**CodeFirst 连数据库的步骤**

Nuget 引用Microsoft.EntityFrameworkCore

如果数据库是sqlserver,引用Microsoft.EntityFrameworkCore.SqlServer

如果数据库是 mysql, 引用 Pomelo.EntityFrameworkCore.Mysql

如果数据库是 oracle,引用

建立实体类和 dbcontext

引用 Microsoft.VisualStudio.Web.CodeGeneration.Design 或者Microsoft.EntityFrameworkCore.Tools.DotNet

在程序包管理控制台上运行 Add-Migration Initial ,Update-Database 将实体映射到数据库的表中

建立控制器，自动生成代码框架

**Ef core 使用 mysql**

建立 aspnet core 项目， 可以是mvc模板

nuget 安装 Microsoft.EntityFrameeworkCore.Tools Pomelo.EntityFrameworkCore.Mysql

打开项目文件，确定 项目文件的内容如下

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>netcoreapp2.0</TargetFramework>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore.All" Version="2.0.0" />

<PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="2.0.0" />

<PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Version="2.0.0" />

</ItemGroup>

<ItemGroup>

<DotNetCliToolReference Include="Microsoft.EntityFrameworkCore.Tools.DotNet" Version="2.0.0" />

</ItemGroup>

</Project> **确保 各依赖的版本 一致**

建立实体类和 DbContext

在 startup.cs中 添加代码 使用 mysql

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc();

//这里就是填写数据库的连接字符串

var connection = "Server=localhost;Port=3306;Database=TestMySql; User=root;Password=;";

services.AddDbContextPool<DataContext>(options => options.UseMySql(connection));

}

在 startup.cs 所在目录 打开 cmd

输入 dotnet ef Migrations add Init

dotnet ef database update

此时会将实体类的结构 同步到数据库

当实体类修改后，可以通过

dotnet ef Migrations add newname

dotnet ef database update

更新到数据库

通过改变字符串和 UseMySql 切换为UseSqlServer的切换，可以切换到 sqlserver