**Change Tracking in Entity Framework Core**

EF Core verfolgt automatisch Änderungen, die an Entitäten vorgenommen werden. Dies geschieht durch den **DbContext**, der den Status jeder Entität überwacht.

**Tracking State ausgeben / abfragen**

Jede Entität hat einen Status, der vom DbContext überwacht wird. Die möglichen Zustände sind:

* **Added:** Die Entität wurde hinzugefügt und wird bei SaveChanges() eingefügt.
* **Modified:** Eine vorhandene Entität wurde geändert.
* **Deleted:** Die Entität wurde markiert, um aus der Datenbank gelöscht zu werden.
* **Unchanged:** Die Entität hat keine Änderungen erfahren.

**Überprüfen des Zustands einer Entität**

var product = new Product

{

ProductName = "Testprodukt",

UnitPrice = 9.99m,

CategoryId = category.CategoryId,

Discontinued = false

};

var entry = context.Entry(produkt);

Console.WriteLine($"Status von Product: {entry.State}");

**Change Tracking manuell deaktivieren**

Sie möchten z.B. eine Liste aller Produkte nur anzeigen (**READONLY**), ohne sie nachträglich zu bearbeiten? Dann brauchen Sie keine Änderungsverfolgung durch Entity Framework Core.

Das ergibt insbesondre viel Sinn, wenn wir große Mengen von Daten abfragen.

var products = context.Products.AsNoTracking().ToList();

**Fazit und Zusammenfassung**

* **Change Tracking** ermöglicht es, Änderungen an Entitäten zu **erkennen** und die Datenbank nur dann zu aktualisieren, wenn Änderungen vorgenommen wurden.
* EF Core **verfolgt den Zustand** der Entitäten **automatisch** und hilft dabei, nur die tatsächlich geänderten Entitäten zu speichern.
* Durch **AsNoTracking()** können wir die Leistung verbessern, wenn wir nur Daten abfragen und keine Änderungen verfolgen müssen.
* Entitäten, die mit **AsNoTracking()** aus DB abgerufen und danach im Code geändert wurden, werden **NICHT** in der DB committed.