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:

- Aprimorar conceitos de backend e typescript.
- Conhecer e revisar conceitos de backend.

Referências básicas:

- Partial: https://www.typescriptlang.org/docs/handbook/utility-types.html#partialtype
- Promise: https://www.educba.com/typescript-promise-type/
- UUID: https://www.npmjs.com/package/uuid
- Como criar foreign key “circular”: https://stackoverflow.com/questions/13413158/how-do-i-create-a-circular-foreign-key-dependency-in-postgresql

- **Backend:**
  - ▸CORS:https://www.codeconcisely.com/posts/how-to-set-up-cors-and- cookie-session-in-express/
  - ▸ COOKIES: https://acervolima.com/cookies-http-em-node-js/
  - ▸SESSIONS:https://medium.com/@evangow/server-authentication-basicsexpress-sessions-passport-and-curl-359b7456003d

Objetivos:
Neste encontro vamos aprofundar nosso contato com o backend do Typescript.
Iremos também rever pg, session, cookies e validators.
Exercicio – Desafio:

- Criar uma API para gerenciamento de funcionários e equipes, deve incluir 3tipos de usuários e suas respectivas permissões. Leia atentamente as regras de negócio e os endpoints antes de começar:

- Tipos de Usuário:

    - Administrador: administra o sistema todo, tem permissão geral. Não é membro de nenhuma equipe.
    - Líder: usuário que é membro de uma equipe e é líder dela (pode liderar uma única equipe).
    - Funcionário: usuário que é membro de uma equipe mas não é líder dela (só pode estar numa única equipe).
    - Autenticado: usuário que não pertence a nenhuma equipe e não é Administrador.

- Devem ter Validators, como os que foram aprendidos ao decorrer das aulas. Todos os dados inseridos devem ser validados pelo backend.

- Cada administrador tem permissão para ver e modificar qualquer dado de qualquer usuário ou equipe, a não ser que a ação viole alguma outra regra aqui descrita. Um administrador não é membro de nenhuma equipe (logo também não é líder de nenhuma equipe).

- Os administradores são pré-cadastrados no sistema e não podem deixar de ser administradores. Mas um administrador pode elevar outro usuário à categoria de Administrador, desde que esse usuário não seja membro de nenhuma equipe (logo também não seja líder de nenhuma equipe).

- Uma equipe não pode ser deletada se tiver membros. Um usuário pode ser deletado mesmo que seja membro de uma equipe (será removido da equipe), exceto se ele for o líder da equipe.

- Cada equipe sempre tem um líder, que tem permissão para ver os dados da equipe e de seus membros, e permissão para alterar os dados da equipe mas não de seus membros. O líder pode adicionar e remover membros da sua equipe, mas não pode sair da equipe. Pode também transferir o cargo de líder para outro membro da equipe. Finalmente, pode também ver os dados de outras equipes e de seus líderes (mas não

demais membros). Nunca vê o password de nenhum outro usuário. Duas equipes não podem ter o mesmo líder.

- Cada funcionário pode alterar seus próprios dados e ver os dados da própria equipe e os outros membros da própria equipe (exceto password). O funcionário não pode mudar ou sair da equipe por conta própria. Também não pode estar em mais de 1 equipe. Finalmente, não pode ser adicionado a uma equipe caso já faça parte de outra, mas pode ser removido da equipe.

- Podem existir usuários sem equipe (é o usuário Autenticado).

Entidades do banco de dados:

- Usuario: id(uuid PK), username(unique), email, first_name, last_name, password, squad(fk), is_admin(boolean)

- Equipe: id(uuid PK), name(unique), leader(fk)

EndPoints:

- GET “/users/me” - Ver seu próprio usuário (Todos). Retorna a entidade Usuario.
- GET “/users/” - Ver todos os usuários (Administrador). Retorna uma lista da entidade Usuario.
- GET “/users/:user_id” - Ver determinado usuário (Administrador, Líder da equipe, Líder das demais equipes somente se “user_id” designar um líder). Retorna a entidade Usuario.
- GET “/teams/” - Ver todas as equipes (Administrador, Líder de qualquer equipe). Retorna uma lista da entidade Equipe.
- GET “/teams/:team_id” - Ver determinada equipe (Administrador, Líder de qualquer equipe, Funcionário da equipe). Retorna a entidade Equipe.
- GET “/teams/:team_id/members” – Ver os membros de determinada equipe (Administrador, Líder da equipe, Funcionário da equipe). Retorna uma lista da entidade Usuario.
- POST “/login” – Se autenticar, ganhando uma sessão (Todos e não autenticado). Retorna um token de acesso.
- POST “/users/” - Criar um novo usuário (Todos e não autenticado). Retorna a entidade Usuario recém-criada

- POST “/teams/” - Criar uma nova equipe (Administrador). Retorna a entidade Equipe recém-criada
- POST “/teams/:team_id/member/:user_id” - Adicionar membro na equipe (Administrador, Líder da equipe). Retorna a entidade Usuario.
- PATCH “/users/:user_id” - Atualizar usuário (Somente o próprio usuário). Retorna a entidade Usuario.
- PATCH “/teams/:team_id” - Atualizar equipe (Administrador, Líder da equipe). Retorna a entidade Equipe.
- DELETE “/teams/:team_id/member/:user_id” - Retirar membro da equipe (Administrador, Líder da equipe). Retorna a entidade Usuario.
- DELETE “/users/:user_id” - Deletar usuário (Administrador). Retorna a entidade Usuario deletada.
- DELETE “/teams/:team_id” - Deletar equipe (Administrador). Retorna a entidade equipe deletada.
- DELETE “/logout” - Deletar sessão (Todos e autenticado)

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!