Mapping Special Scenarios

Torben Jensen
Developer/Cloud Architect

Overview

Foreign keys using shadow properties

Direct many-to-many relationships using property bags

Fine-grained control of mappings using value conversions

Mapping value types using owned entities

Shadow Properties

Map to data in the database

Not defined in entity classes

Maintained by EF Core’s change tracker

Hides data from mapped entities

```
public class Order
{
  public int Id { get; set; }
  public Customer Customer { get; set; }
}
```

◀ Navigation property without corresponding id creates shadow property

```
modelBuilder.Entity<Customer>(entity =>
{
  entity.HasKey(e => e.Id);
})
.Entity<Customer>()
.ToTable(e => e.IsTemporal());
```

- Temporal tables contain a PeriodStart and a PeriodEnd column, which are mapped as shadow properties

Temporal Tables

Querying Data in EF Core 6

Torben Boeck Jensen

Demo

Mapping shadow properties

Configuring Properties

Existing property in Item class

```
entity.Property(e => e.PriceAfterVat)
 .HasComputedColumnSql("[UnitPrice]*1.10");
```

Shadow property for Customer class

```
entity.Property<byte[]>("Checksum")
 .HasComputedColumnSql(
   "CONVERT(VARBINARY(1024),
     CHECKSUM([FirstName],
     [LastName],
     [UserName]))");
```

Updating Shadow Properties

EF Core 6 Fundamentals

Julie Lerman

What Did We Just Do?

Shadow properties

Set up in database context

Accessed using EF Core's change tracker

Indexer Properties

Are used to implement property bags

- Used under the covers by EF Core
- Facilitate direct many-to-many relationships
- We can implement our own property bags

Indexer Properties

Backed by indexer

Add properties to existing entities

Allows class instances to be indexed

Like array[1]

```
public class Customer
{
  /*
    ... MORE CODE ...
  */

  Dictionary<string, object> _data =
    new Dictionary<string, object>();

  public object this[string key]
  {
    get => _data[key];
    set => _data[key] = value;
  }

 /*
    ... MORE CODE ...
  */
}

e.IndexerProperty<DateTime>("LastUpdated");
e.IndexerProperty<string>("ConsumerType");
```

◀ Dictionary for containing indexer properties

◀ Indexer

◀ Set up indexer property in OnModelCreating

```
modelBuilder
  .SharedTypeEntity<
    Dictionary<string, object>>
  ("Tag",
  entity =>
  {
    entity.Property<int>("Id");
    entity.Property<int>("CustomerId");
    entity.Property<string>("Nickname");
    entity.Property<int>("DiscountRate");
  });

DbSet<Dictionary<string, object>> Tags =>
  Set<Dictionary<string, object>>("Tag");
```

- Used for direct many-to-many relationship
- Set up as a shared type entity
- Name of shared type entity
- Property bags only consist of indexer properties
- Defined as a collection on our data context using a call to the Set-method

Demo

Mapping indexer properties

Demo

Mapping property bags

What Did We Just Do?

Indexer properties

Add more properties to our entities

Property bag entities

Owned Entity Types

Navigation properties on other entity types

Only exist in the context of their owner

Can have many owners

Useful for Domain-Driven Design

Demo

Mapping Owned entity types

What Did We Just Do?

Reduce clutter in entity classes

Increases clutter in database tables

Useful depending on how we interact with the database

Value Conversions

Convert from database type to another type on entity classes

Conversion is actually between model type and provider type

Convert from one value to another of the same type

Demo

Mapping enums

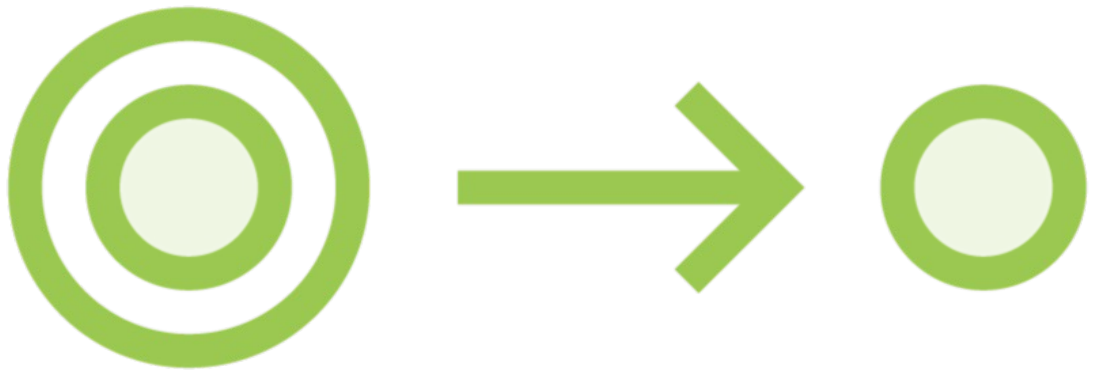
Value converter

Demo

Custom conversions

Pre-convention model configuration

What Did We Just Do?



Value conversions

Predefined conversions

Bulk conversions

Summary

Special scenarios

Shadow properties

Indexer properties

Shadow properties are used behind the scenes

Owned entity types

Value conversions

Remember maintenance

Further Reading

Indexers, https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/indexers/

Shared-type entity types, https://docs.microsoft.com/en-us/ef/core/modeling/entity-types?tabs=data-annotations#shared-type-entity-types