

Gustavo Santos Costa

Tomadas de decisão e da codificação do teste técnico da Samplemed

Resiliência: O que fazer para mitigar possíveis erros e controlar os possíveis erros recebidos da API?

No caso da api do Blog construída foi utilizada de alguns validadores nos serializers para controlar erros na hora de criação dos artigos. Além disso foram feitos testes manuais para ver se de fato a API estava conseguindo realizar as funções no backend de forma correta, alterando variáveis e tamanhos do texto digitado nos campos do JSON. Testes unitários poderiam vir a ser uma opção para controlar ainda mais o limite de erros.

Performance: Quais boas práticas são aplicadas em banco de dados e no código para garantir performance?

Foi utilizado do SQLite3 padrão do django como banco de dados, o qual possui uma boa compatibilidade com o mesmo. O número de tabelas criadas foram 3, e foi utilizada a tabela User vinda do django.contrib.auth.models. Foi utilizado das Generic Views do DRF para facilitar o desenvolvimento e garantir uma performance aceitável.

Segurança: Como garantir segurança para as APIs do sistema?

No quesito de segurança foram utilizados de métodos *IsAuthenticated* e *IsAuthenticatedOrReadOnly* da própria biblioteca de permissões do *rest_framework* bem como a utilização de JWT authentication a partir do *simplejwt* também do *rest_framework*. Isso restringiu que apenas usuários logados ou com o token pudessem realizar funções de CRUD nas APIs. Além disso, foram desenvolvidos 2 métodos no app de *Articles* para aumentar a segurança, sendo eles os: *IsOwner* e o *IsOwnerReadOnly*, que cumprem um papel similar aos da biblioteca do *rest_framework*.

Simultaneidade: Como trabalhar com simultaneidade se milhares de requisições forem solicitadas simultaneamente?

No caso de simultaneidade de requisições talvez fosse interessante trabalhar com um servidor Async que iria processar múltiplas requisições em paralelo com apenas uma Thread. Entretanto, mesmo sem ser um servidor Async, se o servidor de produção for configurado corretamente múltiplas requisições podem ser tratadas ao

mesmo tempo. Utilizando de uma framework estilo a FastAPI ajudaria nesse processo.