.NET Core API for ORM(Entity)

C# (DB) API

API , .

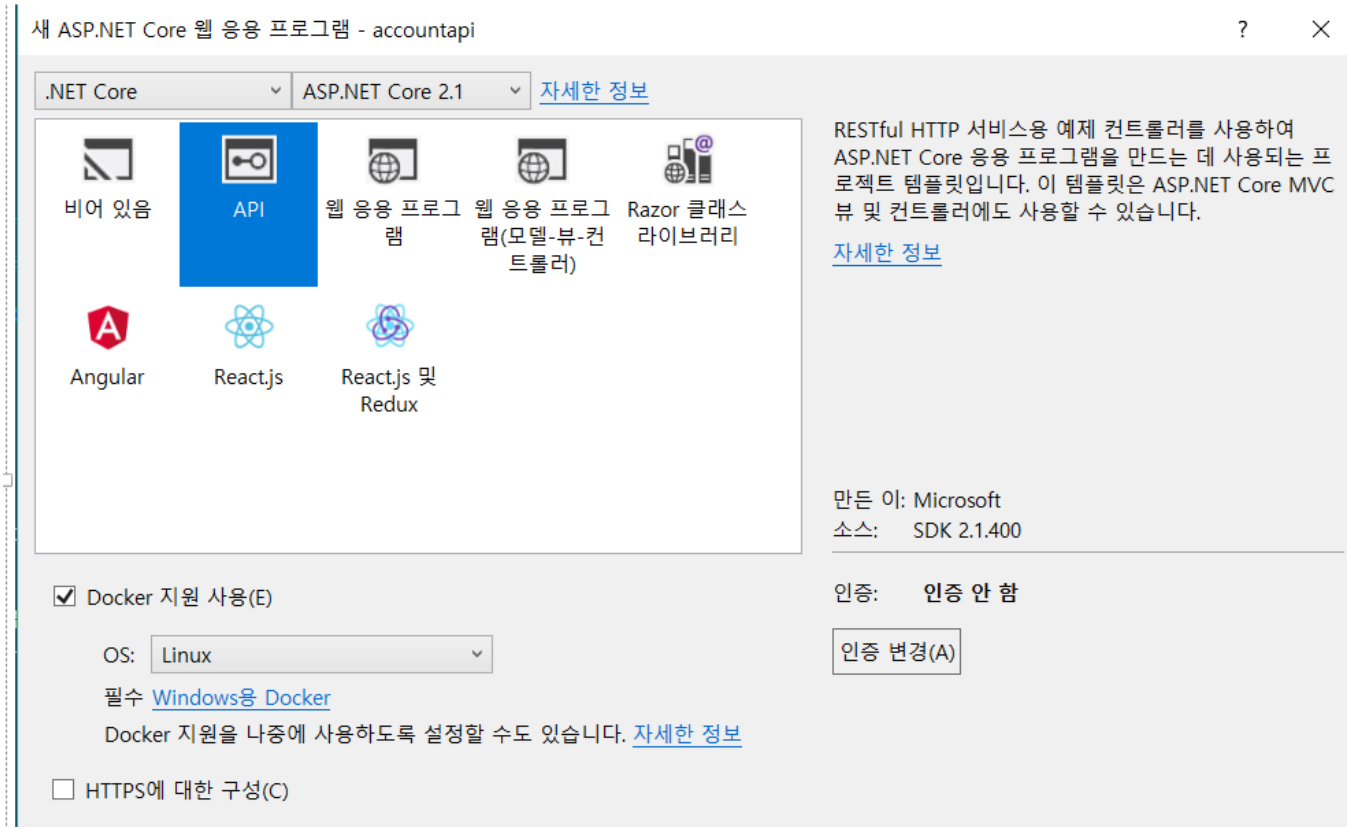
- Docker ( )
- ORM (Mysql + DB)

.net core api .

git : https://github.com/psmon/netcore-restweb-entity-akka

IDE

- Visual Studio 2017 Community
- MySQL WorkBench

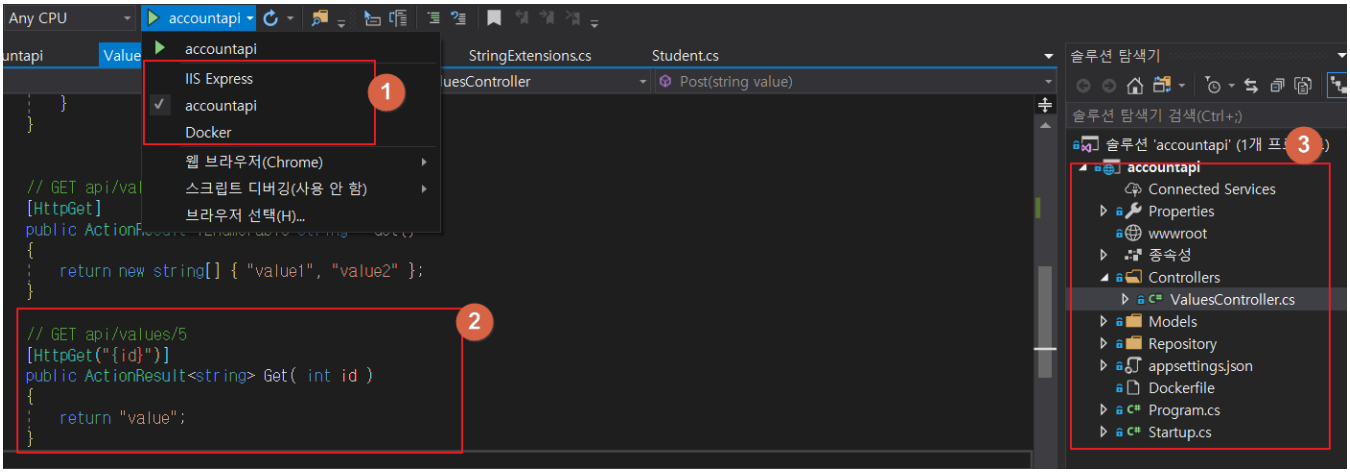


API : API API .

ASP.net Core 2.1 : .net framework 4.? . , Core .

Docker : Docker/ (HyperV for Docker) . .net core .

Https : https . , https haproxy L7 .



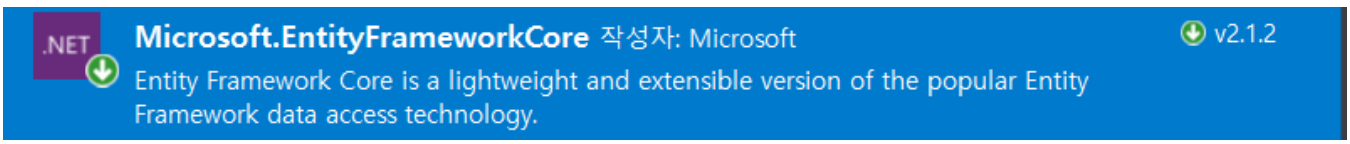
- 1 : . IIS/Console/Docker .
- 2: RestAPI . Endpoint .
- 3: Controllers API .

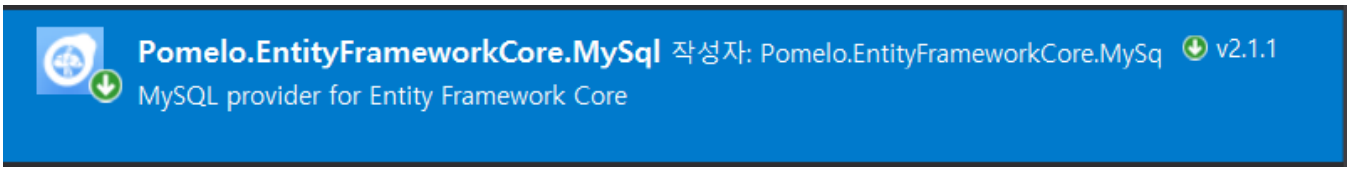
NuGet		/	/	
Microsoft.EntityFrameworkCore.SqlServer	SQL Server 2008	EF Core (Microsoft)		docs
Microsoft.EntityFrameworkCore.Sqlite	SQLite 3.7	EF Core (Microsoft)		docs
Microsoft.EntityFrameworkCore.InMemory	EF Core	EF Core (Microsoft)		docs
Npgsql.EntityFrameworkCore.PostgreSQL	PostgreSQL	Npgsql		docs
Pomelo.EntityFrameworkCore.MySql	MySQL, MariaDB	Pomelo Foundation		readme
Pomelo.EntityFrameworkCore.MyCat	MyCAT Server	Pomelo Foundation	, EF Core 1.1	readme
EntityFrameworkCore.SqlServerCompact40	SQL Server Compact 4.0	Erik Ejlskov Jensen	.NET Framework	wiki
EntityFrameworkCore.SqlServerCompact35	SQL Server Compact 3.5	Erik Ejlskov Jensen	.NET Framework	wiki
MySql.Data.EntityFrameworkCore	MySQL	MySQL (Oracle)		docs
FirebirdSql.EntityFrameworkCore.Firebird	Firebird 2.5 3.x	Jií inura	EF Core 2.0	docs
EntityFrameworkCore.FirebirdSql	Firebird 2.5 3.x	Rafael Almeida	EF Core 2.0	wiki
IBM.EntityFrameworkCore	Db2, Informix	IBM	Windows	
IBM.EntityFrameworkCore-lnx	Db2, Informix	IBM	Linux	
IBM.EntityFrameworkCore-osx	Db2, Informix	IBM	macOS	
Devart.Data.Oracle.EFCore	Oracle 9.2.0.4	DevArt		docs
Devart.Data.PostgreSql.EFCore	PostgreSQL 8.0	DevArt		docs
Devart.Data.SQLite.EFCore	SQLite 3	DevArt		docs
Devart.Data.MySql.EFCore	MySQL 5	DevArt		docs
EntityFrameworkCore.Jet	Microsoft Access	Bubi	EF Core 2.0, .NET Framework	readme

DB , Nuget .

.net core ORM EF(Entity FrameWork) 2.1

EntityCore Mysql Pomelo.EntityFrameworkCore.Mysql .





DB Entity

Entity

C#	SQL-DDL

```
using System;
using System.ComponentModel.DataAnnotations.Schema;


namespace accountapi.Models
{
    [Table("student")]
    public class Student
    {
        [DatabaseGenerated(DatabaseGeneratedOption.None)]
        public int ID { get; set; }
        public string LastName { get; set; }
        public string FirstName { get; set; }
        public DateTime RegDate { get; set; }

    }
}
```

```
CREATE TABLE `student` (
  `id` int(11) NOT NULL,
  `lastname` varchar(45) DEFAULT NULL,
  `firstname` varchar(45) DEFAULT NULL,
  `regdate` datetime DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Entity(C#) DDL

Table SQL .

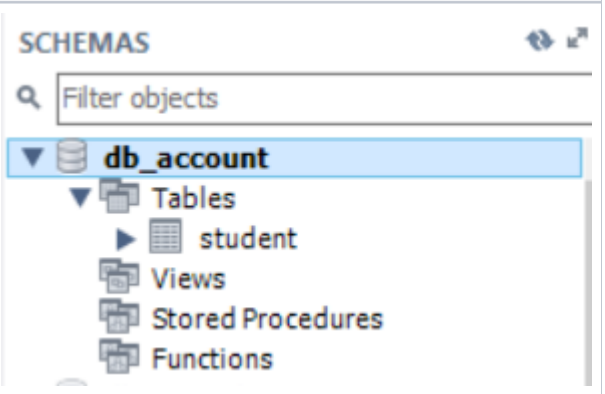
Repository(DBContent)

C#	DB-SCHMEMAS

```
using accountapi.Models;
using Microsoft.EntityFrameworkCore;

namespace accountapi.Repository
{
    public class AccountContent : DbContext
    {
        public DbSet<Student> Students { get; set; }

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



DataBase .

DBContext(Repository) Entity(Table) .

DB DBContext 1:1 .

DBContext .

Controller DBContent

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using accountapi.Models;
using accountapi.Repository;
using Microsoft.AspNetCore.Mvc;

namespace accountapi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ValuesController : ControllerBase
    {
        private readonly AccountContent _context;

        public ValuesController( AccountContent context )
        {
            _context = context;

            if ( _context.Students.Count() == 0 )
            {
                //
                // Create a new Student if collection is empty,
                // which means you can't delete all Student.
                _context.Students.Add(new Student { ID=0, FirstName = "ORM",LastName="Entity" });
                _context.SaveChanges();
            }
        }
```

API Controller DBContext . DBContext

.

ORM , SQL .

SQL SQL Entity

(3 .) (2) / (2) / (2)

SQL 9 , .

ORM CRUD(Create Read Update Delete) SQL .

Document .

- https://docs.microsoft.com/en-us/ef/core/querying/basic
- https://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b

DB

```
using accountapi.Repository;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;

namespace accountapi
{
    public class Startup
    {
        public Startup( IConfiguration configuration )
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices( IServiceCollection services )
        {
            services.AddDbContext<AccountContent>(opt =>
                opt.UseMySql("server=localhost;database=db_account;user=psmon;password=db1234"));

            services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
        }
```

Object (mysql) .

DBContext(Repository) DB , DB DBContext .

DBContext Table

```
    public class Startup
    {
        ......................
        // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
        public void Configure( IApplicationBuilder app, IHostingEnvironment env )
        {
            if ( env.IsDevelopment() )
            {
                using (var serviceScope = app.ApplicationServices.GetService<IServiceScopeFactory>().
CreateScope())
                {
                    var context = serviceScope.ServiceProvider.GetRequiredService<AccountContent>();
                    context.Database.EnsureDeleted();
                    context.Database.EnsureCreated();
                }
                app.UseDeveloperExceptionPage();
            }
            app.UseMvc();
        }
    }
```

DB ,

DBContent Entity, .

DBTable SQL , .

EF Core

URL .

https://docs.microsoft.com/ko-kr/ef/core/managing-schemas/migrations/