C# 应用实验

实验一

1. 输入如下命令

```
/*
create blazor demo
*/
```

```
/*
create blazor demo
*/

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Components;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.HttpsPolicy;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using BlazorDemo.Data;
using Microsoft.EntityFrameworkCore;
using BlazorDemo.Data.Repository;
using BlazorDemo.Data.Repository.IRepository;
using BlazorDemo.Data.UnitOfWork;
using BlazorDemo.Data.UnitOfWork.IUnitOfWork;
using BlazorDemo.Data.Repository.RepositoryBase;
using BlazorDemo.Data.Repository.RepositoryBase.IRepository;
using BlazorDemo.Data.Repository.RepositoryBase.RepositoryBase;

namespace BlazorDemo
{
    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.
        // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddRazorPages();
            services.AddServerSideBlazor();
            services.AddSingleton<WeatherForecastService>();
            services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));
            services.AddScoped<ICategoryRepository, CategoryRepository>();
```

实验二

1. 命令行创建一个 .NET 6 的 Minimal API

```
dotnet new webapi -minimal -o apidemo
```

2. 在 Program.cs 文件 添加如下命令

```
/*
create get city api
*/
```

```
// create get city api
app.Map("/api/cities/{id}", (cityId) =>
{
    cityId.Run(async context =>
    {
        var city = context.GetRouteData().Values["id"];
        var temp = await GetTemperatureAsync(city);
        var summary = summaries[(int)Math.Floor(temp / 10)];
        context.Response.StatusCode = 200;
        await context.Response.WriteAsync($"{city} is {temp:F1} degrees. {summary}");
    });
});
```