EF Core:

1. Reference:
2. Documents: <https://docs.efproject.net/en/latest/>
3. Code First example: <https://www.youtube.com/watch?v=gvYis93MU6s>
4. With VS code: <https://www.youtube.com/watch?v=TZF46oPBHPc>
5. CRUD: <https://www.youtube.com/watch?v=K4x6eoG7hwY>
6. Repository pattern: <http://www.franksworld.com/2016/10/16/repository-pattern-with-c-and-entity-framework/>
7. MSDN Magazine data points: <https://social.msdn.microsoft.com/Search/en-US/magazine?query=data%20points%20ado.net&pgArea=header&Refinement=118&emptyWatermark=true&ac=4>
8. Sqlite on windows ef core <https://docs.efproject.net/en/latest/providers/sqlite/>
9. Install Db browser for Sqlite on mac <http://sqlitebrowser.org/>
10. Steps:
11. Modify project.json:
    1. In “dependencies” section:

### "Microsoft.EntityFrameworkCore.Sqlite": "1.0.0",

### "Microsoft.EntityFrameworkCore.Design": {

### "version": "1.0.0-preview2-final",

### "type": "build"

### }

* 1. “tools”:

"tools": {

"Microsoft.EntityFrameworkCore.Tools": "1.0.0-preview2-final"

}

1. install new package: dotnet restore (mac) nugget restore (win)
2. verify EF already installed: dotnet ef –help
3. create models by adding new class:

**public** **class** **BloggingContext** : DbContext

{

**public** DbSet<Blog> Blogs { **get**; **set**; }

**public** DbSet<Post> Posts { **get**; **set**; }

**protected** **override** **void** **OnConfiguring**(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlite("Filename=./blog.db");

}

}

1. compile the project: dotnet build
2. generate migration: dotnet ef migrations add MyFirstMigration or Add-Migration MyFirstMigration (package manager console)
3. create database files: dotnet ef database update or Update-Database (package manager console)
4. now, use it to add and retrieve db:

**public** **static** **void** **Main**()

{

**using** (**var** db = **new** BloggingContext())

{

db.Blogs.Add(**new** Blog { Url = "http://blogs.msdn.com/adonet" });

**var** count = db.SaveChanges();

Console.WriteLine("{0} records saved to database", count);

Console.WriteLine();

Console.WriteLine("All blogs in database:");

**foreach** (**var** blog **in** db.Blogs)

{

Console.WriteLine(" - {0}", blog.Url);

}

}

}

1. start app: dotnet run
2. with aspnet, register DI (dependency injection) in startup.cs

**using** EFGetStarted.AspNetCore.NewDb.Models;

**using** Microsoft.EntityFrameworkCore;

**public** **void** **ConfigureServices**(IServiceCollection services)

{

**var** connection = @"Server=(localdb)\mssqllocaldb;Database=EFGetStarted.AspNetCore.NewDb;Trusted\_Connection=True;";

services.AddDbContext<BloggingContext>(options => options.UseSqlServer(connection));

1. We now add a controller by select a template

**private** BloggingContext \_context;

**public** **BlogsController**(BloggingContext context)

{

\_context = context;

}

(can use yo with sub generator)

1. And add a view:
2. If we already have a database, here it is:

Scaffold-DbContext "Server=(localdb)\mssqllocaldb;Database=Blogging;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

1. Create a model:
   1. By fluent api

**class** **MyContext** : DbContext

{

**public** DbSet<Blog> Blogs { **get**; **set**; }

**protected** **override** **void** **OnModelCreating**(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Blog>()

.Property(b => b.Url)

.IsRequired();

}

}

* 1. By data annotation

**public** **class** **Blog**

{

**public** **int** BlogId { **get**; **set**; }

[Required]

**public** **string** Url { **get**; **set**; }

}

* 1. Example:

**class** **MyContext** : DbContext

{

**public** DbSet<Blog> Blogs { **get**; **set**; }

**protected** **override** **void** **OnModelCreating**(ModelBuilder modelBuilder)

{

modelBuilder.Entity<AuditEntry>();

}

}

**public** **class** **Blog**

{

**public** **int** BlogId { **get**; **set**; }

**public** **string** Url { **get**; **set**; }

**public** List<Post> Posts { **get**; **set**; }

}

**public** **class** **Post**

{

**public** **int** PostId { **get**; **set**; }

**public** **string** Title { **get**; **set**; }

**public** **string** Content { **get**; **set**; }

**public** Blog Blog { **get**; **set**; }

}

**public** **class** **AuditEntry**

{

**public** **int** AuditEntryId { **get**; **set**; }

**public** **string** Username { **get**; **set**; }

**public** **string** Action { **get**; **set**; }

}

* 1. Relationship: