



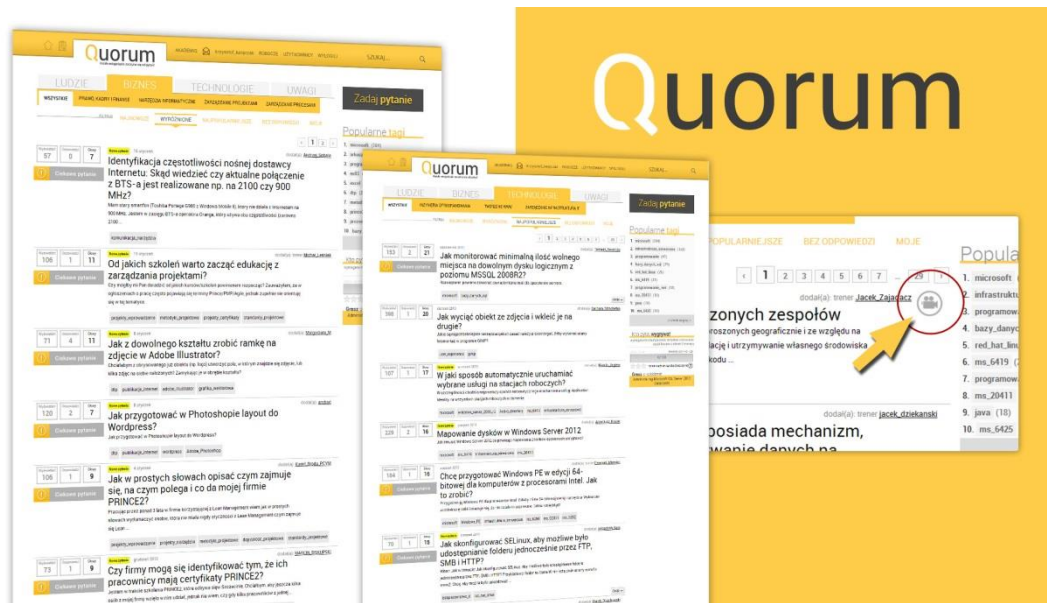
Entity Framework 6.1

marcin.sulecki@altkom.pl

Infolinia: 0 801 258 566
www.altkomakademia.pl



Każde osiągnięcie zaczyna się od pytań!



<https://quorum.akademiiq.pl>



Agenda:

- Wprowadzenie
- Instalacja Entity Framework
- Model First, Database First vs Code First
- Code First
- Inicjalizatory
- Operacje CRUD
- Monitoring
- Migracje
- Konwencje
- Linq
- Audyt
- Wywołanie SQL i procedur składowanych
- Transakcje
- Konkurencyjność
- Operacje asynchroniczne
- Optymalizacja
- Metadane
- Słów kilka o EF Core 1.0



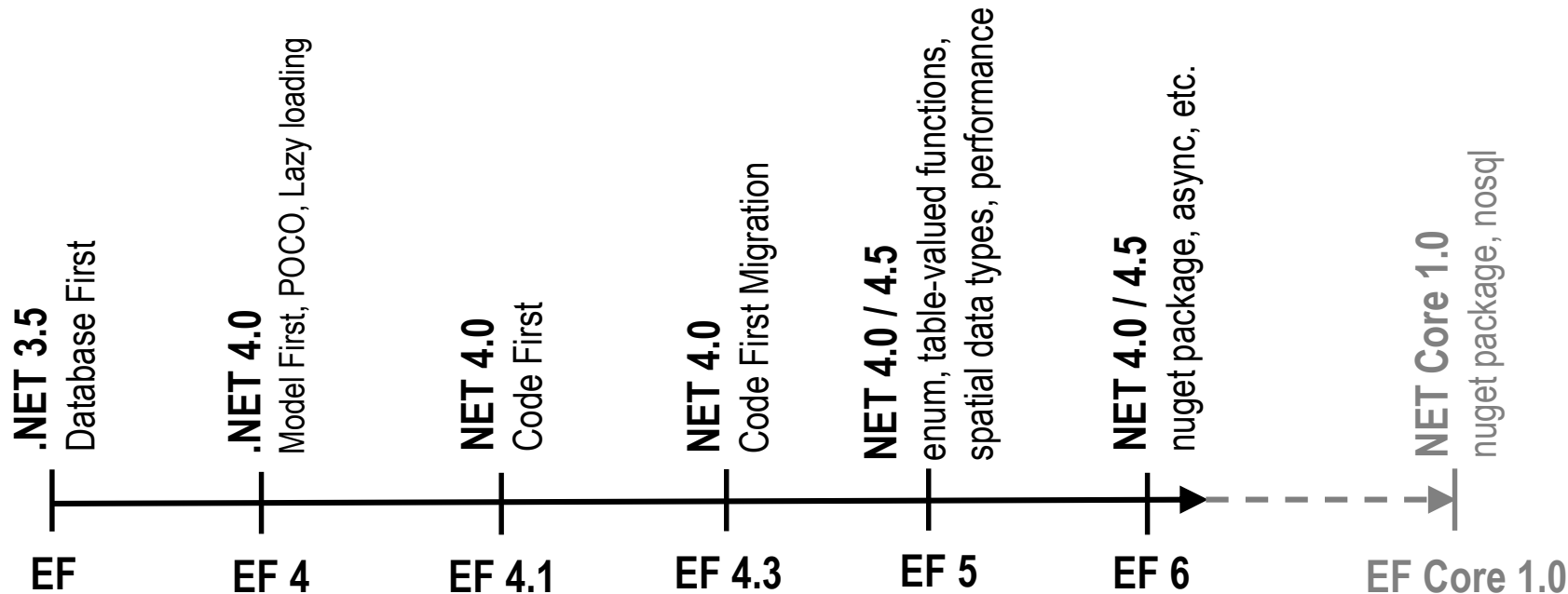
Wprowadzenie

- Entity Framework jako ORM
- Zasada działania:





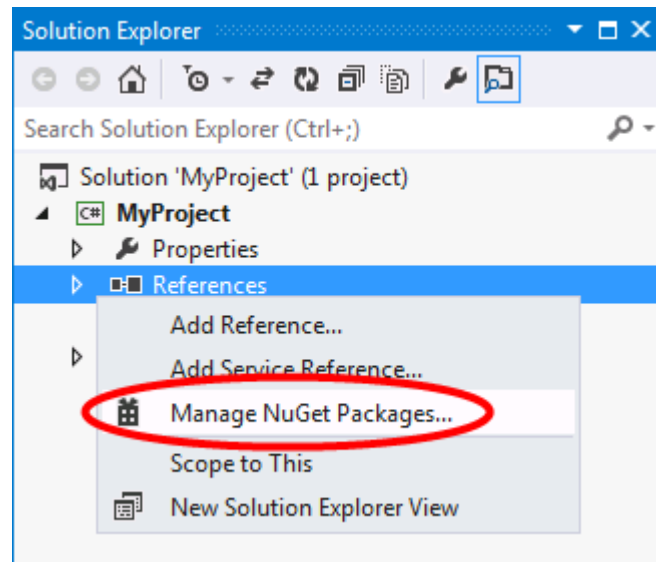
Historia





Instalacja Entity Framework

- Instalacja NuGet Package
- Instalacja z Package Manager Console
 - Install-package EntityFramework





DEMO

Instalacja Entity Framework



Podejścia

Database First

Model First

Code First



Choose Model Contents

What should the model contain?



EF Designer
from database



Empty EF
Designer model



Empty Code
First model



Code First from
database

Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.

< Previous

Next >

Finish

Cancel



Code First

- Mapowanie klas POCO na tabele w bazie danych

```
public class MusicStoreDbContext : DbContext
{
    public DbSet<Album> Albums { get; set; }
}

public class Album
{
    public string Title { get; set; }
    public decimal Cost { get; set; }
}
```

DbContext

POCO (Entity)



DEMO

Code First



Jak się łączy EF do bazy danych?

Connect string named
in DbContext class



Read connect string
Open Connection

Use context class
name to look for
connect string in
app.config



Read connect string
Open Connection

No connect string.
Start database checks



Look for
SQL Express



Look for
(localdb)\11



Look for
(localdb)\mssqllocaldb



Jak się łączy EF do bazy danych?

```
public MusicContext() : base("MusicStoreConnection")  
{  
}
```



Operacje CRUD

- Utworzenie
- Odczyt
- Modyfikacja
- Usunięcie



DEMO

Operacje CRUD



Inicjalizatory bazy danych

- **CreateDatabaseIfNotExists**
tworzy nową bazę danych jeśli nie istnieje.
Jeśli zmieni się model klas wówczas generuje wyjątek.
- **DropCreateDatabaseIfModelChanges**
usuwa i tworzy nową bazę danych jeśli zmieni się model klas.
- **DropCreateDatabaseAlways**
usuwa i tworzy nową bazę danych za każdym razem
- **Własny inicjalizator**



DEMO

Inicjatory



Monitoring

- **Informacje o połączeniu**

`Database.Connection.ConnectionString`

- **Śledzenie zapytań**

`context.Database.Log = s => Console.WriteLine(s)`



DEMO

Monitoring



Migracje

- **Włączenie migracji**
Enable-Migrations
- **Generowanie migracji**
Add-Migration
- **Aktualizacja bazy danych**
Update-Database



Migracje

- **Aktualizacja bazy danych do określonej wersji**
Update-Database –TargetMigration:
- **Generowanie skryptu migracji**
Update-Database –Script
- **Aktualizacja bazy danych podczas uruchomienia aplikacji**
`Database.SetInitializer(new MigrateDatabaseToLatestVersion<Context, Configuration>());`



DEMO

Migracje



Konfigurowanie modelu

- **Data Annotations**
Atrybuty właściwości i klas
- **Fluent API**
Za pomocą wyrażeń lambda



Data Annotations vs Fluent API

- **Dlaczego Data Annotations?**
 - Proste, idealne w prostych scenariuszach
 - Atrybuty są przy właściwościach
- **Fluent API**
 - Klasy modelu są czyste, konfiguracja jest w osobnym pliku
 - Więcej możliwości
 - Czytelność



DEMO

Data Annotations



DEMO

Fluent API



Konwencje

- **Standardowe konwencje**

- Type
- PrimaryKey
- RelationShip
- ComplexType
- ConnectionString

- **Usuwanie konwencji**

`modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();`



DEMO

Usuwanie standardowej konwencji



Własne konwencje

- **Tworzenie konwencji**

```
class MyConvention : Convention
```

- **Dodawanie konwencji**

```
modelBuilder.Conventions.Add(new MyConvention());
```



DEMO

Tworzenie własnych konwencji




Mapowanie dziedziczonych typów na tabele

- **TPH (Table per hierarchy)**
domyślne
- **TPT (Table per type)**
- **TPC (Table per concrete type)**

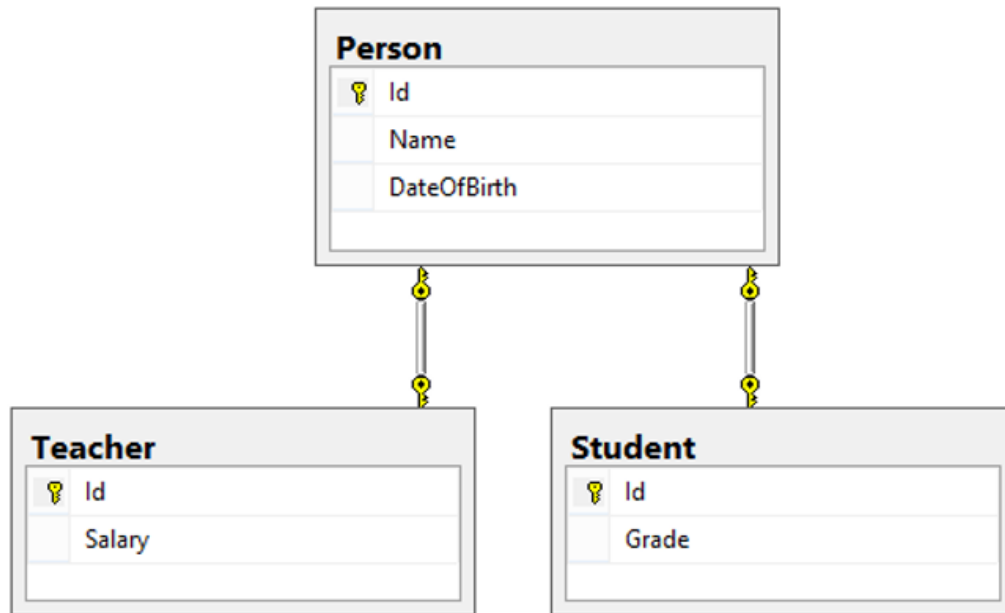


TPH (Table per hierarchy)

Person	
	Id
	Name
	DateOfBirth
	Grade
	Salary
	Discriminator





TPT (Table per type)





TPC (Table per concrete type)

Teachers	
	Id
	Salary
	Name
	DateOfBirth

Students	
	Id
	Grade
	Name
	DateOfBirth



Linq

- Select
 - Where
 - OrderBy
 - GroupBy
 - ToList
- First, FirstOrDefault
 - Single, SingleOrDefault
 - Count, Sum, Average
 - Union, Except
 - Any, All



DEMO

Linq



Attach

- **Przyłączenie encji do kontekstu**
`context.Albums.Attach`
- **Zmiana statusu**
`context.Entry<Album>(album).State = EntityState.Modified`
- **Biblioteka GraphDiff**
`Install-package RefactorThis.GraphDiff`



DEMO

Attach



Audyt

- **Sprawdzenie zmian**
`context.ChangeTracker.HasChanges()`
- **Pobranie stanu encji**
`context.ChangeTracker.Entries()`



DEMO

Audyt



Uruchamianie SQL

- **Wykonanie zapytania SQL**
`context.Database.ExecuteSqlCommand()`
- **Pobranie encji na podstawie zapytania SQL**
`context.Database.SqlQuery<T>()`



DEMO

Uruchamianie SQL



Transakcje

- **Rozpoczęcie transakcji**
`context.Database.BeginTransaction()`
- **Zatwierdzenie transakcji**
`transaction.Commit()`
- **Wycofanie transakcji**
`context.Database.Rollback()`



DEMO

Transakcje



Konkurencyjność

- **Konfiguracja bez pola Timestamp**
Atrybut `ConcurrencyCheck` lub `IsConcurrencyToken()`
- **Konfiguracja z polem Timestamp**
Atrybut `Timestamp` lub `IsConcurrencyToken()`
- **Przechwytywanie informacji o zmianie**
`DbUpdateConcurrencyException`



DEMO

Konkurencyjność



Mapowanie procedur składowanych

- Mapowanie procedur
MapToStoredProcedures



DEMO

Mapowanie procedur składowanych



Operacje asynchroniczne

- **Wyszukiwanie asynchronicznie**
`FindAsync()`
- **Pobieranie listy asynchronicznie**
`ToListAsync()`
- **Zapisywanie zmian asynchronicznie**
`context.SaveChangesAsync()`



DEMO

Operacje asynchroniczne



Ładowanie powiązanych encji

- **Eagerly Loading**

Include()

- **Lazy Loading**

Configuration.LazyLoadingEnabled = true

Configuration.ProxyCreationEnabled = true

- Navigation Property musi być zdefiniowane jako publiczne i wirtualne



DEMO

Ładowanie powiązanych encji



Optymalizacja

- **Wyłączenie śledzenia**
`AsNoTracking()`
- **Generowanie widoków za pomocą Power Tools**
- **Wyłączenie automatycznego wykrywania zmian**
`context.Configuration.AutoDetectChangesEnabled = false`
- **Cache'owanie**
`Install-Package EntityFramework.Cache`



DEMO

Optymalizacja



Metadane

- **Pobranie informacji o modelu**

`workspace.GetItems<EntityType>(DataSpace.SSpace)`



DEMO

Metadane



Spatial Types

- **DbGeometry**
- **DbGeography**

DbGeography.FromText("POINT(54.22 18.40)")



DEMO

Spatial Types



Zasoby

- **Dokumentacja**

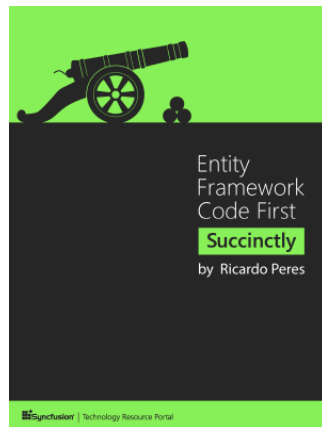
<https://msdn.microsoft.com/en-us/data/ef.aspx>

- **EF Team Blog**

<http://blogs.msdn.com/b/adonet/>



E-books





Dziękuję

Infolinia: 0 801 258 566
www.altkomakademia.pl

marcin.sulecki@altkom.pl