Dokumentacja projektu: Web File Downloader

Informacje o projekcie

Nazwa projektu: Web File Downloader
Autor: Pająk Piotr (numer albumu: 131483)
Wybrany projekt: 3. Rekurencyjne Pobieranie Plików
Kolegium: Kolegium Nauk Przyrodniczych, Uniwersytet Rzeszowski
Przedmiot: Sieci Komputerowe
Prowadzący: Mgr inż. Jarosław Szkoła
Rzeszów 2025

Spis treści

Opis

Aplikacja **Web File Downloader** to narzędzie do automatycznego pobierania plików o określonych rozszerzeniach z wybranych stron internetowych. Program obsługuje zarówno statyczne strony HTML, jak i dynamiczne treści generowane przez JavaScript, dzięki czemu jest wszechstronnym rozwiązaniem do zadań takich jak archiwizacja danych, pobieranie multimediów czy analiza zawartości stron internetowych. Wykorzystując połączenie asynchronicznego pobierania plików (z `aiohttp`) oraz dynamicznego renderowania treści przy użyciu `Selenium`, program skutecznie przetwarza zarówno klasyczne, jak i nowoczesne strony internetowe.

Aplikacja analizuje zawartość stron internetowych, wykrywając linki do plików, obrazy i inne zasoby. Wykorzystując odpowiednią mapę MIME, program pobiera pliki pasujące do zadanych rozszerzeń, co pozwala na precyzyjną selekcję danych do pobrania.

Wspierane rozszerzenia obejmują szeroki wachlarz formatów plików, w tym:

- **Obrazy**: `.png`, `.jpg`, `.gif`, `.bmp`, `.webp`, `.svg`, `.tiff`, `.ico`
- **Dokumenty**: `.json`, `.pdf`, `.docx`, `.xlsx`, `.txt`, `.xml`, `.yaml`
- **Multimedia**: `.mp3`, `.wav`, `.webm`, `.mp4`
- **Kod źródłowy**: `html`, `.js`, `.ts`, `.py`, `.java`, `.css`
- **Pliki archiwów**: `.zip`, `.rar`, `.tar.gz`

Dodatkowe funkcje aplikacji:

- **Pobieranie ciasteczek z popularnych przeglądarek** (Firefox, Chrome, LibreWolf) w celu umożliwienia obsługi uwierzytelnionych sesji,

- **Obsługa wielu jednoczesnych połączeń** z użyciem semaforów, co zwiększa wydajność i przyspiesza proces pobierania,
- **Wykrywanie i zapisywanie osadzonych elementów SVG** w przypadku, gdy rozszerzenie jest uwzględnione w zadanych kryteriach,
- **Ponawianie prób w przypadku nieudanych pobrań**, co zwiększa niezawodność aplikacji przy problemach z połączeniem lub czasem oczekiwania.

Dzięki zastosowaniu tych funkcji, aplikacja jest idealnym narzędziem do pobierania danych z różnorodnych stron internetowych, zarówno statycznych, jak i dynamicznych.

Zastosowane technologie

W projekcie zastosowano następujące technologie:

- **Python**: Główny język programowania, używany do implementacji logiki aplikacji, zarządzania asynchronicznymi operacjami oraz komunikacji z zewnętrznymi serwisami i plikami.
- **aiohttp**: Biblioteka do asynchronicznego wykonywania żądań HTTP, umożliwiająca równoczesne pobieranie zasobów z sieci. Dzięki niej możliwe jest szybkie pobieranie danych w tle, co przyspiesza proces przetwarzania wielu stron.
- **asyncio**: Wbudowana biblioteka do obsługi asynchronicznych operacji w Pythonie, która pozwala na zarządzanie równoległymi zadaniami i lepsze wykorzystanie zasobów systemowych.
- **Selenium**: Narzędzie do automatyzacji przeglądarek internetowych, wykorzystywane do dynamicznego renderowania stron i interakcji z elementami JavaScript. Dzięki Selenium możliwe jest pozyskiwanie treści z witryn, które wymagają interakcji lub dynamicznego ładowania danych.
- **Chromedriver**: Sterownik do przeglądarki Google Chrome, niezbędny do prawidłowego działania Selenium z przeglądarkami opartymi na Chromium. W projekcie użyto wersji 132.0.6834.83.
- **lxml**: Biblioteka do przetwarzania i manipulacji dokumentami XML/HTML. Używana do parsowania treści stron internetowych, w tym ekstrakcji osadzonych elementów SVG i przetwarzania dokumentów HTML.
- **browser-cookie3**: Biblioteka służąca do pobierania ciasteczek z przeglądarek internetowych, takich jak Chrome, Firefox czy LibreWolf. Pozwala to na zachowanie sesji użytkownika i uwierzytelnienie podczas interakcji z dynamicznymi stronami.
- **argparse**: Biblioteka do parsowania argumentów wiersza poleceń, umożliwiająca konfigurację ścieżek, rozszerzeń plików i innych parametrów wykonawczych aplikacji bez konieczności zmiany kodu.
- **hashlib**: Biblioteka do tworzenia funkcji skrótu, używana do generowania unikalnych identyfikatorów (hashy) dla URL, aby zapobiec wielokrotnemu pobieraniu tych samych plików.
- **uuid**: Biblioteka do generowania unikalnych identyfikatorów, wykorzystywana do nadawania plikom unikalnych nazw, gdy brakuje jednoznacznej nazwy.
- **random**: Moduł do generowania losowych liczb, stosowany w projekcie do dodania losowego opóźnienia między operacjami sieciowymi, co pomaga unikać detekcji przez systemy ochrony przed botami.

- **re**: Moduł do pracy z wyrażeniami regularnymi, używany do wyodrębniania linków z treści stron HTML, zarówno statycznych, jak i dynamicznych.

Wykorzystanie technologii:

- **Selenium + Chromedriver**: Używane do renderowania stron wymagających JavaScript i uzyskiwania dynamicznych zasobów, takich jak obrazy, linki, czy elementy osadzone w SVG. Dzięki temu aplikacja może pobierać treści z witryn, które nie ładują się poprawnie w klasyczny sposób.
- **aiohttp + asyncio**: Te dwie biblioteki zapewniają asynchroniczność i wydajność operacji sieciowych. Dzięki im aplikacja może równolegle pobierać wiele plików i przetwarzać strony, co znacznie przyspiesza cały proces.
- **browser-cookie3**: Umożliwia pobieranie ciasteczek z przeglądarek. Dzięki temu aplikacja może działać w kontekście aktywnej sesji użytkownika.

Najważniejsze elementy projektu

1. Mapowanie rozszerzeń plików na typy MIME

Mapa `extension_map` pozwala na jednoznaczne przypisanie rozszerzenia pliku do jego typu MIME. Dzięki temu aplikacja może rozpoznać, które pliki należy pobrać, a które pominąć. Mapa obejmuje szeroką gamę typów plików, od obrazów, przez dokumenty, po audio i wideo, co czyni projekt elastycznym i zdolnym do obsługi różnorodnych zasobów.

2. Dynamiczne renderowanie stron

Funkcja `get_dynamic_page_content` wykorzystuje Selenium oraz Chromedriver do pobierania zawartości stron, które generują treść za pomocą JavaScript. Dzięki temu możliwe jest analizowanie linków, obrazów oraz innych zasobów, które są ładowane dynamicznie. Jest to szczególnie przydatne na stronach, które wymagają interakcji z użytkownikiem lub pełnego wsparcia dla nowoczesnych technologii webowych.

3. Pobieranie plików asynchronicznie

Funkcja `download_with_retry` zapewnia niezawodne pobieranie plików z obsługą ponawiania prób w razie wystąpienia błędów sieciowych. Dzięki zastosowaniu `aiohttp` oraz `asyncio`, aplikacja umożliwia równoległe pobieranie wielu plików, co znacząco przyspiesza proces. Obsługa ponawiania prób sprawia, że proces jest odporny na chwilowe problemy z połączeniem.

4. Obsługa ciasteczek

Funkcja `get_cookies_from_browser` umożliwia pobieranie ciasteczek z przeglądarek (Chrome, Firefox, LibreWolf). Jest to szczególnie przydatne przy pobieraniu zawartości stron wymagających sesji użytkownika.

5. Rekurencyjne przetwarzanie stron

Funkcja `process_page` pozwala na przetwarzanie stron do określonej głębokości, pobieranie plików oraz przechodzenie po znalezionych linkach. Zastosowanie rekurencji umożliwia skuteczne zbieranie zasobów z witryn wielostronicowych. Dodatkowo, mechanizm kontrolowania głębokości przetwarzania zapewnia, że aplikacja nie przechodzi zbyt głęboko w strukturę strony, co zapobiega nadmiernym obciążeniom serwera i niekontrolowanemu zbieraniu zasobów.

6. Wykrywanie pętli nieskończonych

Funkcja `is_symlink_loop` wykrywa pętle symlinków w URL-ach. Przy pomocy `os.path.realpath` identyfikuje rzeczywiste ścieżki odwiedzanych zasobów i porównuje je z wcześniej zapisanymi ścieżkami. Jeśli dany symlink prowadzi do już odwiedzonej ścieżki, funkcja przerywa przetwarzanie danego URL-a, zapobiegając nieskończonym pętlom.

7. Tworzenie katalogu, jeśli nie istnieje

Funkcja `ensure_directory` sprawdza, czy katalog, do którego mają być zapisywane pliki, istnieje. Jeśli katalog nie istnieje, funkcja automatycznie go tworzy. Dzięki temu unikamy błędów związanych z brakiem wymaganej struktury folderów, co pozwala na płynne i automatyczne zapisywanie pobranych plików.

8. Ekstrakcja i zapis osadzonych plików SVG

Funkcja `extract_and_save_svgs` służy do wykrywania i zapisywania osadzonych plików SVG w treści HTML. Jeśli rozszerzenie `svg` jest uwzględnione w mapie rozszerzeń, funkcja przeanalizuje zawartość strony i zapisze wszystkie znalezione elementy SVG do pliku. To istotne, gdy strona zawiera wewnętrzne grafiki SVG, które nie są linkami do zewnętrznych plików.

9. Pobieranie pliku z odpowiednimi nagłówkami i ponawianie prób

Funkcja `download_file` odpowiada za pobieranie plików z danego URL. Sprawdza, czy dany plik był już wcześniej pobrany, aby uniknąć duplikatów. Następnie pobiera plik, rozpoznaje jego MIME type i zapisuje go do odpowiedniego katalogu, dodając odpowiednie rozszerzenie na podstawie typu MIME. Jeśli wystąpi błąd podczas pobierania, funkcja obsługuje ponawianie prób, aby zapewnić niezawodne pobieranie plików.

10. Sprawdzanie typu MIME pliku

Funkcja `get_mime_type` odpowiada za określenie typu MIME pliku na podstawie nagłówka HTTP. Wykorzystuje `aiohttp` do wykonania zapytania HEAD do serwera i analizy nagłówka `Content-Type`, który wskazuje na typ pliku. Jest to kluczowe do określenia, jakie rozszerzenie przypisać do pobranego pliku i czy należy go w ogóle pobierać.

Przykłady działania

1. Podstawowe użycie

Pobieranie obrazów JPEG i PNG z danej strony:

```
python web-downloader.py --url https://example.com --extensions "*.jpg|*.png" --output downloads
```

2. Użycie ciasteczek z przeglądarki

Pobieranie plików PDF z wykorzystaniem ciasteczek z Librewolf do domyślnego folderu pobierania:

```
python web-downloader.py --url https://example.com --extensions "*.pdf" --cookies-from-browser librewolf
```

3. Ograniczenie głębokości rekursji

Pobieranie plików tekstowych do głębokości 2:

```
python web-downloader.py --url https://example.com --extensions "*.txt" --max-depth 2
```

4. Zwiększenie liczby jednoczesnych połączeń

Przetwarzanie z maksymalnie 10 jednoczesnymi połączeniami:

```
python web-downloader.py --url https://example.com --extensions "*.jpg|*.png" --max-workers 10
```

Wnioski

Program efektywnie łączy podejście asynchroniczne i dynamiczne renderowanie stron, co pozwala na pobieranie różnych zasobów z internetu w sposób równoległy i wydajny. Zastosowanie biblioteki `aiohttp` umożliwia asynchroniczne pobieranie plików, a wykorzystanie `Selenium` pozwala na obsługę dynamicznych treści, które mogą być ładowane po początkowym załadowaniu strony. Dodatkowo, mechanizm ponawiania prób przy błędach i zarządzanie równoległymi połączeniami za pomocą semaforów zwiększają niezawodność i efektywność programu. Integracja z przeglądarkami pozwala na pobieranie ciasteczek w celu obsługi uwierzytelnionych sesji, co czyni narzędzie bardziej uniwersalnym.