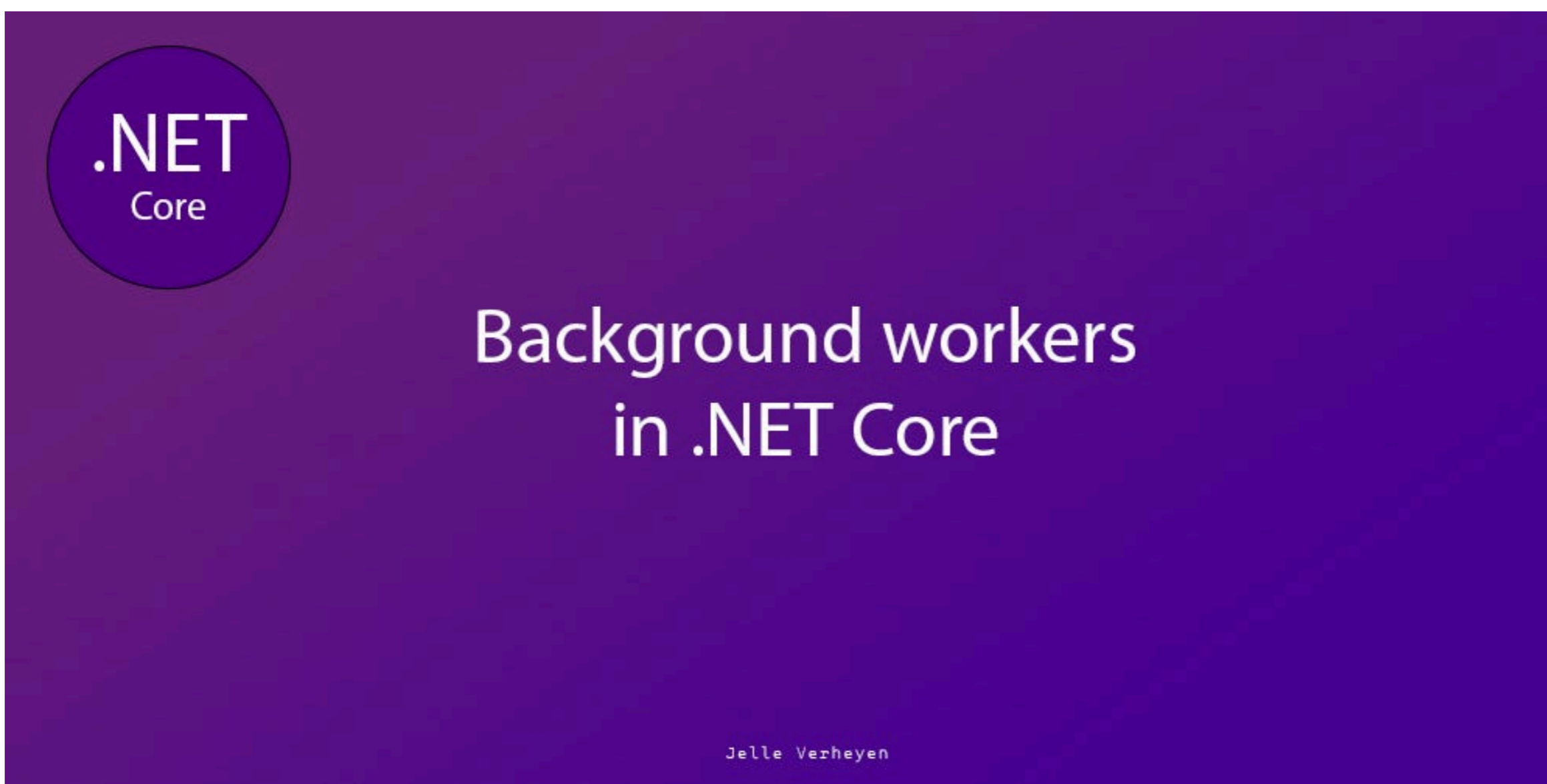


.net core实现后台服务就这么简单，只要实现IHostedService接口就行了，你用过吗？

作者：微信公众号：【架构师老卢】

11-18 8:53

158



概述： `IHostedService` 接口在.NET Core中的主要用途是定义应用程序生命周期内运行的后台服务。这些服务可以执行初始化、长时间运行的任务、定期运行的任务等。通过实例，我们可以演示如何使用 `IHostedService` 来实现一个简单的网络检测服务，并记录日志。

IHostedService 接口在.NET Core中的主要用途是定义应用程序生命周期内运行的后台服务。这些服务可以执行初始化、长时间运行的任务、定期运行的任务等。通过实例，我们可以演示如何使用 **IHostedService** 来实现一个简单的网络检测服务，并记录日志。

首先，我们需要创建一个实现 **IHostedService** 接口的类。在这个类中，我们将使用 **HttpClient** 定期检测网络是否正常，并通过 **ILogger** 记录日志。

```
1 using System;
2 using System.Net.Http;
3 using System.Threading;
4 using System.Threading.Tasks;
5 using Microsoft.Extensions.Hosting;
6 using Microsoft.Extensions.Logging;
7
8 public class NetworkStatusService : IHostedService, IDisposable
9 {
10     private readonly ILogger<NetworkStatusService> _logger;
11     private readonly HttpClient _httpClient;
12     private Timer _timer;
13
14     public NetworkStatusService(ILogger<NetworkStatusService> logger)
15     {
16         _logger = logger ?? throw new ArgumentNullException(nameof(logger));
17         _httpClient = new HttpClient();
18     }
19
20     public Task StartAsync(CancellationToken cancellationToken)
21     {
22         _logger.LogInformation("NetworkStatusService 正在启动.");
23
24         // 使用 Timer 定期执行网络检测任务
25         _timer = new Timer(CheckNetworkStatus, null, TimeSpan.Zero, TimeSpan.FromSeconds(30));
26
27         return Task.CompletedTask;
28     }
29
30     private async void CheckNetworkStatus(object state)
31     {
32         try
33         {
34             // 这里可以替换成实际的网络检测逻辑
35             var response = await _httpClient.GetAsync("https://www.example.com");
36
37             if (response.IsSuccessStatusCode)
38             {
39                 _logger.LogInformation("网络正常.");
40             }
41             else
42             {
43                 _logger.LogWarning("网络异常。 HTTP 状态码: {StatusCode}", (int)response.StatusCode);
44             }
45         }
46         catch (Exception ex)
47         {
48             _logger.LogError(ex, "网络检测时发生错误.");
49         }
50     }
51
52     public Task StopAsync(CancellationToken cancellationToken)
53     {
54         _logger.LogInformation("NetworkStatusService 正在停止.");
55
56         // 在这里执行清理逻辑，停止长时间运行的任务
57         _timer?.Change(Timeout.Infinite, 0);
58
59         return Task.CompletedTask;
60     }
61
62     public void Dispose()
63     {
64         // 释放资源
65         _httpClient.Dispose();
66         _timer?.Dispose();
67     }
68 }
```

接下来，我们需要在 **Startup.cs** 文件中注册这个服务：

```
1 using Microsoft.Extensions.DependencyInjection;
2 using Microsoft.Extensions.Hosting;
3
4 public class Startup
5 {
6     public void ConfigureServices(IServiceCollection services)
7     {
8         // 注册 NetworkStatusService
9         services.AddHostedService<NetworkStatusService>();
10    }
11 }
```

这个示例中，**NetworkStatusService** 类会在应用程序启动时开始执行网络检测任务，每隔30秒执行一次。如果网络正常，将会记录信息级别的日志，如果网络异常，则记录警告级别的日志。这样，我们就使用 **IHostedService** 实现了一个简单的网络检测服务，并通过日志记录器记录相关信息。

