

C# 应用实验

实验一

1. 输入如下命令

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

```
1  /*  
2  
3  create blazor demo  
4  
5  */  
6  
7  using System;  
8  using System.Collections.Generic;  
9  using System.Linq;  
10 using System.Threading.Tasks;  
11 using Microsoft.AspNetCore.Builder;  
12 using Microsoft.AspNetCore.Components;  
13 using Microsoft.AspNetCore.Hosting;  
14 using Microsoft.AspNetCore.HttpsPolicy;  
15 using Microsoft.Extensions.Configuration;  
16 using Microsoft.Extensions.DependencyInjection;  
17 using Microsoft.Extensions.Hosting;  
18 using BlazorDemo.Data;  
19 using Microsoft.EntityFrameworkCore;  
20 using BlazorDemo.Data.Repository;  
21 using BlazorDemo.Data.Repository.IRepository;  
22 using BlazorDemo.Data.UnitOfWork;  
23 using BlazorDemo.Data.UnitOfWork.IUnitOfWork;  
24 using BlazorDemo.Data.Repository.RepositoryBase;  
25 using BlazorDemo.Data.Repository.RepositoryBase.IRepository;  
26 using BlazorDemo.Data.Repository.RepositoryBase.RepositoryBase;  
27  
28 namespace BlazorDemo  
29 {  
30     public class Startup  
31     {  
32         public Startup(IConfiguration configuration)  
33         {  
34             Configuration = configuration;  
35         }  
36  
37         public IConfiguration Configuration { get; }  
38  
39         // This method gets called by the runtime. Use this method to add services to the container.  
40         // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940  
41         public void ConfigureServices(IServiceCollection services)  
42         {  
43             services.AddRazorPages();  
44             services.AddServerSideBlazor();  
45             services.AddSingleton<WeatherForecastService>();  
46             services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));  
47             services.AddScoped<ICategoryRepository, CategoryRepository>();  
48         }  
49     }  
50 }  
51  
52 public class Program  
53 {  
54     public static void Main()  
55     {  
56         CreateHostBuilder().Build().Run();  
57     }  
58  
59     private static IHostBuilder CreateHostBuilder() =>  
60     {  
61         return Host.CreateDefaultBuilder().ConfigureServices(s =>  
62         {  
63             s.AddStartupServices();  
64         }).Build();  
65     }  
66 }  
67  
68 public static class StartupServicesExtensions  
69 {  
70     public static void AddStartupServices(this IServiceCollection services)  
71     {  
72         services.AddScoped<IRepository, RepositoryBase>();  
73         services.AddScoped<IUnitOfWork, UnitOfWork>();  
74     }  
75 }
```

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}");  
    });  
});
```