

Implementing a Round-Robin Load Balancer with YARP



Added 3 sample web API projects and One web API project for loadbalancer proxy

```
"https": {
  "commandName": "Project",
  "dotnetRunMessages": true,
  "launchBrowser": true,
  "launchUrl": "swagger",
  "applicationUrl": "https://localhost:7201;http://localhost:5263",
  "environmentVariables": {
    "ASPNETCORE_ENVIRONMENT": "Development"
  }
},
"https": {
  "commandName": "Project",
  "dotnetRunMessages": true,
  "launchBrowser": true,
  "launchUrl": "swagger",
  "applicationUrl": "https://localhost:7202;http://localhost:5219",
  "environmentVariables": {
    "ASPNETCORE_ENVIRONMENT": "Development"
  }
},
"https": {
  "commandName": "Project",
  "dotnetRunMessages": true,
  "launchBrowser": true,
  "launchUrl": "swagger",
  "applicationUrl": "https://localhost:7203;http://localhost:5131",
  "environmentVariables": {
    "ASPNETCORE_ENVIRONMENT": "Development"
  }
}
```

Updated Launch settings files with port for continence mainAPI-7201, weather1-7202, weather2-7203.



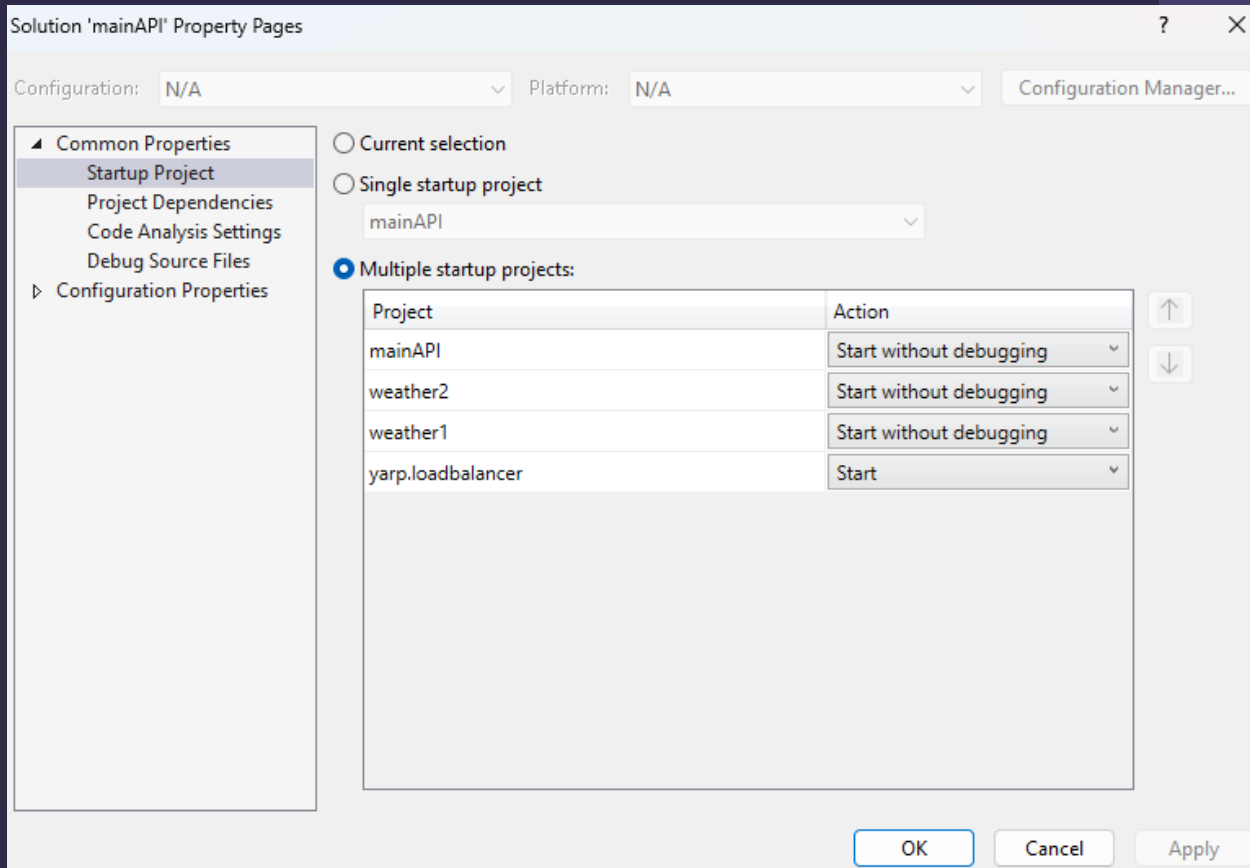
Added Yarp.ReverseProxy Nuget package to load balancer project

```
yarp.loadbalancer
1  var builder = WebApplication.CreateBuilder(args);
2
3  builder.Services.AddEndpointsApiExplorer();
4  builder.Services.AddSwaggerGen();
5  builder.Services.AddReverseProxy()
6    .LoadFromConfig(builder.Configuration.GetSection("ReverseProxy"));
7  builder.Services.AddHealthChecks();
8  var app = builder.Build();
9
10 // Configure the HTTP request pipeline.
11 if (app.Environment.IsDevelopment())
12 {
13     app.UseSwagger();
14     app.UseSwaggerUI();
15 }
16
17 app.UseHttpsRedirection();
18
19 app.MapReverseProxy();
20 app.MapHealthChecks("health");
21 app.Run();
22
```

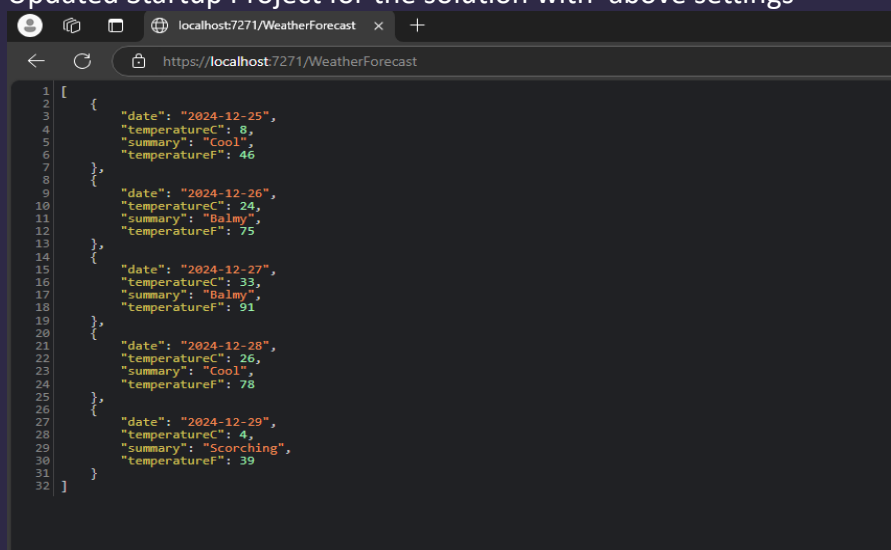
Configured services for yarp and health check inside Program.cs of load balancer project

```
Schema: https://json.schemastore.org/appsettings.json
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft.AspNetCore": "Warning"
6      }
7    },
8    "AllowedHosts": "*",
9    "ReverseProxy": {
10     "Routes": {
11       "weatherRoute": {
12         "ClusterId": "apiCluster",
13         "Match": {
14           "Host": "localhost",
15           "Path": "WeatherForecast/{**catchall}"
16         }
17       }
18     },
19     "Clusters": {
20       "apiCluster": {
21         "LoadBalancingPolicy": "RoundRobin",
22         "Destinations": {
23           "mainapp": {
24             "Address": "https://localhost:7201/"
25           },
26           "weather1": {
27             "Address": "https://localhost:7202/"
28           },
29           "weather2": {
30             "Address": "https://localhost:7203/"
31           }
32         }
33       }
34     }
35   }
36 }
37
```

Added ReverseProxy section to application settings and configure Routes and Clusters.



Updated Startup Project for the solution with above settings



Started Project and Refreshed weather forecast endpoint to check RoundRobin assignment

```
info: Yarp.ReverseProxy.Configuration.ConfigProvider.ConfigurationConfigProvider[1]
  Loading proxy data from config.
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: https://localhost:7271
info: Microsoft.Hosting.Lifetime[14]
  Now listening on: http://localhost:5209
info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
  Content root path: D:\Study Materials\LoadBalancer\yarp.loadbalancer
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7203/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7202/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7201/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7203/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7202/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7201/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[9]
  Proxying to https://localhost:7203/WeatherForecast HTTP/2 RequestVersionOrLower
info: Yarp.ReverseProxy.Forwarder.HttpForwarder[56]
  Received HTTP/2.0 response 200.
```

You can find it is assigning Load balancer according to round robin manner.

Thanking You !