👩🏾‍💻[M4] Mini Mapa de Programação👩🏾‍💻

👩🏻‍🏫 Apresentação de módulo

Em **Programação** vocês serão irão aprender sobre **Back End** e **APIs**. Trabalhando com NodeJS, API's REST, NPM, Promisses, Rest & CRUD, SQLite, DAO, Jest e Supertest.

A1: Hello node 👋🏾

📝 Tópicos da aula:

- Introdução ao módulo;
- Introdução ao Node;
- Introdução ao runtime do Node.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Crie um arquivo Javascript chamado “main.js” que contenha uma função que receba um parâmetro e verifique se ele é par ou ímpar. Se for par, a função retorna “Par”. Se for ímpar, a função retorna “Ímpar”. Em seguida, faça um laço for que se repita 10 vezes e, para cada repetição, ele pegue o valor de i e verifique se é par ou ímpar com a nossa função criada anteriormente, exibindo no console o resultado de cada número. Por fim, execute o arquivo usando o Node.js.

📚 Material complementar:

➔ Tutorial: Node.js no Windows para iniciantes | Microsoft Learn

A2: Biblioteca 📚

📝 Tópicos da aula:

- NPM;
- Introdução ao conceito de pacotes/bibliotecas;

- Bibliotecas + NPM;
- Chalk@4.1.2.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Crie uma função que funcione como termômetro, diante do número recebido ela deve exibir “Paciente com febre” na cor vermelho, ou “Paciente não está com febre” na cor verde.

➔ **Para ir além**

- ◆ Crie outra função, esta funcionar para gerar um número aleatório dentre 10 até 100 e implemente ela juntamente com a de termômetro.

📚 Material complementar:

➔ O que é NPM e como usar uma biblioteca instalada por ele | Blog TreinaWeb

A3: Promessas 💍

📝 Tópicos da aula:

- Revisão de conceitos de chamadas de APIs e de protocolos da Web (HTTP);
- Promises.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Crie uma promessa que verifica se a constante “quantidade” tem o valor menor que 10, se for maior que 10 então a promessa deve esperar 3.5 segundos para enviar a mensagem “Esta tudo bem na estação espacial!”. Porem se acaso for menor que 10 rejeite a promessa enviando a mensagem “Estamos em apuros!” de forma instantânea.

📚 Material complementar:

➔ Promise - JavaScript | MDN

A4: Pesquisando 🔎

📝 Tópicos da aula:

- Fetch;
- Código assíncono;
- Async & Await;
- Revisão try & catch;

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Crie duas requisições, uma usando o fetch com then e catch para a API RandowUser, anote as observações. Logo em seguida, usando a requisição feita como base, crie outra requisição usando o fetch juntamente com o async & await e usando blocos de controle como o try & catch para a prevenção dos erros.

📚 Material complementar:

➔ Async/await no JavaScript: o que é e quando usar a programação assíncrona? | Alura

A5: Configurando ⚙️

📝 Tópicos da aula:

- Aprofundamento no NPM;
- Introdução de conceitos da estrutura de projeto node: package.json, package-lock.json,

node_modules;
- Introdução ao framework express.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Crie um novo projeto node, e usando o express crie uma API que contenha três ou mais rotas get, para um sistema de músicas, sendo cada rota uma funcionalidade, por exemplo:
  - ◆ get “/allmusics” entrega a lista contendo todas as músicas;
  - ◆ get “/randommusic” entrega uma música aleatória;
  - ◆ get “/daymusic” entrega uma música do dia.

📚 Material complementar:

➔ express - npm

➔ CommonJS modules | Node.js v21.1.0 Documentation

A6: Explain Like I'm 5 🕔

📝 Tópicos da aula:

- Explain Like I'm 5;
- Dojo de código.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Sem atividade assíncrona.

📚 Material complementar:

➔ Sem material complementar.

A7: Classificando 😐

📝 Tópicos da aula:

- Revisão dos conceitos de POO;
- Introdução a conceitos de POO em uma API;
- Router.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Usando o express e o router, crie uma API que classifique objetos em diferentes categorias, como roupas, livros, eletrônicos, etc. Cada categoria deve ter uma rota get que retorne uma lista de objetos dessa categoria. Por exemplo: get "/roupas" retorna uma lista de roupas; get "/livros" retorna uma lista de livros; get "/eletronicos" retorna uma lista de eletrônicos.

📚 Material complementar:

➔ Roteamento no Express

➔ https://www.w3schools.com/nodejs/nodejs_classes.asp

➔ https://www.devmedia.com.br/orientacao-a-objetos-com-node-js/38438

A8: Up! 🆙

📝 Tópicos da aula:

- ES6 Modules;
- Arquitetura de API: MVC.

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Usando o ES6 Modules, refatore o código da API que você criou na atividade assíncrona da aula anterior, separando as responsabilidades em diferentes arquivos. Por exemplo, crie um arquivo para as classes dos objetos, outro para as rotas da API, outro para as funções auxiliares, etc. Use o comando import para importar os módulos que você precisa em cada arquivo, e o comando export para exportar os módulos que você criou.

📚 Material complementar:

➔ Módulos JavaScript

➔ https://www.devmedia.com.br/srp-principio-da-responsabilidade-unica/38178

➔ Introdução ao Padrão MVC: Primeiros passos na Arquitetura MVC

A9: Postagens 🚤

📝 Tópicos da aula:

- Nodemon;
- Insomnia & Postman;
- Método POST.

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Usando o Nodemon, crie uma API que permita fazer postagens de textos, imagens ou vídeos. A API

deve ter uma rota POST que receba os dados da postagem, como o autor, o conteúdo, o tipo e a data. A API deve armazenar as postagens em um array ou em um arquivo JSON. A API também deve ter uma rota GET que retorne todas as postagens salvas.

📚 Material complementar:

- ➔ nodemon - npm
- ➔ Postman
- ➔ Insomnia REST
- ➔ POST - HTTP | MDN

A10: No meio do campo ⚽

📝 Tópicos da aula:

- Requisição HTTP review;
- Conceitos e utilização de middlewares;
- Utilização do corpo de uma requisição POST.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

- ➔ Usando o express e middlewares, crie uma API que permita fazer comentários atravez dos método post. A API deve ter uma rota POST que receba os dados do comentário, como o autor, o texto e a postagem relacionada. A API deve armazenar os comentários em um banco de dados mocado. A API também deve ter uma rota GET que retorne todos os comentários salvos. A API deve usar um middleware para validar os dados do comentário, verificando se eles estão no formato correto e se não estão vazios. Se os dados forem inválidos, o middleware deve enviar uma resposta com um código de erro e uma mensagem explicando o problema. Se os dados forem válidos, o middleware deve passar o controle para a próxima função.

📚 Material complementar:

➔ Usando middlewares do Express

➔ Códigos de status de respostas HTTP

A11: De novo! ⏳

📝 Tópicos da aula:

- Explain Like I'm 5;
- Dojo de código.

💻 Material de aula

- Acesse aqui o slide da aula.

📪 Atividades assíncronas:

➔ Sem atividade assíncrona.

📚 Material complementar:

➔ Sem atividade assíncrona.

A12: Armazenamento local 🚩

📝 Tópicos da aula:

- Banco de dados mocado;
- TI Verde e pegada de carbono;
- Models.

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Usando um banco de dados mocado, crie uma API baseando-se em um problema do seu dia a dia, que permita armazenar e manipular dados. A API deve ter no minimo 2 rotas e cada uma deve permitir criar e ler os dados. A API deve usar os models para definir a estrutura e a validação dos dados. Use o Insomnia ou o Postman para testar as suas requisições e verificar se a sua API está funcionando corretamente. Você deve enviar e receber os dados em formato JSON.

📚 Material complementar:

➔ Operadores no Python
➔ model - npm

A13: Descartando 🗑️

📝 Tópicos da aula:

- Método DELETE & Parâmetros em rotas;
- Deleção de usuários;

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Usando o projeto construído na atividade assíncrona passada. Adicione a API uma rota DELETE que receba o id do usuário como um parâmetro de rota. A API deve verificar se o usuário existe no banco de dados mocado e, se existir, removê-lo. A API deve enviar uma resposta com um código de sucesso e uma mensagem confirmando a deleção. Se o usuário não existir, a API deve enviar uma resposta com um código de erro e uma mensagem explicando o problema.

📚 Material complementar:

➔ Roteamento no Express
➔ DELETE - HTTP | MDN

A14: Editando ✏️

📝 Tópicos da aula:

- PUT como verbo de atualização do HTTP;
- Finalização do CRUD;
- Atualização de recursos na API.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Dando continuidade em nossa API construída nas atividades assíncronas passadas, adicione o método PUT para permitir atualizar os dados dos usuários de um banco de dados mocado. A API deve ter uma rota PUT que receba o id do usuário como um parâmetro de rota e os novos dados do usuário no corpo da requisição. Ela deve verificar se o usuário existe no banco de dados e, se existir, atualizar os seus dados e então enviar uma resposta com um código de sucesso e uma mensagem confirmando a atualização. Se acaso o usuário não existir, a API deve enviar uma resposta com um código de erro e uma mensagem explicando o problema.

📚 Material complementar:

➔ PUT - HTTP | MDN
➔ Python: Operadores (devmedia.com.br)

A15: Testando bancos 🏦

📝 Tópicos da aula:

- Revisão comandos SQL;
- Introdução ao SQLite.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Usando o banco SQLite criado em aula e a proposta que o banco propõe para solucionar o problema, criem uma postagem no LinkedIn, juntamente com os participantes de sua Squad e apresente algumas espécies que estão correndo o risco de extinção, o porquê disso, logo após apresente a solução da Squad e consecutivamente o banco de dados criado.

📚 Material complementar:

➔ SQL Tutorial
➔ IUCN Red List Categories and Criteria

A16: Mudando o clima ⛈️🌞

📝 Tópicos da aula:

- ORM;
- Sequelize;
- Criação do banco e das tabelas;
- Mudanças climáticas.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Crie um projeto node usando o Sequelize, e modele um banco de dados que armazene informações sobre as emissões de gases de efeito estufa de diferentes países, como o nome do país, o ano, o tipo de gás, a quantidade emitida, e o setor responsável.

📚 Material complementar:

➔ Getting Started | Sequelize
➔ Crises climática e de perda de biodiversidade estão entre os riscos mais graves da próxima década | WWF Brasil
➔ 5 curiosidades e 30 fatos sobre mudança climática que você precisa saber - Tunes Ambiental

A17: Tudo rodando 🌀

📝 Tópicos da aula:

- Recap de ES6 Modules;
- Recap do MVC;
- Sequelize: Read;
- Sequelize: Create.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Usando o projeto que você criou na atividade assíncrona da aula passada, implemente as seguintes funcionalidades usando o Sequelize:

- ◆ Read: Crie uma rota que retorne todos os países que emitiram mais de um determinado tipo de gás de efeito estufa em um determinado ano. Receba os parâmetros pelo query string da requisição.
- ◆ Create: Crie uma rota que insira um novo país no banco de dados, com seus respectivos dados de emissão. Receba os dados pelo corpo da requisição.
- ◆ Boas práticas: Refatore o seu código, separando as funções de acesso ao banco de dados em um arquivo separado, e importando-as no arquivo principal. Use nomes significativos para as variáveis e funções, e comente o seu código explicando o que cada parte faz.

📚 Material complementar:

➔ Model Querying - Basics | Sequelize
➔ Model Querying - Finders | Sequelize
➔ Model Querying - Basics | Sequelize

A18: No fim da linha 🛤️

📝 Tópicos da aula:

- Sequelize: CRUD review;
- Sequelize: Update;
- Sequelize: Delete.
- **Mudanças climáticas.**

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Usando o projeto que você criou nas atividades assíncronas das aulas anteriores, implemente as seguintes funcionalidades usando o Sequelize:

- ◆ Update: Crie uma rota que atualize os dados de um país no banco de dados, conforme os parâmetros recebidos pelo corpo da requisição. Por exemplo, se você receber o id do pais, o nome do país e o ano, você deve atualizar os dados de emissão desse país nesse ano.
- ◆ Delete: Crie uma rota que delete um país do banco de dados, conforme o parâmetro recebido pelo query string da requisição. Por exemplo, se você receber o id do país, você deve deletar os dados desse país do banco de dados.

📚 Material complementar:

➔ Como combater as mudanças climáticas - eCycle
➔ Special Report on Climate Change and Land — IPCC site
➔ Model Querying - Basics | Sequelize

A19: Re-visitando 🗺️

📝 Tópicos da aula:

- Explain Like I'm 5;
- Dojo de código;
- Poluição plástica.

💻 Material de aula

- Acesse aqui o slide da aula.

📬 Atividades assíncronas:

➔ Usando a API criada em aula, criem uma postagem para o LinkedIn que contenha uma, breve apresentação do que é a poluição plásticas, seus principais problemas e apresente a proposta e o sistema desenvolvido por vocês.

📚 Material complementar:

➔ Não há materiais complementares.

A20: Teste testando 🔦

📝 Tópicos da aula:

- Intro ao Jest;
- Intro supertest.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Usando a API que ajuda a combater a poluição plástica, crie models para todos os dados, que forem inseridos no banco de dados, e dentro dos models crie funções de validações (como por exemplo, uma função que verifica se o e-mail é válido) e teste os models usando o Jest. E crie um teste para cada rota da API usando o Supertest.

📚 Material complementar:

➔

A21:

📝 Tópicos da aula:

- CORS;
- Hospedar API.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔

📚 Material complementar:

➔

A22: Anjo guardião 👼🏽

📝 Tópicos da aula:

- Plantão de ajuda com o projeto

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Não há atividades assíncronas.

📚 Material complementar:

➔ Não há materiais complementares.

A23: Apresentação de projeto final 🖼️

📝 Tópicos da aula:

- Apresentação de projeto final.

💻 Material de aula

- Acesse aqui o slide da aula.

📫 Atividades assíncronas:

➔ Não há atividades assíncronas.

📚 Material complementar:

➔ Não há materiais complementares.