



JSON, async/await e servidor



JSON

+



O que é JSON?

É uma sigla que significa: JavaScript Object Notation.

O JavaScript tem uma biblioteca padrão que nos ajuda a trabalhar com JSON, com esta biblioteca podemos transformar um objeto JSON em uma string e também o sentido contrário, transformar uma string em um JSON.

Como podemos interagir

Para transformar um JSON em string:

```
$JSON.stringify(meuJson);
```

Para transforma uma string em JSON:

```
$JSON.parse(minhaString);
```



Prática

Criar uma classe de Usuario

Criar uma instância de Usuario

Salvar os dados do Usuario em JSON em um arquivo
Apresentar no console quando salvou

Carregar os dados de um Usuario do arquivo JSON
Apresentar no console o resultado

async/await

+



O que é async/await?

É uma forma de tratar promises de uma forma mais limpa, porém travando a aplicação.

Async é uma instrução que adiciono na definição de uma promise para informar que aquela function pode para a aplicação até que obtenha uma resposta de uma Promise

Await é a instrução que eu passo na chamada de uma function para informar que somente deve seguir para próxima linha quando finalizar a execução da function/promise

Como podemos definir async/await

```
const salvaDados = async () => {  
  await salvarAutor();  
  await salvarEditora();  
  await salvarLivro();  
  console.log('Processo finalizado');  
}
```



Prática

Continuando a prática anterior

Crie 10 usuários, adicionado na rotina de salvar:

```
$ setTimeout(() => {}, Math.random(3));
```

Tente consultar um Usuário, que ainda não tenha salvo

Como garantir que execute somente de salvar?

- async/await

Criando um
servidor

+



Como criar um servidor em nodejs?

O NodeJS provê uma estrutura para que consigamos receber requests e definir um retorno para essas requests

Como podemos instanciar um servidor

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, {
    'Content-Type': 'application/json'
  });
  res.end(JSON.stringify({ data: 'Hello World!' }));
});

server.listen(8000);
```



Prática

Vamos instanciar um servidor que tenha dois endpoints:

- Adicionar usuário
- Buscar dados de um usuário

0/
Exercícios
+



Aquecimento

Refatore os exercícios da manhã:

- Criando um endpoint para adicionar aeroporto
- Criando um endpoint para adicionar voo
- Criando um endpoint para adicionar passageiro



Vamos a prática

Crie uma aplicação de gerenciamento de viagens.

Funcionalidades:

- Criar um endpoint que liste todas as rotas disponíveis
- Crie um endpoint para adicionar aeroporto, com as informações de Nome, Código e Endereço. O registro deve ser salvo em JSON em um arquivo
- Crie um endpoint para adicionar voo, com as informações de Código do voo, Código aeroporto origem, Código aeroporto destino e Nome da empresa aérea. O registro deve ser salvo em JSON em um arquivo
- Crie um endpoint para adicionar passageiro em um voo, com as informações de Código do voo, Nome do passageiro. O registro deve ser salvo em JSON em um arquivo
- Crie um endpoint para listar voos, apresentando as informações: código do voo, nome do aeroporto de origem, nome do aeroporto de destino e o nome de todos os passageiros no voo. O registro deve ser salvo em JSON em um arquivo



Conforme a aplicação que foi criada no exercício anterior, pense como poderíamos criar essas outras duas aplicações:

- Crie uma aplicação de gerenciamento de eventos;
- Crie uma aplicação de gerenciamento de filmes.

Criando outras Aplicações

Obrigado!

pulsati⁺

pulsati.com.br