

Arquivo JSON: tudo que o desenvolvedor precisa saber

“ Descubra como o arquivo JSON pode facilitar o desenvolvimento de software. Entenda suas funcionalidades, vantagens e como integrá-lo em seus projetos!

Com a crescente adoção de APIs e a necessidade de uma comunicação rápida e eficiente entre sistemas, o **arquivo JSON** se tornou uma escolha popular. Mas será que você está o utilizando da maneira mais eficiente possível?

Neste artigo, vamos explorar em profundidade o que é JSON, suas funcionalidades, usos práticos e como os desenvolvedores podem utilizar essa tecnologia de forma eficaz em seus projetos.

O que é arquivo JSON?

Arquivo JSON, ou **JavaScript Object Notation**, é um formato leve de intercâmbio de dados que é fácil para humanos lerem e escreverem, e fácil para máquinas interpretarem e gerarem. Resumidamente, ele define uma **formatação para armazenamento e transferência de dados em formato texto**.

Pense no JSON como uma lista organizada de informações, onde cada item é descrito de uma forma clara e precisa. Ou seja, um arquivo JSON é composto por pares de chave-valor, onde a “chave” é o nome do dado e o “valor” é o dado em si.

Vamos ver um exemplo:

```
{
  "nome": "illimitar",
  "cidade": "Rio de Janeiro"
}
```

Neste exemplo, “nome” e “cidade” são as chaves, e “illimitar” e “Rio de Janeiro” são os valores correspondentes.

Histórico

O arquivo JSON surgiu por volta dos anos 2000, mas só ganhou força mesmo a partir da versão 5 do Javascript. Com o seu amadurecimento, é suportado por uma infinidade de linguagens de programação.

Mas não se engane, mesmo se inspirando no JavaScript, um arquivo JSON não é escrito puramente nesta linguagem, tendo suas peculiaridades e sintaxe próprias. Contudo, devido à sua derivação do JavaScript, é fácil perceber que o JSON herdou traços da linguagem que o originou, facilitando ainda mais o aprendizado para muitos programadores.

Facilidade para desenvolvedores

Por seguir um padrão lógico muito simples, é facilmente compreendido pelos desenvolvedores, o que impulsionou seu crescimento e o tornou tão popular no mundo da programação.

Rápido, simples, leve, prático e interpretado velozmente pelas máquinas devido a todos esses atributos, o arquivo JSON vem sendo utilizado amplamente em APIs e outras aplicações na rede, se tornando essencial para o armazenamento e troca de informações na web.

Por que dominar o arquivo JSON?

Quem desenvolve aplicações para web precisa se preocupar com os formatos de dados, suas integrações e seus protocolos de comunicação. Neste sentido, dois formatos dominam este tipo de software, são eles: o XML e o JSON.

Entender o arquivo JSON é importante porque ele é amplamente utilizado em APIs, para armazenar dados e para a troca de informações entre aplicações. E, ainda, tem diversas vantagens:

Leve e rápido

Um arquivo JSON tem menos texto/código que o XML, o que o torna mais rápido para ser transmitido e processado.

Fácil de ler e escrever

A sintaxe é simples, o que facilita a leitura e a escrita tanto para humanos quanto para máquinas.

Ampla compatibilidade

O arquivo JSON pode ser usado com quase todas as linguagens de programação atuais, facilitando a integração entre diferentes sistemas. Veja os exemplos de linguagens de programação que já aceitam o JSON:

- Python
- Rubi
- PHP
- JavaScript

- C++

E várias outras!

Referência no mercado

Além disso tudo, a robustez que o JSON traz em suas funcionalidades vem atraindo a atenção dos gigantes do mercado. O próprio Google já implementa o código em suas aplicações, tendo excelentes resultados no transporte de dados.

Ou seja, por todos os pontos positivos citados, é uma tendência natural que mais e mais aplicações utilizem um arquivo JSON e que o mercado precise cada vez mais de profissionais capacitados neste formato.

O que é arquivo XML?

Extensible Markup Language (XML) é uma linguagem de marcação que define um conjunto de regras para codificação de documentos em um formato que também é legível tanto para humanos quanto para sistemas. Assim como o arquivo JSON, o XML é usado para armazenar e transportar dados, mas possui uma estrutura mais complexa.

Uma aplicação bem comum é a utilização do XML para integração das notas fiscais eletrônicas, pela simplicidade da formatação que utiliza tags para a marcação dos dados.

Qual a diferença entre arquivo JSON e XML?

A principal diferença entre JSON e XML é a simplicidade e a fácil leitura do JSON em comparação com a verbosidade do XML. Enquanto o XML usa tags de abertura e fechamento para definir elementos, o JSON utiliza uma notação de chave-valor dentro de chaves {} e colchetes [] para definir objetos e arrays.

Exemplos

Vamos olhar a sintaxe do XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<profissoes>
  <profissao>
    <nomeProfissao>Professor</nomeProfissao>
    <referencia>PF</referencia>
  </profissao>
  <profissao>
    <nomeProfissao>Programador</nomeProfissao>
    <referencia>PR</referencia>
```

```
</profissao>
<profissao>
  <nomeProfissao>Copywriter</nomeProfissao>
  <referencia>CO</referencia>
</profissao>
</profissoes>
```

Bastante coisa, não é mesmo? O arquivo JSON, por sua vez, utiliza chaves { } para delimitação de objetos. Confira o trecho:

```
{
  "profissao": "Professor"
}
```

Muito mais simples, né?!

Arquivo JSON: Quais são suas vantagens na prática?

Como já vimos, os arquivos JSON possuem várias vantagens:

1. Leveza: Arquivos JSON são menores em tamanho comparados ao XML, o que resulta em tempos de carregamento mais rápidos.
2. Facilidade de uso: A sintaxe simples e a estrutura de chave-valor tornam o JSON fácil de ler e escrever.
3. Compatibilidade: JSON é compatível com a maioria das linguagens de programação modernas, facilitando a integração entre diferentes sistemas.
4. Performance: O arquivo também permite uma comunicação mais rápida e eficiente entre o cliente e o servidor, sendo ideal para APIs RESTful.
5. Interoperabilidade: Por ser um formato texto, JSON pode ser facilmente transmitido e manipulado em rede.

Sintaxe

A sintaxe do JSON é direta e fácil de entender. Aqui estão os elementos básicos:

- Objetos: Definidos por chaves { } e contêm pares de chave-valor.
- Arrays: Definidos por colchetes [] e contêm uma lista ordenada de valores.
- Valores: Podem ser strings, números, objetos, arrays, true, false ou null.

Para mostrar como é fácil interpretar o arquivo JSON, preparamos um pequeno resumo dos principais pontos de atenção de sua sintaxe. Nele fica bem evidente sua praticidade se comparado ao seu principal concorrente, o XML. E também ficam evidentes os traços herdados do JavaScript:

```
{
  "cidade": "Joinville",
  "bairro": "Floresta",
  "endereço": [
    "Rua da Ipanema",
    "sem número"
  ]
}
```

Neste curto trecho, observamos elementos básicos:

- O atributos devem ser seguidos de : (dois pontos)
- Estes atributos devem ser separados por , (vírgulas)
- O último atributo de um código não deve conter vírgula
- Os arrays só podem conter um tipo de dado (texto, número, etc)

Já os tipos de dados primitivos do JSON são:

- Tipo nulo, representado por: Ex: { "cod" : null }
- Tipo booleano, representado por: true ou false
- Tipo número, que é escrito sem aspas e pode representar valores inteiros ou reais; neste último caso, deve-se usar um ponto para separar a parte inteira das casas decimais.
- Tipo string, separados por aspas.

Veja na prática os tipos de dados aplicados neste exemplo:

```
{
  "cor": "amarelo",
  "código": 10,
  "valor": 10.58,
  "status": false,
  "referencia": null
}
```

A partir destes conceitos básicos já é possível começar a desenvolver seus primeiros códigos nesta formação. É importante reforçar que os códigos em JSON são estruturados em objetos, que são simbolizados por { } e os arrays por colchetes [].

Preferência no desenvolvimento

Desenvolvedores preferem JSON devido à sua simplicidade e eficiência. A estrutura compacta e a sintaxe clara ajudam a evitar erros comuns na troca de dados e tornam o processo de depuração mais fácil.

Arquivo JSON e APIs

O arquivo JSON é amplamente utilizado em APIs para armazenamento e transporte de dados entre diferentes ambientes. No dia a dia de um desenvolvedor, tarefas comuns com JSON incluem:

- **Ler dados de um JSON:** Extrair informações específicas de um arquivo JSON.
- **Modificar dados:** Incluir ou alterar valores dentro de um arquivo JSON.
- **Conversão de dados:** Converter texto em um objeto JSON e vice-versa.

Essas operações são simples e diretas, tornando o JSON uma ferramenta eficiente para manipulação de dados. A sintaxe clara e a estrutura leve de JSON facilitam essas atividades.

Arquivo JSON e APIs RESTful

Nas APIs, o JSON é frequentemente utilizado junto com o padrão REST (Representational State Transfer). As APIs RESTful codificam dados em JSON e utilizam o formato UTF-8 para inclusão e retorno de informações. Aqui estão os dois principais tipos de requisições em APIs RESTful:

- **GET (Síncrono):** Quando uma URL específica é requisitada, os resultados são exibidos imediatamente.
- **POST (Assíncrono):** Envia dados para um servidor e o resultado pode ser acessado posteriormente pela URL fornecida no cabeçalho da resposta.

A integração entre API e JSON minimiza as requisições e a quantidade de dados transmitidos entre cliente e servidor, resultando em maior eficiência sem comprometer a flexibilidade, segurança e legibilidade.

Origem: [Redação da Tecnospeed](#)

Revisão #: contagem de revisões

Criado: duração de tempo por usuário

Atualizado: duração de tempo por usuário