

# System do analizy zachowania użytkowników na stronach WWW

*Kacper Łukawski*

Zrealizowany projekt jest aplikacją pozwalającą na śledzenie akcji wykonywanych przez użytkowników odwiedzających stronę internetową. Zgromadzone dane służą do stworzenia grafu i możliwa jest ich dalsza analiza z użyciem algorytmów grafowych.

## Konfiguracja serwera

Serwer aplikacji został stworzony z użyciem technologii node.js oraz bazy danych neo4j. Z tego powodu, do jego uruchomienia niezbędne jest posiadanie node.js (<http://nodejs.org/>) w wersji 0.10.24 lub nowszej, npm (<https://www.npmjs.org/>) oraz JRE6 (<http://www.oracle.com/technetwork/java/javase/downloads/jre6-downloads-1637595.html>). Po zainstalowaniu wszystkich niezbędnych zależności należy, z głównego katalogu aplikacji, uruchomić polecenie `npm install`. Dzięki temu zainstalujemy wszystkie niezbędne pakiety node.js. Do uruchomienia aplikacji potrzebne będzie wywołanie komendy `node app.js`. Domyślnie aplikacja wystartuje serwis śledzący na porcie 3000.

## Integracja z serwisem WWW

Do uruchomienia systemu na wybranej stronie internetowej służy dostarczona biblioteka JavaScript - *ba-lib.js*, którą należy dołączyć w sekcji `<head>` dokumentu HTML.

```
<script type="text/javascript" src="ba-lib.js"></script>
<script type="text/javascript">
    ba.setTrackingUrl('//www.example.com:3000/track'); // adres pod którym został
                                                       // zainstalowany serwer
</script>
```

Wszystkie zdarzenia, które chcemy śledzić muszą wysyłać żądania do serwera korzystając z metody dostarczanej przez bibliotekę JavaScript. Służy do tego metoda:

```
ba.trackEvent(eventType, parameters)
```

Pierwszy parametr to nazwa śledzonego zdarzenia, a drugi jest mapą parametrów opisujących aktualne zdarzenie. Całość będzie stanowił pojedynczy wierzchołek grafu, jaki zostanie stworzony przez aplikację, i który będzie można następnie analizować.

Poniższy przykład ilustruje śledzenie wszystkich kliknięć w odnośniki (do prawidłowego działania wymagana jest także biblioteka jQuery).

```
<script type="text/javascript">
    $(document).ready(function(){
        $('a').click(function(){
```

```

        ba.trackEvent('link_click', {
            'href': $(this).attr('href')
        });
    });
});
</script>

```

## Komunikacja z API

API systemu jest oparte o protokół HTTP i format JSON. Dostępne metody:

metoda HTTP	adres	komentarz
GET	/api/all	Pobranie listy wszystkich wierzchołków i połączeń grafu
GET	/api/cycles/:eventId	Pobranie wszystkich cykli przebiegających przez wybrany wierzchołek
GET	/api/shortestPath/:startEventId/:endEventId	Pobranie najkrótszej (w sensie najmniejszej liczby wierzchołków odwiedzonych) ścieżki prowadzącej pomiędzy wybranymi wierzchołkami
GET	/api/mostProbableMove/:eventId	Wybiera najbardziej prawdopodobną akcję jaką wykona użytkownik po wykonaniu wskazanego zdarzenia
GET	/api/mostProbablePath/:startEventId/:endEventId	Pobranie najbardziej prawdopodobnej ścieżki prowadzącej między wybranymi wierzchołkami

## Demo systemu

Zasadniczą częścią stworzonej aplikacji jest API, które pozwala na analizę danych i ich interpretację przez zewnętrzne systemy. Aby umożliwić prosty podgląd danych, jakie są przechowywane, stworzona została prosta aplikacja webowa dostępna pod tym samym adresem, pod którym uruchomiliśmy serwer agregujący (domyślnie <http://localhost:3000/>).

Domyślnie otrzymamy graf prezentujący wszystkie zdarzenia, które zostały zarejestrowane. Po kliknięciu na wybrany wierzchołek otrzymamy listę wszystkich możliwych opcji. Najechanie na krawędź umożliwia podgląd przechowywanych prawdopodobieństw oraz ilości przejść z wierzchołka A do B.

