

Rocketseat / bootcamp-gostack-desafios

<> Code

! Issues 7

🔗 Pull requests 1

▶ Actions

📁 Projects

📖 Wiki

🛡 Security

👤 master ▾

bootcamp-gostack-desafios / desafio-conceitos-nodejs /

...

Gitpod



StefanoSaffran committed on 25 May ...

🕒 History

..



assets

3 months ago



README.en.md

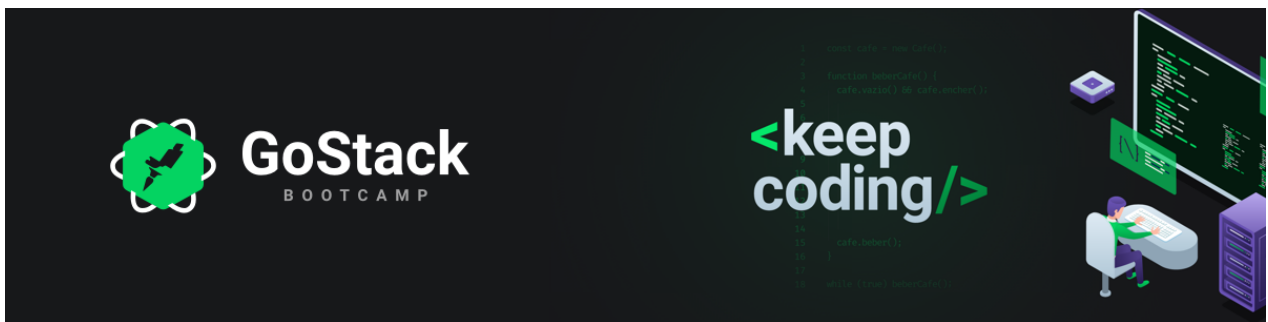
last month



README.md

last month

README.md



Desafio 02: Conceitos do Node.js

"Não espere para plantar, apenas tenha paciência para colher!"

languages 0

made by

Rocketseat

license

MIT

🌟 Stars

488

[Sobre o desafio](#) | [Entrega](#) | [Licença](#)

Sobre o desafio

Nesse desafio, você deve criar uma aplicação para treinar o que você aprendeu até agora no Node.js!

Essa será uma aplicação para armazenar repositórios do seu portfólio, que irá permitir a criação, listagem, atualização e remoção dos repositórios, e além disso permitir que os repositórios possam receber "likes".

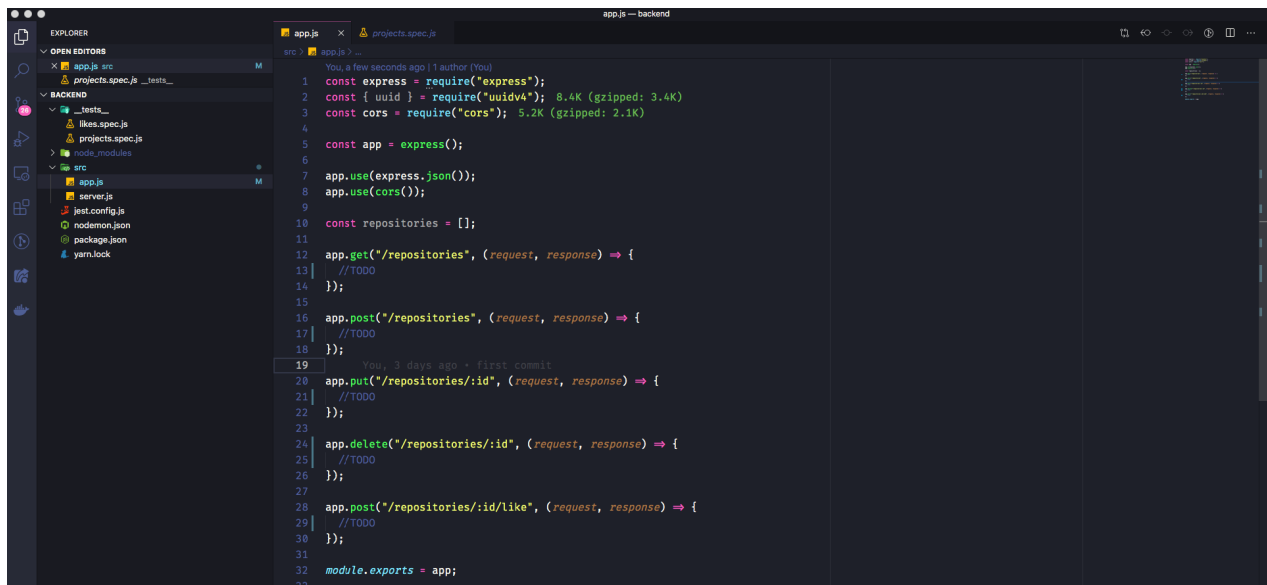
Template da aplicação

Para te ajudar nesse desafio, criamos para você um modelo que você deve utilizar como um template do Github.

O template está disponível na seguinte url: [Acessar Template](#)

Dica: Caso não saiba utilizar repositórios do Github como template, temos um guia em [nosso FAQ](#).

Agora navegue até a pasta criada e abra no Visual Studio Code, lembre-se de executar o comando `yarn` no seu terminal para instalar todas as dependências, e você terá algo parecido com isso:



Rotas da aplicação

Agora que você já está com o template clonado, e pronto para continuar, você deve abrir o arquivo `app.js`, e completar onde não possui código com o código para atingir os objetivos de cada rota.

- **POST /repositories** : A rota deve receber `title`, `url` e `techs` dentro do corpo da requisição, sendo a URL o link para o github desse repositório. Ao cadastrar um novo projeto, ele deve ser armazenado dentro de um objeto no seguinte formato: `{ id: "uuid", title: 'Desafio Node.js', url: 'http://github.com/...', techs: ["Node.js", "..."], likes: 0 }`; Certifique-se que o ID seja um UUID, e de sempre iniciar os likes como 0.
- **GET /repositories** : Rota que lista todos os repositórios;
- **PUT /repositories/:id** : A rota deve alterar apenas o `title`, a `url` e as `techs` do repositório que possua o `id` igual ao `id` presente nos parâmetros da rota;
- **DELETE /repositories/:id** : A rota deve deletar o repositório com o `id` presente nos parâmetros da rota;
- **POST /repositories/:id/like** : A rota deve aumentar o número de likes do repositório específico escolhido através do `id` presente nos parâmetros da rota, a cada chamada dessa rota, o número de likes deve ser aumentado em 1;

Dica: Acima utilizamos `POST` em uma rota, mesmo ela alterando o número de likes do repositório sem criar nada diretamente.

Se dividirmos semanticamente as responsabilidades da nossa aplicação em entidades, o `like` seria uma entidade, e `repository` seria outra entidade.

Com essa separação, temos diferentes regras de negócio para cada entidade, assim, ao chamar a rota de `like` e adicionamos apenas um like, podemos interpretar que estamos criando um novo like, e não atualizando os likes.

Então por que não usar `PUT` no lugar de `POST` ? Justamente por estarmos "criando" UM novo like, e não atualizando o número de likes para qualquer outro valor.

Talvez fique difícil enxergar por ser apenas um número, mas pense que cada like seja salvo em uma tabela no banco junto do usuário que realizou esse like. Agora fica mais claro que você está criando um novo like, certo?

Bons estudos <3

Especificação dos testes

Em cada teste, tem uma breve descrição no que sua aplicação deve cumprir para que o teste passe.

Caso você tenha dúvidas quanto ao que são os testes, e como interpretá-los, dê uma olhada em [nosso FAQ](#).

Para esse desafio temos os seguintes testes:

- **should be able to create a new repository** : Para que esse teste passe, sua aplicação deve permitir que um repositório seja criado, e retorne um json com o projeto criado.
- **should be able to list the repositories** : Para que esse teste passe, sua aplicação deve permitir que seja retornado um array com todos os repositórios que foram criados até o momento.
- **should be able to update repository** : Para que esse teste passe, sua aplicação deve permitir que sejam alterados apenas os campos `url` , `title` e `techs` .
- **should not be able to update a repository that does not exist** : Para que esse teste passe, você deve validar na sua rota de update se o id do repositório enviado pela url existe ou não. Caso não exista, retornar um erro com status `400` .
- **should not be able to update repository likes manually** : Para que esse teste passe, você não deve permitir que sua rota de update altere diretamente os likes desse repositório, mantendo o mesmo número de likes que o repositório já possuía antes da atualização. Isso porque o único lugar que deve atualizar essa informação é a rota responsável por aumentar o número de likes.
- **should be able to delete the repository** : Para que esse teste passe, você deve permitir que a sua rota de delete exclua um projeto, e ao fazer a exclusão, ele retorne uma resposta vazia, com status `204` .
- **should not be able to delete a repository that does not exist** : Para que esse teste passe, você deve validar na sua rota de delete se o id do repositório enviado pela url existe ou não. Caso não exista, retornar um erro com status `400` .

- **should be able to give a like to the repository** : Para que esse teste passe, sua aplicação deve permitir que um repositório com o id informado possa receber likes. O valor de likes deve ser incrementado em 1 a cada requisição, e como resultado, retornar um json contendo o repositório com o número de likes atualizado.
- **should not be able to like a repository that does not exist** : Para que esse teste passe, você deve validar na sua rota de like se o id do repositório enviado pela url existe ou não. Caso não exista, retornar um erro com status `400`.

Entrega

Esse desafio deve ser entregue a partir da plataforma Skylab, envie o link do repositório que você fez suas alterações. Após concluir o desafio, fazer um post no LinkedIn e postar o código no Github é uma boa forma de demonstrar seus conhecimentos e esforços para evoluir na sua carreira para oportunidades futuras.

Licença

Esse projeto está sob a licença MIT. Veja o arquivo [LICENSE](#) para mais detalhes.

Feito com  by Rocketseat  [Entre na nossa comunidade!](#)