

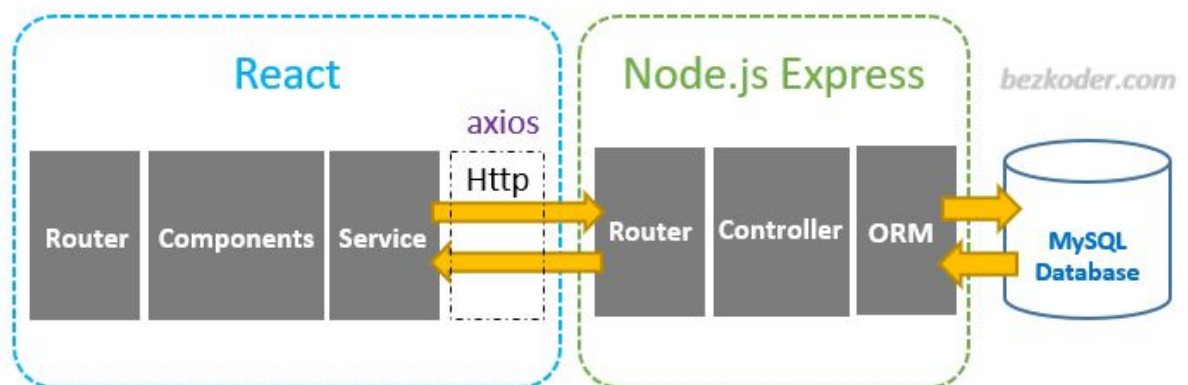
Documentação do Projeto React PokemonApp

O projeto React Pokemon App consiste num CRUD desenvolvido pelo framework React.js com Axios, React-Router e Bootstrap na parte do Front-end, o Node.js como servidor back-end, framework express.js para a API MySQL como banco de dados.

Neste projeto foi construído uma aplicação Full-Stack onde:

1. O Crud tem Nome, Tipo e Status de Capturado
2. O usuário pode cadastrar, deletar e editar as informações.
3. Existe também uma barra de pesquisa para filtragem.
4. Mudar o status de published/pending usando publish/unbplish botão
5. Atualizar o objeto no banco de dados e deleta-lo do banco.

O Projeto desenvolvido usará esta arquitetura:



1. O Node.js exporta os serviços de REST da API interagindo com o banco de dados usando o Sequelize ORM.
2. O React manda o protocolo HTTP para o cliente e retorna usando o Axios, consumindo o dado vindo dos componentes. React-Router é usado para as rotas das páginas.

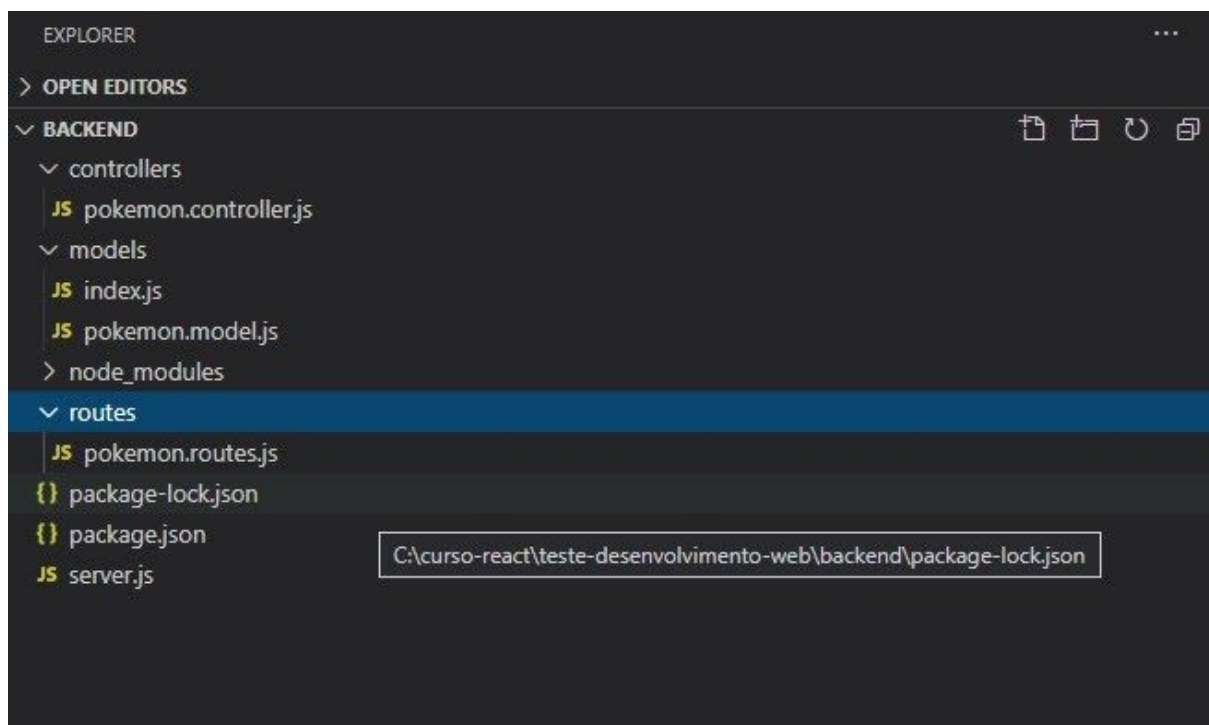
○ Node.js e Express.js Back-End

Este serão as rotas que a API irá exportar

Método	URL	Ações
GET	api/pokemons	pegar todos os pokemons

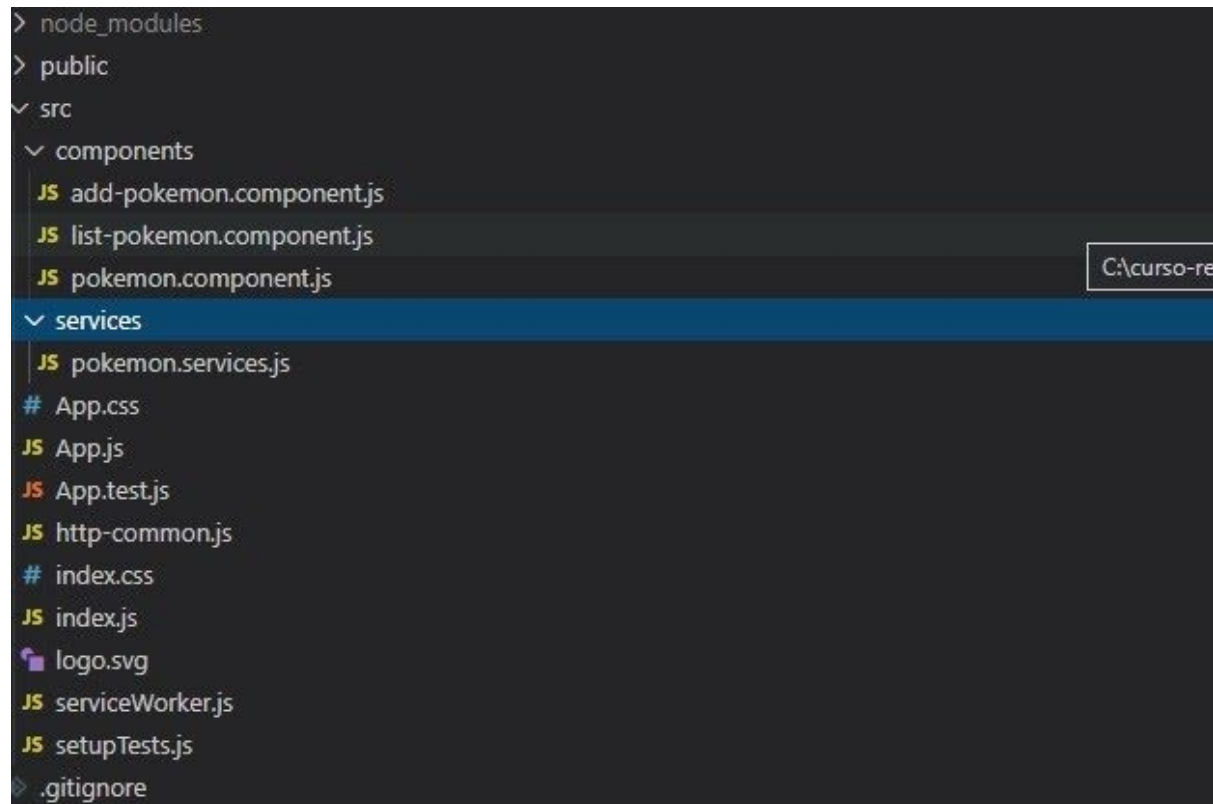
GET	api/pokemons/:id	pega os pokemons pelo id
POST	api/pokemons	adiciona um novo objeto
PUT	api/pokemons/:id	atualiza o pokemon
DELETE	api/pokemons	remove todos
DELETE	api/pokemons/:id	remove pelo id
GET	api/pokemons?nome=[kw]	filtrar o pokemon pelo nome

- **Estrutura da API**



1. O arquivo db.config.js será o arquivo que fará conexão com o banco de dados MySQL, exportando os parametros e o Sequelize é que realizará as conexões do banco.
2. Os models serão os que farão as configurações para o banco de dados no model/index.js e criando os modelos de dados do models/pokemon.model.js para o banco de dados.
3. Os controllers no pokemon.controller.js
4. Rotas estarão no routes/pokemon.routes.js

- **Estrutura do projeto no Front-end**



- O arquivo package.json é que contém os módulos Axios, React-Router-dom e o bootstrap para o design
- Os componentes add-pokemon.component.js, list-pokemon.component.js e pokemon.component.js
- O arquivo http-common.js inicializa o protocolo http baseado nas url e cabeçalhos
- O pokemonServices envia os métodos HTTP comunicar com a API.
- o env configura a porta para a aplicação.

- **Como Rodar**

1. abra o prompt de comando
2. vá no diretório do projeto
3. roda o comando npm install para baixar as dependências
4. e em seguida vá para o diretório da API
5. abre um segundo prompt

6. Rode o comando `node server` para subir o servidor.
7. Abra um terceiro prompt para criar o banco
8. crie o banco `pokemonDB`, `usuario:root`, `senha:123456`
9. crie uma tabela chamada `pokémon` com os seguintes atributos: `Nome`, `Tipo`, `Capturado`