Controller:

```csharp
1   using ExampleAppService.Rest.Database;
2   using ExampleAppService.Rest.Models;
3   using ExampleAppService.Rest.Database.Entities;
4   using Microsoft.AspNetCore.Mvc;
5   using Microsoft.EntityFrameworkCore;
6
7   namespace ExampleAppService.Rest.Controllers;
8
9   [ApiController]
10  [Route("[controller]")]
    0 references
11  public class PeopleController : ControllerBase
12  {
        4 references
13      private readonly PeopleDb db;
        0 references
14      public PeopleController(PeopleDb db)
15      {
16          this.db = db;
17      }
18
19      [HttpGet]
        0 references
20      public async Task<IActionResult> GetPeople()
21      {
22          var people = await db.People.ToListAsync();
23          return Ok(people);
24      }
25
26      [HttpPost]
        0 references
27      public async Task<IActionResult> Add([FromBody] CreatePersonRequest request)
28      {
29          if(!request.Validate()) {
30              throw new Exception("Incorrect model.");
31          }
32
33          var personEntity = new PersonEntity(request.FirstName, request.LastName, request.PhoneNumber);
34          db.People.Add(personEntity);
35          await db.SaveChangesAsync();
36
37          return Ok();
38      }
39  }
40
```

Model:

```
namespace ExampleAppService.Rest.Models
{
    public class CreatePersonRequest
    {
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string PhoneNumber { get; set; }

        public bool Validate()
        {
            return string.IsNullOrEmpty(FirstName) && string.IsNullOrEmpty(LastName) && string.IsNullOrEmpty(PhoneNumber);
        }
    }
}
```

peopleDb:

```
using Microsoft.EntityFrameworkCore;
using ExampleAppService.Rest.Database.Entities;

namespace ExampleAppService.Rest.Database
{
    public class PeopleDb : DbContext
    {
        public PeopleDb(DbContextOptions options) : base(options)
        {
        }

        public DbSet<PersonEntity> People { get; protected set; }

        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            base.OnModelCreating(modelBuilder);
            var personEntity = modelBuilder.Entity<PersonEntity>();
            personEntity.ToTable("Person");
            personEntity.HasKey(pk => pk.PersonId);
            personEntity.Property(p => p.FirstName).HasMaxLength(250);
            personEntity.Property(p => p.LastName).HasMaxLength(250);
            personEntity.Property(p => p.PhoneNumber).HasMaxLength(12);
        }
    }
}
```

personEntity:

```csharp
namespace ExampleAppService.Rest.Database.Entities
{
    4 references
    public class PersonEntity
    {
        1 reference
        protected PersonEntity()
        {
            CreatedOn = DateTime.UtcNow;
        }
        1 reference
        public PersonEntity(string firstName, string lastName, string phoneNumber) : this()
        {
            FirstName = firstName;
            LastName = lastName;
            PhoneNumber = phoneNumber;
        }

        1 reference
        public int PersonId { get; protected set; }
        2 references
        public string FirstName { get; protected set; }
        2 references
        public string LastName { get; protected set; }
        2 references
        public string PhoneNumber { get; protected set; }
        1 reference
        public DateTime CreatedOn { get; protected set; }

    }
}
```

Program.cs:

```csharp
using ExampleAppService.Rest.Database;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddDbContext<PeopleDb>(options =>
{
    var connectionString = builder.Configuration.GetConnectionString("PeopleDb");
    options.UseSqlServer(connectionString);
});

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```