6.05.2022

- Wybierz stronę WWW zawierającą jakąś potencjalnie potrzebną informację (aktualna temperatura w Krakowie, kurs dolara, itp.), a następnie napisz program ściągający tę stronę i wyłuskujący z niej te dane.

  Możesz użyć dowolnego języka programowania i bibliotek. Oprócz bibliotek do wykonywania zapytań HTTP warto użyć parsera dokumentów HTML, np. *Beautiful Soup* (Python) lub *JSoup* (Java).

  Program powinien obsługiwać możliwe błędy na wszystkich poziomach, np.

  - błędy przewidziane przez protokół HTTP zasygnalizowane przez serwer (np. zasób niedostępny),
  - niedostępność samego serwera lub niedostępność sieci,
  - niemożliwość odnalezienia właściwych danych na pobranej stronie.

  Proszę również sprawdzać, czy pobrana strona rzeczywiście jest tą spodziewaną, a nie np. stroną z komunikatem o wewnętrznym błędzie serwera WWW.

  Przydatny zasób - lista kodów zwrotnych zdefiniowanych w protokole HTTP (Zastanów się, które z nich mogą wystąpić podczas próby automatycznego pobrania strony?):

  - w dokumencie RFC dla HTTP 1.1
  - lub na stronach MDN

  Uwaga: Rozwiązanie tego zadania proszę przesłać przez formularz w MS Teams do czwartku 19 maja. Rozwiązanie powinno zawierać program wraz z plikami potrzebnymi do jego zbudowania i wykonania, a także plik README z krótkim opisem tego, jaka strona została wybrana i jakie informacje są z niej wyłuskiwane.

Dodatkowe uwagi:

- Takie automatyczne pozyskiwanie informacji ze stron WWW jest podstawą techniki zwanej *"web scraping"*. Pobieranie zawartości stron WWW przy pomocy programu może łatwo prowadzić do obciążenia serwera HTTP znacznie większego niż generowane przez przeglądarki internetowe. Przy ręcznym testowaniu nie powinno to być problemem, należy jednak unikać tworzenia programów wysyłających zapytania co sekundę itp.

- W celu powstrzymania automatycznego pobierania zasobów stron WWW tam, gdzie nie jest to pożądane, istnieje konwencja definiowania które zasoby mogą być przeglądane przez "roboty" a które nie, w pliku `robots.txt`. Bardzo możliwe, że konwencja ta niedługo stanie się standardem, dlatego warto się z nią zapoznać:

  - https://www.robotstxt.org/
  - propozycja przyszłego standardu

- Zapoznaj się również z uwagami Wykładowcy do analogicznego zadania:

  > Wyłuskiwanie danych ze strony HTML jest dość kruchą techniką, bo witryna może nieoczekiwanie (dla nas) zmienić wygląd bądź zawartość. Biorąc to pod uwagę łatwo zauważyć, że podejście typu „zwróć bajty od 5078 do 5081” jest skazane na rychłą porażkę; „zwróć zawartość czwartego elementu `<p>` znajdującego się wewnątrz elementu `<div>` o identyfikatorze »temp«” jest lepsze. Warto postarać się o to, aby program zauważał nieoczekiwane bądź podejrzane sytuacje i je zgłaszał (np. jeśli w tym czwartym `<p>` jest ciąg znaków nie będący liczbą, to raczej nie jest to temperatura; jeśli znaleziona liczba wykracza poza przedział [-30, 40] to raczej nie jest to temperatura w stopniach Celsjusza).

Nawigację po treści strony ułatwia zbudowanie drzewa obiektów reprezentujących elementy strony HTML; ponownie polecam Waszej uwadze bibliotekę Beautiful Soup (Python) i jej odpowiednik JSoup (Java). Do sprawdzania czy łańcuch znaków pasuje do zadanego wzorca dobrze nadają się wyrażenia regularne.

Może się zdarzyć, że traficie na stronę, która wewnątrz przeglądarki wyświetla potrzebne nam informacje, ale gdy się ją ściągnie przez HTTP to nigdzie w jej treści nie można ich znaleźć. Prawie na pewno przyczyną jest jej dynamiczna, AJAX-owa natura. Takie strony mają puste miejsca, wypełniane zawartością ściąganą przez javascriptowy kod z innych URL-i.

Współczesne przeglądarki mają wbudowane narzędzia deweloperskie, jednym z nich jest analizator połączeń sieciowych. Można przy jego pomocy spróbować odnaleźć rzeczywiste źródło potrzebnych nam danych. Poszukiwania proponuję zacząć od połączeń zainicjowanych przez XHR (czyli javascriptową klasę XMLHttpRequest) i/lub tych, które ściągały dokumenty typu JSON.

Udane znalezienie JSON-owego (lub XML-owego) źródła surowych danych jest dobrą wiadomością, bo ryzyko zmiany formatu tych danych jest znacznie mniejsze niż ryzyko zmiany struktury strony HTML.

- Proszę zapoznać się z listą kodów zwrotnych zdefiniowanych w protokole HTTP:
  - w dokumencie RFC dla HTTP 1.1
  - lub na stronach MDN

  Zastanów się, które z nich mogą się pojawić przy próbie pobrania strony w HTML za pomocą prostego zapytania GET.