



APIS WEB

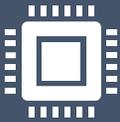
prof. Me. Hélio Esperidião



API - *Application Programming Interface*



API significa Interface de Programação de Aplicações. É um conjunto de definições e protocolos que permite a comunicação entre diferentes softwares.



Uma API define como diferentes componentes de software devem interagir uns com os outros.



Uma API pode ser entendida como uma ponte que permite que diferentes sistemas, aplicativos ou serviços se comuniquem e compartilhem recursos e funcionalidades entre si.

APIs

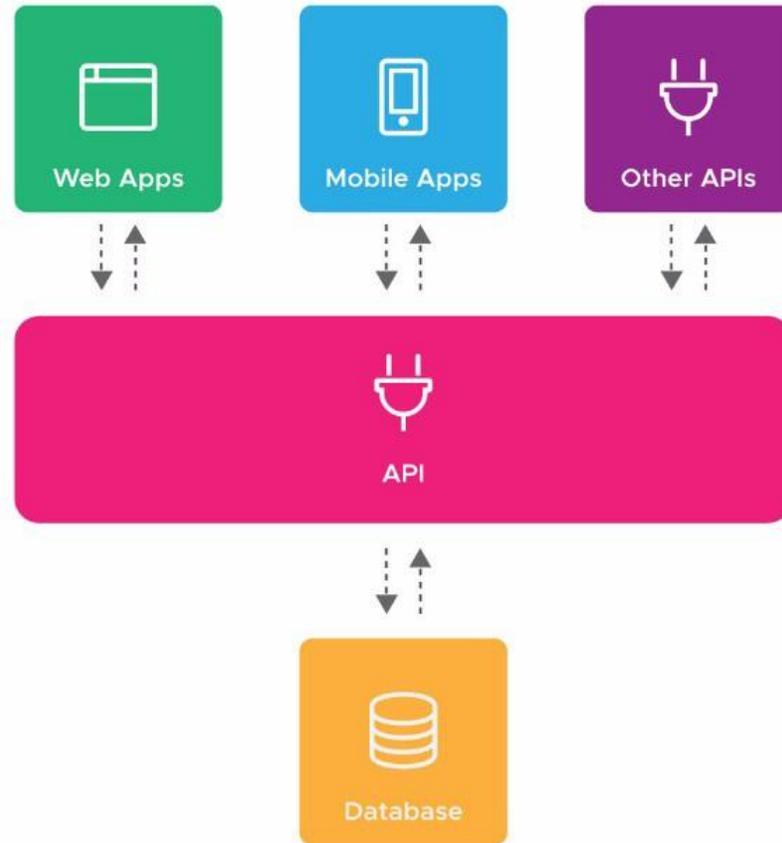


API é um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos.



Por meio de APIs, desenvolvedores podem criar novos softwares e aplicativos capazes de se comunicar com outras plataformas

Esquema de funcionamento de apis



JSON - JavaScript Object Notation

É uma formatação leve de troca de dados.

Para seres humanos, é fácil de ler e escrever.

Para máquinas, é fácil de interpretar e gerar.

É baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999.

JSON

01

JSON é em formato texto e completamente independente de linguagem

02

Formato ideal de troca de dados

03

é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida (parsing) entre sistemas

ESTRUTURA

Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, record, struct, dicionário, hash table, keyed list, ou arrays associativas.

Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência.

Exemplos de json

```
{"altura":57.7,"largura":55}
```

```
{"nome":"helio","cargo":"professor"}
```

Como Criar JSON em PHP com a classe StdClass

- O que é StdClass?
 - É uma classe genérica vazia do PHP
 - Ideal para criar objetos de forma simples e dinâmica
 - Muito usada para montar dados antes de converter para JSON

Criando um Objeto JSON Simples

```
<?php
$obj = new stdClass();
$obj->nome = "João";
$obj->idade = 30;
$obj->ativo = true;

$json = json_encode($obj);
echo $json;

?>
```



```
{
  "nome": "João",
  "idade": 30,
  "ativo": true
}
```

Estruturas mais elaboradas

```
<?php
$obj = new stdClass();
$obj->nome = "Maria";
$obj->email = "maria@email.com";
$obj->cargo = new stdClass();
    $obj->cargo->id = 1;
    $obj->cargo->nome = "Técnico em Informática";
echo json_encode($obj);
?>
```

```
{
  "nome": "Maria",
  "email": "maria@email.com",
  "cargo": {
    "id": 1,
    "nome": "Técnico em Informática"
  }
}
```

Observe que há uma hierarquia de objetos

Vetor de objetos JSON

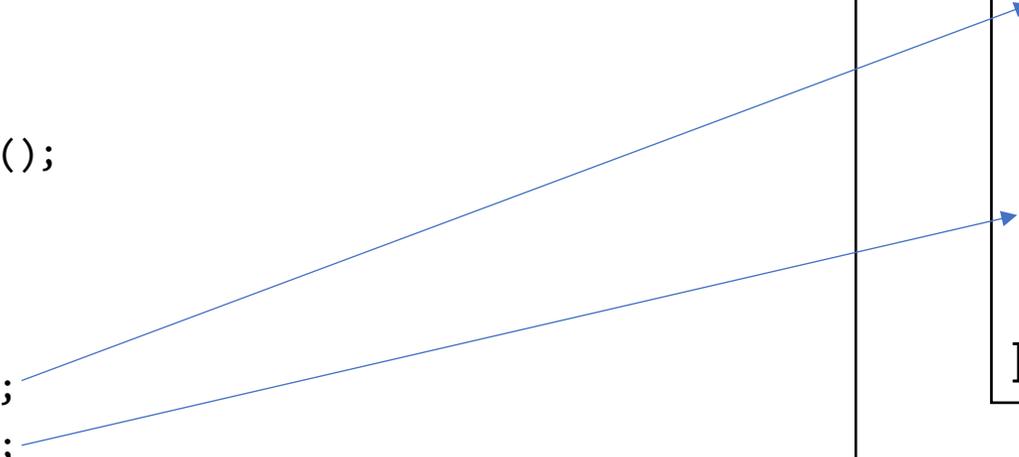
```
<?php
$usuarios = [];
$u1 = new stdClass();
$u1->nome = "Carlos";
$u1->idade = 28;

$u2 = new stdClass();
$u2->nome = "Ana";
$u2->idade = 32;

$usuarios[0] = $u1;
$usuarios[1] = $u2;

echo json_encode($usuarios);
?>
```

```
[
  {
    "nome": "Carlos",
    "idade": 28
  },
  {
    "nome": "Ana",
    "idade": 32
  }
]
```



Aplicações modernas.



Tipicamente em uma aplicação moderna o frontend que não utiliza o **get** ou **delete** envia um json com os dados para um controle que utiliza as classes necessárias para resolver o problema.

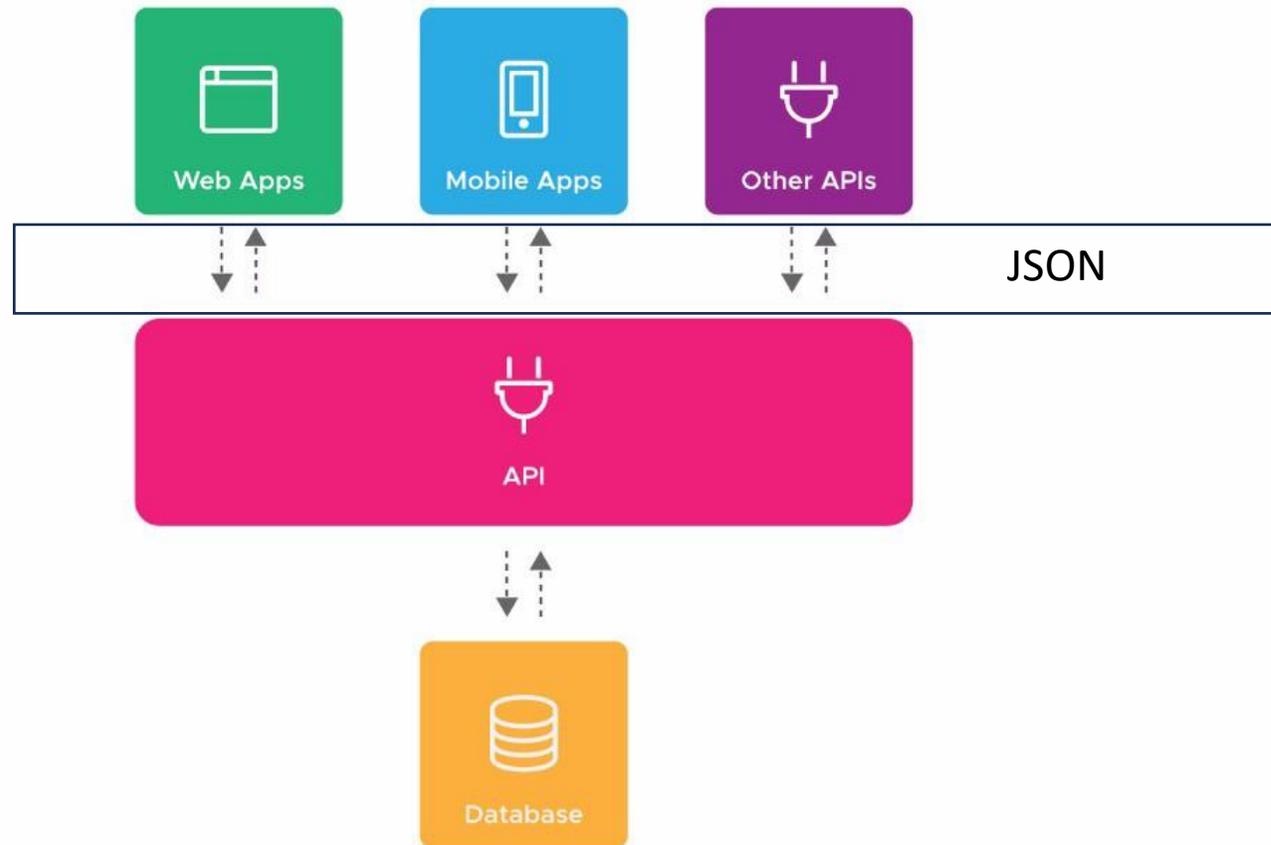


O controle também responde para o front-end um json.



Se o front trabalhar com o método get tipicamente os dados são passados na uri.

Comunicação da API



apis interessantes.

- <https://api.adviceslip.com/advice>
- <https://deividfortuna.github.io/fipe/>
 - <https://parallelum.com.br/fipe/api/v1/carros/marcas>
 - <https://parallelum.com.br/fipe/api/v1/carros/marcas/59/modelos>

insomnia

- É um software que nos permite fazer requisições à APIs.
- Ajuda no desenvolvimento do Backend (ex.: testar as chamadas para cada **endpoint** sem necessitar de um Frontend)
- Download free:
<https://insomnia.rest/>

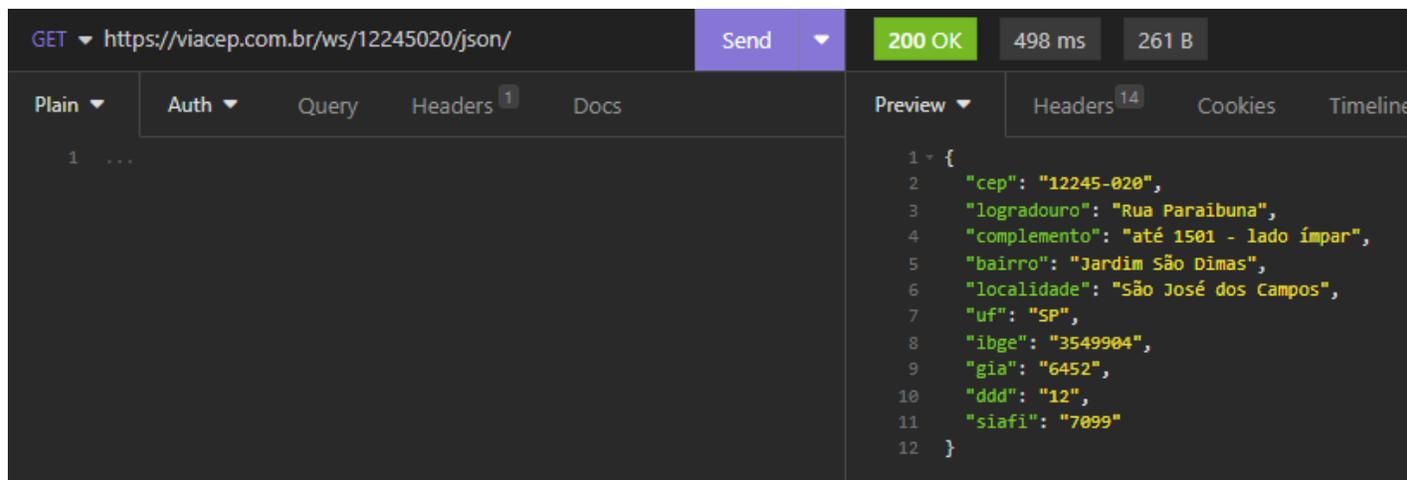


Insomnia

Via cep:

<https://viacep.com.br/ws/12245020/json/>

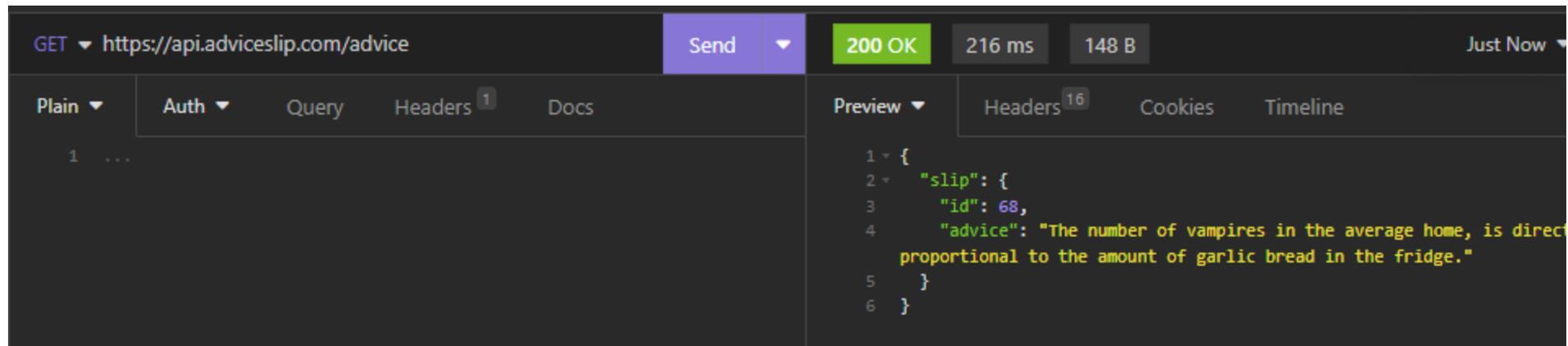
- Observe que a ferramenta envia um get para o endereço:
 - <https://viacep.com.br/ws/12245020/json/>
- Observe o formato da url, **não** é utilizado o padrão "? &" para identificar as variáveis e dados que serão enviados para o servidor.
- É possível notar que o cep é enviado na url.



```
GET https://viacep.com.br/ws/12245020/json/ 200 OK 498 ms 261 B
Plain Auth Query Headers 1 Docs Preview Headers 14 Cookies Timeline
1 ...
2 {
3   "cep": "12245-020",
4   "logradouro": "Rua Paraibuna",
5   "complemento": "até 1501 - lado ímpar",
6   "bairro": "Jardim São Dimas",
7   "localidade": "São José dos Campos",
8   "uf": "SP",
9   "ibge": "3549904",
10  "gia": "6452",
11  "ddd": "12",
12  "siafi": "7099"
}
```

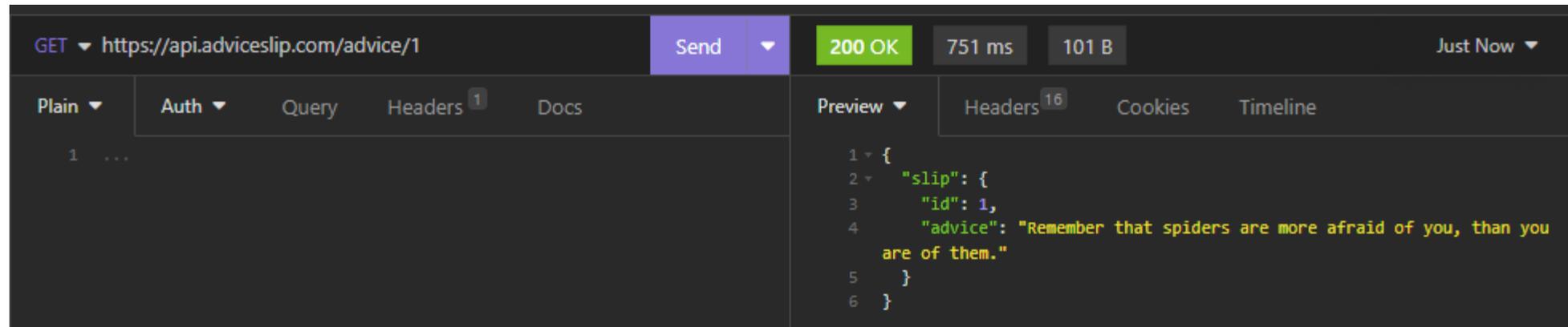
Testando apis:

- Acesse a URL:
 - <https://api.adviceslip.com/advice>



https://api.adviceslip.com/advice/1

- Observe que é enviado o número 1 pela a url, é possível notar que o 1 é referente a uma chave primária em um banco de dados e que você pode escolher outros números correspondentes a conelhos diferentes.



```
GET https://api.adviceslip.com/advice/1 200 OK 751 ms 101 B Just Now
```

```
Plain Auth Query Headers 1 Docs Preview Headers 16 Cookies Timeline
```

```
1 ...
```

```
1 {
2   "slip": {
3     "id": 1,
4     "advice": "Remember that spiders are more afraid of you, than you
5               are of them."
6   }
}
```

Apis e Aplicações Modernas

- Tipicamente quando construímos APIs web um dos objetivos é separar completamente o front-end do backend.
- Utilizando O MVC é possível separar o front-end do back end.
- Muitos desenvolvedores tem preferido trabalhar com o JSON para comunicação entre o back-end (POST, PUT) e o front-end.
- Os fronts (sites, televisores, carros, dispositivos eletronicos) dados para o back que responde com um JSON.
- Isso garante um isolamento ainda maior do front-end.