

Wsparcio - Backend Repository



Wparcio - agent AI do surfowania po sieci!

Intro

Aplikacja wystawia dwa endpointy dla aplikacji frontend.

Endpoints

GET `api/v1/crawler/crawl`

Endpoint przyjmuje jeden request param - `url`, specyfikujący konkretną stronę internetową, na której będzie operował agent.

Logika za endpointem rekurencyjnie pobiera content html strony głównej oraz jej podstron. Wynikiem operacji jest wypełnienie nimi bazy danych.

W bazie danych trzymane jest url każdej z podstron oraz jej treść html.

PROF

Przechowywanie tych danych w bazie ma na cel nie tylko nośnika danych między siecią web, a czatem GPT, ale też cache'owania kontentu w celu minimalizacji liczby wykonań operacji crawlowania/scrapowania. Czas przechowywania stron w bazie jest możliwy do sparametryzowania (raz dziennie / raz tygodniowo etc...).

Po zapisaniu danych w bazie backend jest gotowy do podjęcia konwersacji z użytkownikiem gdzie tematem jest dane url.

POST `/api/v1/conversation`

Endpoint wystawiony w celu podjęcia konwersacji na temat danego url. `url` przekazywane jest `url_path`.

Backend wykonuje API call do czatu GPT przesyłając mu content stron html w celu odnalezienia:

1. **Kluczowych informacji** - w zależności od typu strony może to być inny zestaw informacji. Dla stron firm usługowych będzie to adres, numer kontaktowy, oferta. Dla stron wydarzenia będzie

to opis wydarzenia, jego data, możliwość zapisu itp.

2. **Możliwych akcji** - Czat GPT rozpoznaje możliwe akcje do wykonania przez użytkownika na danej stronie.

Zwracana do frontendu jest wiadomość do wyświetlenia w czatbocie (**kluczowe informacje**) oraz lista propozycji jakie wiadomości może wysłać user (**możliwe akcje**).

Od tej pory dana konwersacja identyfikowana jest jednoznacznie przez parę `threadId` oraz `assistantId`. Dwójka ta zwracana jest przez endpoint w reponse i używana przez frontend do dalszej konwersacji.

Backend w tym czasie rozpoczął konwersację z czatem GPT, który ma załadowany kontekst ze stron html.

POST

`/api/v1/messages/thread/{threadId}/assistant/{assistantId}`

Endpoint używany do dalszej konwersacji.

Wiadomości od użytkownika zawierane są w body żądania HTTP.

W przyszłości do implementacji jest przesyłanie frontendowi akcji jakie powinien wykonywać (za pomocą funkcji `querySelector`).

Dokumentacja

Dokładna dokumentacja znajduje się pod adresem: <http://ec2-3-71-86-240.eu-central-1.compute.amazonaws.com:8080/swagger-ui/index.html#/message-controller/createMessage>.

Z racji ograniczonych zasobów maszyny rekomendujemy testowanie przy użyciu odpowiednio "małych" stron, takich jak:

- <http://www.cornerburger.pl/> (jedliśmy wczoraj, polecamy)*
- <https://megal.pl/>

*niestety nieco obciążyliśmy im stronę podczas hackatonu 😅