



Java

CQRS





O que é CQRS e qual é a sua importância na arquitetura de software?

CQRS, ou Command Query Responsibility Segregation, é um padrão de arquitetura que separa as operações de leitura (queries) das operações de escrita (comandos) em um sistema.

Isso permite otimizar cada caminho de maneira independente, melhorando o desempenho e a escalabilidade do sistema.



Quais são os principais componentes de um sistema CQRS em Java e como eles se comunicam?

Geralmente consiste em duas partes principais: o lado de comando (Command) e o lado de consulta (Query). O lado de comando é responsável por receber, processar e armazenar comandos, enquanto o lado de consulta é responsável por fornecer dados otimizados para leitura.

Eles se comunicam por meio de eventos ou bancos de dados separados.



Quais são os benefícios e desafios de implementar CQRS em uma aplicação Java?

Os benefícios de implementar CQRS em Java incluem o aumento da escalabilidade, a capacidade de otimizar as operações de leitura e escrita independentemente, e uma maior flexibilidade na modelagem de domínio.

Mas, os desafios podem incluir a complexidade adicional, a necessidade de sincronização de dados entre os lados de comando e consulta, e um maior overhead de desenvolvimento.



Qual é a diferença entre CQRS e Event Sourcing, e em que situações você escolheria um sobre o outro?

CQRS e Event Sourcing são frequentemente usados juntos, mas eles abordam problemas diferentes. CQRS se concentra na separação de comandos e consultas, enquanto Event Sourcing é uma técnica que consiste em armazenar todos os eventos de mudança de estado em um sistema.

Você escolheria CQRS quando desejar separar claramente comandos e consultas, enquanto Event Sourcing seria usado quando precisar rastrear o histórico completo de alterações no estado do sistema.



luisfabriciodellamas