FRONT - git@github.com:tryber/herocker-exercise-frontend.git

BACK - git@github.com:tryber/herocker-exercise-backend.git

Agora, a prática

Para os exercícios que vão fazer um grupo, use os seguintes repositórios:

- Front-end
- Back-end

1. Realize o deploy de uma aplicação Front-end no Heroku utilizando um build-pack
2. Realize o deploy de uma API Node no Heroku utilizando Docker
3. Crie um banco de dados no SUPABASE
4. Configure as variáveis de ambiente do Back-end, o arquivo de configuração do sequelize está no caminho `/src/sequelize/config/config.js`
5. Conecte sua aplicação back-end ao banco de dados.

- Use o script `utils/testSequelizeConnection.js` para testar sua conexão <!-- - Crie uma rota para fornecer as informações das pessoas usuárias retornadas diretamente do banco de dados.-->

6. Popule o banco de dados com informações de pessoas usuárias ou qualquer tipo de dados que viram uma tabela
    - Valide os próximos passos pela interface do SUPABASE
    - Use `npx sequelize-cli db:migrate` para fazer a migração e criar as tabelas no banco
    - Para o popular o banco você pode usar `npx sequelize-cli db:seed:all`
7. Consuma sua API pela aplicação front-end, renderize todas as informações retornadas da API.
    - Será necessário configurar o CORS -> link aqui
8. Crie Actions do Github de lint para Front-end e Back-end;

LINK: http://expressjs.com/en/resources/middleware/cors.html

Bônus

Usando os últimos exercícios, temos alguns desafios aqui:

- Front-end: Action para gerar preview
- Back-end: Action de teste com Jest
- Criar uma pipeline no Heroku
- Criar aplicação Front-end usando Dockerfile

https://kledenai.medium.com/deploy-reactjs-com-docker-1769cf7a5f74