

Driven

JSON e serialização

Carlos André Lima de Matos - T6
18 de julho de 2022

<https://driven.com.br/>

O que é serialização e deserialização

Serialização: É um processo para converter uma estrutura de dados ou um objeto em um formato que possa ser armazenado ou transferido.

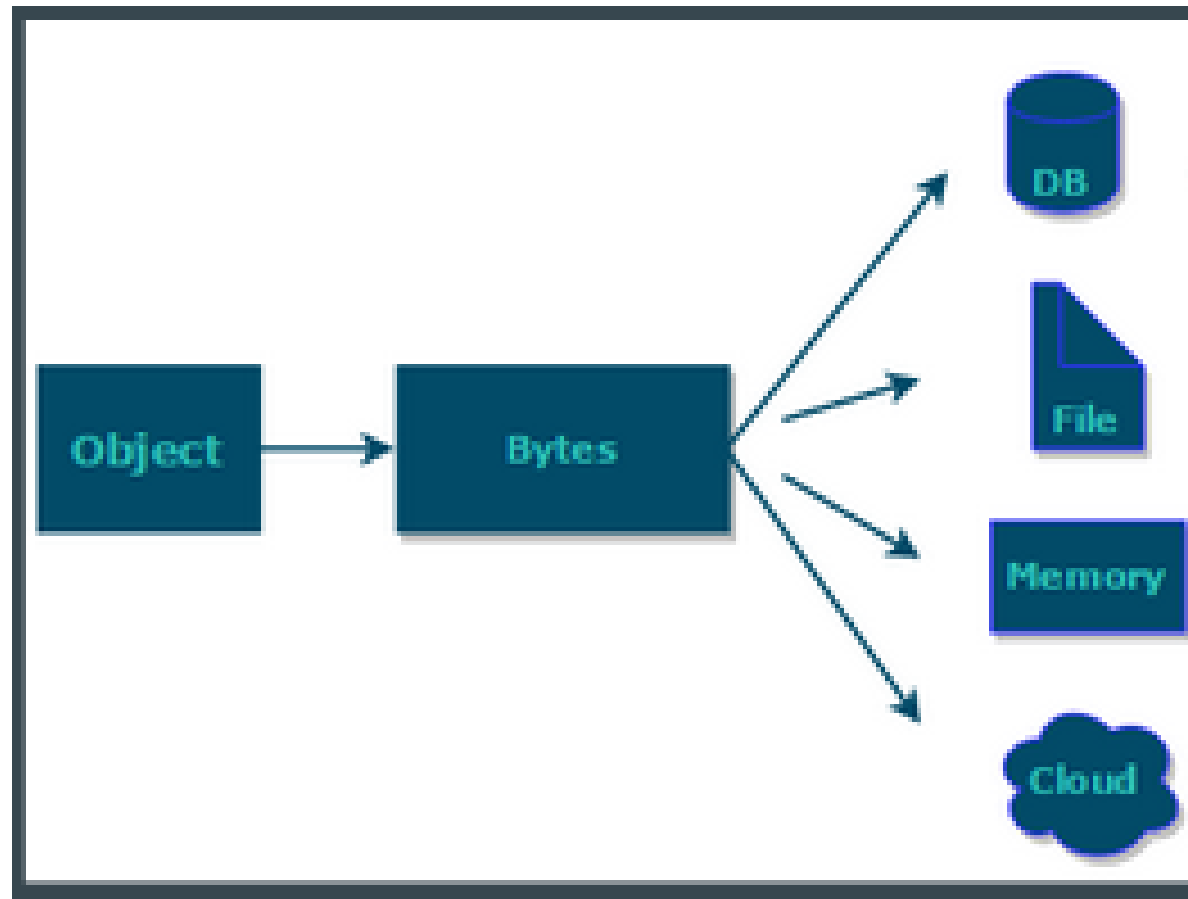
Deserialização: Refere-se ao processo inverso da serialização, ou seja, é restaurar o objeto/estrutura de dados recebido sem alterar seus atributos e valores.

Processo importante para garantir a **comunicação entre aplicações**.

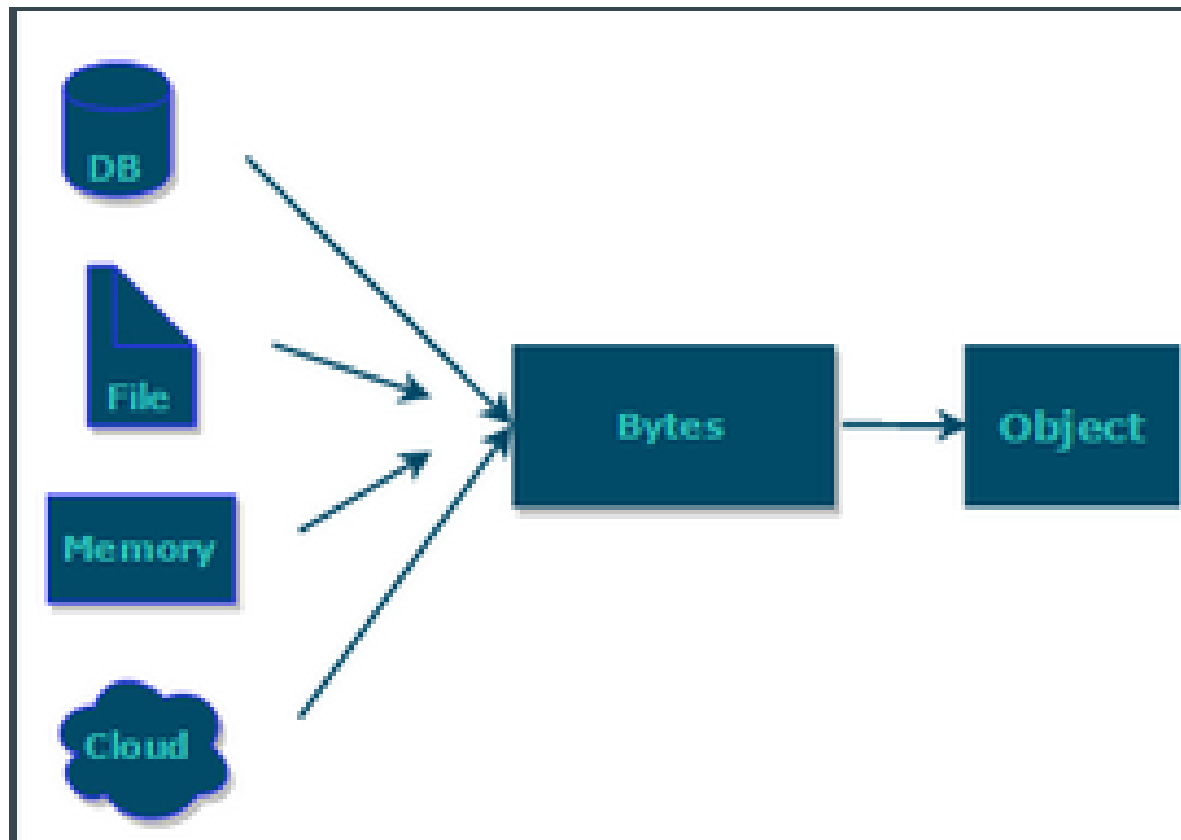
Aplicação

Envio de objetos e suas propriedades através de uma rede, ou salvá-lo em bancos de dados. Vejamos a imagem abaixo:

Serialização



Deserialização



O que é JSON

O formato **JSON** (*JavaScript Object Notation*) é, como a grafia do nome sugere, uma forma de notação de objetos **JavaScript** de modo que eles possam ser representados de uma forma comum a diversas linguagens.

O que é e quais as regras do formato JSON?

As regras para a sintaxe JSON são um subconjunto das regras de sintaxe do *JavaScript*. Vejamos:

1. Atributos devem estar dentro de aspas duplas;

```
Exemplo: "name", "age";
```

2. Os dados devem estar no formato **atributo: valor**;

```
Exemplo: "name": "Carlos", "age": 32;
```

3. Valores que são aceitos para os atributos;

- *Strings*: devem estar entre "", não pode ser '";
- *Number*;
- *Object*;
- *Array*;
- *Boolean*;
- *null*.

Não são aceitos como valores em JSON: **função, data, undefined**.

4. Objetos JSON devem estar dentre de chaves;

```
Exemplo: {"name": "Carlos"}
```

5. Arrays em JSON devem estar dentre de colchetes;

```
Exemplo: [ {"day":12}, {"day":13}, {"day":14} ]
```

6. Dados dentro de objetos ou arrays devem ser separados por vírgula.

```
{  
  "name": "Carlos",  
  "age": 32,  
  "student": false  
}
```

Como podemos converter um objeto JS em JSON e vice versa?

Para converter um objeto JS em JSON, utilizamos a função:

```
JSON.stringify(<objeto>);
```

Para converter de JSON para um objeto JS, utilizamos a função:

```
JSON.parse(<dados-serializados>);
```

O que acontece quando serializamos objetos que tenham uma função dentro? Por quê?

Verificamos que a função irá desaparecer na versão serializada em JSON, tanto seu valor/sua chamada quanto o nome do seu atributo.

Isso acontece porque funções não são aceitas pelo JSON como valores de atributos.

O que é XML?

O XML, sigla para *eXtensible Markup Language*, foi criada para armazenar e transportar dados, com o objetivo de ser facilmente entendida tanto por humanos quando máquinas.

Não existe uma função nativa para serializar um objeto JS em XML. Temos bibliotecas de terceiros, como é o caso dessa para Node.js: <https://github.com/wankdanker/node-object-to-xml>

E sites, como por exemplo: <https://www.convertsimple.com/convert-javascript-object-to-xml/>

Regras do formato XML

- Toda tag aberta (<tag>) deve ser fechada (</tag>);
- Todas as tags são case-sensitive, ou seja, <tag> é diferente de <TAG>;
- Os atributos devem estar entre aspas. Exemplo: <contact type="phone">(75) 98562 5262</contact>;
- Os atributos devem ser únicos, não pode haver mais de um atributo com o mesmo nome.

O que é YAML? Como serializamos um objeto JS em YAML?

O YAML é uma linguagem de serialização de dados usada na escrita de arquivos de configuração. Sua estrutura é baseada em indentação e, por isso, entre outros motivos, pode ser considerada mais legível para humanos que o JSON e o XML.

Não existe uma função nativa para serializar um objeto JS em YAML. Podemos utilizar bibliotecas de terceiros:

<https://github.com/eemeli/yaml> e site: <http://www.json2yaml.com/>.

Regras do formato YAML

- Comentários no YAML começam com (#) e devem ser separados de outros tokens por espaços em branco;
- A estrutura do documento é composta por indentação com espaços em branco;
- Os elementos da lista são iniciados com hífen (-) sendo um elemento em cada linha, ou entre colchetes [] e separados por vírgula e espaço (,).

Comparando as três formas de serialização

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<site>
  <titulo>Carlos Andre - Desenvolvedor Web</titulo>
  <url>https://github.com/snowslaura</url>
  <post>
    <titulo>JSON e Serialização de Objetos - Parte 1</titulo>
    <data>18/07/2022</data>
  </post>
  <post>
    <titulo>JSON e Serialização de Objetos - Parte 2</titulo>
    </data>18/07/2022</data>
  </post>
</site>
```

JSON

```
{
  "titulo": "Carlos Andre - Desenvolvedor Web",
  "url": "https://github.com/carlosandretaiga",
  "posts": [
    {
      "titulo": "JSON e Serialização de Objetos - Parte 1",
      "data": "18/07/2022"
    },
    {
      "titulo": "JSON e Serialização de Objetos - Parte 2",
      "data": "18/07/2022"
    }
  ]
}
```

YAML

```
site:
  titulo: Carlos Andre - Desenvolvedor Web
  url: https://github.com/carlosandretaiga
  post:
    titulo: JSON e Serialização de Objetos - Parte 1
    data: 18/07/2022
  post:
    titulo: JSON e Serialização de Objetos - Parte 2
    data: 18/07/2022
```

Obrigado!

Referências

- W3Schools. **JSON Syntax**. Disponível em: < https://www.w3schools.com/js/js_json_syntax.asp >. Acesso em: 27 jul. 2022.
- W3Schools. **JSON Stringify**. Disponível em: < https://www.w3schools.com/js/js_json_stringify.asp >. Acesso em: 27 jul. 2022.

- W3Schools. **XML Tutorial**. Disponível em: < <https://www.w3schools.com/xml/default.asp> >. Acesso em: 18 jul. 2022.
- StackOverflow. **How to convert JSON to YAML in JavaScript?**. Disponível em : < <https://stackoverflow.com/questions/38781929/how-to-convert-json-to-yaml-in-javascript> >. Acesso em: 18 jul. 2022.
- The YAML serialization format. Disponível em: < > <https://doc.sitecore.com/xp/en/developers/93/sitecore-experience-manager/the-yaml-serialization-format.html> >. Acesso em: 18 jul. 2022.