

Les API, un enfer ?

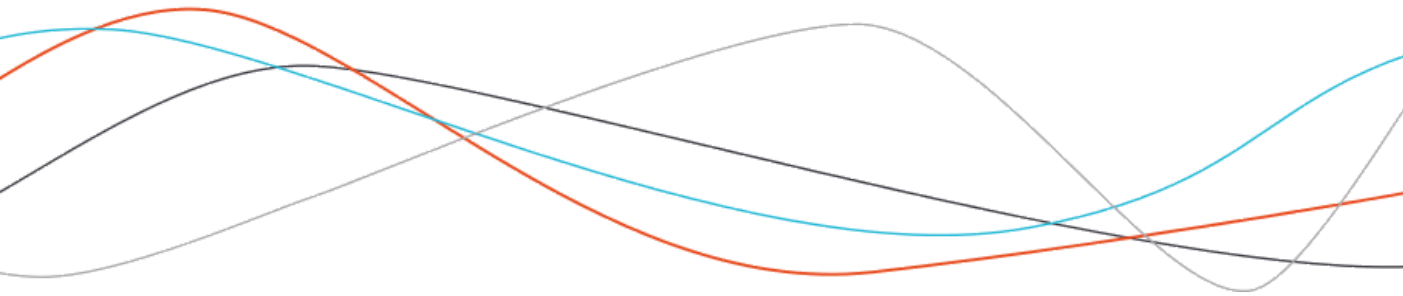
Colin FAY

2017/06/30

Colin FAY - ThinkR

@_colinfay / @thinkR_fr

colin@thinkr.fr - www.thinkr.fr



Colin FAY

ThinkR

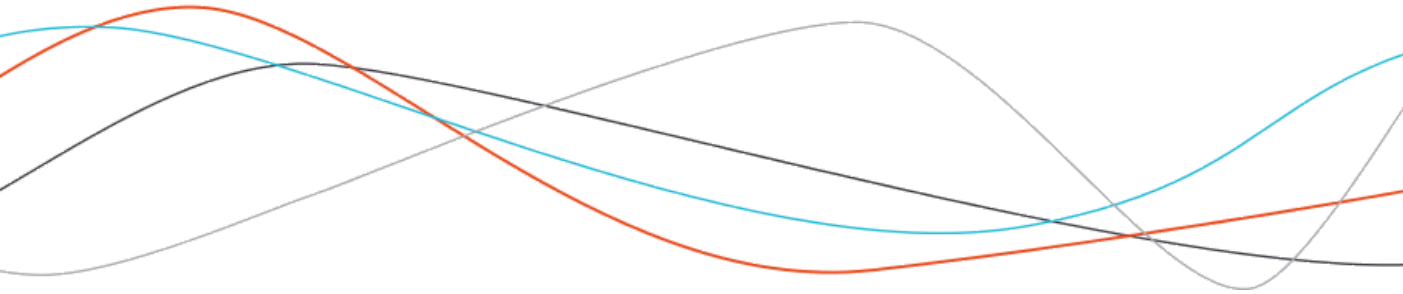
Développeur de :

`rpinterest, rgeoapi & languagelayer.`

Four decorative wavy lines in red, cyan, black, and grey cross the lower half of the slide.

Au programme :

- Une API, qu'est-ce que c'est ?
- Une API, comment ça marche ?
- R et les API



Pourquoi ?

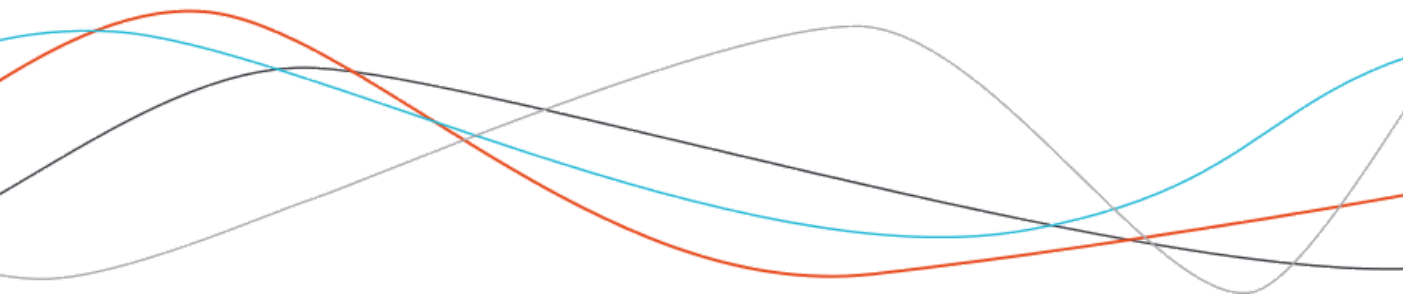
Les API, les premiers temps, c'est souvent...



C'est quoi une API ?

Contraction de Application Programming Interface, l'API permet de faire communiquer un logiciel A (le fournisseur) et un logiciel B (le consommateur).

En théorie, on accède à une API en plongeant dans la documentation. En pratique...



Interroger une API

Deux types de requêtes :

- GET : va "chercher" la ressource sur le serveur distant.
- POST : va "poster" sur le serveur distant.

À noter qu'il en existe d'autres, notamment PUT, DELETE ou encore PATCH.

Anatomie d'un échange

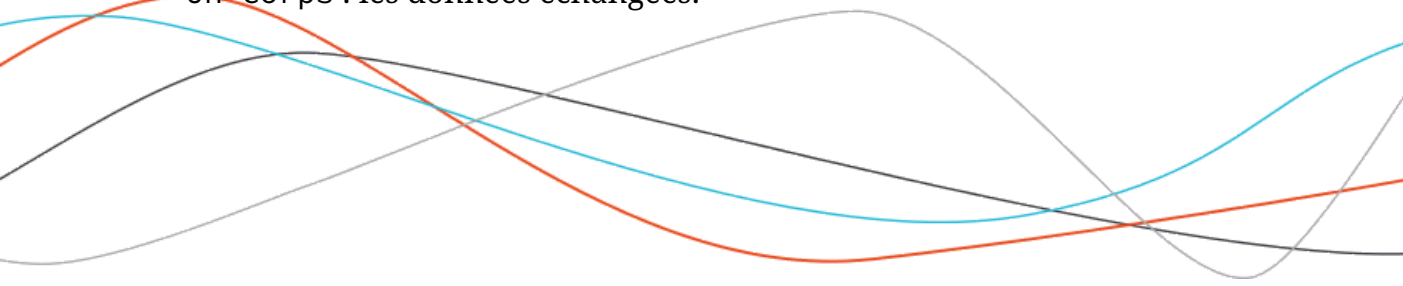
Côté requête

- Une `url` : l'adresse unique d'un emplacement.
- Une `méthode` : GET, POST, DELETE...
- Des `headers` : les métainformations sur l'échange.
- Un `corps` : les données échangées.

À noter qu'il en existe d'autres, notamment PUT et POST.

Anatomie d'un échange

Côté réponse

- Un statut : les informations sur l'état de l'échange (200,404...).
 - Des headers : les métainformations sur l'échange.
 - Un corps : les données échangées.
- 

Géo API

<https://api.gouv.fr/api/api-geo.html>

```
data <- fromJSON("https://geo.api.gouv.fr/communes?codePostal=35200&t")

# Construire une fonction qui appelle l'API
get_by_code <- function(query){
  url <- paste("https://geo.api.gouv.fr/communes?codePostal=", query,
  result <- fromJSON(url)
  return(result)
}
get_by_code("35200")
```

Géo API

```
## Par étape
query <- "35200"
url <- paste("https://geo.api.gouv.fr/communes?codePostal=", query, "&")

# Appel de l'url avec GET
response <- httr::GET(url)
response
> Response [https://geo.api.gouv.fr/communes?codePostal=35200&fields=
  Date: 2017-07-03 12:50
  Status: 200
  Content-Type: application/json; charset=utf-8
  Size: 135 B
```

Géo API

```
# Récupération du contenu  
response2 <- response$content  
response2
```

```
> [1] 5b 7b 22 6e 6f 6d 22 3a 22 52 65 6e 6e 65 73 22 2c 22 63 6f 64  
[26] 33 35 32 33 38 22 2c 22 63 6f 64 65 73 50 6f 73 74 61 75 78 22  
[51] 35 30 30 30 22 2c 22 33 35 32 30 30 22 2c 22 33 35 37 30 30 22  
[76] 6f 64 65 44 65 70 61 72 74 65 6d 65 6e 74 22 3a 22 33 35 22 2c  
[101] 65 52 65 67 69 6f 6e 22 3a 22 35 33 22 2c 22 70 6f 70 75 6c 61  
[126] 22 3a 32 31 31 33 37 33 7d 5d
```

Géo API

```
# Transformation en contenu lisible
response3 <- rawToChar(response2)
response3

> [1] "[{"nom\":"Rennes\","code\":"35238\","codesPostaux\":[\"35238\", \"35200\", \"35700\"]}"

# Depuis JSON

response4 <- fromJSON(response3)
response4

>      nom      code      codesPostaux codeDepartement codeRegion popul
1 Rennes 35238 35000, 35200, 35700          35           53      211

## En une seule fois

GET(url) %>% content()
```

HavelBeenPwned

<https://haveibeenpwned.com/>

Étape 1

```
data <- fromJSON("https://haveibeenpwned.com/api/v2/breachedaccount/t  
datatable(data[,1:4])
```

Construire sa requête dans R

```
query <- "test@example.com"  
url <- paste("https://haveibeenpwned.com/api/v2/breachedaccount/", que  
fromJSON(url) %>% datatable()
```

HavelBeenPwned

```
# Et si nous avons une liste ?
```

```
query <- list("test@example.com", "colin@thinkr.fr", "vincent@thinkr.fr")
```

```
lapply(query, FUN = function(x){  
  url <- paste("https://haveibeenpwned.com/api/v2/breachedaccount/", x,  
    fromJSON(url)  
})
```

```
> Error in open.connection(con, "rb") : HTTP error 429.
```

Language layer

<https://languagelayer.com>

```
# Créer deux variables : access_key et query
access_key <- "fd82b89719707d36fe030538c2985d24"
query <- "Anglet c'est fun"
# Les coller dans l'url
url <- paste("http://apilayer.net/api/detect?access_key =", access_key, "&q=", query)
# Effectuer la requête
data <- fromJSON("http://apilayer.net/api/detect?access_key=fd82b89719707d36fe030538c2985d24&q=Anglet c'est fun")
data$results
```


Microsoft Computer Vision

```
# Construire l'ensemble de la requête POST

#L'url à appeller
url <- "https://westcentralus.api.cognitive.microsoft.com/vision/v1.0/analyze?api-version=2017-08-01"

# Le JSON de requête
request_body <- data.frame(url = 'https://pbs.twimg.com/profile_image/1234567890/normal.jpg')
request_body_json <- toJSON(request_body, auto_unbox = "TRUE")
request_body_json <- gsub("\\[|\\]", "", request_body_json)
```

```
# L'API key
api_key <- "c3628ce7b1ab4ef58c2abfd426223bec"

# Et enfin, la requête
result <- POST(url,
               body = request_body_json,
               add_headers(.headers = c("Content-Type"=Contenttype, "X-Api-Key"=api_key),
               content())
```

```
# L'API key
api_key <- "c3628ce7b1ab4ef58c2abfd426223bec"

# Et enfin, la requête
result <- POST(url,
               body = request_body_json,
               add_headers(.headers = c("Content-Type"=Contenttype, "X-Api-Key"=api_key),
               content())
```

Merci !

Colin FAY - ThinkR

@_colinfay / @thinkR_fr

colin@thinkr.fr - www.thinkr.fr

