

Acessando Dados da Web em R

APIs

Tiago Ventura | venturat@umd.edu

University of Maryland, College Park

Planos para Hoje.

1. Introdução a APIs. O que fazem? Onde vivem? O que comem?
2. Acessando APIs via http.
 - Estudo de caso na API do Fogo Cruzado.
3. Exemplos de pacotes de R para acessar APIs
 - Fogo Cruzado
 - Dados do Congresso (bRasilLegis)
4. Exercício
5. Utilizando a API do Twitter.

Introdução a APIs. O que fazem? Onde vivem? O que comem?

1. Acessar dados compartilhados por Empresas, ONGs e Governo.
 - Twitter, Facebook, Spotify, Yelp.
 - Portal da Transparência, IPEA.
 - Fogo Cruzado.
2. Processar nossos dados em Algoritmos desenvolvidos por terceiros.
 - PeRsPEctive.
 - Classificação de imagens.
 - Serenata

Uso de API para acesso à dados.

Em seu formato mais simples, uma API é somente uma url. Veja o exemplo abaixo:

http://meusite.com/key?param_1;param_2

Principais elementos:

- **http://meusite.com/**: a base da API, chamado nas documentações de end-point.
- **key**: as credenciais que muitos sites pedem - nem todos, como veremos.
- **?param_1;param_2** parametros, ou filtros para refinar a busca da API.

Passo-a-Passo para acessar API

1. Verificar se a API está disponível, e solicitar credenciais.
2. Verificar se há um pacote de R disponível para acessar a API
3. Ler a documentação da API, identificar a informações de seu interesse.
4. Se existir um pacote de R, use-o.
5. Se não existir, escreva seu código e limpe os dados.

Fogo Cruzado API

Acessando sua Senha.

```
# Pacotes para acessar APIs
library(httr)
library(jsonlite)
library(tidyverse)

# Solicitar a senha
get_jwt <- POST("https://api.fogocruzado.org.br/
               api/v1/auth/login",
               query = list(
                 email = "venturat@umd.edu",
                 password = "xxxxxxx"))

# Pegar a senha
token <- content(get_jwt)$access_token

# token = sua senha de acesso à API. Não compartilhe por aí.
```


Base da API (end-point)



Iniciando

Introdução

Autenticação

Endpoints

Sobre

Filtros

Fontes de

Ocorrências

Presença de Agente

de Segurança

Sobre

Endpoints são as URL's que a API provê para o retorno dos dados.

A API tem 5 endpoints, 3 deles permitem filtros pela URL de acordo com os padrões de JSON API Rest.

Url e Versionamento

A URL base da aplicação é:

<http://api.fogocruzado.org.br/>

A aplicação tem um controle de versão onde a mesma é incrementada, caso a API seja atualizada de maneira a impactar nas aplicações cliente. A versão é sempre colocada após a URL base.

Documentação.

Passo 1: Construa o Endereco.

```
# Passo 1. Crie a Url
base_url <- "https://api.fogocruzado.org.br/api/v1"
occurrences <- "/occurrences"
api <- paste0(base_url, occurrences)
print(api)
```

```
## [1] "https://api.fogocruzado.org.br/api/v1/occurrences"
```

Passo 2: GET

```
# Passo 2: Acesse a API
```

```
response <- GET(api,  
               add_headers('Authorization' =  
                           paste("Bearer",  
                                 token, sep = " ")))
```

```
# Qual o resultado?
```

```
response
```

```
## Response [https://api.fogocruzado.org.br/api/v1/occurrences]
```

```
##   Date: 2019-12-17 19:45
```

```
##   Status: 200
```

```
##   Content-Type: application/json
```

```
##   Size: 18.9 MB
```

Passo 3: JSON

```
[{"id_ocorrencia":8758,
  "local_ocorrencia":"Mar\u00e9, Rio de Janeiro - RJ, Brasil",
  "latitude_ocorrencia":-22.8574962616,
  "longitude_ocorrencia":-43.2430152893,
  "data_ocorrencia":"2018-01-09 15:57:00",
  "hora_ocorrencia":"15:57:00",
  "presen_agen_segur_ocorrencia":0,
  "qtd_morto_civil_ocorrencia":0,
  "qtd_morto_agen_segur_ocorrencia":0,
  "qtd_ferido_civil_ocorrencia":0,
  "qtd_ferido_agen_segur_ocorrencia":0,
  "fonte_ocorrencia":0,
  "estado_id":19,
  "cidade_id":3661,
  "nome_cidade":"Rio de Janeiro",
  "cod_ibge_cidade":3304557,
  "gentilico_cidade":"carioca",
  "populacao_cidade":6320446,
  "area_cidade":119746,
  "densidade_demo_cidade":"5.27",
  "nome_estado":"Rio de Janeiro",
  "uf_estado":"RJ"}]
```

```
# Converter para um json
json_fogo_cruzado <- content(response,
                             as="text", encoding = "UTF-8")
```

```
output <- fromJSON(json_fogo_cruzado) %>%
  tibble::as_tibble()
```

glimpse(output)

```
## Observations: 26,952
## Variables: 23
## $ id_ocorrendia <int> 8758, 10431, 10430,
## $ local_ocorrendia <chr> "Maré, Rio de Janeiro", "Rio de Janeiro", "Rio de Janeiro",
## $ latitude_ocorrendia <dbl> -22.85750, -22.93538, -22.93538,
## $ longitude_ocorrendia <dbl> -43.24302, -43.19463, -43.19463,
## $ data_ocorrendia <chr> "2018-01-09 15:57:00", "2018-01-09 20:54:00", "2018-01-09 20:54:00",
## $ hora_ocorrendia <chr> "15:57:00", "20:54:00", "20:54:00",
## $ presen_agen_segur_ocorrendia <int> 0, 0, 0, 1, 1, 1, 0,
## $ qtd_morto_civil_ocorrendia <int> 0, 0, 0, 0, 0, 0, 0,
## $ qtd_morto_agen_segur_ocorrendia <int> 0, 0, 0, 1, 0, 0, 0,
## $ qtd_ferido_civil_ocorrendia <int> 0, 0, 0, 0, 0, 0, 0,
## $ qtd_ferido_agen_segur_ocorrendia <int> 0, 0, 0, 0, 0, 0, 0,
## $ fonte_ocorrendia <int> 0, 0, 0, 1, 0, 0, 0,
## $ estado_id <int> 19, 19, 19, 19, 19,
## $ cidade_id <int> 3661, 3661, 3602, 3602,
## $ nome_cidade <chr> "Rio de Janeiro", "Rio de Janeiro", "Rio de Janeiro", "Rio de Janeiro",
## $ cod_ibge_cidade <int> 3304557, 3304557, 3304557, 3304557, 3304557,
```

Adicionar Filtros

```
query_list <- list(presen_agen_segur_ocorrencia="1",
                  uf_estado="RJ")

response <- GET(api,
               add_headers('Authorization' =
                           paste("Bearer", token, sep = " ")),
               query=query_list)

output <- fromJSON(httr::content(response,
                                  as="text", encoding = "UTF-8")) %>%
  as_tibble()
```



```
glimpse(output)
```

```
## Observations: 5,710
## Variables: 23
## $ id_ocorrendia <int> 16, 17, 18, 34, 44,
## $ local_ocorrendia <chr> "Santa Cruz, Rio de
## $ latitude_ocorrendia <dbl> -22.91965, -22.89864
## $ longitude_ocorrendia <dbl> -43.68439, -43.35236
## $ data_ocorrendia <chr> "2016-07-05 06:02:00
## $ hora_ocorrendia <chr> "6:02:00", "13:44:00
## $ presen_agen_segur_ocorrendia <int> 1, 1, 1, 1, 1, 1, 1,
## $ qtd_morto_civil_ocorrendia <int> 0, 0, 0, 1, 0, 0, 0,
## $ qtd_morto_agen_segur_ocorrendia <int> 1, 0, 0, 0, 0, 0, 0,
## $ qtd_ferido_civil_ocorrendia <int> 0, 0, 0, 0, 0, 0, 1,
## $ qtd_ferido_agen_segur_ocorrendia <int> 0, 0, 0, 0, 0, 0, 0,
## $ fonte_ocorrendia <int> 1, 0, 0, 0, 0, 0, 0,
## $ estado_id <int> 19, 19, 19, 19, 19,
## $ cidade_id <int> 3661, 3661, 3661, 36
## $ nome_cidade <chr> "Rio de Janeiro", "R
## $ cod_ibge_cidade <int> 3304557, 3304557, 33
```

Cidades com Mais Ocorrências no Rio de Janeiro

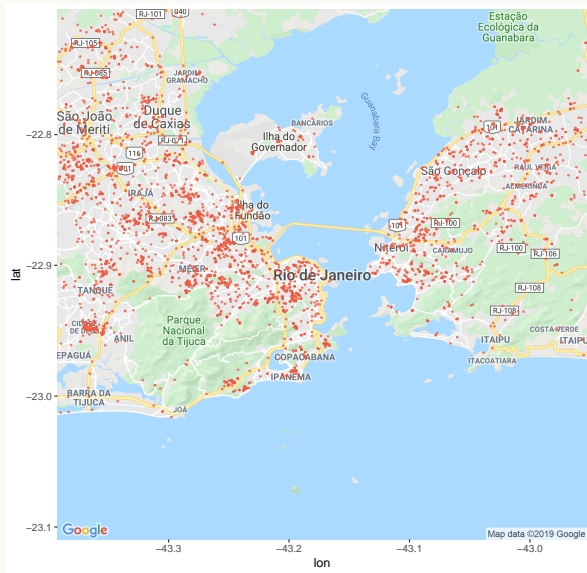
```
mais_ocorrencias <- output %>%  
  count(nome_cidade) %>%  
  top_n(10) %>%  
  arrange(n) %>%  
  ggplot(., aes(y=n, x=nome_cidade)) +  
  geom_col() + theme_minimal() + coord_flip() +  
  ylab("Tiroteios com Presença Policial") +  
  xlab(" Cidades do Rio de Janeiro")
```

```
## Selecting by n
```

Visualizacao com Mapas

```
library(ggmap)
library(RColorBrewer)

ggmap(get_googlemap("rio de janeiro",
                    zoom = 11, maptype = "roadmap", scale=2)) +
  geom_point(data = output,
            aes(x = longitude_ocorrencia,
                y = latitude_ocorrencia),
            color="tomato2", alpha=.8, size = 0.3)
```



Pacote de R para APIs.

1. Antes de acessar uma API, sempre busque se alguém já criou um pacote de R para isso.
 - Sim. Quase sempre, há um pacote pre-programado.
2. Onde encontrar?
 - Primeiro, google o nome da api + R
 - CRAN
 - Github
3. **Crossfire**

Instalar o pacote

```
#Instalação do pacote  
install.packages("devtools") # pacote para acessar o github  
devtools::install_github("voltdatalab/crossfire")
```

Acessar Dados

```
library(crossfire)

# Registra usuario e senha, e envia sua senha da API

fogocruzado_signin(email = "venturat@umd.edu",
                    password = "xxxxx")

# Extrair os dados que pegamos manualmente antes

fogocruzado_rj <- get_fogocruzado(state= "RJ",
                                  security_agent = 1)
```

Muito mais fácil!

Exemplo II: congressbr

1. Pacote de R para acessar API do Congresso e Senado.
2. Este [artigo](#) sobre o pacote oferece uma excelente introdução à como utilizá-lo.
3. [Documentação](#)

- `cham_votes()`: retorna informação sobre votações na Câmara voting information from the Chamber of
- `cham_bills()`: retorna informação sobre atividade legislativa na Câmara.
- `sen_votes()`: retorna informação de voto no Senado.
- `sen_bill_search()`: Procura por atividade legislativa no Senado.

```
# Instale o pacote
```

```
install.packages("congressbr")
```

```
devtools::install_github("RobertMyles/congressbr")
```

```
all <- cham_legislator_list()
glimpse(all)
```

```
## Observations: 513
## Variables: 13
## $ legislator_id <chr> "73701", "73696", "73801"
## $ legislator_status <chr> "Titular", "Titular", "Ti
## $ legislator_name <chr> "BENEDITA SOUZA DA SILVA
## $ legislator_cham_name <chr> "BENEDITA DA SILVA", "ANG
## $ legislator_photo_url <chr> "http://www.camara.gov.br
## $ legislator_gender <chr> "feminino", "feminino", "
## $ legislator_state <chr> "RJ", "SC", "PE", "RJ", "
## $ legislator_party <chr> "PT", "PP", "PCdoB", "PCd
## $ legislator_cabinet <chr> "330", "252", "915", "622
## $ legislator_building_address <chr> "4", "4", "4", "4", "4",
## $ legislator_telephone_number <chr> "3215-5330", "3215-5252",
## $ legislator_email <chr> "dep.beneditadasilva@cama
## $ legislator_comissions <chr> "", "", "", "", "", "", ""
```

Leis Votadas por Ano

```
cham_plenary_bills(2010) %>% slice(1:5)
```

```
## # A tibble: 5 x 3
##   bill_id bill_name      vote_date
##   <chr>   <chr>         <chr>
## 1 21571    PLP 168/1993 04/05/2010
## 2 21571    PLP 168/1993 04/05/2010
## 3 21571    PLP 168/1993 05/05/2010
## 4 21571    PLP 168/1993 05/05/2010
## 5 21571    PLP 168/1993 05/05/2010
```

Detalhes de uma Lei

```
lei_3962 <- cham_bill_info(type = "PL",  
                           number = "3962", year = "2008")  
glimpse(lei_3962)
```

```
## Observations: 1
## Variables: 19
## $ bill_type      <chr> "PL          "
## $ bill_number    <chr> "3962"
## $ bill_year      <chr> "2008"
## $ bill_id        <chr> "408406"
## $ bill_name      <chr> "PL 3962/2008"
## $ type_name      <chr> "Projeto de Lei"
## $ bill_subject   <chr> "Administracao Publica; Previdenci
## $ ementa_txt     <chr> "Cria a Superintendencia Nacional
## $ ementa_explanation <chr> "\n  "
## $ author_name    <chr> "Poder Executivo"
## $ author_id      <chr> "\n  "
## $ author_state   <chr> "\n  "
## $ author_party   <chr> "\n  "
## $ bill_index     <chr> "Criacao, recriacao, Superintenden
## $ presentation_date <chr> "29/08/2008"
## $ complete_link  <chr> "http://www.camara.gov.br/proposic
## $ situation_desc <chr> "Transformada no(a) Lei Ordinaria 91
## $ last_action_desc <chr> "As Comissoes de \nSeguridade Soc
```

```
votos <- cham_votes(type = "PL",  
                    number = "1992", year = "2007")  
glimpse(votos)
```



```
## Observations: 2,052
## Variables: 31
## $ rollcall_id <chr> "PL-1992-2007-1",
## $ decision_summary <chr> "Aprovada a Subem
## $ decision_date <chr> "28/2/2012", "28/
## $ decision_time <chr> "20:26", "20:26",
## $ rollcall_subject <chr> "SUBEMENDA SUBSTI
## $ session_id <chr> "4531", "4531", "
## $ DEM_orientation <chr> "Nao", "Nao", "Na
## $ GOV_orientation <chr> "Sim", "Sim", "Si
## $ Minoria_orientation <chr> "Liberado", "Libe
## $ PDT_orientation <chr> "Nao", "Nao", "Na
## $ PMDB_orientation <chr> "Sim", "Sim", "Si
## $ PMN_orientation <chr> "Nao", "Nao", "Na
## $ PP_orientation <chr> "Sim", "Sim", "Si
## $ PRB_orientation <chr> "Sim", "Sim", "Si
## $ PrPtdobPrpPhsPtcPslPrtb_orientation <chr> "Sim", "Sim", "Si
## $ PsbPtbPcdob_orientation <chr> "Liberado", "Libe
## $ PSC_orientation <chr> "Sim", "Sim", "Si
## $ PSD_orientation <chr> "Liberado", "Libe
```

Exercício 1

1. Encontre uma API com um pacote de R escrito. Instale o pacote, e utilize-o corretamente. A **API do Ipea** pode ser uma opção, ou esse **pacote** para acessar dados do Lattes, **new york times**, ou a **congressbr** são boas opções.
2. Encontre uma API para você praticar o use da função GET. Pode ser qualquer uma. A guisa de sugestão, você pode utilizar a API do **Portal da Transparência**. É bem simples de acessar.
 - Dica 1 : Escolha o tema, e vá em tentar agora. Depois que você encontrar o resultado, basta copiar para o R.
 - Dica 2 : a url base é <http://www.transparencia.gov.br/api-de-dados/>
3. Colete uma informação desta API. Qualquer uma. Faça um GET que retorne um resultado interessante