

INSTITUTO MAUÁ DE TECNOLOGIA



**MAUÁ**

# Pós Graduação Inteligência Artificial com Deep-learning Instituto Mauá de Tecnologia



## Agenda:

- Serviços
- Requisições HTTP

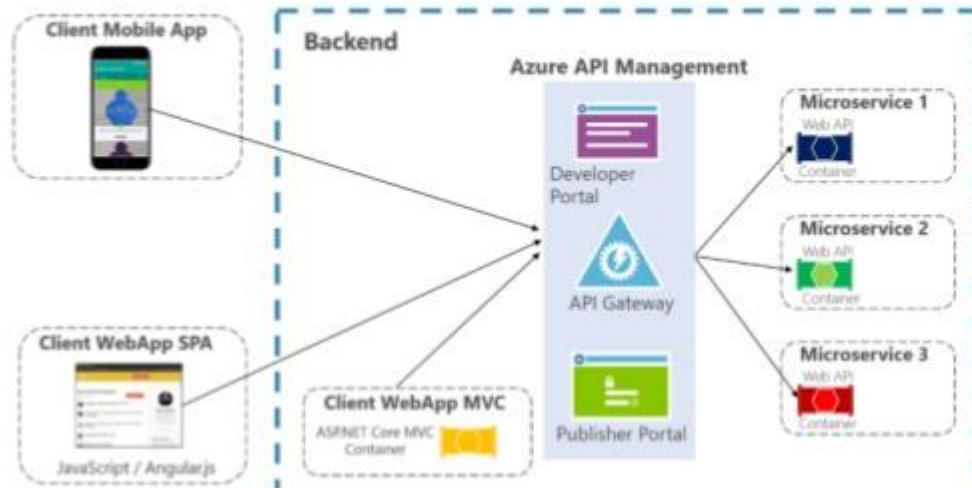
- “SaaS, ou Software como Serviço, é um **modelo de uso de software baseado na nuvem**, ou seja, o sistema fica alojado remotamente e pode ser acessado via Internet.” [Retirado de <https://www.totvs.com/blog/negocios/software-as-a-service/>, 14/06/2020]
- “**Software as a service** (SaaS [/sæəs](#)<sup>[1]</sup>) (also known as subscribeware or rentware) is a software [licensing](#) and [delivery](#) model in which [software](#) is licensed on a [subscription](#) basis and is centrally [hosted](#). It is sometimes referred to as "on-demand software", and was formerly referred to as "software plus services" by [Microsoft](#).” [Retirado de [https://en.wikipedia.org/wiki/Software as a servisse](https://en.wikipedia.org/wiki/Software_as_a_service), 14/06/2020]

- “An **application programming interface (API)** is a [computing interface](#) which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. It can also provide extension mechanisms so that users can extend existing functionality in various ways and to varying degrees.” [Retirado de [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface), 14/06/2020]

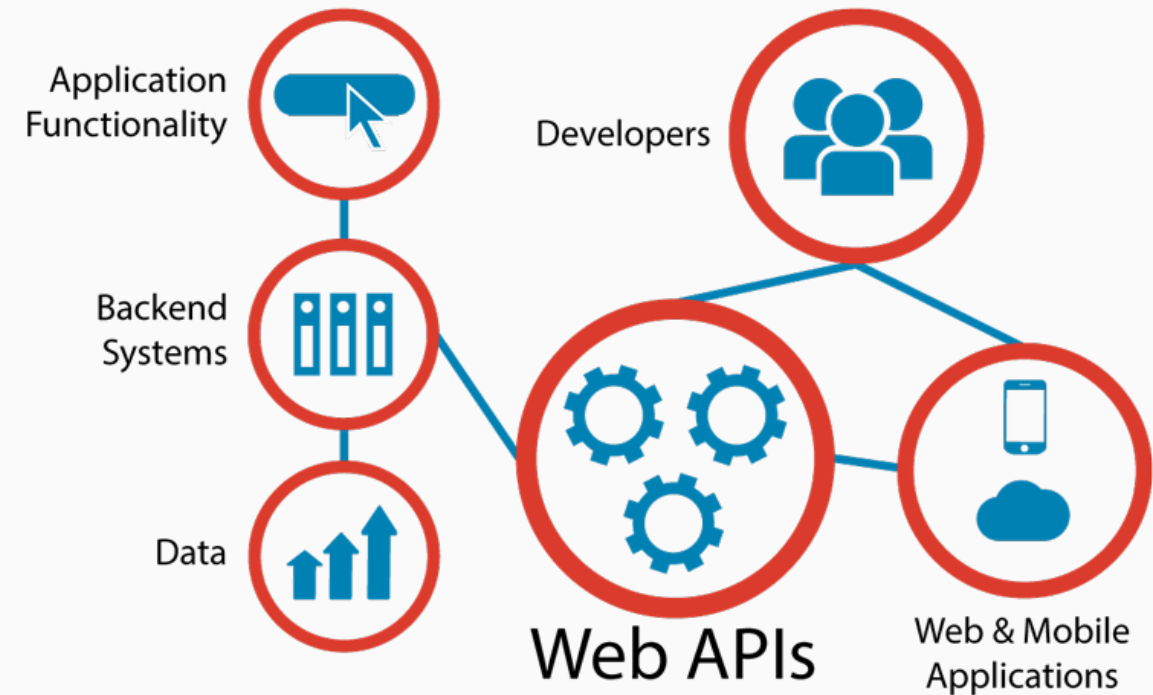


# API: Application Programming Interface

## Microservices – API Gateway Pattern API Gateway with Azure API Management Architecture



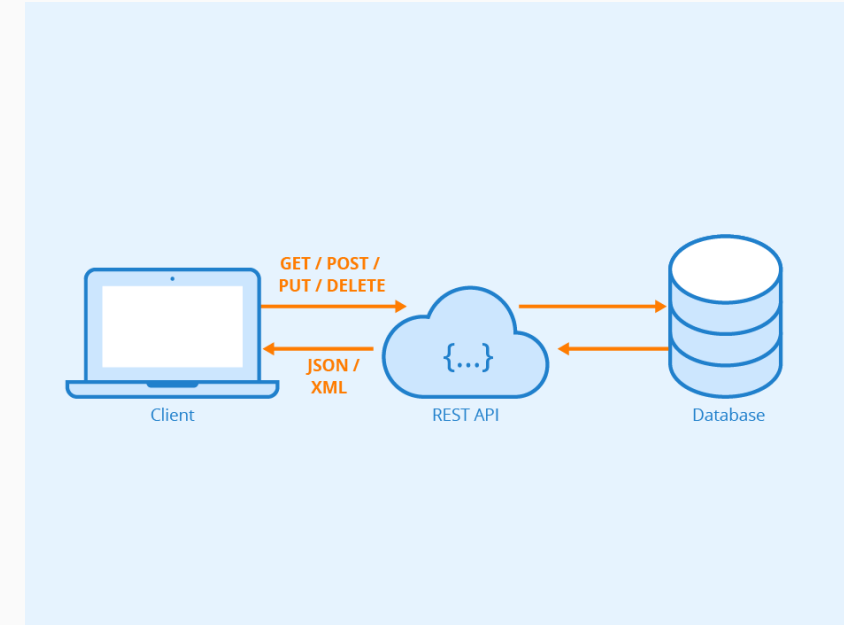
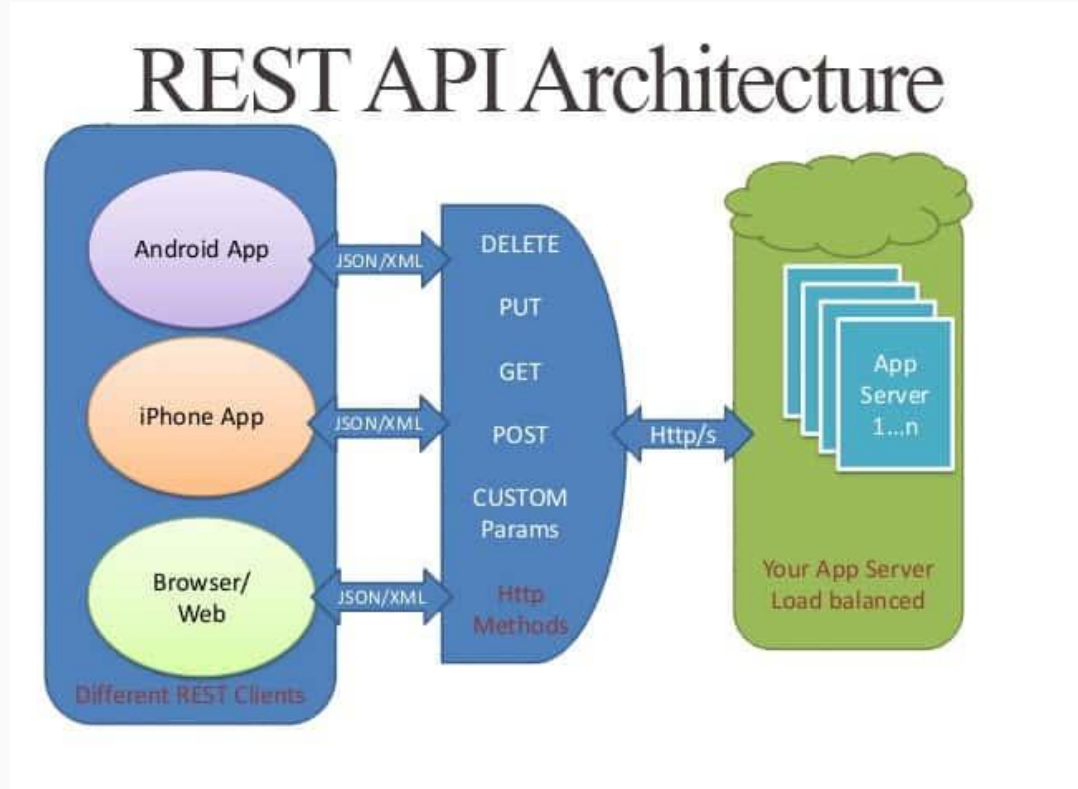
Retirado de (<https://image.slidesharecdn.com/netinsiderio-170921195925/95/net-inside-microservices-net-core-e-serverless-19-638.jpg?cb=1506024147>), em 14/06/2020



Retirado de ([https://nrelatorios.com.br/images/2017/09/20/web-apis-v5\\_0.png](https://nrelatorios.com.br/images/2017/09/20/web-apis-v5_0.png)), em 14/06/2020

- “API RESTful é uma interface que fornece dados em um formato padronizado baseado em requisições HTTP.” [Retirado de <https://www.hostgator.com.br/blog/api-restful/>, 14/06/2020]
- “Rest, que é a abreviatura de Representational State Transfer, é um conjunto de restrições utilizadas para que as requisições HTTP atendam as diretrizes definidas na arquitetura.” [Retirado de <https://rockcontent.com/blog/rest-api/>, em 14/06/2020]

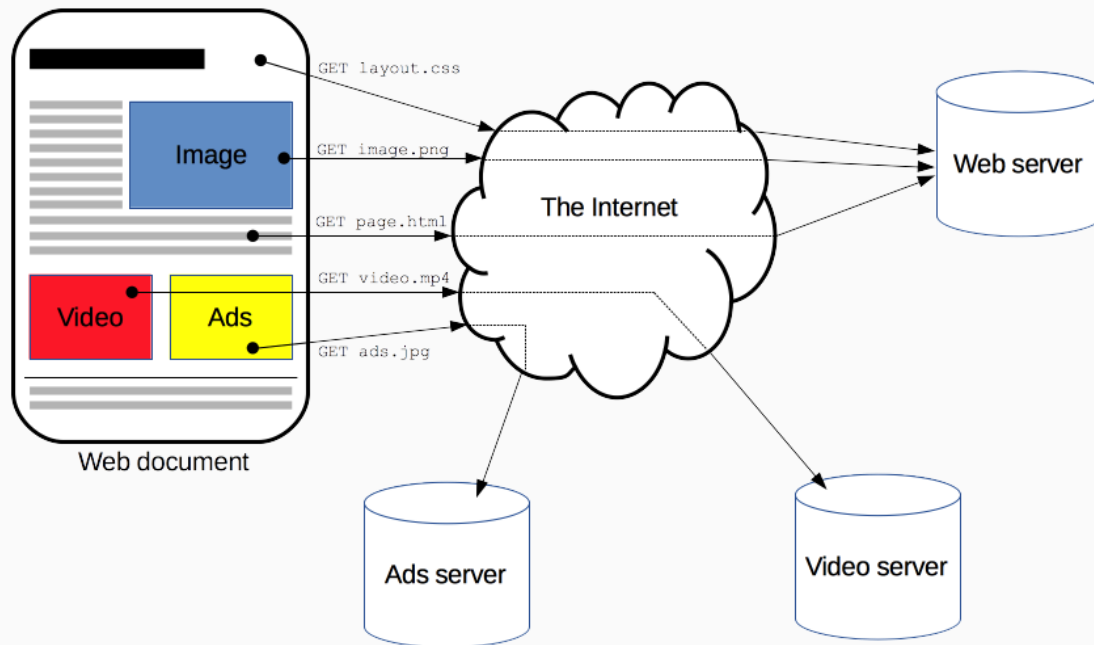
# REST: Representational State Transfer



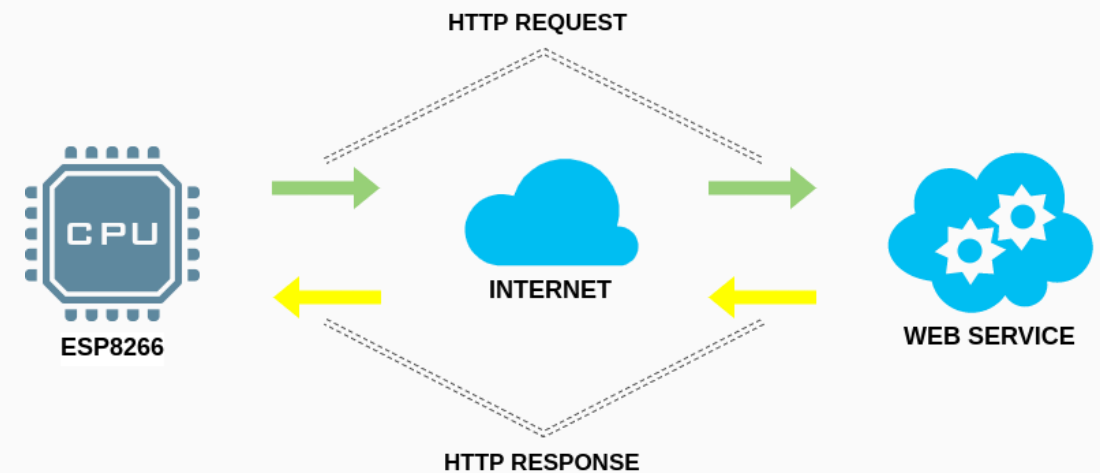
Retirado de (<https://www.astera.com/wp-content/uploads/2020/01/rest.png>), em 14/06/2020

Retirado de (<https://shareurcodes.com/photos//rest-api.jpg>), em 14/06/2020

# Requisições HTTP



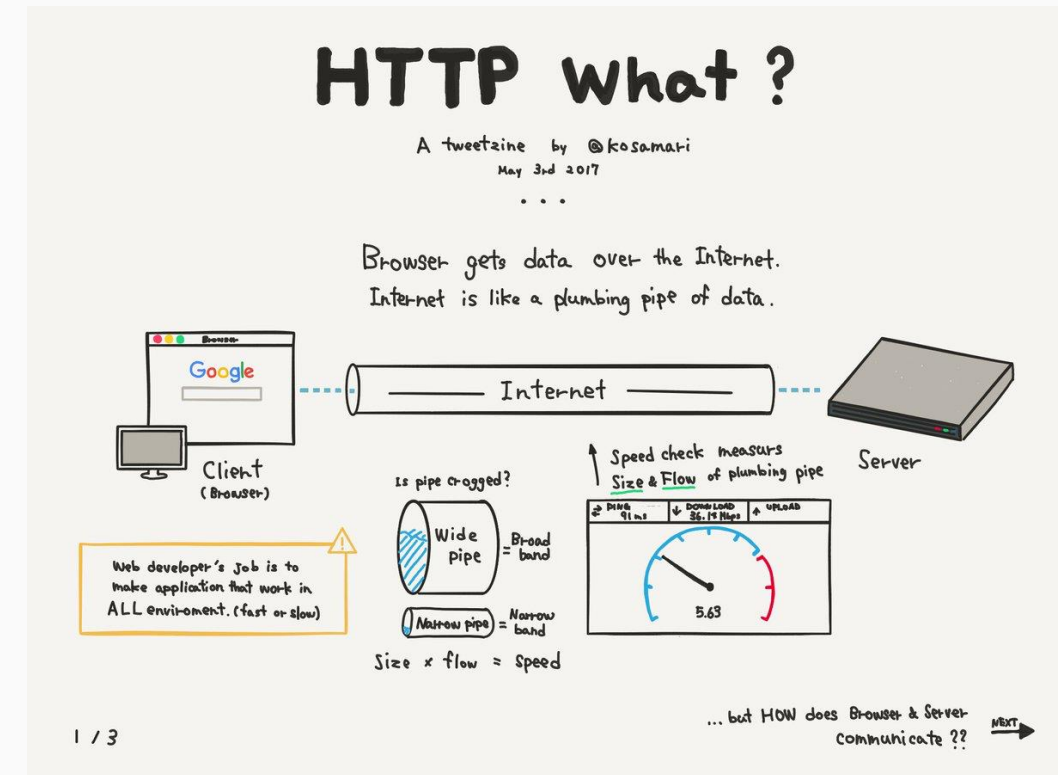
Retirado de  
[https://mdn.mozillademos.org/files/13677/Fetching\\_a\\_page.png](https://mdn.mozillademos.org/files/13677/Fetching_a_page.png), em 28/02/2019



Retirado de  
<https://raw.githubusercontent.com/douglaszuqueto/esp8266-http-request/master/files/esp8266-request.png>, em 16/06/2020

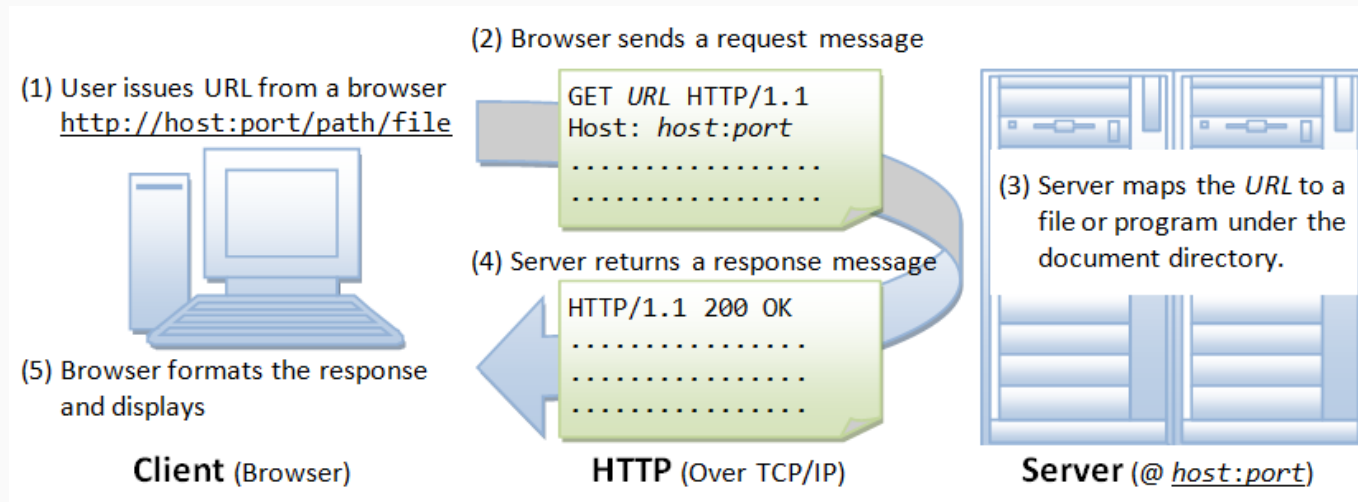


- O protocolo HTTP é a forma mais utilizada para trocar dados na internet.
- Maiores informações sobre seu funcionamento podem ser encontradas em:
  - <https://howhttps.works/why-do-we-need-https/>
  - <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>
  - [https://pt.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://pt.wikipedia.org/wiki/Hypertext_Transfer_Protocol)



Retirado de (<https://igluonline.com/wp-content/uploads/2018/12/como-funciona-http.jpg.jpg>), em 16/01/2020

- O protocolo utilizar a arquitetura cliente-servidor, onde o servidor responde as requisições feitas pelos clientes.
- O protocolo é do tipo pergunta-resposta.

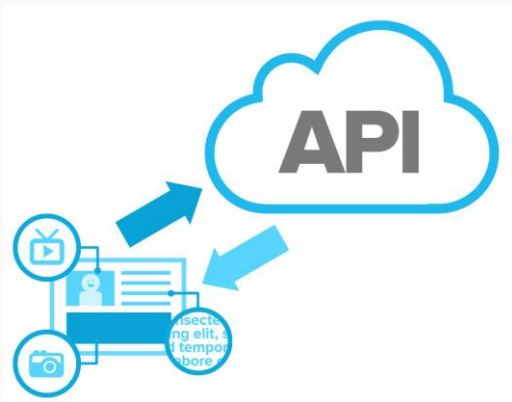


Retirado de  
([https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/images/HTTP\\_Steps.png](https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/images/HTTP_Steps.png)), em 28/02/2019

- Os métodos utilizados nas requisições HTTP são:

- **GET** – Utilizado para requisitar dados de um servidor ou serviço.
- **POST** – Envia dados para um servidor ou serviço. Os dados são enviados no corpo da requisição. Cria uma nova inserção de dados
- **PUT** – Funciona de forma similar ao POST, mas com a capacidade de atualizar os dados no servidor, sem ter que criar um novo registro.
- **DELETE** – Utilizado para excluir algum recurso
- **HEAD** – Variação do GET, mas sem retorno de dados
- **TRACE** – Verifica (eco) o caminho por onde o pacote anda.
- **OPTIONS** – Retorna quais métodos o servidor aceita.
- **CONNECT** – Cria uma conexão segura entre o cliente e o servidor.

- Formato mais utilizado para troca de mensagens entre serviços e programas/aplicações.
- É um formato bastante simples, que separa os valores da mensagem utilizando uma codificação chave:valor.



Retirado de  
(<https://ilovecode.com.br/wp-content/uploads/2016/05/Api.jpg>),  
em 28/02/2019



Retirado de  
(<https://camo.githubusercontent.com/1d51dc68a789839e809d7546692359305dd726e8/68747470733a2f2f7261772e6769746875622e636f6d2f6a646f726e2f6a736f6e2d656469746f722f6d61737465722f6a736f6e656469746f722e706e67>), em 28/02/2019



Clouds(overcast clouds) **18°C**

Pressure **1014.0 hPa**

Humidity **43.0%**

Wind **1.76 mps 266.003°**

```
{
  "coord":{"lon":12.4958,"lat":41.903},
  "sys":{"country":"Italy","sunrise":1369107818,
  "weather":[{"
    "id":802,"main":"Clouds","description":"
    "icon":"03d"}]},
  "base":"global stations",
  "main":{"
    "temp":290.38,
    "humidity":68,
    "pressure":1015,
    "temp_min":287.04,
```



A screenshot of a web browser window. The address bar shows the URL "http://services...et/iso2code/IN" with a red box highlighting the path "/country/get/iso2code/IN". A red arrow points from the text "REST API Call" to this highlighted path. The browser's main content area displays a JSON response. The JSON structure is as follows:

```
{
  "RestResponse" : {
    "messages" : [ "More webservises are available at http://www
/services.htm", "Country found matching code [IN]." ],
    "result" : {
      "name" : "India",
      "alpha2_code" : "IN",
      "alpha3_code" : "IND"
    }
  }
}
```

Retirado de  
([https://2.bp.blogspot.com/-kOQfsnZK3qI/WJ0SAAtOr6SI/AAAAAAAAAH5Q/UQYdu7QBXXwclqhQwhiiEFODc9Wo2fjzwCLcB/w1200-h630-p-k-no-nu/RESTful%2BWeb%2BSer vice%2BREsponse%2BSpring%2BRestTemplate%2BExample.png](https://2.bp.blogspot.com/-kOQfsnZK3qI/WJ0SAAtOr6SI/AAAAAAAAAH5Q/UQYdu7QBXXwclqhQwhiiEFODc9Wo2fjzwCLcB/w1200-h630-p-k-no-nu/RESTful%2BWeb%2BSer vice%2BREsponse%2BSprin g%2BRestTemplate%2BExample.png)), em 28/02/2019

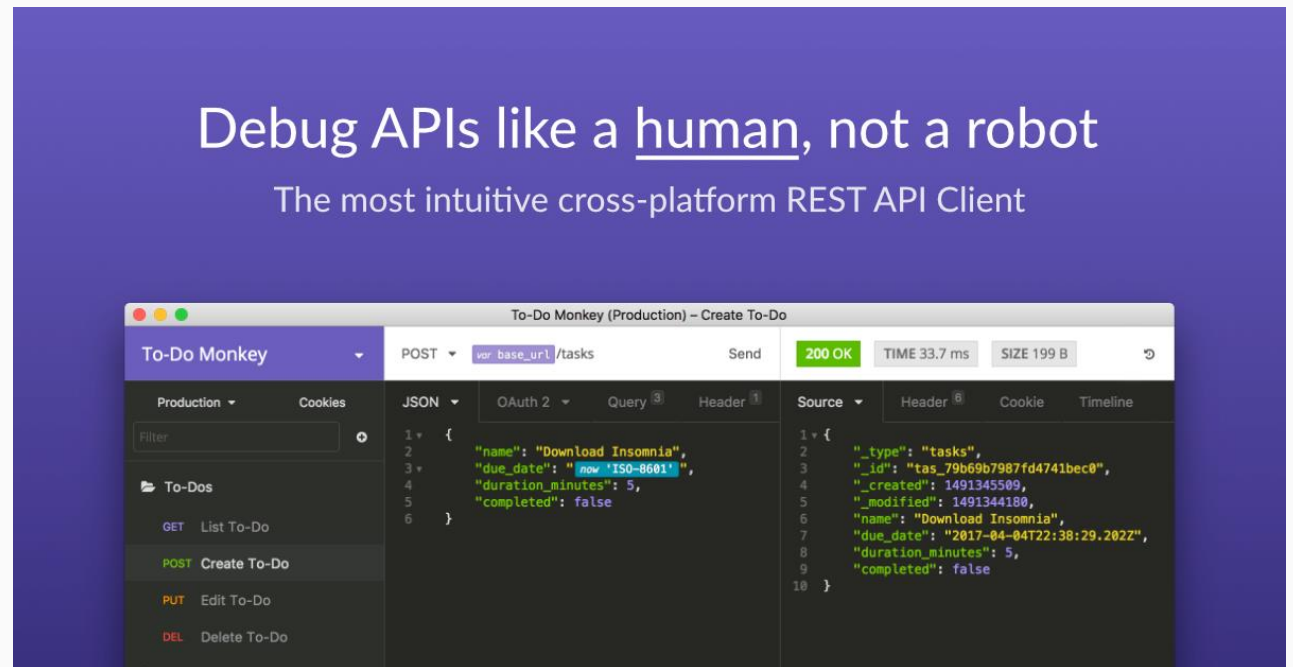
# JSON:

## E como utilizar esses elementos?

- Existem algumas ferramentas para realizar as requisições e testar a forma como os dados são realizados trocados utilizando as APIs.

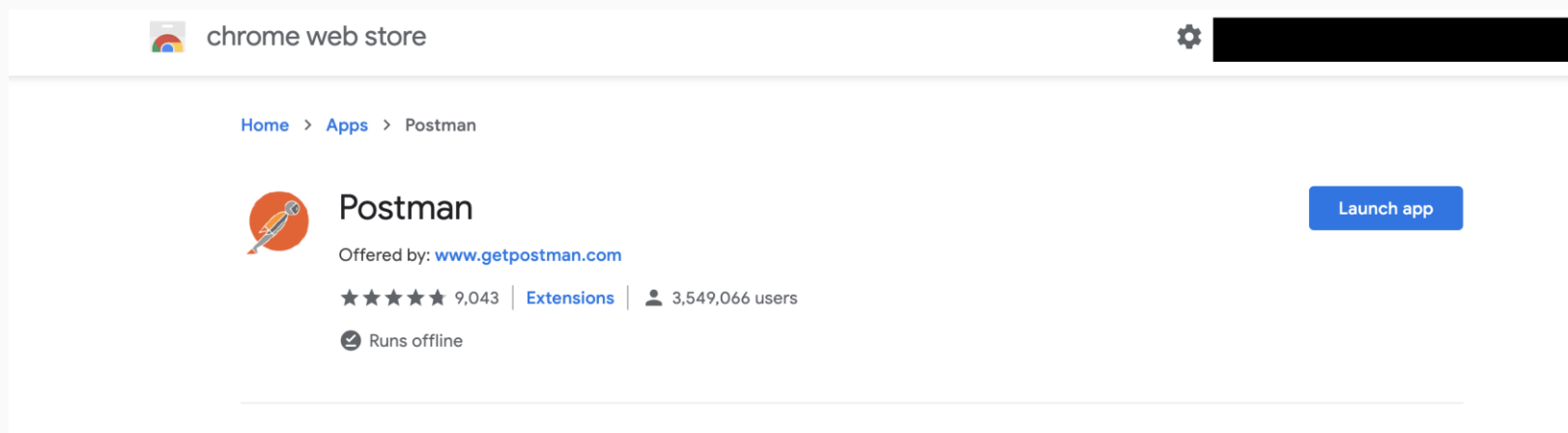


Retirado de  
(<https://www.getpostman.com/img/v2/logo-glyph.png>), em 16/01/2020

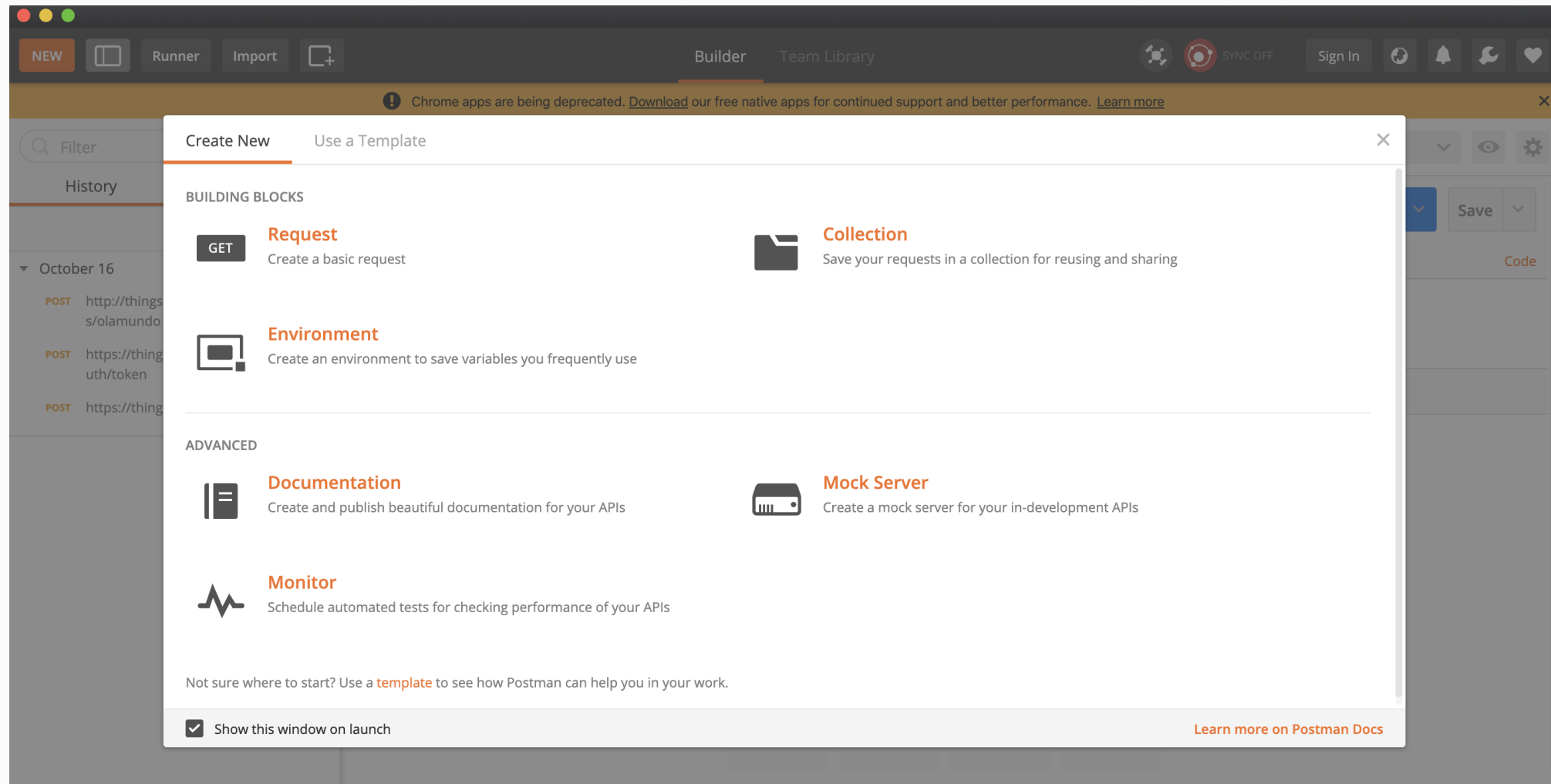


Retirado de  
(<https://insomnia.rest/images/twitter-promo.png>), em 16/01/2020

- O Postman pode ser instalado no navegador (Chrome App) ou utilizado como App nativo do sistema operacional. A versão Chrome App foi descontinuada, mas ainda pode ser baixada em (<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcdcdcbncdddomop?hl=en>).



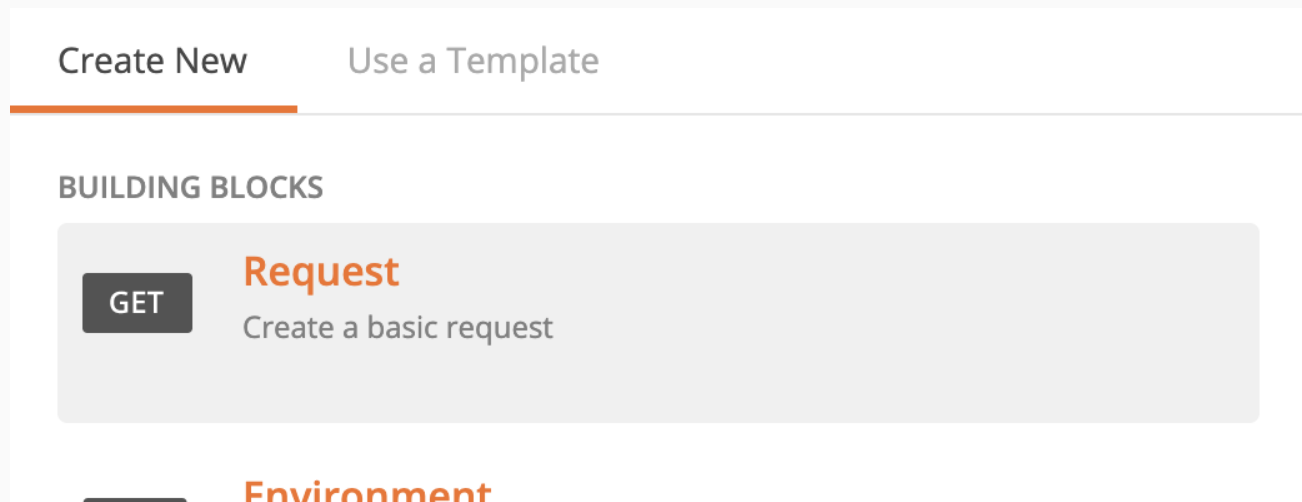
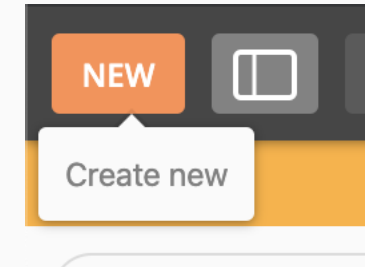
# JSON: Utilizando o POSTMAN no Navegador



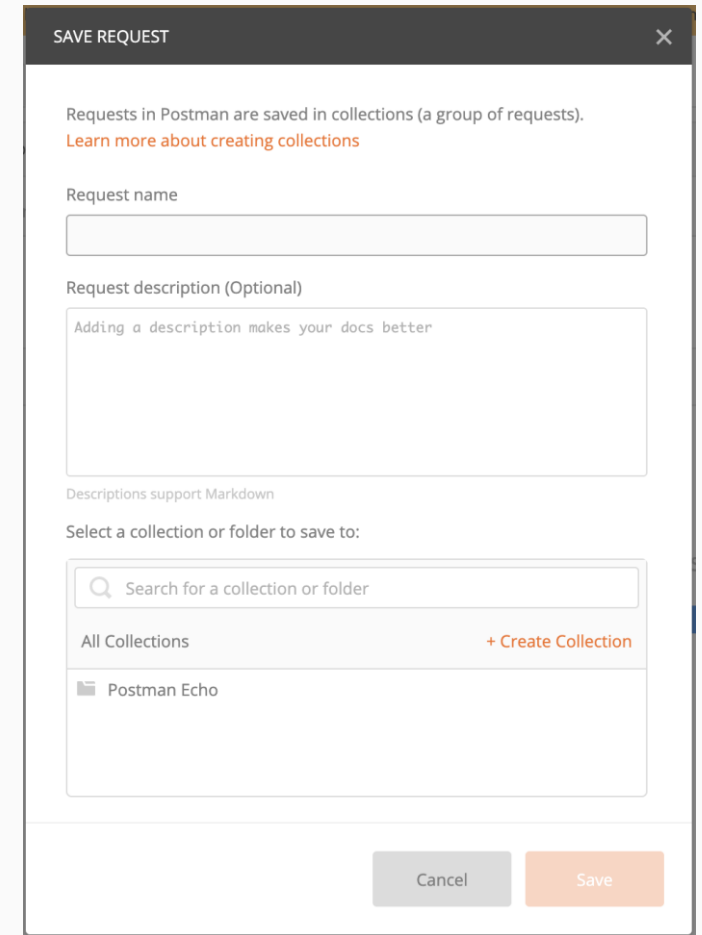
- Para verificar as funcionalidades do Postman, vamos utilizar ele para realizar algumas requisições para uma API dos Correios (<https://viacep.com.br>).
- A API do ViaCEP devolve um JSON com os dados do CEP enviado para ela por uma requisição GET. Se algum erro for encontrado, o JSON de retorno vai apresentar esse erro.



- Com o Postman, escolher o botão “*New*”.
- Selecionar a opção “Request”.



- É possível agrupar as requisições realizadas em coleções no Postman. Verificar a documentação para maiores detalhes.
- Dar um nome para a requisição e criar uma coleção para salvar elas.



SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).  
[Learn more about creating collections](#)

Request name

Request description (Optional)

Adding a description makes your docs better

Descriptions support Markdown

Select a collection or folder to save to:

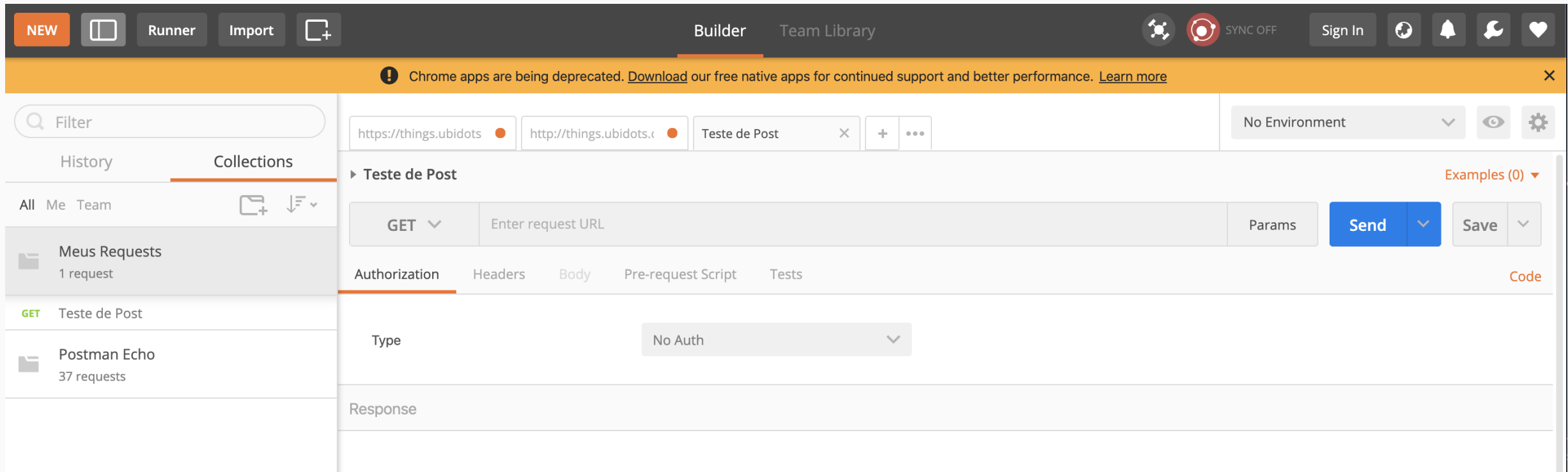
Search for a collection or folder

All Collections [+ Create Collection](#)

Postman Echo

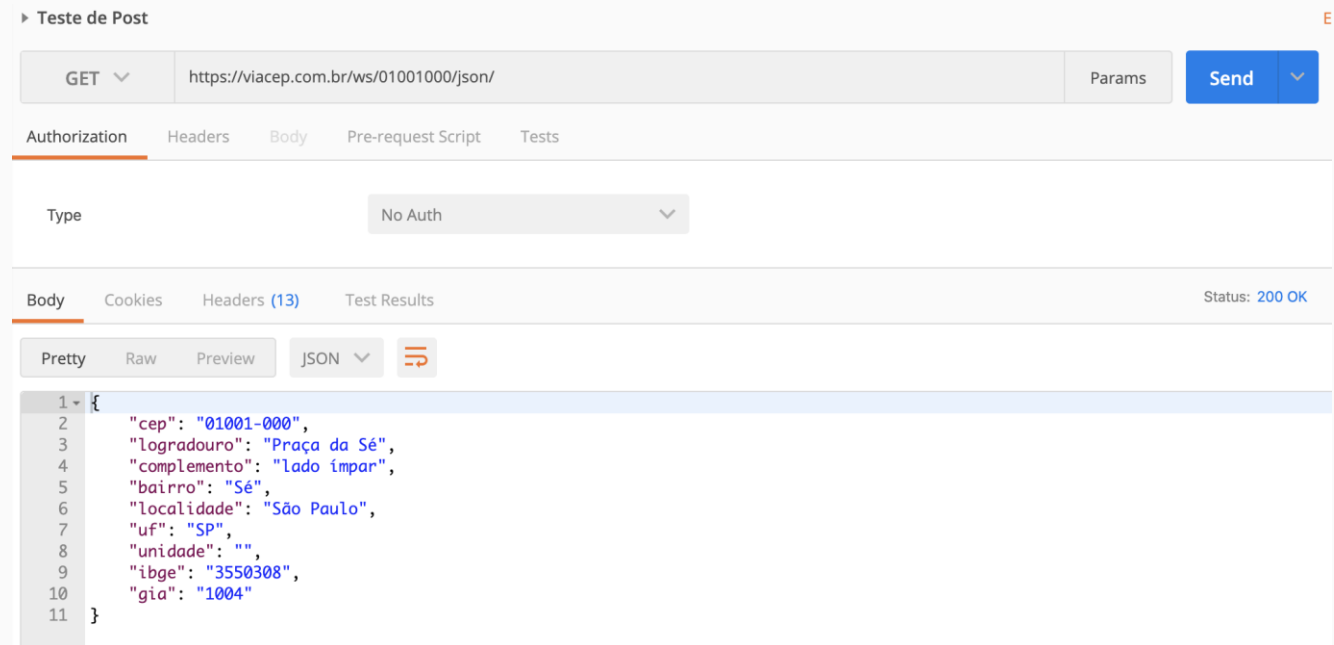
Cancel Save

# JSON: Utilizando o POSTMAN no Navegador

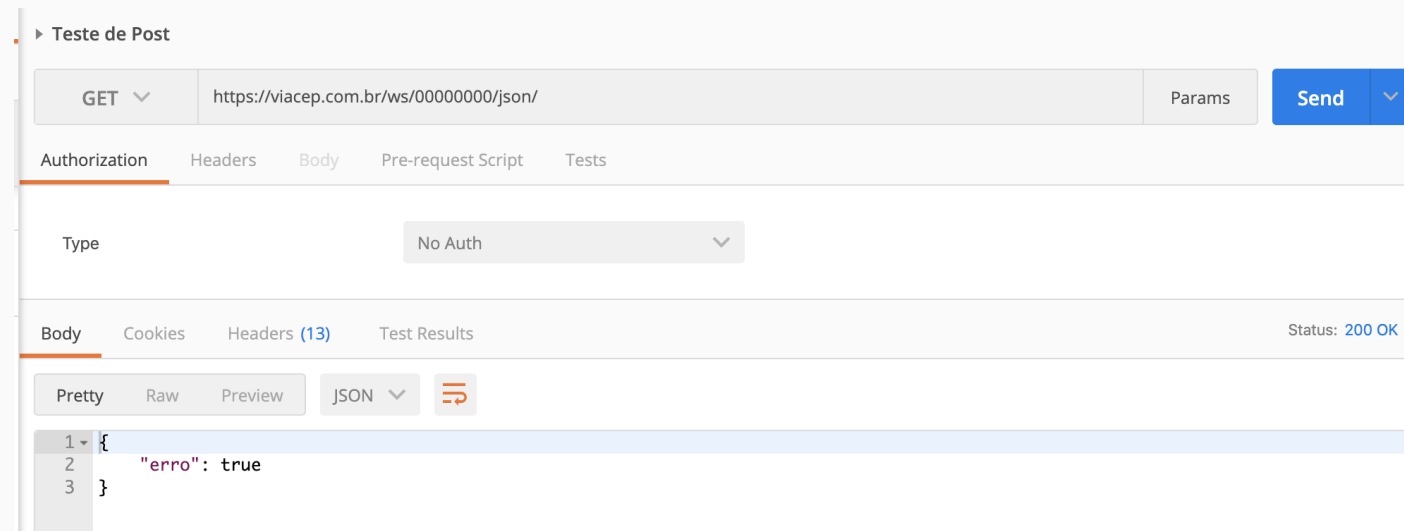


The screenshot displays the Postman web application interface. At the top, a dark navigation bar includes buttons for 'NEW', 'Runner', 'Import', and 'Builder', along with a 'Team Library' link and a 'Sign In' button. A yellow banner below the navigation bar contains a warning about Chrome apps being deprecated. The left sidebar features a 'Filter' search bar and two tabs: 'History' and 'Collections'. Under 'Collections', there are two folders: 'Meus Requests' (containing 1 request) and 'Postman Echo' (containing 37 requests). The main workspace shows a request named 'Teste de Post' selected. The request details panel on the right includes a dropdown for the HTTP method (currently 'GET'), a text input for the request URL, and buttons for 'Params', 'Send', and 'Save'. Below this, the 'Authorization' tab is active, showing a 'Type' dropdown set to 'No Auth'. The 'Response' tab is also visible at the bottom.

- Agora colocar a URL dentro do Postman para realizar a requisição.
- O botão SEND envia a requisição.

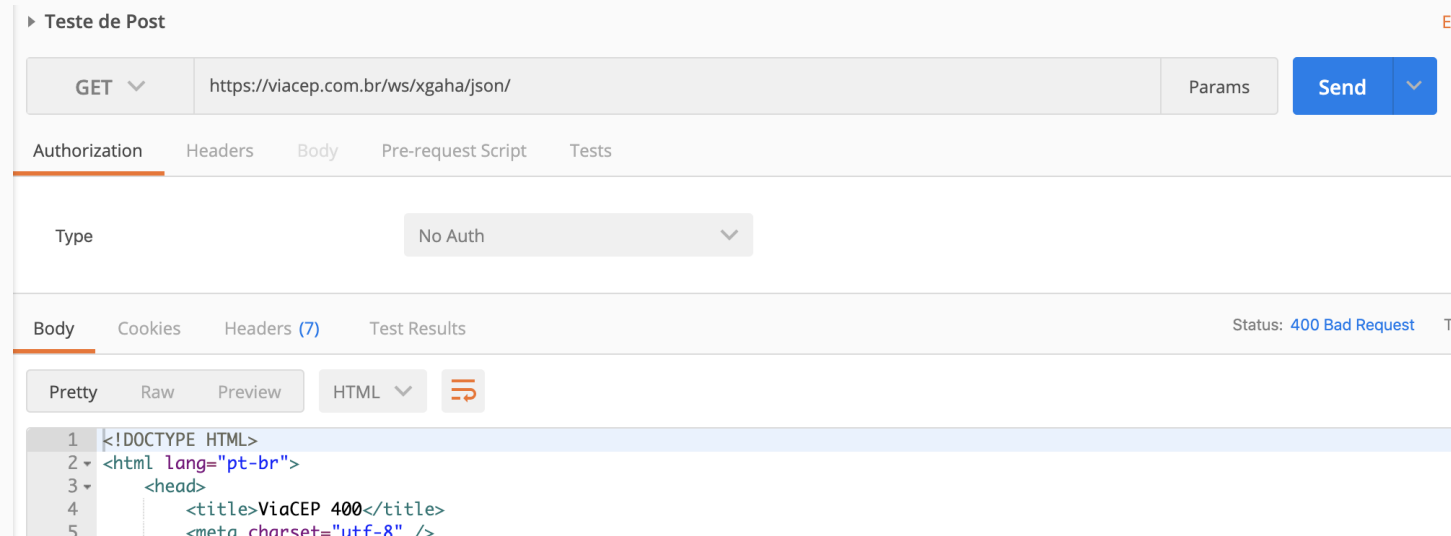


- Cada API é projetada de uma forma. No caso do ViaCEP, quando um CEP inválido, mas no formato correto, é enviado, ele retorna a seguinte informação.





- Quando uma requisição é realizada no formato incorreto, o código do status dela é diferente de 200.

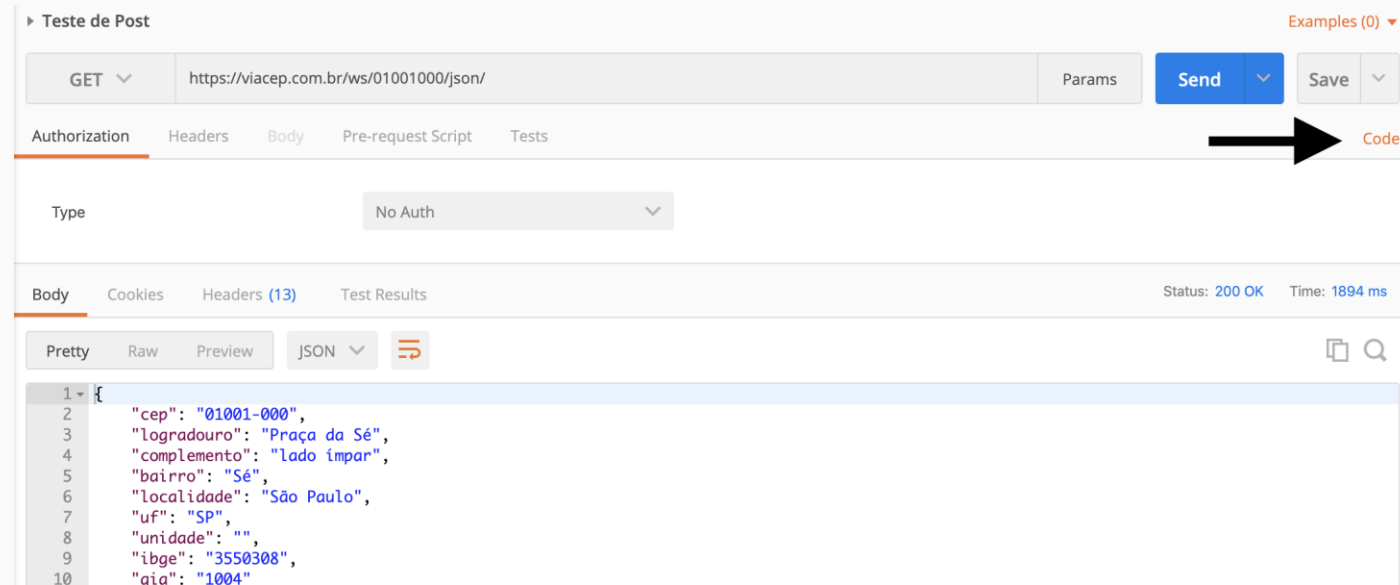


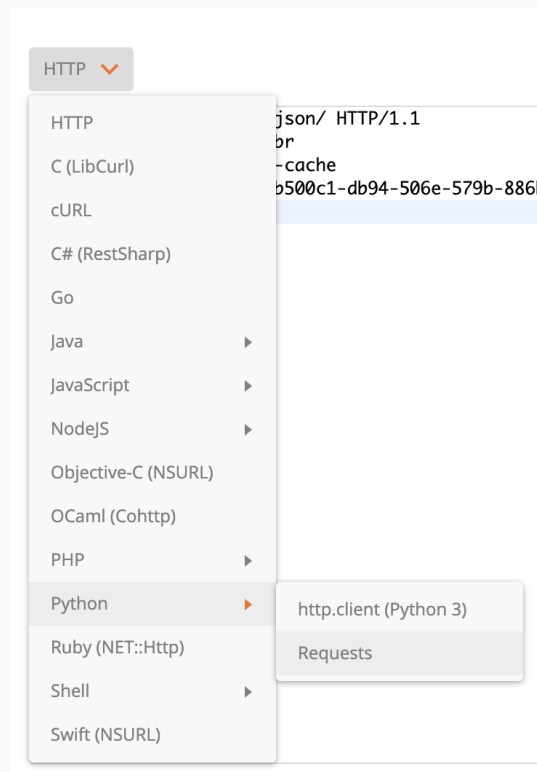
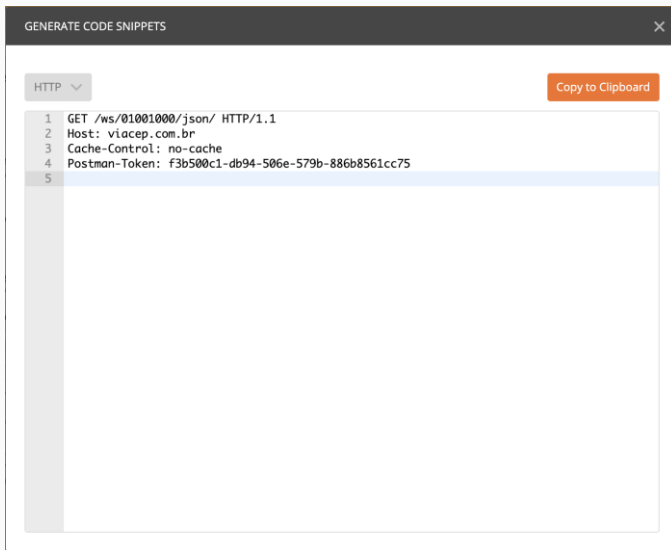
## HTTP Status Codes

| Code | Description        | Code | Description           |
|------|--------------------|------|-----------------------|
| 200  | OK                 | 400  | Bad Request           |
| 201  | Created            | 401  | Unauthorized          |
| 202  | Accepted           | 403  | Forbidden             |
| 301  | Moved Permanently  | 404  | Not Found             |
| 303  | See Other          | 410  | Gone                  |
| 304  | Not Modified       | 500  | Internal Server Error |
| 307  | Temporary Redirect | 503  | Service Unavailable   |

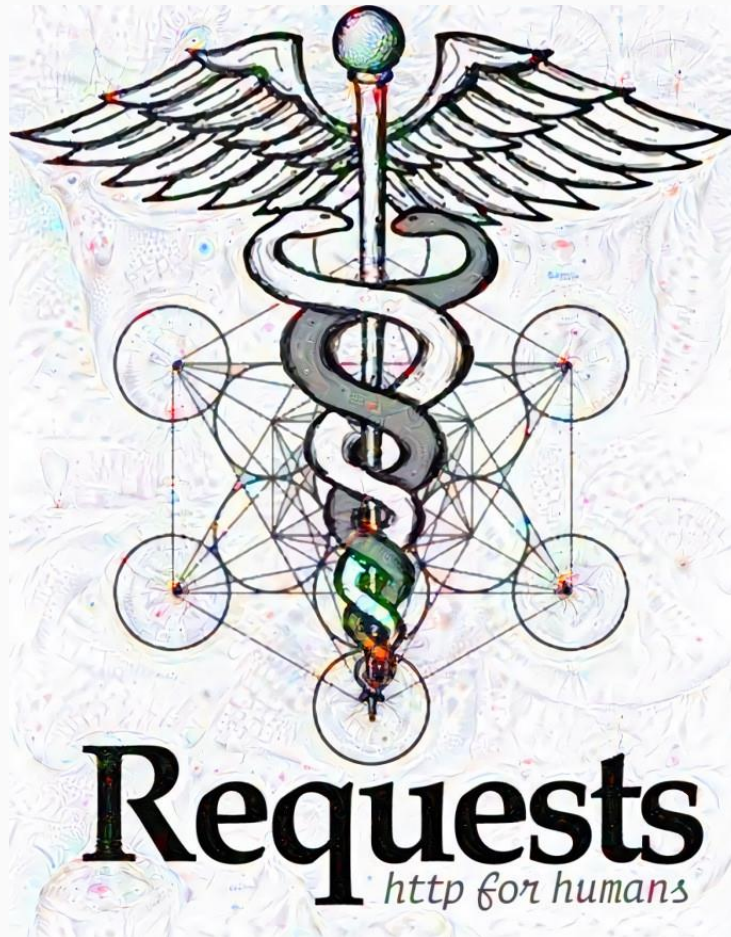
Retirado de  
(<http://image.slidesharecdn.com/restfulwebservicewithspringmvc-130630030659-phpapp02/95/restful-web-services-with-spring-mvc-28-638.jpg?cb=1372561944>),  
em 16/01/2020

- Um recurso do Postman é a geração de código da requisição realizada.





# Requests: Fazendo requisições HTTP com Python



- Biblioteca para realizar requisições HTTP (<https://requests.readthedocs.io/en/master/>).
- Já disponível no Colab para utilização.
- Facilita a utilização das requisições feitas a serviços e outros servidores.

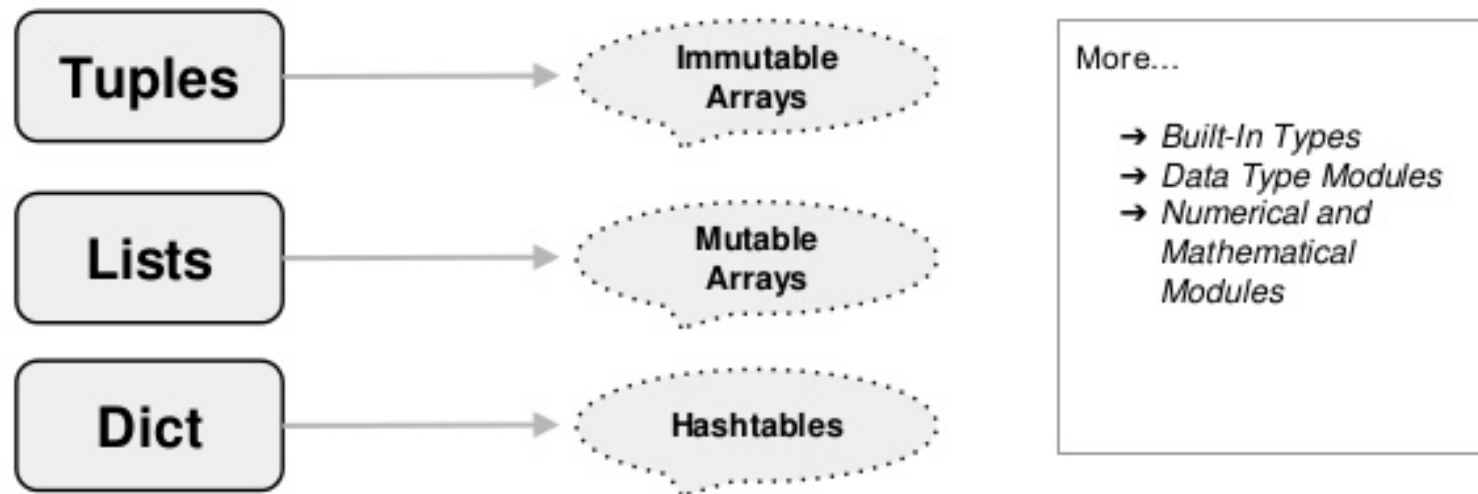


## Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type":"User"...}'
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

## Python Data Structures - Primer

*A Refresher to Python Data Structures*



Adaptado de (<https://image.slidesharecdn.com/pythondatastructures-bestinclassfordataanalysisandvisualization-160804022623/95/python-data-structures-best-in-class-for-data-analysis-3-638.jpg?cb=1470278367>), em 14/06/2020

# Observação Rápida: Dicionários em Python

```
1 meu_dicionario = {"chave1": 45.8, "chave2":89, "chave3":-9.2}
2
3 for chave in meu_dicionario.keys():
4     print("Minha chave: %s - Meu valor: %.3f" % (chave, meu_dicionario[chave]))
5
6 meu_dicionario["novaChave"] = 78.2
7 meu_dicionario["chave1"] = 67.3
8
9 print(meu_dicionario)
10
11 if "chave1" in meu_dicionario.keys():
12     print("chave1 estava aqui!")
13
14 try:
15     print(meu_dicionario["chave que nao existe"])
16 except:
17     print("Não é possível acessar chaves inválidas!")
18
19 print(meu_dicionario.get("chave99", "VALOR PADRÃO"))
```

```
Minha chave: chave1 - Meu valor: 45.800
Minha chave: chave2 - Meu valor: 89.000
Minha chave: chave3 - Meu valor: -9.200
{'chave1': 67.3, 'chave2': 89, 'chave3': -9.2, 'novaChave': 78.2}
chave1 estava aqui!
Não é possível acessar chaves inválidas!
VALOR PADRÃO
```



**“Talk is  
cheap. Show  
me the code.”**

## **Linus Torvalds**

Retirado de (<https://kwize.com/pics/Linus-Torvalds-quote-about-talking-1a9797.jpg>), em 08/08/2019

- O Google disponibilizou uma ferramenta chamada CoLab, que permite que desenvolvedores possam escrever código de maneira colaborativa.
- Link:  
<https://colab.research.google.com/>



- O permite site (Ergast Developer API) utilizar informações estruturadas sobre corridas de formula 1.
- Link: <https://ergast.com/mrd/>



## API Documentation

The Ergast Developer API is an experimental [web service](#) which provides a historical record of motor racing data for non-commercial purposes. Please

## Index

[API Documentation](#)  
[Season List](#)

- Verificar os resultados obtidos em:  
<http://ergast.com/api/f1/2018/20/results>

### Query Results

| Query Details |        |       |         |
|---------------|--------|-------|---------|
| Series        | Season | Round | Results |
| f1            | 2018   | 20    | 20      |

| 2018 Brazilian Grand Prix |    |                |             |      |      |             |          |    |
|---------------------------|----|----------------|-------------|------|------|-------------|----------|----|
| Race Results              |    |                |             |      |      |             |          |    |
| Pos                       | No | Driver         | Constructor | Laps | Grid | Time        | Status   | Po |
| 1                         | 44 | Lewis Hamilton | Mercedes    | 71   | 1    | 1:27:09.066 | Finished | 25 |
| 2                         | 33 | Max Verstappen | Red Bull    | 71   | 5    | +1.469s     | Finished | 18 |
|                           |    | Kimi           |             |      |      |             |          |    |

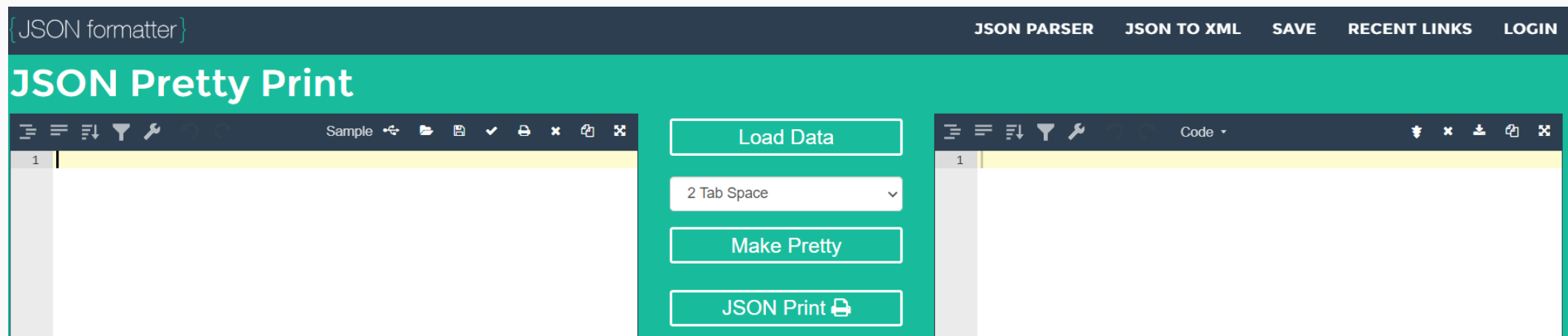


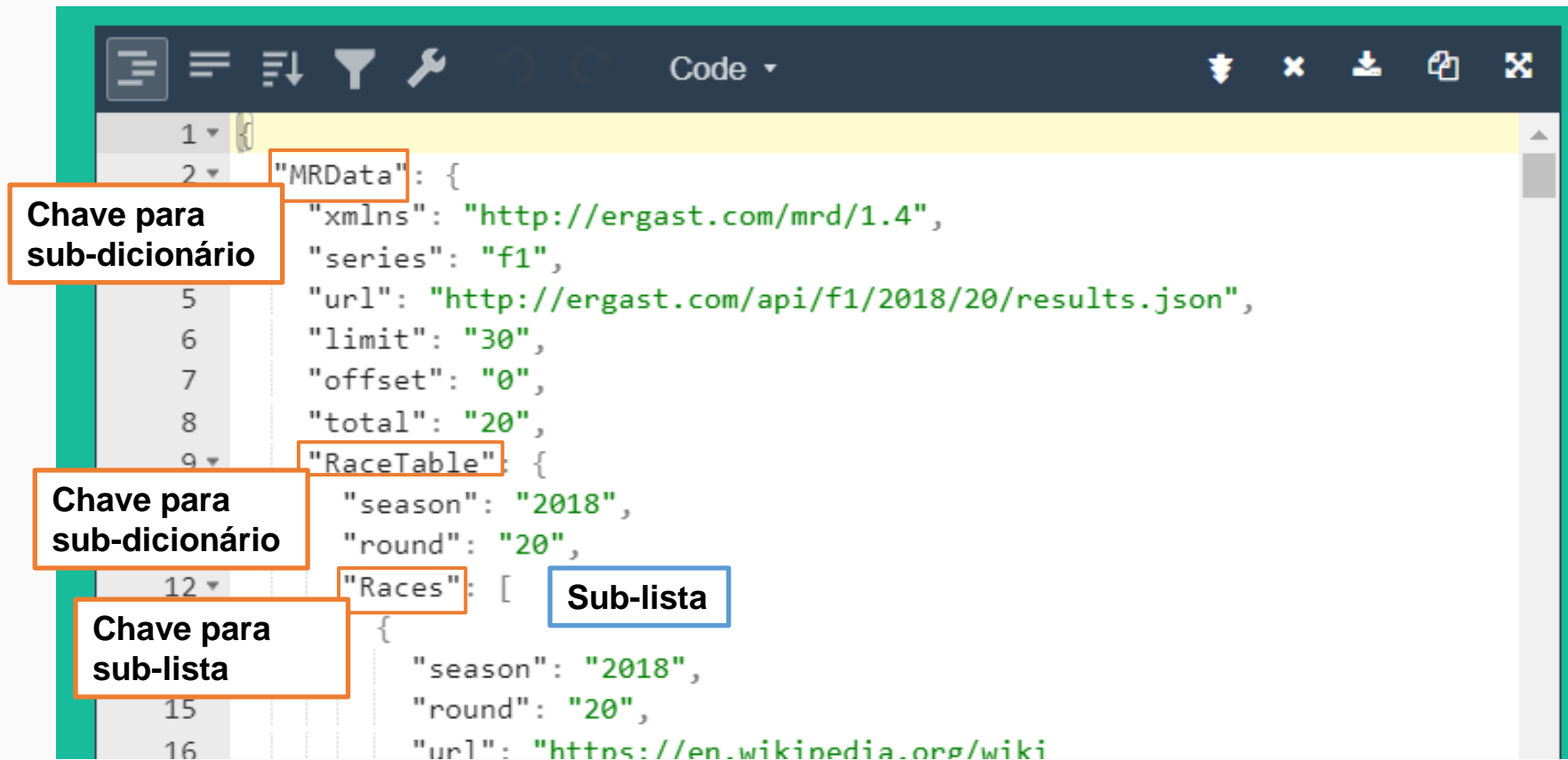
- Agora vamos trabalhar com esses dados no nosso programa:

<http://ergast.com/api/f1/2018/20/results.json>

```
{ "MRData":
  { "xmlns": "http://ergast.com/mrd/1.4", "series": "f1", "url": "http://ergast.com/api/f1/2018/20/results.json", "limit": "30", "offset": "0", "total": "20", "RaceTable": { "season": "2018", "round": "20", "Races":
    [ { "season": "2018", "round": "20", "url": "https://en.wikipedia.org/wiki/2018_Brazilian_Grand_Prix", "raceName": "Brazilian Grand Prix", "Circuit":
      { "circuitId": "interlagos", "url": "http://en.wikipedia.org/wiki/Aut%C3%B3dromo_Jos%C3%A9_Carlos_Pace", "circuitName": "Autódromo José Carlos Pace", "Location": { "lat": "-23.7036", "long": "-46.6997", "locality": "São Paulo", "country": "Brazil" } }, "date": "2018-11-11", "time": "17:10:00Z", "Results":
        [ { "number": "44", "position": "1", "positionText": "1", "points": "25", "Driver":
            { "driverId": "hamilton", "permanentNumber": "44", "code": "HAM", "url": "http://en.wikipedia.org/wiki/Lewis_Hamilton", "givenName": "Lewis", "familyName": "Hamilton", "dateOfBirth": "1985-01-07", "nationality": "British" }, "Constructor":
```

- Diversas ferramentas on-line possibilitam visualizar melhor os dados obtidos de requisições JSON, como: <https://jsonformatter.org/json-pretty-print>



A screenshot of a code editor window with a dark theme. The editor displays a JSON object. On the left side of the editor, there is a line number column with numbers 1 through 16. The JSON code is as follows:

```
1 {  
2   "MRData": {  
3     "xmlns": "http://ergast.com/mrd/1.4",  
4     "series": "f1",  
5     "url": "http://ergast.com/api/f1/2018/20/results.json",  
6     "limit": "30",  
7     "offset": "0",  
8     "total": "20",  
9     "RaceTable": {  
10      "season": "2018",  
11      "round": "20",  
12      "Races": [  
13        {  
14          "season": "2018",  
15          "round": "20",  
16          "url": "https://en.wikipedia.org/wiki"
```

Annotations are present: three orange boxes on the left with the text "Chave para sub-dicionário" pointing to the "MRData" key (line 2), the "RaceTable" key (line 9), and the "Races" key (line 12); and one blue box on the right with the text "Sub-lista" pointing to the "Races" array (line 12).

- Manipulando os índices do resultado obtido, conseguimos chegar nas informações necessárias para nossa aplicação.
- Em geral as APIs permitem obter os dados de forma parcial.

```
object ► MRData ► RaceTable ► Races ► 0 ► Results ► 0 ► Driver ►
```

```
driverId : hamilton
```

```
object ► MRData ► RaceTable ► Races ► 0 ► Results ► 8 ► Driver ►
```

```
driverId : kevin_magnussen
```

- Exibir a colocação e o nome dos pilotos da corrida 20, de 2018.

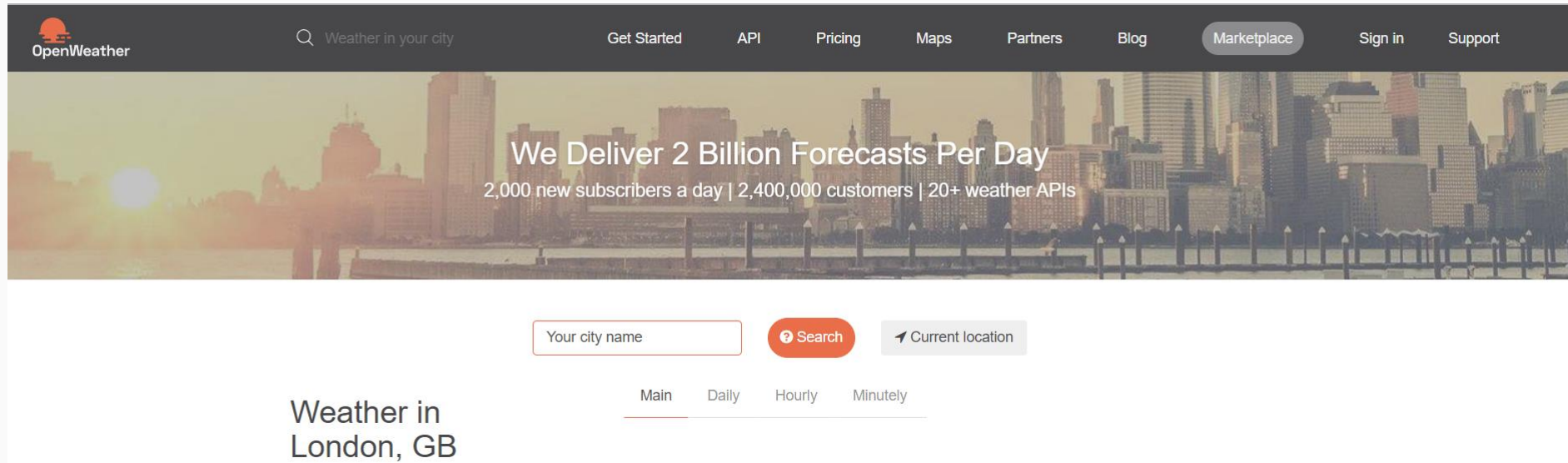
```
1 import requests
2
3 URL = "http://ergast.com/api/f1/2018/20/results.json"
4
5 resultado = requests.get(URL)
6
7 if resultado.status_code == 200:
8     print("Classificação dos Pilotos:")
9     for piloto in resultado.json()["MRData"]["RaceTable"]["Races"][0]["Results"]:
10         print("%s - %s" % (piloto["position"], piloto["Driver"]["driverId"]))
```

Disponível em: <https://gist.github.com/Murilo-ZC/ed5f1516e37c494b1382068c221f64f6>

```
Classificação dos Pilotos:
1 - hamilton
2 - max_verstappen
3 - raikkonen
4 - ricciardo
5 - bottas
6 - vettel
7 - leclerc
8 - grosjean
9 - kevin_magnussen
10 - perez
11 - brendon_hartley
12 - sainz
13 - gasly
14 - vandoorne
15 - ocon
16 - sirotkin
17 - alonso
18 - stroll
19 - hulkenberg
20 - ericsson
```

- Por questões estratégicas, de recursos disponíveis ou mesmo de segurança, diversas API não disponibilizam seu acesso sem uma CHAVE DE ACESSO.
- A forma como a chave de acesso pode ser gerada varia de acordo com o serviço que está sendo utilizado.

- API com Registro: <https://openweathermap.org/>





We have sent the confirmation link to [murilo.carvalho@maua.com](mailto:murilo.carvalho@maua.com). Please check your email.

[New Products](#) [Services](#) [API keys](#) [Billing plans](#) [Payments](#) [Block logs](#) [My orders](#) [My profile](#)



## Historical weather for any location

Our new technology, Time Machine, has allowed us to enhance the data in the [Historical Weather Collection](#).

[New Products](#) [Services](#) [API keys](#) [Billing plans](#) [Payments](#) [Block logs](#) [My orders](#) [My profile](#)

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key

Name

Create key



Default



API key name

Generate

- Na aba API, é possível ver que existem algumas formas de realizar a requisição dos dados de temperatura.
- Por exemplo:  
<https://openweathermap.org/current>
- Formato da requisição:  
`api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={your api key}`

### By geographic coordinates

API call:

```
api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={your api key}
```

Parameters:

**lat, lon** coordinates of the location of your interest

Examples of API calls:

```
api.openweathermap.org/data/2.5/weather?lat=35&lon=139
```

API response:

```
{ "coord": { "lon": 139, "lat": 35 },  
  "weather": [  
    {  
      "id": 800,  
      "main": "Clear",  
      "description": "clear sky",  
      "icon": "01n"  
    }  
  ],  
  "base": "stations",  
  "main": {  
    "temp": 281.52,
```

- Ainda temos um problema, precisamos dos dados da latitude e da longitude do local. Como podemos resolver isso? Utilizando outra API!
- Verificar o portal: <https://cepaberto.com/>

[CEP Aberto](#) [Doar](#) [Colaborar](#) [Login](#) [Registrar-se](#)

## CEP Aberto

É um projeto que visa prover acesso gratuito e construir de maneira colaborativa uma base de dados com os Codigos de Enderecamento Postal (CEP) **geolocalizados** de todo o Brasil. [Colabore](#) melhorando a informação de CEPs que você conhece. [Registre-se](#) gratuitamente para ter acesso a API e poder baixar a base de dados. O CEP Aberto possui informação de 1137130 CEPs distribuídos em 10659 cidades e municípios.

[Compartilhar](#)

### O problema do CEP no Brasil

Independente da [requisição popular \(retirada do ar\)](#) pela abertura da base de dados de CEP, a mesma continua fechada. Isto significa que somente mediante [pagamento de licença](#), as pessoas podem ter acesso a um conjunto de dados que deveria ser público. Ou seja, a base de dados de CEPs é vendida como um produto monopolizado pelos Correios.



[Colabore](#)

```
// our motto
if not cep.open()
  cep.aberto.run()
end
```

[Use a API](#)



[Baixe a base de dados](#)

- Depois de fazer o registro no portal CEP aberto, podemos utilizar sua API para pegar os dados de latitude e longitude com base em um CEP informado.

### Busca do CEP pelo número

Dado um número de CEP busca os dados respectivos

- GET `https://www.cepaberto.com/api/v3/cep`
- Parametro aceito `cep`

Exemplo de busca pelos dados do CEP `01001000` em Python:

```
import requests

url = "https://www.cepaberto.com/api/v3/cep?cep=01001000"
# O seu token está visível apenas pra você
headers = {'Authorization': 'Token token=...'}
response = requests.get(url, headers=headers)

print(response.json())
```

### Resultado

Para o exemplo anterior, o resultado será:

```
{"cidade": {"ibge": "3550308", "nome": "São Paulo", "ddd": 11}, "estado": {"sigla": "SP"}, "altitude": 760
```

- Agora com as duas informações, integrar as duas APIs para exibir a temperatura atual de um local com base em um CEP.

```
1  import requests
2
3  url = "https://www.cepaberto.com/api/v3/cep?cep=09051110"
4  headers = {'Authorization': 'Token token=COLOCAR_SUA_APIKEY_CEPABERTO'}
5  response = requests.get(url, headers=headers)
6
7  lat = response.json()["latitude"]
8  lon = response.json()["longitude"]
9
10
11  URL = "http://api.openweathermap.org/data/2.5/weather?lat=%s&lon=%s&appid=COLOCAR_SUA_APIKEY_OPENWEATHER" % (lat, lon)
12
13
14  resultado = requests.get(URL)
15
16  if resultado.status_code == 200:
17      print(float(resultado.json()['main']['temp'])-273.18)
```

Disponível em:

<https://gist.github.com/Murilo-ZC/112e6340063a9b73b05f70c34112b052>