

academictwitterR Twitch talk

Christopher Barrie and Justin Chun-ting Ho

2021-05-20

**Intro**

# Structure of intro. talk

- Installation and authorization
- Data storage
- Convenience functions
- Query building

# Installation and authorization

```
library(academictwitteR)
```

OR

```
devtools::install_github("cjbarrie/academictwitteR", build_vignettes = T)
```

Authorization instructions available as **CRAN vignette**.

## Consumer Keys ⓘ

API Key and Secret

Regenerate

## Authentication Tokens ⓘ

Bearer Token

*Generated April 19, 2021*

Regenerate

Revoke



## Helpful docs

[How to use projects](#)

[App permissions](#)

[Authentication overview](#)

[Authentication best practices](#)

# Installation and authorization

```
bearer_token <- "" # Insert bearer token
```

# Data storage

1. Storage in .rds format
2. Returned as data.frame object

```
tweets <-  
  get_all_tweets(  
    "#BLM OR #BlackLivesMatter",  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token,  
    file = "blmtweets"  
  )
```

# Data storage

1. Storage in JSON format
2. In named directory
3. Recovery functions
4. Bind functions

*# Note: not passing to an object because nothing returned*

```
get_all_tweets(  
    "#BLM OR #BlackLivesMatter",  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token,  
    data_path = "data/"  
    bind_tweets = FALSE  
)
```



# Data storage

```
resume_collection(data_path = "data", bearer_token)
```

# Data storage

```
update_collection(data_path = "data", end_tweets = "2020-01-10T00:00:00Z", bearer_token)
```

# Data storage

```
tweets <- bind_tweet_jsons(data_path = "data/")
```

# Main functions

## Getting tweets

```
tweets <-  
  get_all_tweets(  
    "#BLM OR #BlackLivesMatter",  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token,  
    data_path = "data/"  
    bind_tweets = FALSE  
  )
```

# Main functions

## Getting user tweets

```
users <- c("TwitterDev", "jack")
tweets <-
  get_user_tweets(
    users,
    "2010-01-01T00:00:00Z",
    "2020-01-01T00:00:00Z",
    bearer_token,
    data_path = "data/"
    bind_tweets = FALSE
  )
```

# Convenience functions

## Getting image tweets

```
tweets <-  
  get_image_tweets(  
    "#BLM OR #BlackLivesMatter",  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token,  
    data_path = "data/"  
    bind_tweets = FALSE  
  )
```

# Convenience functions

## Getting video tweets

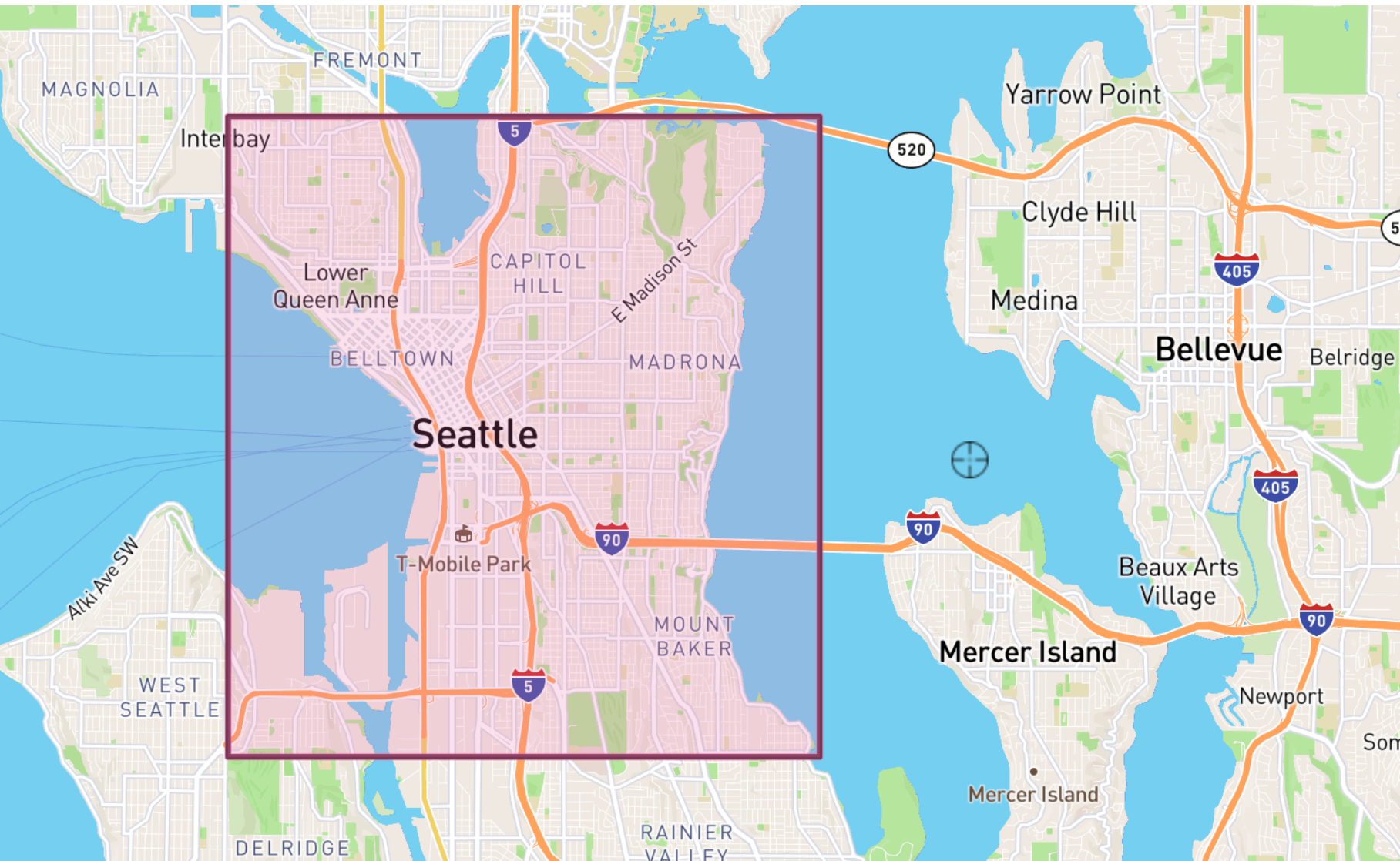
```
tweets <-  
  get_video_tweets(  
    "#BLM OR #BlackLivesMatter",  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token,  
    data_path = "data/"  
    bind_tweets = FALSE  
  )
```

# Query building

## Building manually

```
tweets <-  
  get_all_tweets(  
    "#BLM OR #BlackLivesMatter place_country:US lang:en -is:retweet has:images",  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token  
  )
```





# Query building

## Building manually

```
tweets <-  
  get_all_tweets(  
    "#BLM OR #BlackLivesMatter bounding_box:[-122.375679 47.563554 -122.266159 47.643417] lang:e  
    "2020-01-01T00:00:00Z",  
    "2020-01-05T00:00:00Z",  
    bearer_token  
  )
```

# Query building

## Using query builder

```
query <- build_query(query = "#BLM", geo_query = TRUE)
```

Which geo buffer type do you want?

- 1: Point radius
- 2: Bounding box

Selection: 2

What is west longitude? -122.375679

What is south latitude? 47.563554

What is east longitude? -122.266159

What is north latitude? 47.643417

```
query
```

```
## [1] "#BLM -is:retweet bounding_box:[-122.375679 47.563554 -122.266159 47.643417] has:images lang:en"
```