ASP.NET Core

[运行状况监视](https://docs.microsoft.com/zh-cn/dotnet/standard/microservices-architecture/implement-resilient-applications/monitor-app-health)

技术点：ASP.NET Core 服务中实现运行状况检查;使用监视程序（AspNetCore.Diagnostics.HealthChecks、AspNetCore.HealthChecks.UI）

[BETTER TIMEOUT HANDLING WITH HTTPCLIENT](https://thomaslevesque.com/2018/02/25/better-timeout-handling-with-httpclient/)

技术点：HttpClient设置超时最佳实践

[ASP.NET Core官方技术文档参考](https://docs.microsoft.com/zh-cn/aspnet/core/security/?view=aspnetcore-2.2)

技术点：ASP.NET Core官方技术文档

[确保 .NET 微服务和 Web 应用的安全性](https://docs.microsoft.com/zh-cn/dotnet/standard/microservices-architecture/secure-net-microservices-web-applications/)

技术点：ASP.NET Core身份认证、授权。

[异步编程中的最佳做法](https://msdn.microsoft.com/zh-cn/magazine/jj991977.aspx)

技术点：async await最佳做法、误区解答

[Designing Evolvable Web APIs with ASP.NET: Harnessing the Power of the Web](https://books.google.com.hk/books?id=xZ0TAwAAQBAJ&pg=PT491&lpg=PT491&dq=httpclient+CancellationToken&source=bl&ots=JIuxiwMjFZ&sig=ACfU3U3VkZGbNim8pkytwZsE60eCl8Wq9g&hl=zh-CN&sa=X&ved=2ahUKEwjbtvSE_OXiAhUF-2EKHdX_A104ChDoATAAegQIBhAB#v=onepage&q=httpclient%20CancellationToken&f=false)

技术点：HttpClient

[Introduction to ASP.NET Web API](https://dotnettutorials.net/lesson/web-api-architecture/)

技术点：ASP.NET Web API示例教程

[ASP.NET Core 数据保护](https://docs.microsoft.com/zh-cn/aspnet/core/security/data-protection/introduction?view=aspnetcore-2.2)

.net core应用程序中的数据保护、加解密；

加解密中的salt(盐)：

// generate a 128-bit salt using a secure PRNG

byte[] salt = new byte[128 / 8];

using (var rng = RandomNumberGenerator.Create())

{

rng.GetBytes(salt);

}

[Microsoft.AspNetCore.DataProtection-示例0001](https://stackoverflow.com/questions/38795103/encrypt-string-in-net-core)

技术点：.net core数据保护、加解密

.Net Core依赖注入代码片段集锦：

public static void Main(string[] args)

{

// add data protection services

var serviceCollection = new ServiceCollection();

serviceCollection.AddDataProtection();

var services = serviceCollection.BuildServiceProvider();

// create an instance of MyClass using the service provider

var instance = ActivatorUtilities.CreateInstance<MyClass>(services);

instance.RunSample();

}

public class MyClass

{

public void RunSample()

{

}

}

.net core跨域设置代码片段：

public void Configure(IApplicationBuilder app)

{

app.UseCors(SpecificOrigins.SpecificOriginName);

}

public IServiceProvider ConfigureServices(IServiceCollection services)

{

services.AddCors(options =>

{

options.AddPolicy(SpecificOrigins.SpecificOriginName,

builder =>

{

//builder.WithOrigins("http://signin.yunexpress.com", "https://signin.yunexpress.com")

builder.AllowAnyOrigin()

.AllowAnyHeader()

.AllowAnyMethod();

});

});

return services.BuildServiceProvider();

}