OpenStreetMap API

Mauricio Eduardo Landim

2024-04-19

Contents

Introdução **1**

Esquema do Banco de Dados **1**

API's do OpenStreetMaps **2**
API de Dados (RESTful) 3
API Nominatim 4
API Overpass 5
API de Estilos 7

Uso de caso para API RESTful **8**

Uso de caso da Overpass API **10**

Introdução

O intuito deste texto é explorar a variedade de dados que podemos requisitar a partir da API do OpenStreetMap.

Esquema do Banco de Dados

Nesta primeira seção gostaria de expor o esquema do banco de dados do OpenStreetMaps e seus respectivos elementos. Podemos ter um entendimento melhor do esquema do banco de dados através do link (https://wiki.openstreetmap.org/wiki/Openstreetmap-website/Database_schema) e pela imagem (https://wiki.openstreetmap.org/w/images/1/10/ERD_of_OSM_DB.svg). O OpenStreetMaps tem 3 tipos de dados primitivos, que seriam os nodes, ways e relations. Um melhor detalhamento sobre estes elementos pode ser visto no link (https://wiki.openstreetmap.org/wiki/Elements). Além disso, existem tabelas que são relacionadas as experiencias do usuário e suas atividades dentro da plataforma do OSM, assim como a users, changesets, gpx, notes, diary_entries. As demais tabelas dizem respeito as atividades de back-end e funções de desenvolvedores.

Dentro do esquema temos tabelas pai e tabelas filhos, estas segundas que desempenham funções suplementares as funções da tabela pai.

Tabelas principais:

1. users (https://wiki.openstreetmap.org/wiki/User)

- user_tokens
- user_roles
- user_blocks
- friends

- messages
- reports
- issues
- issue_comments

2. changesets (https://wiki.openstreetmap.org/wiki/Changeset)

- changesets_subscribers
- changesets_comments
- changesets_tags

3. nodes (https://wiki.openstreetmap.org/wiki/Node)

- node_tags
- current_nodes
- current_node_tags

4. ways (https://wiki.openstreetmap.org/wiki/Way)

- way_tags
- way_nodes
- current_ways
- current_way_nodes
- current_way_tags

5. relations (https://wiki.openstreetmap.org/wiki/Relation)

- relation_tags
- relation_members
- current_relations
- current_relation_members
- current_relation_tag

6. redactions (https://wiki.openstreetmap.org/wiki/Redaction_bot_progress_map)
7. gpx_files

- gpx_file_tags
- gps_points

8. notes

- note_comments

9. diary_entries

- diary_comments
- diary_entry_subscriptions
- languages

Tabelas do OAuth:

- client_applications
- oauth_tokens
- oauth_applications
- oauth_access_tokens
- oauth_access_grants

API's do OpenStreetMaps

O OpenStreetMap oferece várias APIs que permitem acessar e contribuir com os dados do mapa de diferentes maneiras. Aqui estão os principais tipos de APIs fornecidos pelo OpenStreetMap:

1. API de Dados (RESTful):

- Esta é a API principal que permite acessar os dados brutos do mapa do OpenStreetMap, como ruas, estradas, edifícios, pontos de interesse, etc.
- É uma API RESTful que fornece endpoints para recuperar, criar, atualizar e excluir elementos do mapa, bem como para executar consultas de pesquisa e recuperação de dados.
- Os dados são geralmente retornados em formato XML ou JSON.
- Exemplo de URL: https://api.openstreetmap.org/api/0.6/map

2. API Nominatim:

- Esta API é usada para geocodificação/reversão geográfica, ou seja, converter endereços em coordenadas geográficas (latitude e longitude) e vice-versa.
- Também pode ser usada para pesquisar locais e obter informações sobre lugares específicos.
- Exemplo de URL: https://nominatim.openstreetmap.org/search

3. API Overpass:

- Esta API oferece uma interface mais avançada para consultar dados do OpenStreetMap.
- É especialmente útil para consultas complexas que envolvem filtragem, agrupamento e processamento de dados geoespaciais.
- Os resultados podem ser retornados em formatos como XML, JSON ou GeoJSON.
- Exemplo de URL: https://overpass-api.de/api/interpreter

4. API de Estilos:

- Esta API permite criar e personalizar estilos de renderização para os dados do OpenStreetMap.
- Os estilos definem como os elementos do mapa são exibidos visualmente em mapas interativos ou estáticos.
- Exemplo de URL: https://openstreetmap.org/styles

API de Dados (RESTful)

```
library(httr)

# Definir as coordenadas de latitude e longitude da área desejada
min_lat <- -23.60
max_lat <- -23.55
min_lon <- -46.70
max_lon <- -46.65

# Montar a URL da API com os parâmetros necessários
url <- paste0("https://api.openstreetmap.org/api/0.6/map?bbox=", min_lon, ",", min_lat, ",", max_lon, "
# Fazer a solicitação GET para a API
resposta <- GET(url)

# Verificar se a solicitação foi bem-sucedida
# stop_for_status(resposta)

# Extrair o conteúdo da resposta
conteudo <- content(resposta, "text")

# Exibir os dados brutos do mapa
print(conteudo)
```

```
## [1] "You requested too many nodes (limit is 50000). Either request a smaller area, or use planet.osm
```

API Nominatim

```r
library(httr)
library(jsonlite)

# Definir as coordenadas de latitude e longitude do ponto de interesse
latitude <- -23.55052  # Exemplo: São Paulo, Brasil
longitude <- -46.633308

# Montar a URL da API com os parâmetros necessários
url <- paste0("https://nominatim.openstreetmap.org/reverse?lat=", latitude, "&lon=", longitude, "&forma

# Fazer a solicitação GET para a API
resposta <- GET(url)

# Verificar se a solicitação foi bem-sucedida
stop_for_status(resposta)

# Extrair o conteúdo da resposta
conteudo <- content(resposta, "text", encoding = "UTF-8")

# Converter o conteúdo JSON em um data frame
nominatim_eg <- fromJSON(conteudo)

# Visualizar os dados
str(nominatim_eg)
```

```
## List of 15
##  $ place_id    : int 7595760
##  $ licence     : chr "Data © OpenStreetMap contributors, ODbL 1.0. http://osm.org/copyright"
##  $ osm_type    : chr "way"
##  $ osm_id      : int 75488634
##  $ lat         : chr "-23.550389799999998"
##  $ lon         : chr "-46.633080956332904"
##  $ class       : chr "tourism"
##  $ type        : chr "attraction"
##  $ place_rank  : int 30
##  $ importance  : num 0.431
##  $ addresstype : chr "tourism"
##  $ name        : chr "Praça da Sé"
##  $ display_name: chr "Praça da Sé, Rua Venceslau Brás, Glicério, Sé, São Paulo, Região Imediata de Sã
##  $ address     :List of 13
##   ..$ tourism       : chr "Praça da Sé"
##   ..$ road          : chr "Rua Venceslau Brás"
##   ..$ suburb        : chr "Glicério"
##   ..$ city          : chr "São Paulo"
##   ..$ municipality  : chr "Região Imediata de São Paulo"
##   ..$ county        : chr "Região Metropolitana de São Paulo"
##   ..$ state_district: chr "Região Geográfica Intermediária de São Paulo"
##   ..$ state         : chr "São Paulo"
##   ..$ ISO3166-2-lvl4: chr "BR-SP"
##   ..$ region        : chr "Região Sudeste"
##   ..$ postcode      : chr "01017-000"
##   ..$ country       : chr "Brasil"
```

```
##   ..$ country_code  : chr "br"
##  $ boundingbox : chr [1:4] "-23.5517242" "-23.5491629" "-46.6342888" "-46.6319455"
```

API Overpass

```r
library(httr)
library(jsonlite)

# Definir a consulta Overpass
consulta_overpass <- '[out:json];(node(around:1000,-23.55052,-46.633308)["amenity"];);out;'

# Codificar a consulta para que seja usada corretamente na URL
consulta_codificada <- URLencode(consulta_overpass)

# Montar a URL da API Overpass
url <- paste0("https://overpass-api.de/api/interpreter?data=", consulta_codificada)

# Fazer a solicitação GET para a API
resposta <- GET(url)

# Verificar se a solicitação foi bem-sucedida
stop_for_status(resposta)

# Extrair o conteúdo da resposta
conteudo <- content(resposta, "text", encoding = "UTF-8")

# Converter o conteúdo JSON em um data frame
overpass_eg <- fromJSON(conteudo)

# Visualizar os dados
str(overpass_eg)
```

```
## List of 4
##  $ version  : num 0.6
##  $ generator: chr "Overpass API 0.7.62.1 084b4234"
##  $ osm3s    :List of 2
##   ..$ timestamp_osm_base: chr "2024-04-19T16:44:44Z"
##   ..$ copyright         : chr "The data included in this document is from www.openstreetmap.org. The
##  $ elements :'data.frame':   416 obs. of  5 variables:
##   ..$ type: chr [1:416] "node" "node" "node" "node" ...
##   ..$ id  : num [1:416] 5.99e+08 5.99e+08 5.99e+08 5.99e+08 5.99e+08 ...
##   ..$ lat : num [1:416] -23.6 -23.6 -23.6 -23.6 -23.6 ...
##   ..$ lon : num [1:416] -46.6 -46.6 -46.6 -46.6 -46.6 ...
##   ..$ tags:'data.frame': 416 obs. of  131 variables:
##   .. ..$ amenity                      : chr [1:416] "restaurant" "parking" "parking" "taxi" ...
##   .. ..$ name                         : chr [1:416] "Maria Paula Bar e Restaurante" NA NA NA ...
##   .. ..$ addr:city                    : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:country                 : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:housenumber             : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:street                  : chr [1:416] NA NA NA NA ...
##   .. ..$ brand                        : chr [1:416] NA NA NA NA ...
##   .. ..$ brand:wikidata               : chr [1:416] NA NA NA NA ...
##   .. ..$ brand:wikipedia              : chr [1:416] NA NA NA NA ...
##   .. ..$ operator                     : chr [1:416] NA NA NA NA ...
```

```
##   .. ..$ access:gender               : chr [1:416] NA NA NA NA ...
##   .. ..$ beds                        : chr [1:416] NA NA NA NA ...
##   .. ..$ building                    : chr [1:416] NA NA NA NA ...
##   .. ..$ website                     : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:postcode               : chr [1:416] NA NA NA NA ...
##   .. ..$ atm                         : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:suburb                 : chr [1:416] NA NA NA NA ...
##   .. ..$ phone                       : chr [1:416] NA NA NA NA ...
##   .. ..$ source                      : chr [1:416] NA NA NA NA ...
##   .. ..$ wikidata                    : chr [1:416] NA NA NA NA ...
##   .. ..$ wikimedia_commons           : chr [1:416] NA NA NA NA ...
##   .. ..$ wikipedia                   : chr [1:416] NA NA NA NA ...
##   .. ..$ capacity                    : chr [1:416] NA NA NA NA ...
##   .. ..$ access                      : chr [1:416] NA NA NA NA ...
##   .. ..$ layer                       : chr [1:416] NA NA NA NA ...
##   .. ..$ opening_hours               : chr [1:416] NA NA NA NA ...
##   .. ..$ parking                     : chr [1:416] NA NA NA NA ...
##   .. ..$ surface                     : chr [1:416] NA NA NA NA ...
##   .. ..$ cuisine                     : chr [1:416] NA NA NA NA ...
##   .. ..$ fee                         : chr [1:416] NA NA NA NA ...
##   .. ..$ supervised                  : chr [1:416] NA NA NA NA ...
##   .. ..$ takeaway                    : chr [1:416] NA NA NA NA ...
##   .. ..$ entrance                    : chr [1:416] NA NA NA NA ...
##   .. ..$ payment:telephone_cards     : chr [1:416] NA NA NA NA ...
##   .. ..$ ref                         : chr [1:416] NA NA NA NA ...
##   .. ..$ denomination                : chr [1:416] NA NA NA NA ...
##   .. ..$ religion                    : chr [1:416] NA NA NA NA ...
##   .. ..$ description                 : chr [1:416] NA NA NA NA ...
##   .. ..$ social_facility             : chr [1:416] NA NA NA NA ...
##   .. ..$ social_facility:for         : chr [1:416] NA NA NA NA ...
##   .. ..$ alt_name                    : chr [1:416] NA NA NA NA ...
##   .. ..$ end_date                    : chr [1:416] NA NA NA NA ...
##   .. ..$ female                      : chr [1:416] NA NA NA NA ...
##   .. ..$ male                        : chr [1:416] NA NA NA NA ...
##   .. ..$ note                        : chr [1:416] NA NA NA NA ...
##   .. ..$ healthcare                  : chr [1:416] NA NA NA NA ...
##   .. ..$ short_name                  : chr [1:416] NA NA NA NA ...
##   .. ..$ operator:wikidata           : chr [1:416] NA NA NA NA ...
##   .. ..$ operator:wikipedia          : chr [1:416] NA NA NA NA ...
##   .. ..$ bus                         : chr [1:416] NA NA NA NA ...
##   .. ..$ lit                         : chr [1:416] NA NA NA NA ...
##   .. ..$ public_transport            : chr [1:416] NA NA NA NA ...
##   .. ..$ bicycle_parking             : chr [1:416] NA NA NA NA ...
##   .. ..$ level                       : chr [1:416] NA NA NA NA ...
##   .. ..$ smoking                     : chr [1:416] NA NA NA NA ...
##   .. ..$ covered                     : chr [1:416] NA NA NA NA ...
##   .. ..$ disused:amenity             : chr [1:416] NA NA NA NA ...
##   .. ..$ delivery                    : chr [1:416] NA NA NA NA ...
##   .. ..$ outdoor_seating             : chr [1:416] NA NA NA NA ...
##   .. ..$ name:ja                     : chr [1:416] NA NA NA NA ...
##   .. ..$ gtfs_id                     : chr [1:416] NA NA NA NA ...
##   .. ..$ highway                     : chr [1:416] NA NA NA NA ...
##   .. ..$ network                     : chr [1:416] NA NA NA NA ...
##   .. ..$ route_ref                   : chr [1:416] NA NA NA NA ...
```

```
##   .. ..$ wheelchair                  : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:housename              : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:website             : chr [1:416] NA NA NA NA ...
##   .. ..$ name:pt                     : chr [1:416] NA NA NA NA ...
##   .. ..$ email                       : chr [1:416] NA NA NA NA ...
##   .. ..$ tourism                     : chr [1:416] NA NA NA NA ...
##   .. ..$ bench                       : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:phone               : chr [1:416] NA NA NA NA ...
##   .. ..$ government                  : chr [1:416] NA NA NA NA ...
##   .. ..$ office                      : chr [1:416] NA NA NA NA ...
##   .. ..$ shelter                     : chr [1:416] NA NA NA NA ...
##   .. ..$ addr:state                  : chr [1:416] NA NA NA NA ...
##   .. ..$ display                     : chr [1:416] NA NA NA NA ...
##   .. ..$ support                     : chr [1:416] NA NA NA NA ...
##   .. ..$ fixme                       : chr [1:416] NA NA NA NA ...
##   .. ..$ name:en                     : chr [1:416] NA NA NA NA ...
##   .. ..$ phone_1                     : chr [1:416] NA NA NA NA ...
##   .. ..$ diet:vegan                  : chr [1:416] NA NA NA NA ...
##   .. ..$ diet:vegetarian             : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:email               : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:facebook            : chr [1:416] NA NA NA NA ...
##   .. ..$ toilets:wheelchair          : chr [1:416] NA NA NA NA ...
##   .. ..$ wheelchair:description      : chr [1:416] NA NA NA NA ...
##   .. ..$ official_name               : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:flickr              : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:twitter             : chr [1:416] NA NA NA NA ...
##   .. ..$ contact:youtube             : chr [1:416] NA NA NA NA ...
##   .. ..$ internet_access             : chr [1:416] NA NA NA NA ...
##   .. ..$ start_date                  : chr [1:416] NA NA NA NA ...
##   .. ..$ drive_through               : chr [1:416] NA NA NA NA ...
##   .. ..$ historic                    : chr [1:416] NA NA NA NA ...
##   .. ..$ healthcare:speciality       : chr [1:416] NA NA NA NA ...
##   .. ..$ townhall:type               : chr [1:416] NA NA NA NA ...
##   .. ..$ branch                      : chr [1:416] NA NA NA NA ...
##   .. ..$ brand:short                 : chr [1:416] NA NA NA NA ...
##   .. .. [list output truncated]
```

API de Estilos

```r
library(httr)

# Definir o nome do estilo desejado
nome_do_estilo <- "mapnik"  # Mapnik é um estilo de renderização padrão do OpenStreetMap

# Montar a URL da API de Estilos
url <- paste0("https://openstreetmap.org/styles/", nome_do_estilo, "/style.json")

# Fazer a solicitação GET para a API
resposta <- GET(url)

# Verificar se a solicitação foi bem-sucedida
# stop_for_status()
```

```r
# Extrair o conteúdo da resposta (o estilo de renderização)
estilos_eg <- content(resposta, "text", encoding = "UTF-8")

# Exibir o conteúdo (o estilo de renderização)
str(estilos_eg)
```

```
##  chr ""
```

Uso de caso para API RESTful

Começarei recolhendo as informações do usuario.

```r
library(httr)
library(jsonlite)

# Defina o UID do usuário do OpenStreetMap que você deseja consultar
uid_usuario <- 21160844

# Faça a solicitação à API de Edições para obter o histórico de edições do usuário pelo UID
url <- paste0("https://api.openstreetmap.org/api/0.6/changesets?user=", uid_usuario)
resposta <- GET(url)

# Verifique se a solicitação foi bem-sucedida
stop_for_status(resposta)

# Extrair o conteúdo da resposta (JSON)
conteudo <- content(resposta, "text", encoding = "UTF-8")

# Analisar o JSON
dados_json <- fromJSON(conteudo)

# Extrair os IDs dos changesets do JSON
ids_changesets <- dados_json$changeset$id

# Exibir os IDs dos changesets
print(ids_changesets)
```

```
## [1] 149053068 149052433 149051809 149049540 148254832
```

Convertendo o arquivo JSON em CSV

```r
dados_df <- as.data.frame(dados_json)

str(dados_df)
```

```
## 'data.frame':    5 obs. of  18 variables:
##  $ version                  : chr  "0.6" "0.6" "0.6" "0.6" ...
##  $ generator                : chr  "OpenStreetMap server" "OpenStreetMap server" "OpenStreetMap serv
##  $ copyright                : chr  "OpenStreetMap and contributors" "OpenStreetMap and contributors"
##  $ attribution              : chr  "http://www.openstreetmap.org/copyright" "http://www.openstreetma
##  $ license                  : chr  "http://opendatacommons.org/licenses/odbl/1-0/" "http://opendataco
##  $ changesets.id            : int  149053068 149052433 149051809 149049540 148254832
##  $ changesets.created_at    : chr  "2024-03-23T14:24:01Z" "2024-03-23T14:11:22Z" "2024-03-23T13:54:53
##  $ changesets.open          : logi  FALSE FALSE FALSE FALSE FALSE
##  $ changesets.comments_count: int  0 0 0 0 0
```

```
##  $ changesets.changes_count : int  5 57 99 51 50
##  $ changesets.closed_at     : chr  "2024-03-23T14:24:02Z" "2024-03-23T14:11:23Z" "2024-03-23T13:54:5
##  $ changesets.min_lat       : num  -12.9 -12.9 -12.9 -12.9 -12.9
##  $ changesets.min_lon       : num  -38.4 -38.4 -38.4 -38.4 -38.4
##  $ changesets.max_lat       : num  -12.9 -12.9 -12.9 -12.9 -12.9
##  $ changesets.max_lon       : num  -38.4 -38.4 -38.4 -38.4 -38.4
##  $ changesets.uid           : int  21160844 21160844 21160844 21160844 21160844
##  $ changesets.user          : chr  "Mauricio Ed" "Mauricio Ed" "Mauricio Ed" "Mauricio Ed" ...
##  $ changesets.tags          :'data.frame':   5 obs. of  10 variables:
##   ..$ changesets_count                       : chr  "5" "4" "3" "2" ...
##   ..$ comment                                : chr  "#osmus-tasks-538, Mapatona YouthMappers UFBA BA
##   ..$ created_by                             : chr  "iD 2.21.1" "iD 2.21.1" "iD 2.21.1" "iD 2.21.1"
##   ..$ hashtags                               : chr  "#osmus-tasks-538" "#osmus-tasks-538" "#osmus-ta
##   ..$ host                                   : chr  "https://tasks.openstreetmap.us/projects/538/map/
##   ..$ imagery_used                           : chr  "Esri World Imagery;.gpx data file" "Esri World
##   ..$ locale                                 : chr  "pt-BR" "pt-BR" "pt-BR" "pt-BR" ...
##   ..$ source                                 : chr  "aerial imagery;Bing;Esri" NA "aerial imagery;Bi
##   ..$ review_requested                       : chr  NA NA NA NA ...
##   ..$ warnings:crossing_ways:building-highway: chr  NA NA NA NA ...
```

Agora irei puxar os valores fornecidos quando fizermos uma requisição para cada valor da coluna changesets.id

```
changeset_id <- 149053068

url <- paste0("https://api.openstreetmap.org/api/0.6/changesets/#", changeset_id)

resposta <- GET(url)

stop_for_status(resposta)

conteudo <- content(resposta, "text", enconding = "UTF-8")

dados_jdad <- fromJSON(conteudo)
```

Transformando os dados JSON em uma tabela

```
changeset_id_df <- as.data.frame(dados_jdad)

str(changeset_id_df)
```

```
## 'data.frame':    100 obs. of  18 variables:
##  $ version                  : chr  "0.6" "0.6" "0.6" "0.6" ...
##  $ generator                : chr  "OpenStreetMap server" "OpenStreetMap server" "OpenStreetMap serv
##  $ copyright                : chr  "OpenStreetMap and contributors" "OpenStreetMap and contributors"
##  $ attribution              : chr  "http://www.openstreetmap.org/copyright" "http://www.openstreetma
##  $ license                  : chr  "http://opendatacommons.org/licenses/odbl/1-0/" "http://opendatac
##  $ changesets.id            : int  150233029 150233028 150233027 150233026 150233025 150233024 15023
##  $ changesets.created_at    : chr  "2024-04-19T16:45:41Z" "2024-04-19T16:45:38Z" "2024-04-19T16:45:3
##  $ changesets.open          : logi  TRUE FALSE TRUE FALSE FALSE FALSE ...
##  $ changesets.comments_count: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ changesets.changes_count : int  1 6 1 11 10 14 17 213 2 1 ...
##  $ changesets.min_lat       : num  45.13 33.87 33.8 -8.69 -4.06 ...
##  $ changesets.min_lon       : num  20.1 -84.7 -118.3 36.7 143.1 ...
##  $ changesets.max_lat       : num  45.14 33.87 33.8 -8.68 -4.06 ...
##  $ changesets.max_lon       : num  20.1 -84.7 -118.3 36.7 143.1 ...
##  $ changesets.uid           : int  21310460 21326147 13990595 18953770 20978051 4515353 14174445 119
```

```
##  $ changesets.user          : chr  "kentakta" "Wayne561651313" "KelsonV" "Kharandesert" ...
##  $ changesets.tags          :'data.frame':   100 obs. of  32 variables:
##   ..$ changesets_count                          : chr  "219" "1" NA NA ...
##   ..$ comment                                   : chr  "update" "Corrected street name" "Specify fire l
##   ..$ created_by                                : chr  "iD 2.28.1" "iD 2.28.1" "StreetComplete 57.2" "
##   ..$ host                                      : chr  "https://www.openstreetmap.org/edit" "https://w
##   ..$ imagery_used                              : chr  "OpenStreetMap (Standard)" "Bing Maps Aerial" NA
##   ..$ locale                                    : chr  "en-US" "en-US" "en-US" NA ...
##   ..$ review_requested                          : chr  NA "yes" NA NA ...
##   ..$ source                                    : chr  NA NA "survey" "Esri World Imagery" ...
##   ..$ StreetComplete:quest_type                 : chr  NA NA "AddFireHydrantType" NA ...
##   ..$ hashtags                                  : chr  NA NA NA "#adt" ...
##   ..$ ideditor:walkthrough_progress             : chr  NA NA NA NA ...
##   ..$ ideditor:walkthrough_started              : chr  NA NA NA NA ...
##   ..$ resolved:crossing_ways:building-building: chr  NA NA NA NA ...
##   ..$ bundle_id                                 : chr  NA NA NA NA ...
##   ..$ resolved:crossing_ways:building-highway : chr  NA NA NA NA ...
##   ..$ warnings:crossing_ways:highway-waterway : chr  NA NA NA NA ...
##   ..$ imagery_used:1                            : chr  NA NA NA NA ...
##   ..$ imagery_used:2                            : chr  NA NA NA NA ...
##   ..$ imagery_used:2:uuid                       : chr  NA NA NA NA ...
##   ..$ imagery_used:3                            : chr  NA NA NA NA ...
##   ..$ imagery_used:3:uuid                       : chr  NA NA NA NA ...
##   ..$ operator                                  : chr  NA NA NA NA ...
##   ..$ source:date                               : chr  NA NA NA NA ...
##   ..$ url                                       : chr  NA NA NA NA ...
##   ..$ closed:note                               : chr  NA NA NA NA ...
##   ..$ source:date:addr                          : chr  NA NA NA NA ...
##   ..$ type                                      : chr  NA NA NA NA ...
##   ..$ resolved:unsquare_way:building            : chr  NA NA NA NA ...
##   ..$ resolved:crossing_ways:highway-highway  : chr  NA NA NA NA ...
##   ..$ warnings:close_nodes:vertices             : chr  NA NA NA NA ...
##   ..$ resolved:disconnected_way:highway         : chr  NA NA NA NA ...
##   ..$ warnings:crossing_ways:highway-highway  : chr  NA NA NA NA ...
##  $ changesets.closed_at     : chr  NA "2024-04-19T16:45:39Z" NA "2024-04-19T16:45:35Z" ...
```

Uso de caso da Overpass API

```r
library(httr)
library(jsonlite)

# Defina o ID do changeset desejado
changeset_id <- "#149053068"

# Construa a consulta Overpass QL
overpass_query <- paste0('[out:json][timeout:25];',
                         'changeset(', changeset_id, ');',
                         'out;')

# Envie a consulta para a Overpass API
response <- POST("https://overpass-api.de/api/interpreter", body = overpass_query)

# Verifique se a solicitação foi bem-sucedida
```

```
# stop_for_status(response)

# Analise a resposta JSON
# changeset_info <- content(response, "text") %>%
  # fromJSON()

# Exiba as informações do changeset
# print(changeset_info)
```