HTML Parser

Roshan Bolonna

LucyTech’s HTML Parser Practical Test

1/11/25

Contents

Introduction

This application is designed to efficiently crawl through the provided URL and extract essential information from the web page. The server is capable of parsing HTML content and provides the following information:

- **Internal Links**: Extracts and lists all internal links present on the page.
- **External Links**: Extracts and lists all external links leading to different domains.
- **HTML Version**: Identifies and displays the HTML version used by the web page.
- **Page Title**: Retrieves and shows the title of the web page.
- **Login Page**: Identify login page.

With these capabilities, our HTML Parser Server aims to simplify web scraping and data extraction tasks, making it a valuable tool for developers and analysts.

Project Structure & Specification

Project composed of Frontend and Backend services.

Frontend

Dependencies

This project uses the following dependencies:

Core Dependencies

- **React** version 18.3.1: A JavaScript library for building user interfaces.
- **React DOM** version 18.3.1: Provides DOM-specific methods for React.
- **React Router DOM** version 7.1.1: A complete routing library for React.
- **Ant Design (antd)** version 5.22.7: A popular design system for React.
- **Axios** version 1.7.9: A promise-based HTTP client for the browser and Node.js.

Development Dependencies

- **TypeScript** version ~5.6.2: A typed superset of JavaScript that compiles to plain JavaScript.
- **Vite** version 6.0.5: A build tool that aims to provide a faster and leaner development experience for modern web projects.
- **ESLint** version 9.17.0: A pluggable and configurable linter tool for identifying and reporting on patterns in JavaScript.
- **TypeScript ESLint** version 8.18.2: ESLint tooling for TypeScript.
- **ESLint Plugin React Hooks** version 5.0.0: ESLint rules for React hooks.
- **ESLint Plugin React Refresh** version 0.4.16: ESLint plugin for React fast refresh.
- **Globals** version 15.14.0: Global variable definitions for various JavaScript environments.
- **@vitejs/plugin-react-swc** version 3.5.0: Vite plugin for React with SWC.
- **@eslint/js** version 9.17.0: ESLint configuration.
- **@types/react** version 18.3.18: TypeScript definitions for React.
- **@types/react-dom** version 18.3.5: TypeScript definitions for React DOM.

Scripts

- **dev**: Starts the development server using Vite.
- **build**: Compiles the project using TypeScript and Vite.
- **lint**: Runs ESLint to lint the codebase.
- **preview**: Previews the production build using Vite.

Backend

Dependencies

This project uses the following dependencies:

- **Go** version 1.23.4

Direct Dependencies:

- github.com/PuerkitoBio/goquery v1.10.1: A little like jQuery, but for Go.
- github.com/chromedp/chromedp v0.11.2: Package chromedp is a high-level Chrome DevTools Protocol.
- github.com/gorilla/mux v1.8.1: A powerful URL router and dispatcher for matching incoming requests to their respective handler.
- golang.org/x/net v0.33.0: Supplementary Go networking libraries.

Indirect Dependencies:

- github.com/andybalholm/cascadia v1.3.3: CSS selector library.
- github.com/chromedp/cdproto v0.0.0-20241022234722-4d5d5faf59fb: Package cdproto contains the generated DevTools Protocol domains.
- github.com/chromedp/sysutil v1.1.0: System utilities for Chromedp.
- github.com/gobwas/httphead v0.1.0: Fast HTTP headers parsing library.
- github.com/gobwas/pool v0.2.1: Resource pools for efficient HTTP processing.
- github.com/gobwas/ws v1.4.0: A fast WebSocket library.
- github.com/josharian/intern v1.0.0: Efficient string interning.
- github.com/mailru/easyjson v0.7.7: Fast JSON serialization for golang.
- golang.org/x/sys v0.28.0: Low-level interaction with the operating system.

Setup & Run

Clone the backend and frontend GIT repos.

https://github.com/RoshanGerard/lucytech-html-parser.git

How to Run

Non-Docker Setup

Backend

Run the **main.go** using go cli command.

```
cd ~/<project>
go mod tidy
go run ./cmd/main.go
```

Note:
Make sure to run go mod tidy to keep your go.mod file updated with only the necessary dependencies.

Frontend

Prepare to run

When the repo is cloned, please install the dependencies prior to start running the application, this is a onetime process unless **node_modules** directory doesn't exist or deleted.

```
1. cd ~/<project>
2. npm install
```

Run the App

Here we only provide instructions how to run frontend in development mode, please use the below command to run.

```
1. cd ~/<project>
2. npm run dev
```

Docker Setup & Run

Be sure to install latest docker & docker compose v2 in order to build and run the docker based html parser application.

~~**IMPORTANT NOTE:**~~

~~***Following docker compose build and run process has known issues with core dependencies used in golang backend service causing an error where service to hang indefinitely root cause is originating from chromed used for parsing the HTML documents which doesn't seems to compatible with headless chromium version for linux, therefore docker deployment is not suitable for testing the functions,***~~

~~***Unfortunately, I was unable to resolve this issue within the given timeline, apologies in advance.***~~

Step 1: Navigate to the HTML Parser Docker Distribution repo.

lucytech-html-parser/lucytech-docker-html-parser

Step 2: Provide execution permissions to bash scripts.

```
sudo chmod +x lucytech-docker-html-parser/bin/build.sh
```

Step 3: Build Docker Images

Run below command to build docker images.

```
lucytech-docker-html-parser/bin/build.sh
```

Step 4: Run the App Services

Run the below docker compose command to start the services.

```
docker compose -f lucytech-docker-html-parser/docker-compose/docker-compose.yml up -d
```

Testing Instructions

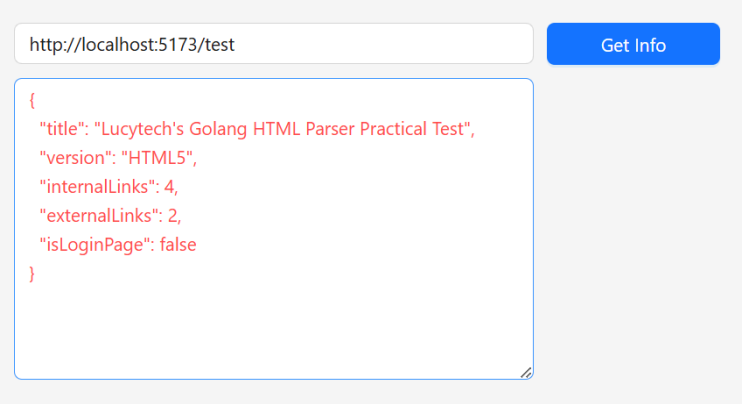
Please follow the testing instructions for testing the following areas featured in the HTML Parser.

- **Internal Links**: Extracts and lists all internal links present on the page.
- **External Links**: Extracts and lists all external links leading to different domains.
- **HTML Version**: Identifies and displays the HTML version used by the web page.
- **Page Title**: Retrieves and shows the title of the web page.
- **Login Page**: Identify login page.

Test: Link Detection, HTML Version, Page Title

Use following links to test the functionalities.

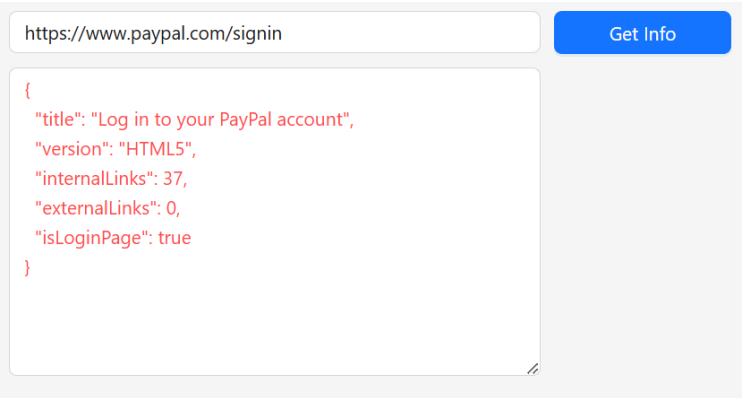
- https://www.google.com
- http://localhost:5173
- http://localhost:5173/test



Test: Detect Login Page Signatures

Use the following links to test the login page signature detection functionalities.

- http://localhost:5173/login
- https://www.paypal.com/signin



Limitations & Inconsistencies

Following limitations and inconsistencies corresponds to the requirement specifications are present.

- Count Headers
- Detect Header Depth
- Dockerization
- CI/CD