

# Treinow - Node Challenger

## Descrição do Problema

A **Treinow** está começando a sair do campo das ideias e para isso precisamos da sua ajuda. Toda plataforma precisa ter uma API completa e robusta, que responda às requisições rapidamente. E para dar início a esse projeto, queremos que você nos dê alguma contribuição.

Desenvolva uma API que atenda aos seguintes requisitos:

- A API precisa ser REST
- Você pode utilizar o quaisquer bibliotecas no seu projeto, a única restrição é que seja **feito utilizando Node**
- A API deve se comunicar com um banco de dados
  - Nesse caso, é necessário que seja um banco de dados embutido, como SQLite
- A API deve possuir um módulo de usuários
  - Cadastro. Não deve ser possível cadastrar dois usuários com o mesmo email, nem com o mesmo documento. Serão registrados as seguintes informações do usuário:
    - Nome
    - Email
    - Documento
    - Tipo (Personal Trainer ou Aluno)
    - Senha
  - Login. Após cadastrado, o usuário terá disponível um **endpoint para fazer login, se autenticar** (GET /auth).
- A API deve possuir um **mecanismo de autenticação**, ou seja, os endpoints precisam estar protegidos, **apenas usuários autenticados terão acesso às respostas da API**
  - Para esse mecanismo de autenticação, utilize o JSON Web Token
- Além do módulo de usuários, a API deve possuir os seguintes endpoints:

**ATENÇÃO! Os endpoints abaixo são voltados para os usuários do tipo ALUNO**

- Registro de treino (*POST /store/workout*)
  - Nesse endpoint **o usuário que está autenticado** poderá registrar que iniciou um treino, **gravando data e hora de início da atividade**. Não precisa se preocupar em gravar quando o treino terminou

- Deletar conta (*DELETE /account/delete*)
  - Nesse endpoint o usuário poderá deletar a sua conta, apagando os seus registros pessoais, bem como todos os dados que estão vinculados a ele, como os registros de treinos, por exemplo.
- Avaliar personal (*POST /personal/<personal\_id>/rating*)
  - Esse endpoint possui um parâmetro na rota, o *personal\_id*, trata-se do identificador, no banco de dados, do personal que usuário vai avaliar, o usuário irá passar no *body* dessa requisição uma nota entre 1-5, para avaliar o personal em questão. **Para avaliar um personal, o aluno precisa ser aluno desse personal**

**ATENÇÃO! Os endpoints abaixo são voltados para os usuários do tipo PERSONAL TRAINER**

- Cadastro de aluno (*POST /personal/student*)
  - Nesse endpoint o personal irá cadastrar um usuário, **que precisa ser do tipo aluno**, como seu aluno. Ele passará o identificador do usuário (que será seu aluno) no *body* da requisição
- Cadastro de treino (*POST /personal/workout*)
  - Nesse endpoint o personal irá **cadastrar treinos**, que ficará disponível para consultas posteriores. Os seguintes dados serão registrados:
    - Duração (em minutos)
    - Nome (ex: Posteiros)
    - Data (Registra o dia em que o treino foi cadastrado)
- Consulta de treinos (*GET /personal/workout*)
  - Nesse endpoint são **listados todos os treinos cadastrados pelo personal que está autenticado**

## Considerações Importantes

- A interpretação desta descrição faz parte da avaliação
- Cada endpoint descrito, possui entre parentes o tipo da requisição (GET, POST, DELETE) e o caminho da mesma, o endereço que fornece aquele recurso, por exemplo: <*api\_address*>/personal/workout. Siga esse padrão ao desenvolver a sua solução.
- O código do projeto deve ser disponibilizado em um repositório Git (Github, Gitlab, Bitbucket, etc).
- O desafio deve ser **implementado até o dia 27/2 (segunda-feira) às 23h59min**;
  - Toda implementação é válida, então se não completar o desafio, envia tudo o que conseguiu, isso pode fazer total diferença;