAI's ChatGPT model. This enables the creation of a chatbot that can answer questions about the scraped website's content.

2. System Architecture

- Web Scraping Module: Uses Playwright for browser automation
- Data Processing Module: Cleans and formats scraped data
- OpenAI API Integration: Feeds processed data to ChatGPT
- Chatbot Interface: Handles user queries and displays responses

3. Web Scraping Process

1. Define target URL and crawling parameters
2. Use Playwright to navigate and interact with web pages
3. Extract relevant data using CSS selectors
4. Store scraped data in a structured format (e.g., JSON)

4. Data Processing

Clean and normalize extracted data to ensure consistency and remove irrelevant information. This process may include text cleaning, entity recognition, and content categorization.

5. OpenAI API Integration

Develop a module to interact with the OpenAI API, sending processed website data as context for the ChatGPT model. Multiple files can be added as a knowledge base.

6. Chatbot Implementation

Create a user interface for the chatbot, allowing users to ask questions about the scraped website. The chatbot uses the ChatGPT model to generate responses based on the provided website data.

7. Future Improvements

- Implement incremental crawling to keep data up-to-date
- Develop a user-friendly interface for configuring crawl parameters
- Create a UI for the tool

This documentation provides a high-level overview of the GPT Crawler project, outlining the approach to scrape website data and create a ChatGPT-powered chatbot for answering questions related to the scraped content.

8. Limitations

- Requires OpenAI subscription
- Crawling time is lengthy for websites with numerous URLs or subpages
- The approximate upper limit of pages to scrape should be known

9. Tech Stack

• TypeScript

• PlaywrightCrawler

• JavaScript for building UI and API chatbot integration

• Node.js

• OpenAI API

10. Features

- Crawl websites to generate knowledge files for custom GPTs
- Configurable crawling parameters (URL, selector, page limit, etc.)
- Support for running locally, in Docker, or as an API
- Integration with OpenAI for creating custom GPTs and assistants

11. Installation and Setup

Clone the repository

Ensure Node.js ≥ 16 is installed, then run:

```
git clone https://github.com/builderio/gpt-crawler
```

Install dependencies

```
npm i
```

Configure the crawler

Edit the `config.ts` file to set crawling parameters:

```
export const defaultConfig: Config = {
  url: "https://www.example.com",
  match: "https://www.example.com/**",
  selector: `.main-content`,
  maxPagesToCrawl: 50,
  outputFileName: "output.json",
};
```

Run the crawler

```
npm start
```

12. Alternative Running Methods

Docker Container

For containerized execution, use the `containerapp` directory and modify its `config.ts`.

API Server

To run as an API server:

```
npm run start:server
```

Access the API docs at the `/api-docs` endpoint.

13. Creating Custom GPTs

1. Go to https://chat.openai.com/
2. Click your name in the bottom left corner
3. Choose "My GPTs" in the menu
4. Click "Create a GPT"
5. Choose "Configure"
6. Under "Knowledge," upload the generated output file

14. Creating Custom Assistants

1. Visit https://platform.openai.com/assistants
2. Click "+ Create"
3. Choose "upload" and select the generated file

15. Limitations and Considerations

- File size limits may require splitting or tokenization of large outputs
- Crawling time increases with the number of pages or subpages
- PDFs and images in the website needs to be uploaded separately.
- Requires an OpenAI subscription for custom GPT creation
- Vector Embeddings is not necessary for fine tuning as the vectorization process is done by CHATGPT MODEL . This process supports various file formats . JSON is one among them.
- Thus we go with the approach of uploading the JSON file directly as the vector base in the assistant UI provide by OpenAi.