



Curso Técnico em Informática

## UC13: Executar os processos de codificação, manutenção e documentação de aplicativos computacionais para internet

Instrutor: Anderson Oliveira da Silva





# Tipos de Resposta em Node.js com Express.js

Quando um servidor Express recebe uma requisição HTTP, ele precisa responder de alguma forma ao cliente que fez a requisição. Dependendo do tipo de informação que você deseja enviar de volta, existem diferentes métodos de resposta que você pode usar. Esses métodos ajudam a enviar texto, JSON, arquivos, redirecionamentos, e até mesmo a renderizar visualizações dinâmicas.

## Principais Tipos de Resposta

`res.send()`: Este método é utilizado para enviar uma resposta de texto ou HTML. Ele é bastante versátil e pode ser usado para enviar strings, buffers e objetos.

`res.json()`: Quando você precisa enviar dados em formato JSON, `res.json()` é a escolha ideal. Ele automaticamente define o cabeçalho de conteúdo como `application/json` e serializa o objeto JavaScript em JSON.

`res.sendFile()`: Se você precisa enviar um arquivo como resposta, `res.sendFile()` é o método apropriado. Ele permite especificar o caminho do arquivo e o envia para o cliente.

`res.redirect()`: Este método é utilizado para redirecionar a requisição para outra URL. É útil em casos onde, por exemplo, você deseja redirecionar o usuário após um login bem-sucedido.

`res.render()`: Ele permite renderizar uma visualização e enviar o HTML resultante ao cliente, passando variáveis para o template.

## O Que é JSON?

JSON (JavaScript Object Notation) é um formato leve de troca de dados que é fácil de ler e escrever tanto para humanos quanto para máquinas. Ele foi derivado de uma subset da linguagem de programação JavaScript, mas é independente de linguagem, o que significa que pode ser usado em praticamente qualquer linguagem de programação moderna.

```
{  
  "name": "John Doe",  
  "age": 30,  
  "isStudent": false,  
  "courses": ["Math", "Science", "History"],  
  "address": {  
    "street": "123 Main St",  
    "city": "Anytown",  
    "postalCode": "12345"  
  }  
}
```

### Estrutura de JSON

**Pares Chave-Valor:** Cada item de dados é representado como um par chave-valor, onde a chave é uma string e o valor pode ser uma string, número, booleano (true/false), array ou outro objeto.

**Arrays:** Listas de valores ordenados, que podem ser de qualquer tipo, incluindo outros objetos JSON.

**Objetos Aninhados:** Objetos podem conter outros objetos, permitindo a criação de estruturas de dados complexas.

## Manipulação de Cabeçalhos e Status:

`res.set()`: Define cabeçalhos personalizados na resposta.

`res.status()`: Define o código de status HTTP da resposta.

`res.type()`: Define o tipo MIME da resposta.

Cabeçalhos HTTP são metadados enviados junto com uma requisição ou resposta HTTP. Eles fornecem informações adicionais sobre a requisição ou resposta, como o tipo de conteúdo, métodos de autenticação, políticas de cache, e muito mais.

### **Exemplos de Cabeçalhos Comuns:**

- Content-Type: Indica o tipo de mídia do recurso.
- Authorization: Contém credenciais para autenticação.
- Cache-Control: Define políticas de cache.
- User-Agent: Identifica o cliente HTTP (navegador, por exemplo).

### **Status da Requisição**

Status HTTP são códigos numéricos que indicam o resultado de uma requisição HTTP. Eles ajudam a comunicar se a requisição foi bem-sucedida, se houve um erro, ou se algum tipo de redirecionamento é necessário.

### **Categorias de Códigos de Status:**

- 2xx (Sucesso): A requisição foi bem-sucedida (200 OK).
- 3xx (Redirecionamento): A ação adicional é necessária (301 Moved Permanently).
- 4xx (Erro do Cliente): A requisição contém um erro (404 Not Found).
- 5xx (Erro do Servidor): O servidor encontrou um erro (500 Internal Server Error).



Siga o Senac em Minas nas Redes Sociais:

