



growdev



Programa Starter Full Stack Web Developer

Todo o conteúdo deste documento é propriedade da Growdev. O mesmo pode ser utilizado livremente para estudo pessoal. É proibida qualquer utilização desse material que não se enquadre nas condições acima sem o prévio consentimento formal, por escrito, da Growdev. O uso indevido está sujeito às medidas legais cabíveis.

Exercício ORM - Sequelize

Agora que entendemos o que é um ORM e como utilizar o Sequelize. Vamos exercitar o que aprendemos.

O objetivo deste exercício é fixar aquilo que vimos em aula.

Vamos relembrar os seguintes pontos:

- Como iniciar um aplicação NodeJS.
- Como configurar o Sequelize no projeto.
- Como utilizar as models do Sequelize.



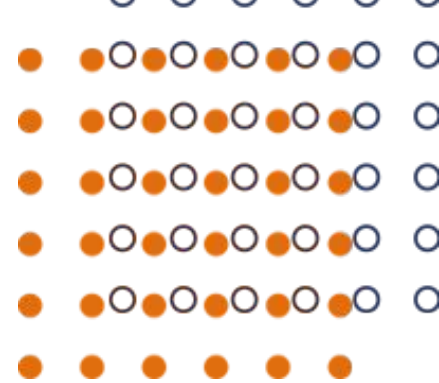
O que vamos fazer?

- 1 - Crie uma aplicação NodeJS.
- 2 - Crie um banco de dados no Heroku.
- 3 - Configure o sequelize no seu projeto.
- 4 - Crie uma migration para criar um usuário. (Detalhes próximo slide).
- 5 - Crie uma migration para criar recados. (Detalhes no próximo slide).

-> Veja próximo slide.



Detalhes das Models



- User (Usuário).
 - Table "users".
 - Columns:
 - uid
 - nome
 - email
 - created_at
 - update_at.

- Post (Recado).
 - Table "posts".
 - Columns:
 - uid
 - content
 - user_uid
 - created_at
 - update_at

Controllers

Cada model anterior precisar ter suas controllers.

Uma controller deve conter apenas os seguintes métodos:

- index
- show
- store
- update
- delete

Bora implementar todos eles...



Rotas, precisa criar?

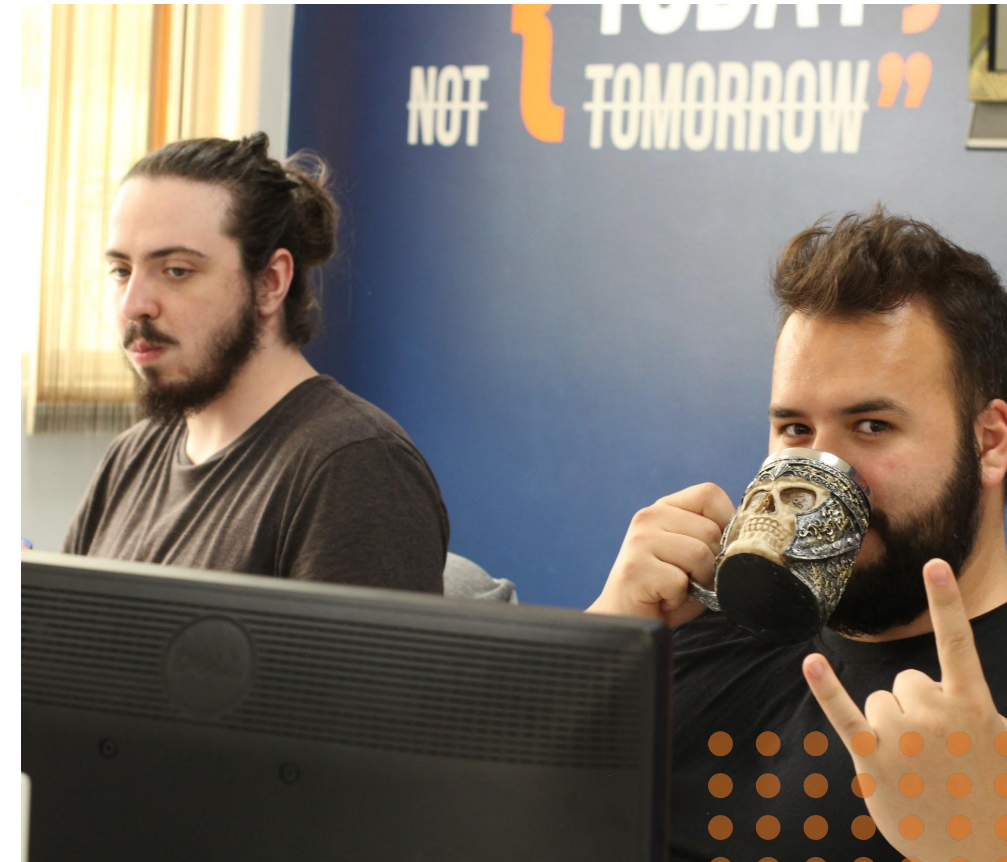
Nesta aplicação precisamos criar as rotas de acesso. Quais são:

- Users
 - [GET] /users (index).
 - [POST] /users (store).
 - [PUT] /users/:uid (update).
 - [GET] /users/:uid (show).
 - [DELETE] /users/:uid (delete).
- Posts
 - [GET] /posts (index).
 - [POST] /posts (store).
 - [PUT] /posts/:uid (update).
 - [GET] /posts/:uid (show).
 - [DELETE] /posts/:uid (delete).



Deverá ter

- Validações dos valores de entrada
- Validações se os recursos que serão alterados existem, antes de realizar cada operação
- Status de retorno de acordo com a operação realizada
- Padronização nos retornos de dados
- O métodos "index" do recurso user, ou seja, a listagem, deverá contemplar paginação e filtros por nome e e-mail,
- O métodos "index" do recurso post, ou seja, a listagem, deverá contemplar paginação e filtros por conteúdo e usuário (id)



Correção, vai ter?

A correção e dúvidas será na aula ao vivo, aproveitem e perguntem o máximo para seu mentor. Não fique com dúvidas.



#boragrowdevers

