

Módulo 16: TypeScript (Parte 06)



Boas-vindas!

Bem-vindo caro aspirante a dev!

Chegamos ao módulo de TypeScript, um superset da linguagem Javascript fortemente tipada e usualmente utilizada com classes. Neste módulo vamos aprender um pouco sobre os tipos de dados em TypeScript, enums, classes, tipos genéricos e interfaces. Também como criar seu primeiro servidor backend com TypeScript.

Ao final deste encontro você deverá:

- Entender o uso de Typescript para backend
- Instalar os módulos @types

Referências básicas:

- **Partial:**
<https://www.typescriptlang.org/docs/handbook/utility-types.html#partialtype>
- **Promise:** <https://www.educba.com/typescript-promise-type/>
- **Array:**
<https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#arrays>
- **UUID:** <https://www.npmjs.com/package/uuid>
- **SOLID:**
 - <https://www.youtube.com/watch?v=6SfrO3D4dHM>
 - <https://www.youtube.com/watch?v=mkx0CdWiPRA>
- **Backend:**
 - <https://acervolima.com/como-usar-o-typescript-no-back-end/>
 - <https://vidafullstack.com.br/javascript/como-usar-typescript-com-nodejs-e-express/>
 - **BASIC:** https://www.youtube.com/watch?v=r_wyef7ZK2g
 - **SOLID:** <https://www.youtube.com/watch?v=vAV4Vy4jfk&t=12s>

Objetivos:

Neste encontro iniciaremos nosso primeiro contato com o lado do backend do Typescript. Neste encontro iremos rever node, express, uuid e boas práticas com Typescript.

Exercícios:

1. Você deve criar um novo projeto e enviar os arquivos com zip (exceto os módulos do node), seguindo estes passos:
 - Instalar as dependências:
 - i. `express pg dotenv cors cookie-parser` ;
 - Instalar as devDependencies:
 - i. `typescript @types/node @types/express @types/cors` ;
 - ii. `@types/dotenv @types/uuid` ;
 - iii. `@types/cookie-parser nodemon concurrently` ;
 - Crie no Postgres um database chamado “db_typescript” e nele uma tabela chamada “accounts” com:
 - i. id, email, name e password;
 - Aplique os comandos e códigos necessário para levantar seu servidor na porta 8000;
 - Compile seu código typescript para JS;
2. Você deve utilizar o projeto da lição anterior e enviar os arquivos com zip (exceto os módulos do node), seguindo estes passos:

Os regex necessários serão:

- i. email:

```
/^(\w{1,}@\w{1,}\. (\w{3})) (\.\w{2}){0,1})$/gim
```

- ii. password:

```
/^\w{1,}$/gim
```

- iii. name:

```
/^[a-z]{1,}$/gim
```

Exercícios (Continuação):

- Assim como validamos nossos dados de entrada na lição passada(no frontend), você deve criar os **validators** para necessários cada dado recebido;
- Crie uma rota de **criação de usuários**, você deve utilizar os conceitos de **classe** e **regex** aprendidos até aqui para a validação.
 - i. **Nesta lição não é necessário processar os dados recebidos, apenas retornar os dados esperados.**
 - ii. As rotas devem ser:
 - 1. **Cadastro** - POST: “/accounts/”
 - a. Recebe: email, name e password;
 - b. Retorna: id, email, name;
 - 2. **Login** - POST: “/accounts/login”
 - a. Recebe: email e password;
 - b. Retorna: id;
 - c. Cookie: token = sessionID
 - 3. **Update** - PATCH (~~autenticada~~): “/accounts/”
 - a. Recebe: email, name e/ou password;
 - b. Retorna: id, email, name;
- Compile seu código typescript para JS;

Em síntese:

Nesta parte do módulo você conheceu um pouco sobre TypeScript, como conceitos básicos, os tipos nativos e iniciar um projeto.

Lembre-se que o foco deste estudo é sempre aprofundar o seu conhecimento em web e, portanto, não se limite às referências deste módulo e mergulhe no conhecimento sobre TypeScript assistindo tutoriais e manuais disponíveis na internet.

Na próxima aula continuaremos nossos estudos sobre o TypeScript. Bons estudos!