

Performance Evaluation of REST and GraphQL APIs

Searching Nested Objects

Nesse trabalho, os autores compararam o desempenho de duas arquiteturas cliente-servidor para API's: GraphQL e REST. Realizando um experimento para comparar o tamanho, tempo, e processamento para se gerar uma resposta do servidor.

A primeira arquitetura analisada é a REST, do qual utiliza de HTTP GET e POST para realiza a comunicação do cliente com o servidor, via uma url. Essa arquitetura por si só já apresenta alguns problemas, visto que o cliente precisa fazer várias requisições ao servidor quando o objeto de desejo é aninhado com outros. Outro problema dessa arquitetura citado pelo autor é o overfatching. Dados que não são tão uteis para o cliente acabam por ser enviados juntos. O versionamento também é um fator que não torna o REST uma boa solução, já que ao criar uma nova versão, clientes antigos podem acabar sendo afetados.

Já a segunda arquitetura, a GraphQL foi criada pelo Facebook em 2012 para circular alguns dos problemas que o REST possui, também funciona utilizando HTTP GET e POST. Porém ao invés do cliente enviar uma requisição solicitando um documento inteiro para a API, o cliente solicita apenas o que ele deseja, similar a uma consulta em um banco de dados. Eliminando os problemas citados anteriormente que estão presentes na arquitetura REST, como por exemplo, o envio de informações que o cliente não necessita.

O autor realizou um experimento usando dois protótipos criado por ele, duas API's, uma REST e uma GraphQL, onde ambas implementavam uma aplicação de gerenciamento de armazém. Nesse armazém haviam prateleiras, níveis e produtos. Foram considerados cenários onde haviam 1000, 10000 e 30000 registros. Duas consultas foram submetidas as API's: qual item ocupada a maior quantidade de prateleiras do armazém e qual o endereço de um item x no armazém. Os resultados mostraram que o tempo de utilização de CPU era maior com REST (no servidor), assim como o consumo de memória. O tamanho da resposta e enviada pelo Servidor e o tempo era menor usando GraphQL em todos os cenários.

Logo, ele concluiu que o GraphQL apresentou um desempenho maior que o REST, assim como esperado. Além de mencionar também que o uso do GraphQL vem aumentando consideravelmente nos últimos anos, mas que mesmo assim, o REST continua ser usado, devido a sua facil implementação.