

Desafio Final

No desafio final você criará uma API REST para um **sistema de agendamentos de compromissos** (calendário). Além de criar a API REST seria muito interessante que você criasse uma **interface para o usuário** que consuma essa API em qualquer ferramenta front-end que você já conheça como ReactJS, VueJS, React Native, Angular, etc... Caso você ache que não tenha conhecimento suficiente para montar a interface do usuário, tudo bem, entregue apenas a API.

Em anexo com esse documento deixei um **exemplo de layout** de uma aplicação mobile que consumiria essa API REST criada nesse desafio. Lembrando que esse layout é para auxílio e não precisa ser seguido para o projeto, a interface pode ser da maneira que você quiser desde que cubra todas rotas listadas abaixo.

A aplicação deve possuir testes **unitários** e de **integração** cobrindo **no mínimo 80%** do código medindo pela ferramenta Istanbul (nyc).

No fim você deve publicar a aplicação em um endereço online acessível publicamente para que eu possa realizar os testes visuais. Caso não tenha criado interface, me envie apenas o código do projeto.

Requisitos funcionais

1. O usuário deve poder criar uma conta com nome, e-mail e senha;
2. O usuário deve poder se autenticar na aplicação com e-mail e senha;
3. O usuário deve poder alterar seu nome e senha informando a senha antiga, a senha nova e a confirmação da senha nova;
4. O usuário deve poder cadastrar eventos em seu calendário com título, localização, data e horário;
5. O usuário deve poder listar os eventos cadastrados por data;
6. O usuário deve poder excluir um compromisso;
7. O usuário deve poder compartilhar um compromisso informando o e-mail do destinatário. Assim que compartilhado, o destinatário deve receber todas informações do evento por e-mail;
8. O usuário deve receber notificações dos eventos por e-mail 30 minutos antes do mesmo;

Requisitos não-funcionais

1. A aplicação pode utilizar banco NoSQL ou SQL, você escolhe;

Regras de negócio

1. O e-mail do usuário é único;
2. O usuário não pode alterar seu e-mail;
3. Não deve ser possível cadastrar dois eventos no mesmo horário no calendário de um usuário;

Dicas

- Para fazer a funcionalidade de notificações 30 minutos antes do evento você precisará utilizar o recurso de CRON, para isso utilize a biblioteca: <https://github.com/kelektiv/node-cron>.

Entrega

Para realizar a entrega e receber correção basta, após pronto, adicioná-lo ao seu Github utilizando Git ou compactá-lo em formato ZIP ou RAR **removendo a pasta node_modules** e enviar para o e-mail gonode@rocketseat.com.br com o assunto: **Desafio Final GoNode - SEU NOME**

No corpo da mensagem você pode adicionar dúvidas ou qualquer outro tipo de consideração.

Boa sorte!

“O sucesso não é feito de presentes, mas de recompensas”!