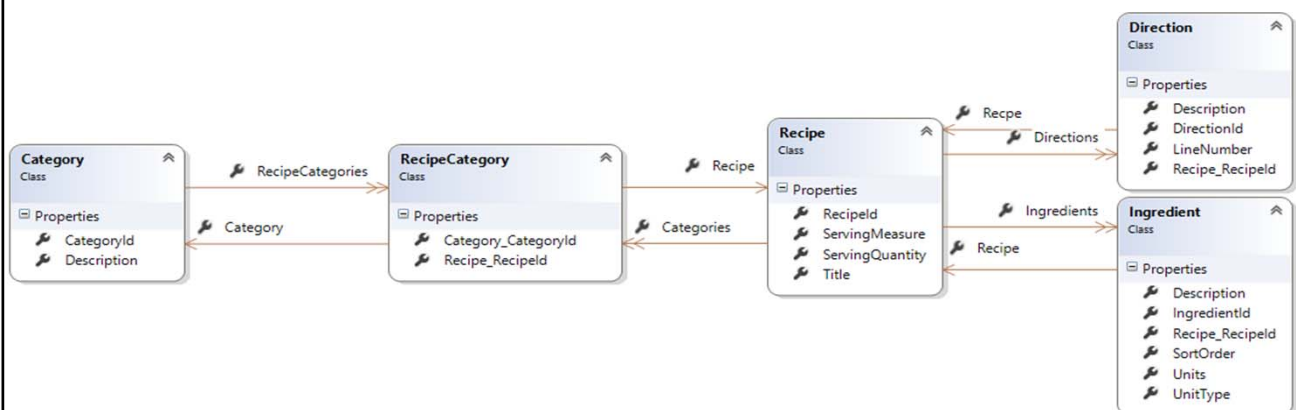


Getting started with Entity Framework Core

Jim Wooley
Solution Architect,
Slalom

Demo Data Model



Getting started

DEMO



Configuration

```
<ItemGroup>
  <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer.Design"
    Version="2.0.0-preview1-final" />
  <PackageReference Include="Microsoft.EntityFrameworkCore.Design"
    Version="2.0.0-preview1-final" />
</ItemGroup>
<ItemGroup>
  <DotNetCliToolReference Include="Microsoft.EntityFrameworkCore.Tools.DotNet"
    Version="2.0.0-preview1-final" />
</ItemGroup>
```



Construction

```
public class RecipeContext : DbContext
{
    public DbSet<Recipe> Recipes { get; set; }
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder
            .UseSqlServer(@"Server=.;Database=Recipe;Trusted_Connection=True;");
    }
}
// ASP.Net
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<RecipeContext>(options =>
        options.UseSqlServer(Configuration.GetConnectionString("RecipeDb")));
    services.AddDbContextPool<RecipeContext>(options =>
        options.UseSqlServer(ConnectionString)); //(EF 2)
}
```



Migrations

- Add-Migration {name}
- Remove-Migration {name}
- Update-Database
- Script-Migration
- dotnet ef migrations add {name}
- dotnet ef migrations remove
- dotnet ef database update
- dotnet ef database script



Scaffolding

```
>dotnet ef dbcontext scaffold
    "data source=.;initial catalog=NWind;integrated security=true;"
    Microsoft.EntityFrameworkCore.SqlServer
    -o Models
    -c NorthwindContext
    -a
    -startup-project ..\MigrationsConsole
```

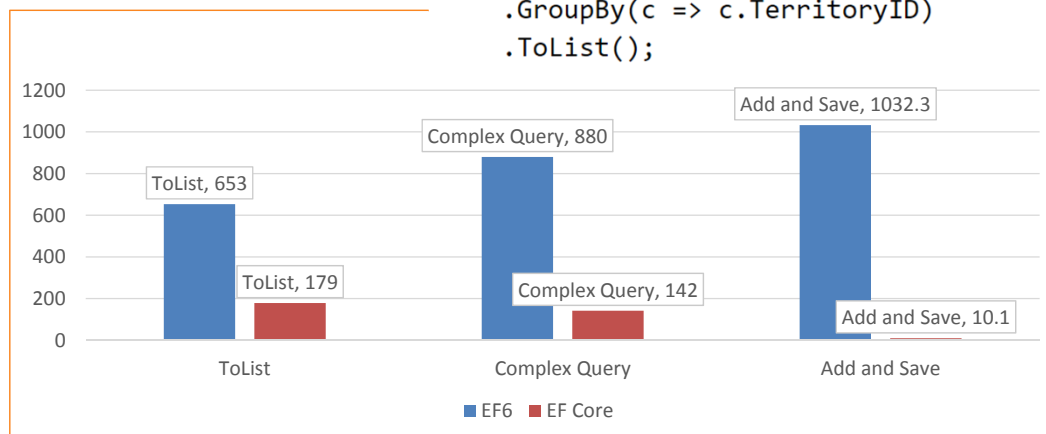
```
>Scaffold-DbContext
-Connection "data source=.;initial catalog=NWind;integrated security=true;"
-Provider Microsoft.EntityFrameworkCore.SqlServer
-OutputDir Models
-Context NorthwindContext
-DataAnnotations
```

<https://ef.readthedocs.io/en/latest/miscellaneous/cli/powershell.html>



Performance

```
db.Customers
    .Where(c => !c.AccountNumber.EndsWith("1"))
    .OrderBy(c => c.AccountNumber)
    .ThenBy(c => c.ModifiedDate)
    .Skip(100)
    .GroupBy(c => c.TerritoryID)
    .ToList();
```



DEMO

Performance



Shadow Properties

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Blog>()
        .Property<DateTime>("LastUpdated");
}

context.Entry(myBlog).Property("LastUpdated").CurrentValue = DateTime.Now;
```

<https://docs.efproject.net/en/latest/modeling/shadow-properties.html>



Providers

- Microsoft SQL Server
- SQLite
- Npgsql (PostgreSQL)
- MySQL – Official, Pomelo, Sapient Guardian
- Microsoft SQL Server Compact Edition
- IBM Data Servers
- InMemory (for Testing)
- Devart (MySQL, Oracle, PostgreSQL, SQLite, DB2, SQL Server, and more)
- Oracle (~~Coming Soon~~) (not yet available)

<https://docs.efproject.net/en/latest/providers/>



In Memory Provider

```
public RecipeContext(DbContextOptions<RecipeContext> options)
    : base(options) { }
```

```
private static DbContextOptions<RecipeContext> CreateNewContextOptions()
{
    var serviceProvider = new ServiceCollection()
        .AddEntityFrameworkInMemoryDatabase()
        .BuildServiceProvider();
    var builder = new DbContextOptionsBuilder<RecipeContext>();
    builder.UseInMemoryDatabase();
    builder.UseInternalServiceProvider(serviceProvider);
    return builder.Options;
}
```

<https://docs.efproject.net/en/latest/miscellaneous/testing.html>



EF Core 2

- EF.Functions.Like("%value%")
- Owned entities and table splitting
- Global Query Filters
- DbContext Pooling
- Manually Compiled Queries
- String Interpolation in FromSql
- Database Scalar Functions

<https://github.com/aspnet/EntityFramework/wiki/Roadmap#ef-core-20>



Core 2.0 - EF.Functions

```
var brownies =  
    from recipe in dc.Recipes  
    where EF.Functions.Like(recipe.Title, "%brownie%")  
    select recipe.Title;
```



EF Core 2.0 – Owned Entities

```
modelBuilder.Entity<Customer>().OwnsOne(c => c.WorkAddress);  
public class Customer  
{  
    public int CustomerId { get; set; }  
    public Address WorkAddress { get; set; }  
}  
public class Address  
{  
    public string Line { get; set; }  
    public string PostalOrZipCode { get; set; }  
    public string StateOrProvince { get; set; }  
    public string CityOrTown { get; internal set; }  
}
```



EF 2.0 - Global Query Filters

```
modelBuilder.Entity<Recipe>().HasQueryFilter(r => !r.IsDeleted);
```



Comparison

EF6

- EDMX
- Many-many
- Lazy Loading
- TPT/TPC
- Complex/Value types
- Spatial
- Complex LINQ
- Stored procs
- GroupBy

EF Core

- Scaffolding
- Migrations
- Batch updates
- Providers
- Dotnet Core/UWP
- Shadow Properties
- Field mapping
- Performance Optimizations

<https://docs.efproject.net/en/latest/efcore-vs-ef6/index.html>

Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Links

<http://docs.efproject.net>

<http://github.com/aspnet/EntityFramework>

<https://ef.readthedocs.io/en/latest/>

EF Core Roadmap - <http://bit.ly/efcoreroadmap>

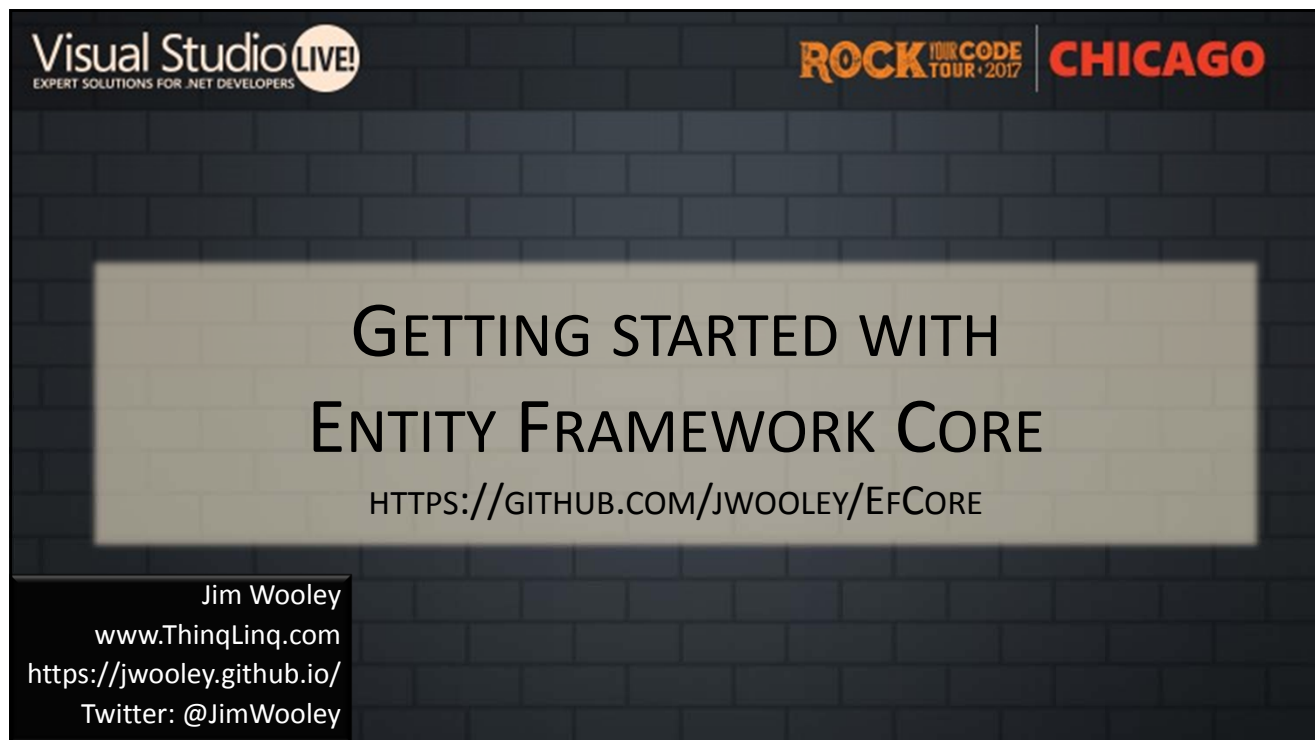
<https://github.com/rowanmiller/Demo-EFCore>

<https://blogs.msdn.microsoft.com/dotnet/2016/10/25/announcing-entity-framework-core-1-1-preview-1/>

<https://blogs.msdn.microsoft.com/dotnet/2017/05/12/announcing-ef-core-2-0-preview-1/>

<https://blogs.msdn.microsoft.com/dotnet/2017/06/28/announcing-ef-core-2-0-preview-2/>

Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS



The slide features a dark blue brick wall background. At the top left is the 'Visual Studio LIVE!' logo with the tagline 'EXPERT SOLUTIONS FOR .NET DEVELOPERS'. At the top right are the 'ROCK YOUR CODE TOUR 2017' and 'CHICAGO' logos. The main title 'GETTING STARTED WITH ENTITY FRAMEWORK CORE' is centered in a light gray rectangular box. Below the title is the GitHub repository URL. In the bottom left corner, a black box contains the speaker's name and contact information.

Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

ROCK YOUR CODE
TOUR 2017

CHICAGO

GETTING STARTED WITH ENTITY FRAMEWORK CORE

[HTTPS://GITHUB.COM/JWOOLEY/EFCore](https://github.com/jwooley/efcore)

Jim Wooley
www.ThinqLinq.com
<https://jwooley.github.io/>
Twitter: @JimWooley