

## O que é JSON?

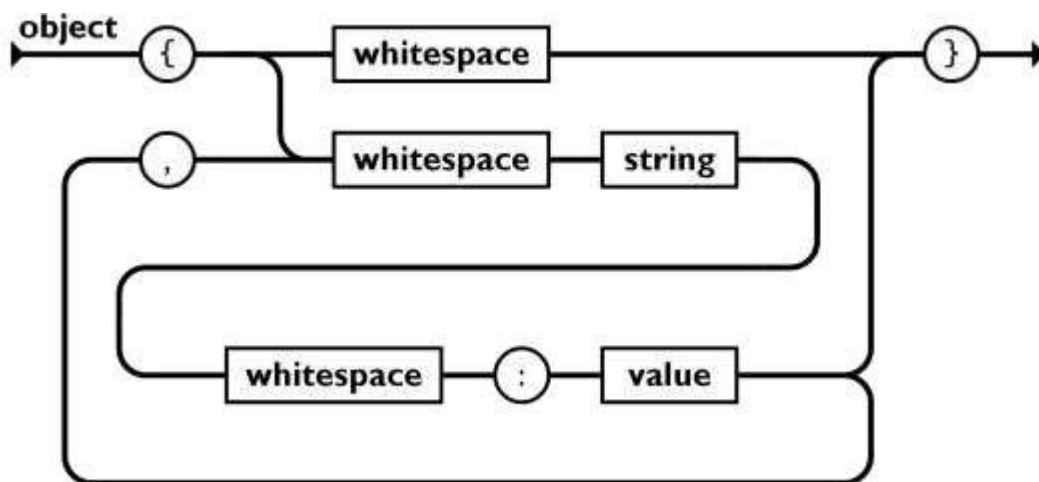
*JSON*, acrônimo de *JavaScript Object Notation*, em uma tradução literal “Notação de Objetos JavaScript” é, segundo seu site oficial, “... uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar.”. **(Disponível em: <https://json.org/json-pt.html>).**

Sendo assim, podemos dizer que o JSON é um formato de troca de informações entre sistemas computadorizados, sendo seu formato simples e claro, podendo ser lido por um ser humano, mesmo ele sendo leigo em teorias computacionais, ou em linguagens de programação.

Atualmente o *JSON* é um formato é o mais utilizado para a troca de informações entre sistemas, transcendendo os limites da Web e passando a ser utilizado em aplicações desktop e mobile. Sendo assim, conhecer suas nuances e características é fundamental.

### Objeto JSON

Conforme podemos ver na figura, um objeto JSON é formado por um conjunto de elementos “chave/valor”, sendo que a chave corresponde ao identificador do dado e o valor é o próprio dado em si. Um objeto começa com { e termina com }. Cada nome é seguido por : e os pares nome/valor são seguidos por ,.

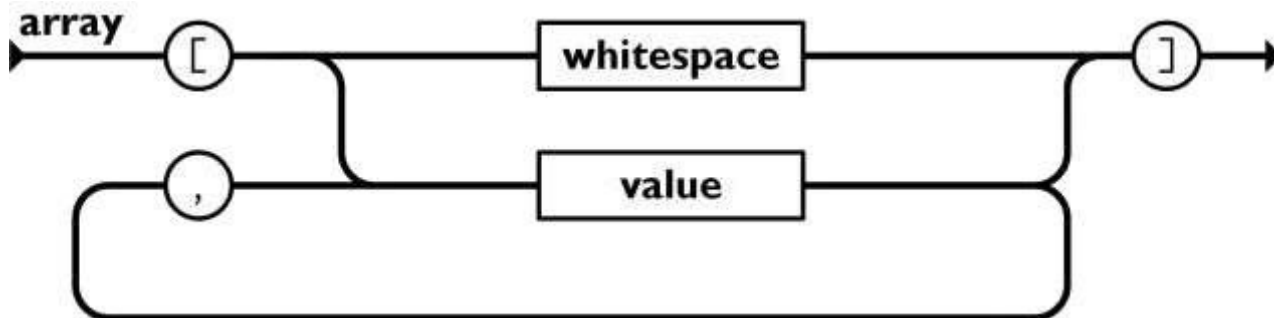


*Figura - Estrutura de um objeto JSON*

Agora que já entendemos a estrutura de um objeto *JSON*, podemos evoluir em nossos estudos e aprendermos sobre conjuntos de objetos *JSON*, ou seja, um *array JSON*.

A estrutura de um array de objetos JSON não é muito diferente de um objeto único. Um array JSON nada mais é que um conjunto ordenado de objetos. O array começa com

[ e termina com ]. Sendo que cada objeto é separado por uma ,.



*Figura - Estrutura de um array de objetos JSON*

### ***Biblioteca convert***

Esteja planejando uma pequena ou grande aplicação, uma coisa é verdade, ela precisará lidar com algum tipo de webservice. E como bem sabemos, atualmente a linguagem mais utilizada para a transferência de dados entre uma API e seus clientes é o formato JSON. Dessa forma, é primordial que saibamos lidar com dados nesse formato.

Mas como sempre, o Dart não nos deixa na mão quando o assunto é implementação de JSON, seja para converter um objeto Dart em *JSON*, como vice-versa. Já que o Dart em seu *core* tem uma biblioteca para conversão de dados. Essa biblioteca vai além de simplesmente lidar com string *JSON*. Mas neste momento falaremos apenas sobre essa parte da biblioteca.

### ***json\_encode***

O método `json_encode`, como o próprio nome já sugere, nos permite fazer um “encode” de um objeto para o formato JSON de forma tão simples e fácil que é difícil de acreditar.

```
1 import 'dart:convert';
2
3 void main() {
4   Map<String,dynamic> mapa = {"nome": "Anderson", "idade": 29};
5
6   dynamic json = jsonEncode(mapa);
7
8   print(json is Map);
9   print(json);
10 }
11
```

### *json\_decode*

O método `json_decode`, como o próprio nome já sugere, nos permite fazer um “decode” de um objeto para o formato JSON de forma tão simples e fácil que é difícil de acreditar.

A screenshot of a code editor with a dark background and light-colored text. The code is written in Dart and demonstrates the use of the `json_decode` method. It includes an import statement for `dart:convert`, a `main` function, and a JSON string. The JSON string is decoded into a dynamic map, which is then printed to the console. The code is as follows:

```
1 import 'dart:convert';
2
3 void main() {
4   // Escape a string JSON com três aspas simples
5   String jsonString = '''{
6     "nome": "Anderson",
7     "idade": 29
8   }''';
9
10  dynamic mapa = jsonDecode(jsonString);
11
12  print(mapa is Map);
13  print("Meu nome é ${mapa["nome"]}, e tenho ${mapa["idade"]} anos");
14}
15
```