

Para saber mais: como o ChangeTracker funciona

Nossas classes não tem nenhuma lógica adicional para facilitar o monitoramento de mudanças em suas propriedades. Isto é, elas não herdam de nenhuma classe com esse propósito, e não possuem nenhuma propriedade que registra se foram modificadas (por exemplo, *IsDirty*). São classes simples que usam o padrão POCO (Plain Old C# Objects).

Então, como o ChangeTracker sabe que, quando uma propriedade foi alterada, ele deve fazer um `UPDATE` no banco?

O Entity guarda um *snapshot* dos valores dos objetos por padrão. Quando aquele objeto começa a ser monitorado pelo Entity, seja através de métodos que recuperam objetos do banco via `SELECT` (por exemplo `ToList`, `First`, `Find`, etc.), seja através do método `Entry` que cria uma entrada no ChangeTracker para o objeto passado como argumento do método.

E chama o método `DetectChanges` ao executar o `SaveChanges`. O que esse método faz? `DetectChanges` verifica diferenças entre os valores atuais das propriedades da entidade e os valores originais guardados no *snapshot* quando ela foi anexada ao contexto.

É possível desligar o monitoramento automático de mudanças através de uma propriedade *booleana* no ChangeTracker chamada `AutoDetectChangesEnabled`. Quando isso é necessário? Quando você tiver uma gravação massiva de objetos através do `SaveChanges`, a performance pode sofrer impacto considerável, uma vez que o método `DetectChanges` será chamado e o ChangeTracker irá percorrer toda a lista de objetos sendo monitorados.

Maiores informações neste artigo [aqui \(https://blog.oneunicorn.com/2016/11/16/notification-entities-in-ef-core-1-1/\)](https://blog.oneunicorn.com/2016/11/16/notification-entities-in-ef-core-1-1/) (em inglês).