

15 - JSON

JSON, que significa JavaScript Object Notation, é uma das pedras angulares na troca de dados, especialmente em aplicações Node.js.

O que é JSON?

JSON é um formato leve para troca de dados. Por ser fácil de ler e escrever para humanos e ao mesmo tempo fácil de interpretar e gerar para máquinas, tornou-se o formato padrão para intercâmbio de dados, especialmente na web.

Estrutura Básica do JSON

```
{
  "nome": "João",
  "idade": 25,
  "interesses": ["programação", "jogos"],
  "ativo": true
}
```

Este é um exemplo de como um objeto JSON pode parecer. É muito semelhante aos objetos JavaScript, mas com algumas diferenças, como todas as chaves e strings precisam estar entre aspas duplas.

Trabalhando com JSON em JavaScript

JavaScript tem métodos nativos para converter objetos em JSON e vice-versa, o que é fundamental para o desenvolvimento backend.

Convertendo Objeto para JSON

```
let pessoa = {
  nome: "Ana",
  idade: 30,
  interesses: ["leitura", "viagens"]
};

let json = JSON.stringify(pessoa);
console.log(json); // {"nome":"Ana","idade":30,"interesses":["leitura","viagens"]}
```

JSON.stringify() converte um objeto JavaScript em uma string JSON. Isso é útil quando você precisa enviar dados para um servidor ou salvá-los de maneira formatada.

Convertendo JSON para Objeto

```
let json = '{"nome":"Carlos","idade":28}';  
let pessoa = JSON.parse(json);  
console.log(pessoa.nome); // Carlos
```

JSON.parse() faz o inverso, transformando uma string JSON em um objeto JavaScript. Essencial para receber e trabalhar com dados de um servidor ou API.

Considerações Importantes

- **Aspas Duplas:** No JSON, as chaves e strings devem estar sempre entre aspas duplas.
- **Dados Primitivos:** JSON pode representar dados como números, strings, booleanos, arrays, objetos e **null**.
- **Não Suporta Comentários:** JSON não permite o uso de comentários dentro do formato.

Usos Comuns do JSON

JSON é amplamente utilizado em:

- **APIs e Web Services:** Para enviar e receber dados.
- **Configurações de Projeto:** Como em arquivos **package.json** em projetos Node.js.
- **Armazenamento de Dados:** Por ser um formato leve e fácil de ler.

Dicas para Trabalhar com JSON

- **Validação:** Sempre valide o JSON antes de usá-lo. Existem várias ferramentas online para verificar se seu JSON está formatado corretamente.
- **Tratamento de Erros:** Use try-catch ao fazer parse de JSON, pois erros de formatação são comuns.