

Serviços Assíncronos

sexta-feira, 17 de setembro de 2021 13:30

- Comunicação síncrona:
 - Uma chamada é feita para um servidor remoto. Enviamos blocos até que a operação seja concluída;
 - Pode ser mais fácil de raciocinar. Sabemos quando as coisas foram concluídas com sucesso ou não.
- Comunicação assíncrona:
 - O chamador não espera a conclusão da operação antes de retornar. Pode até não se importar se a operação será concluída ou não.
 - Pode ser muito útil para trabalhos de longa execução, mas manter uma conexão aberta por um longo período de tempo entre o cliente e o servidor é impraticável. Também funciona muito bem quando você precisa de baixa latência, onde bloquear uma chamada enquanto espera pelo resultado mas pode tornar as coisas mais lentas.

Disparar solicitações e presumir que as coisas funcionaram, a menos que seja informado o contrário, pode garantir que a experiência do usuário permaneça responsiva mesmo quando a rede estiver muito lenta. Por outro lado, a tecnologia para lidar com a comunicação assíncrona pode ser um pouco mais envolvente, mas é um conceito muito importante de entender e nosso uso para plataforma de enfileiramento de mensagens como RabbitMQ torna esse padrão assíncrono muito mais fácil de implementar.

Esses dois modos diferentes de comunicação podem permitir dois estilos diferentes de colaboração:

- solicitação e resposta ou
- baseado em eventos

Com solicitação e resposta, um cliente inicia uma solicitação e espera pela resposta. Este modelo se alinha bem com a comunicação síncrona, mas também pode funcionar com a comunicação assíncrona. Posso iniciar uma operação e registrar um retorno de chamada solicitando ao servidor que me informe quando minha operação for concluída.

Com uma colaboração baseada em eventos, invertemos as coisas. Em vez de o cliente iniciar solicitações pedindo coisas a serem feitas, ele diz que isso aconteceu e espera que outras partes saibam o que fazer. Os sistemas baseados em eventos, por natureza, são assíncronos. Isso significa que o processamento dessas mensagens não precisa ser centralizado em nenhum lugar. Em vez disso, podemos ter muitos consumidores para processar mensagens.

A colaboração baseada em eventos também é altamente dissociada. O cliente que emite um evento não tem como saber quem ou o que irá reagir a ele, o que também significa que você pode adicionar novos assinantes ao evento sem que o cliente precise saber.