



Tymoteusz Makowski
Tomasz Klonecki

<https://tmakowski.github.io/senderR>

Plan prezentacji

1. Motywacja do zrobienia nowych funkcjonalności.
2. Działanie ulepszonej wersji pakietu:
 - 2.1. Wrapper do klasycznego *lapply*.
 - 2.2. Ustawienie domyślnych wartości argumentów w kliencie.
 - 2.3. Pasek postępu dla dowolnych funkcji
3. Plany na przyszłość

Motywacja

Jak było	Jak jest
<ul style="list-style-type: none">W funkcji <code>send_message</code>, za każdym razem trzeba było wpisywać wszystkie argumenty.	<ul style="list-style-type: none">Po stworzeniu (lub w trakcie tworzenia) <code>clienta</code>, możemy ustawić domyślne argumenty i wywoływać funkcję <code>send_message</code> tylko z np. jednym argumentem
<ul style="list-style-type: none">Korzystając z funkcji <code>lapply</code> i <code>txtProgressBar</code> w bazowym R mogliśmy wysłać wiadomość w osobnej funkcji.	<ul style="list-style-type: none">Stworzyliśmy wrappery do funkcji <code>lapply</code> i <code>txtProgressBar</code>, które pozwalają na wykorzystanie tejże funkcji i wysłania wiadomości w jednej linijce.
<ul style="list-style-type: none">Mała ilość dependencji	<ul style="list-style-type: none">Mała ilość dependencji!

Przykłady działania set_fields()

Jak było:

```
client <- client_telegram("my_token")

send_message(client, message = "Hello world!", destination = 12345)

send_message(client, message = "Hello world!", destination = 12345)
```

Jak jest:

```
client <- client_telegram("my_token", destination = 12345)

client <- set_fields(client, message = "Hello world!")

send_message(client)

send_message(client, message = "Hi world!")
```

Przykłady działania sendeR_lapply()

Jak było:

```
client <- client_telegram("my_token")

my_func <- function(x, text) { print(paste(text, x)); Sys.sleep(x) }

lapply(client, 1:10, my_func, "Sleeping:")

send_message(client, message = "sendeR_lapply done!", destination = 12345)
```

Jak jest:

```
client <- client_telegram("my_token")

my_func <- function(x, text) { print(paste(text, x)); Sys.sleep(x) }

sendeR_lapply(client, 1:10, my_func, "Sleeping:", destination = 12345,

               message = "sendeR_lapply done!")
```

Przykłady działania sender_txtProgressBar()

Jak było:

```
pb <- txtProgressBar(...)
For (i in 1:10) {
  Sys.sleep(0.5);
  setTxtProgressBar(pb, i / 10)
}
send_message(client, message = "Hello world!", destination = 12345)
```

Jak jest:

```
client <- client_telegram("my_token", message = "Progress bar has finished!", destination = 12345)
pb <- sender_txtProgressBar(client)

for (i in 1:10) {
  Sys.sleep(0.5)
  setTxtProgressBar(pb, i / 10)
}
```

Plany na przyszłość

- Akceptacja pakietu przez CRAN.
- Stworzenie klientów dla większej ilości komunikatorów.
- Tworzenie dodatkowych wrapperów w miarę potrzeb użytkowników.

Dziękujemy za uwagę

```
library(devtools)
```

```
install_github("tmakowski/sendR")
```