

Online on-demand Project Support



# Entity Framework Core

## Aus der Praxis



Thorsten Kansy (tkansy@dotnetconsulting.eu)

# Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,  
Entwickler, Trainer & Fachautor



Azure Cosmos DB



# Mein Service- Ihr Benefit

- Individuelle Inhouse Trainings
- (Online on-demand) Projektbegleitung
- Beratung
  - Problemanalyse und Lösungen
  - Technologieentscheidungen



# Trigger-Warnung

Eigene Meinung aus “eigenen” (Kunden-)Projekten

Über dem Daumen? Mehr  als  ?

Nicht so



Image from [www.pexels.com](http://www.pexels.com)

Auch nicht so



Image from [www.pexels.com](http://www.pexels.com)

Dann lieber so



Image from [www.pexels.com](http://www.pexels.com)

Eine Datenbank, ein DbContext?

# Eine Datenbank, ein DbContext



DI sorgt für überschaubare Klassen/ Projekte/ Libraries und nun aber ein DbContext für die gesamte Datenbank?

Übersichtlichkeit  
Refactoring?

Self-contained Type configuration!

# Self-contained Type Configuration

## Trennung von Entitäten-Konfiguration und Kontext

```
public class SpeakerEFConfiguration : IEntityConfiguration<Speaker>
{
    public void Configure(EntityTypeBuilder<Speaker> builder)
    {
        builder.Property(s => s.Infos)
            .IsRequired()
            .HasDefaultValue("(Keine Infos)");

        // ...
    }
}

// In OnModelCreating()
modelBuilder.ApplyConfiguration(new SpeakerEFConfiguration());
```

Eine Tabelle, eine Entität?

# Eine Tabelle, eine Entität



Eigentlich auch keine gute Idee, aber wenn sich dies wirklich negativ auswirkt, ist das Datenbank-Design eher das Problem

Migration?

# Migration



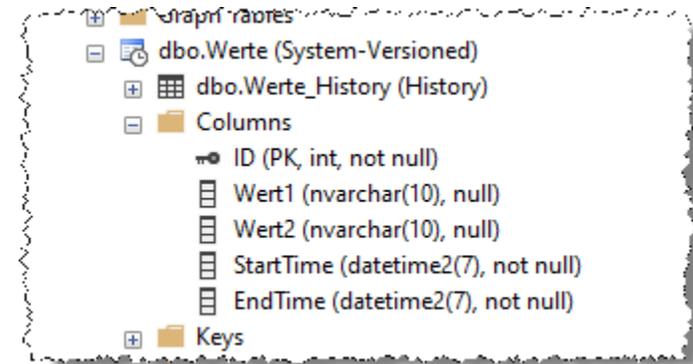
- Datenbank ist viel mehr als nur Tabellen, Sichten, Prozeduren, ...
- Berechtigungen?
- Änderungen von Datentypen/ Referenzen?
- Ohne Migration kann das Model einfacher werden
  
- Aber: Migrations bundles (EF Core 6.0)
  
- Alternativen?
  - DB Schema/ Data Compare
  - DACPAC & BACPACs (SQL Server)

# Temporal Tables?

# Temporal Tables (SQL Server)



- Automatisierte Historisierung
- Handling liegt zentral auf dem Server
- Zeitliche Abfragen erst mit EF 6.0



Sichten/ Prozeduren/ AdHoc?

# Sichten/ Prozeduren/ AdHoc



EF Core als O/R-Mapper ist das primäre Zugriffsmittel,  
aber nicht das Einzige!

Wenn LINQ to Entities zu umständlich/ komplex/ langsam ist

Wenn nur ein DB-Server Typ

Database-Interface und...

# ToSqlQuery()

```
C# Copy  
  
protected override void OnModelCreating(ModelBuilder modelBuilder)  
{  
    modelBuilder.Entity<Post>().ToSqlQuery(  
        @"SELECT Id, Name, Category, BlogId FROM posts  
        UNION ALL  
        SELECT Id, Name, ""Legacy"", BlogId from legacy_posts");  
  
    modelBuilder.Entity<Blog>().ToFunction("BlogsReturningFunction");  
}
```

AdHocs als Embedded Resource!?

# FromSqlRaw

C#

Copy

```
var blogs = context.Blogs
    .FromSqlRaw("SELECT * FROM dbo.Blogs")
    .ToList();
```

Raw SQL queries can be used to execute a stored procedure.

C#

Copy

```
var blogs = context.Blogs
    .FromSqlRaw("EXECUTE dbo.GetMostPopularBlogs")
    .ToList();
```

Global Query Filter?

# Global Query Filter



“Entfernen” logisch gelöschter Daten

# HasQueryFilter (Fluent API)

- Globale Einschränkung für alle Abfragen
  - Kann ignoriert werden (`IgnoreQueryFilters()`)
- Standard: –

```
modelBuilder.Entity<Session>()  
    .HasQueryFilter(p => p.IsDeleted == false);
```

```
SELECT TOP(1) [s].[Id], [s].[ContentDescription], [s].[Begin], [s].[Created], [s].[Difficulty], [s].[Duration], [s]  
].[End], [s].[EventId], [s].[IsDeleted], [s].[SpeakerId], [s].[TechEventId], [s].[Title], [s].[Updated]  
FROM [dnc].[Sessions] AS [s]  
WHERE ([s].[IsDeleted] = 0) AND ([s].[Id] = @_get_Item_0)
```



(Automatic) Lazy Loading?

# (Automatic) Lazy Loading



Häppchenweise Daten laden => unzählige Datenbankzugriffe

N+1 Problem!

Zum Glück kein Default mehr im EF Core

Eager Loading?

# Eager Loading



Genau steuern, wann welche Daten aus der Datenbank geladen werden

## Includes

- Include
- Filtered Include (EF Core 5.0)
- Split Query (EF Core 5.0)
- Etc.

# Filtered Includes

C#

Copy

```
using (var context = new BloggingContext())
{
    var filteredBlogs = context.Blogs
        .Include(
            blog => blog.Posts
                .Where(post => post.BlogId == 1)
                .OrderByDescending(post => post.Title)
                .Take(5))
        .ToList();
}
```

As(No)Tracking?

# As(No)Tracking



Ressourcen für die Erkennung potentieller Datenänderungen, die nicht kommen, sparen

`AsNoTracking()`

`AsTracking()`

`AsNoTrackingWithIdentityResolution()`

ToQueryString()?

# ToQueryString()



Abfragetext ermitteln (ab 5.0)

```
C# Copy  
  
Console.WriteLine(  
    ctx.Artists  
        .Where(a => a.Name == "Pink Floyd")  
        .ToQueryString());
```

Codeänderung nötig

Besser für SQL Server Entwickler: SQL Server Profiler

EF Core & Cosmos DB?

# EF Core & Cosmos DB



EF Core ist klasse

Cosmos DB ist klasse

Nur passen die flexible NoSQL Datenbank mit den starren Entitäten nicht so recht zusammen.

# Fragen?

# Links



[www.dotnetconsulting.eu](http://www.dotnetconsulting.eu)



[@Tkansy](https://twitter.com/Tkansy)



[tkansy@dotnetconsulting.eu](mailto:tkansy@dotnetconsulting.eu)

Images from [www.pexels.com](http://www.pexels.com)