.DOT LIVE

Caching com ASP.NET Core

AO VIVO

INSTRUTOR

LUIS FELIPE, 3x Microsoft MVP

Sobre a DotLive

Sobre a DotLive

- Série de aulas semanais sobre carreira e programação com .NET, com conteúdo de qualidade e atualizado
- Sempre nas quartas-feiras às 20h, aqui no Zoom
- **As aulas ficam gravadas? Sim e não.**
    - Sim, mas apenas para alunos e mentorandos!

O que vamos ver hoje

O que veremos hoje?

- O que é Caching
- Benefícios
- Caching em Memória
- Caching Distribuído com Redis

O que é Caching

O que é Caching

- Caching é uma técnica que melhora a performance de aplicações através de armazenamento e acesso de dados em um cache
- O cache é um componente que vai armazenar dados de maneira a permitir seu acesso mais rápido no futuro
- Origem: cache foi criado para servir dados de maneira mais rápida ao armazenar em uma memória mais eficiente, ao invés de uma memória persistente, que é mais lenta
- Vamos tratar dos usos de caching mais direcionados a desenvolvimento de aplicações

O que é Caching

- Um exemplo do uso de caching em aplicações é no armazenamento de dados que são originados de bancos de dados, ou mesmo dados que simplesmente tem um processamento/cálculo grande para ser gerado
    - No caso de bancos de dados, o caching diminuiria a quantidade de requisições ao banco de dados, tendo um menor consumo de recursos dele e aumentando a velocidade da requisição
    - Já no caso de processamento ou cálculos mais complexas, caso o dado for utilizado em outros momentos, refazê-los resultaria em re-trabalho que poderia ser evitado com o armazenamento de seu resultado em um cache

O que é Caching

- Quando utilizar
  - Consultas no banco de dados
  - Requisições HTTP
  - Cálculos computacionais intensivos
  - Não necessariamente APENAS com dados que mudam pouco
    - Às vezes, mesmo com o dado sendo alterado de vez em quando, simplesmente queremos mais performance!

O que é Caching

- Existem dois tipos principais de Caching:
  - Em memória
  - Distribuído

Mas quais são os seus benefícios?

Benefícios de Caching

- Melhora de performance da aplicação, com latências menores
- Redução de custos de banco de dados
- Melhoria de performance no banco de dados, por redução de carga
- Aumentar a taxa de transferência

Caching em Memória

Caching em Memória

- O caching em memória utiliza a memória do servidor para armazenar dados, resultando em performance melhor que acesso a banco de dados
- Porém, aqui temos um ponto de atenção: ela é adequada para aplicações que contém uma única instância, onde os dados em memória são consistentes para ela
- Em um cenário com mais de uma instância, como um escalonamento horizontal, os dados passam a ser inconsistentes entre instâncias, podendo gerar retrabalho e redundância desnecessária
    - Isso se torna ainda mais crítico se a aplicação utilizar configurações importantes na cache, que não podem estar desatualizadas
- Além disso, cache em memória não é indicado em cenários onde esses dados precisam ser acessados por múltiplos sistemas

Caching em Memória

- Com o .NET é possível utilizar a interface IMemoryCache para a interação com a cache de aplicação;
- Basta configurar adequadamente na classe Program, com o uso do método AddMemoryCache, e já fica disponível para uso via injeção de dependência ao longo da aplicação;

Caching Distribuído

Caching Distribuído

- O Redis é uma estrutura de dados em memória, que pode ser utilizada para caching, banco de dados, entre outras formas
- Ele é reconhecido por sua alta performance, e contém algumas funcionalidades como:
  - Caching
  - Transações
  - Replicação
  - Mensageria
  - Definição de tempo de expiração para dados
- Em contraste ao cache em memória, o Redis é indicado para sistemas distribuídos, onde os dados podem ser acessados por múltiplos sistemas, incluindo múltiplas instâncias da mesma aplicação

Caching Distribuído

- É capaz de lidar com grandes quantidades de dados em tempo real
- Permite incrementos atômicos (INCR)
- Importante destacar capacidades importantes suas, como tolerância a falhas e replicação, se tornando uma ferramenta resiliente para sistemas distribuídos

DEMO

Concluindo