

# Generator ruchu Google Analytics

## Iteracja I architektura systemu

Bartłomiej Dalak      Bartłomiej Karwowski  
Bartosz Gromek      Tomasz Kanas

10 kwietnia 2018

### Wstęp

Dokument architektury systemu ma na celu przedstawienie wizji architektury. Opisana architektura może ulec zmianom w fazie implementacji.

### Opis elementów architektury

#### Aplikacja

Aplikacją w naszym przypadku będzie skrypt “generate\_ga\_traffic.py”, generujący ruch według podanych przez użytkownika wartości. Będzie on uruchamiany przez konsolę.

#### Język

Wykorzystany zostanie Python w wersji 3.6.

#### Użyte biblioteki

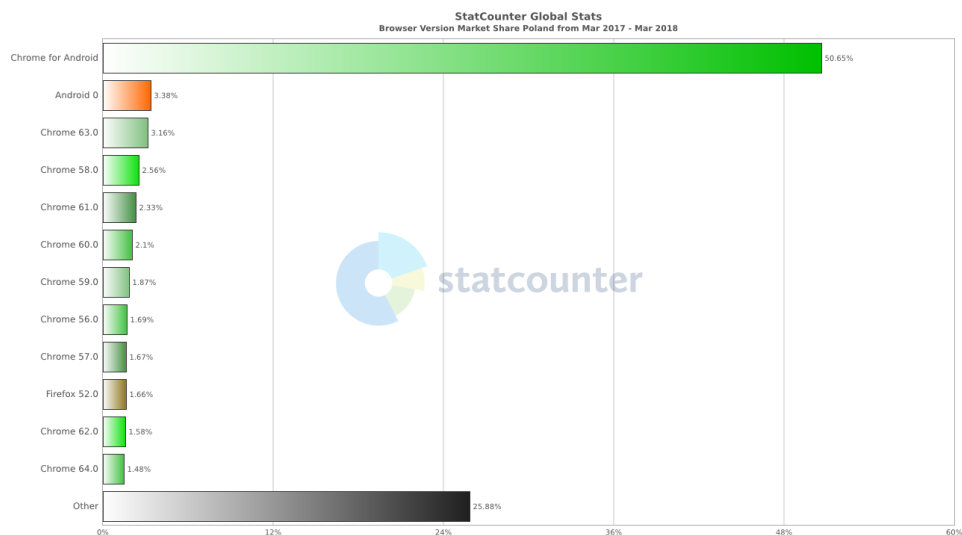
- requests: wysyłanie zapytań do GA
- pandas: do obsługi pliku ‘browser.csv’, w którym mamy rozkład przeglądarek na terenie Polski.

#### send\_requests\_api

API służące do komunikacji z GA, generuje potrzebne dane oraz je wysyła.

## Metody

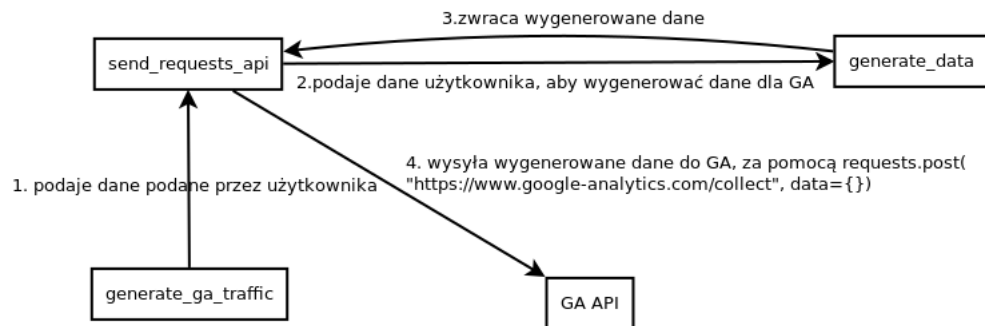
- `send(tracking_id, url, visits_no, time)`: sprawdza czy dane użytkownika mogą zostać wygenerowane, poprzez sprawdzanie czy dane są w poprawnym zakresie i mają poprawny format, oraz wysłanie zapytania do Google'owego weryfikacyjnego API, w celu zweryfikowania poprawności TrackingID oraz adresu strony. Jeśli dane przejdą tą weryfikację to przekazuje je do niżej opisanej funkcji `generate_data`. Po ponownym ich odebraniu, próbuje przesłać je bezpośrednio do GA. Po otrzymaniu (lub nie) odpowiedzi zwraca jeden z 3 możliwych kodów WRONG DATA, OK, CONNECTION PROBLEM.
- `generate_data(visits_no)`: metoda wołana przez `send()`, generujące odpowiednie dane do wysłania. Po odebraniu informacji przekazanych przez użytkownika, do odpowiedniej ilości zapytań przypisuje dane przygotowane z wiarygodnym rozkładem. Informacje do tego potrzebne zostaną czytane z pliku 'browser.csv', który zostanie pobrany ze strony GlobalStats StatCounter (dane dot. oprogramowania użytkowników witryny). Wykres użytkowania przeglądarek na terenie Polski.



## Schemat działania

### Generowanie i wysyłanie danych

Dane otrzymane przez użytkownika zostaną przekazane do napisanego przez nas API: `send_requests_api`. API wygeneruje zestaw wejść, które będą wysyłane do GA, a następnie za pomocą metody `post` z biblioteki `requests` wyśle je.



## Wykrywanie braku połączenia z internetem

Jeśli `send_requests_api` przez 10 minut nie uda się wysłać żadnego requesta do `GA API`, w takim wypadku zostaje zatrzymane wysyłanie oraz zwrócony do `generate_ga_traffic` komunikat `CONNECTION_PROBLEM`. Wtedy użytkownik zostaje o tym poinformowany i może wybrać jedną z trzech możliwości co chce robić dalej.