

# Twitter with R

*Piotr Smuda*

*7 listopada 2015*

## Stworzenie aplikacji Twitter

Aby stworzyć aplikację Twitter wchodzimy na stronę: <https://apps.twitter.com/>.

Będąc na stronie logujemy się na swoje konto, a następnie klikamy na przycisk *Create New App*.

## Twitter Apps

Create New App

W kolejnym oknie podajemy kolejno nazwę, opis i stronę aplikacji, a następnie akceptujemy regulamin i klikamy *Create your Twitter application*:

## Create an application

**Application Details**  
**Name \***  
  
Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.  
**Description \***  
  
Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.  
**Website \***  
  

AS OF THE FIRST DATE THAT YOU USE THE LICENSED MATERIAL ("EFFECTIVE DATE").

IF YOU ARE AN INDIVIDUAL REPRESENTING AN ENTITY, YOU ACKNOWLEDGE THAT YOU HAVE THE APPROPRIATE AUTHORITY TO ACCEPT THIS AGREEMENT ON BEHALF OF SUCH ENTITY. YOU MAY NOT USE THE LICENSED MATERIAL AND MAY NOT ACCEPT THIS AGREEMENT IF YOU ARE NOT OF LEGAL AGE TO FORM A BINDING CONTRACT WITH TWITTER, OR YOU ARE

☒ Yes, I agree

Create your Twitter application

W ten sposób stworzyliśmy aplikację Twitter:

## tweetsDS

Test OAuth

Details

Settings

Keys and Access Tokens

Permissions



tweets download

<https://github.com/smudap>

## Konfiguracja aplikacji Twitter pod R

Na początku zaczynamy od instalacji pakietu *ROAuth* z *CRANu*.

```
install.packages("ROAuth")  
library("ROAuth")
```

### Permanentny dostęp

Po wczytaniu pakietu wchodzimy na stronę aplikacji i wybieramy zakładkę *Keys and Access Tokens*. Kopiujemy *Consumer Key* oraz *Consumer Secret* i przypisujemy je do jakichś zmiennych, np. *consumer\_key* oraz *consumer\_secret*. Następnie klikamy w przycisk *Create my access token* w sekcji *Your Access Token* i zapisujemy *Access Token* oraz *Access Token Secret* pod jakimiś zmiennymi, np. *oauth\_key* oraz *oauth\_secret*.

Details

Settings

Keys and Access Tokens

Permissions

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)

Consumer Secret (API Secret)

### Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

### Token Actions

Create my access token

## Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token

Access Token Secret

Oprócz tych zmiennych potrzebujemy jeszcze stworzyć zmienne z adresami stron internetowych dla tokenów i autoryzacji:

- `https://api.twitter.com/oauth/request_token`
- `https://api.twitter.com/oauth/access_token`
- `https://api.twitter.com/oauth/authorize`

Kod R do stworzenia dotychczasowych zmiennych wygląda następująco:

```
request_URL <- "https://api.twitter.com/oauth/request_token"
access_URL <- "https://api.twitter.com/oauth/access_token"
auth_URL <- "https://api.twitter.com/oauth/authorize"
consumer_key <- "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
consumer_secret <- "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
oauth_key <- "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
oauth_secret <- "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

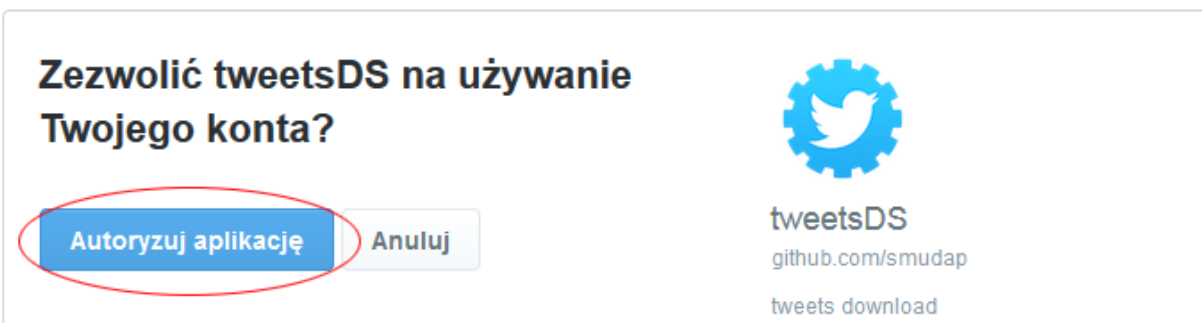
Potem tworzymy jakąś zmienną środowiskową służącą za połączenia z naszą aplikacją:

```
tw_oauth <- OAuthFactory$new(consumerKey = consumer_key, consumerSecret = consumer_secret,
                             oauthKey = oauth_key, oauthSecret = oauth_secret, requestURL = request_URL,
                             accessURL = access_URL, authURL = auth_URL)
```

Następnie autoryzujemy manualnie poprzez kod:

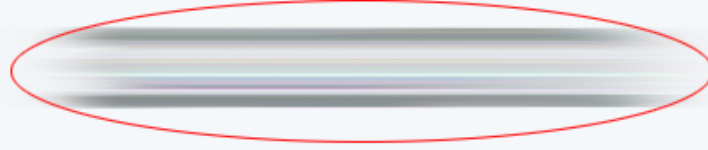
```
tw_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package="RCurl"))
```

Zostaniemy przeniesieni na stronę internetową, gdzie będziemy musieli zaautoryzować aplikację. Klikamy *Autoryzuj aplikację* i kopiujemy przypisany nam *код PIN* do konsoli w RStudio:



Zezwolono na dostęp do tweetsDS!

Następnie wróć do tweetsDS i wprowadź ten kod PIN aby zakończyć proces autoryzacji:



```
> tw_oauth$handshake(cainfo = system.file("curlSSL", "cacert.pem", package="RCurl"))  
To enable the connection, please direct your web browser to:  
https://api.twitter.com/oauth/authorize?oauth\_token=...  
when complete, record the PIN given to you and provide it here:
```

W tym momencie autoryzacja została zakończona i możemy już pobierać tweety. Ale jeśli nie chcemy przechodzić przez proces autoryzacji za każdym razem, jak chcemy coś pobrać z Twittera, to musimy zapisać zmienną `tw_oauth` do pliku, który wystarczy później wczytać by mieć dokonaną autoryzację. Służy do tego np. kod:

```
save(tw_oauth, file = paste0(getwd(), "/tw_oauth"))  
load(paste0(getwd(), "/tw_oauth"))
```

## Pobieranie tweetów

Na początku zaczynamy od instalacji pakietu *streamR* z *CRANu*.

```
install.packages("streamR")  
library("streamR")
```

Pakiet ten umożliwia pobieranie tweetów w formacie *json* i parsowanie ich do R. Kod służący do pobierania, to:

```
filterStream(file.name = NULL, track = NULL,  
             follow = NULL, locations = NULL, language = NULL,  
             timeout = 0, tweets = NULL, oauth = NULL,  
             verbose = TRUE)
```

Gdzie:

- `file.name` - nazwa/ścieżka pliku, gdzie tweety mają być zapisane
- `track` - napis lub wektor słów, po których będą wyłuskwane tweety
- `follow` - wektor z IDkami użytkowników, których chcemy pobrać dane
- `locations` - wektor z szerokością i długością geograficzną, które nas interesują
- `language` - język w jakim zostały napisane tweety (format BCP 47)
- `timeout` - długość pobierania tweetów w sekundach
- `tweets` - maksymalna liczba tweetów, która ma zostać pobrana
- `oauth` - zmienna służąca do połączenia z bazą
- `verbose` - czy ma być wyświetlona informacja o pobranych tweetach

Przykładowy kod do pobrania danych o polityce przez 5 minut:

```
filterStream(file = paste0("tweety-politycznie-", Sys.Date(), ".json"),
  track = c("pis, po, kukiz, korwin, partia razem,
    zjednoczona lewica, psl, petru, nowoczesna"),
  timeout = 5*60, oauth = tw_oauth, language = "pl")
```

Natomiast kod służący do parsowania tweetów do Ra do ramki danych to:

```
parseTweets(tweets, simplify = FALSE, verbose = TRUE)
```

Gdzie:

- tweets - nazwa/ścieżka pliku z tweetami
- simplify - czy ma zostać zwrócona ramka danych bez informacji geograficznych itp.
- verbose - czy ma być wyświetlona informacja o sparsowanych tweetach

Przykładowy kod do sparsowania tweetów:

```
parsed_tweets <- parseTweets(paste0("tweety-politycznie-", Sys.Date(), ".json"),
  simplify = FALSE, verbose = TRUE)
```