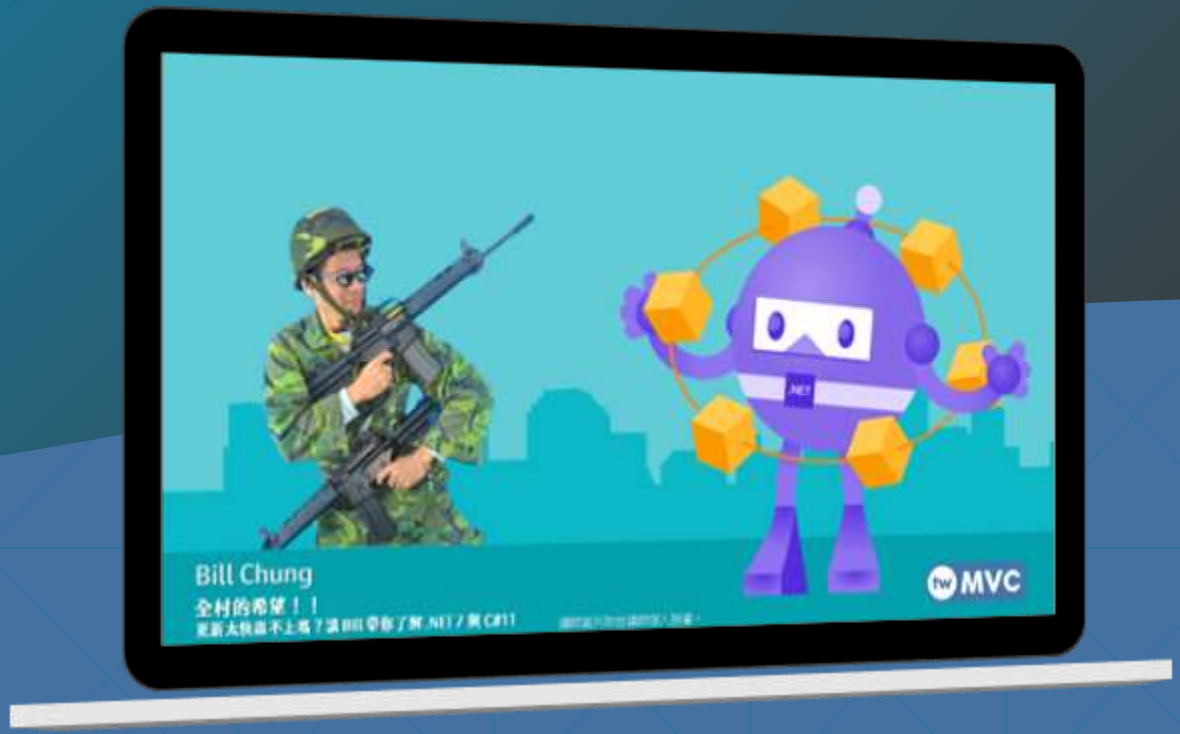
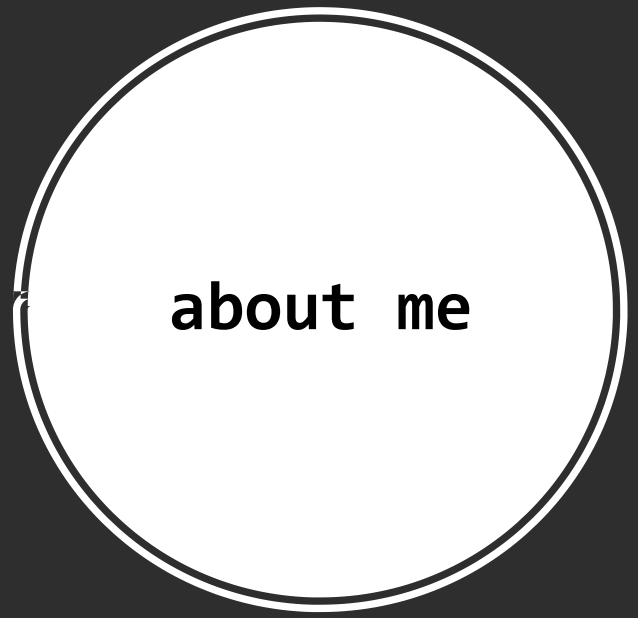
全村的希望

一探 C# 11 與 .NET 7 的神奇

講者：Bill Chung



歡迎參加我們的每週四固定聚會



- Bill Chung
- 專長是說故事
- 海角點部落 https://dotblogs.com.tw/billchung
- github https://github.com/billchungiii

Agenda







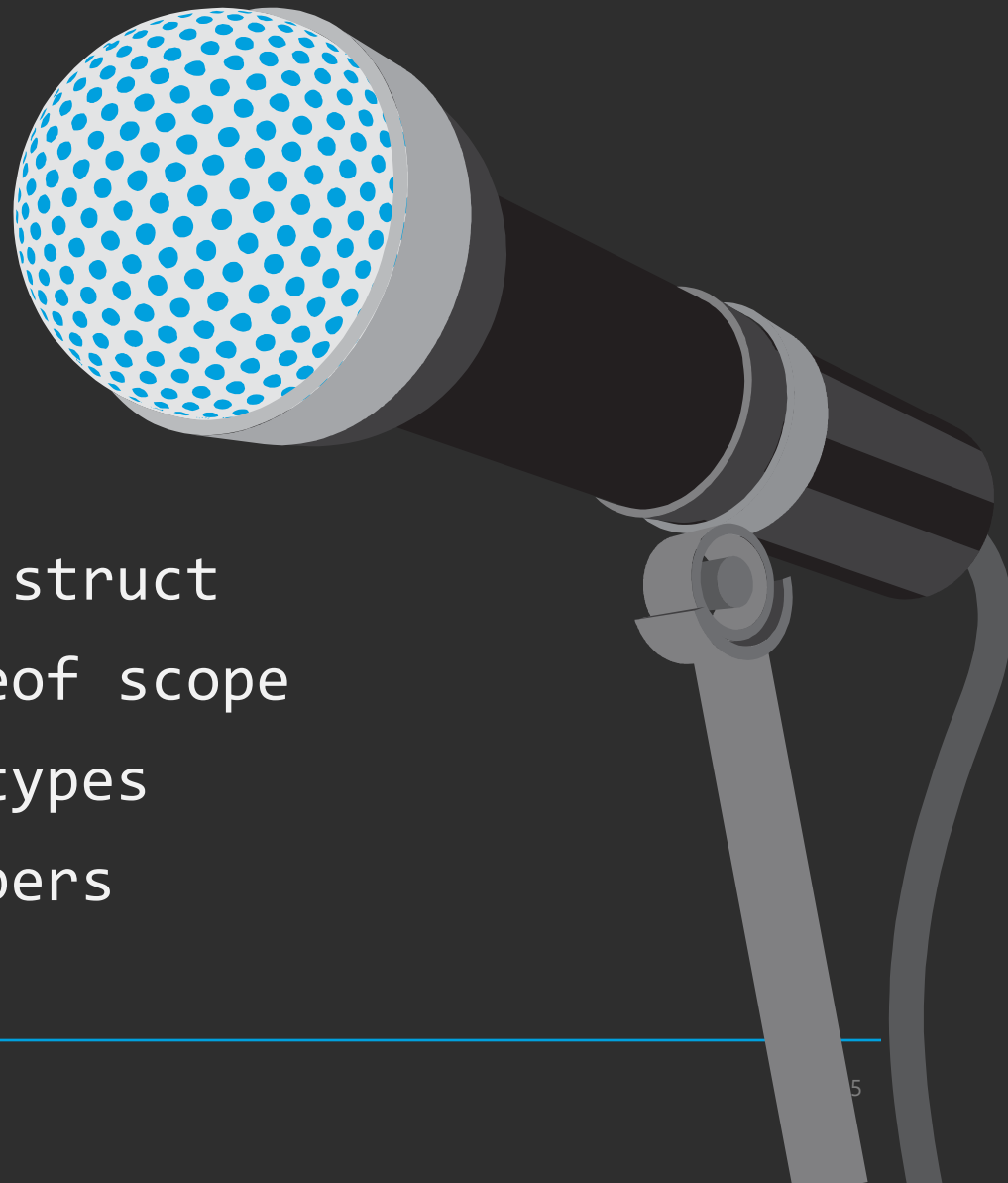
Agenda

.NET 7

- .NET MAUI
- ASP.NET Core Migrations
- Cloud Native
- Performance
- WebSockets over HTTP/2
- New APIs



Agenda



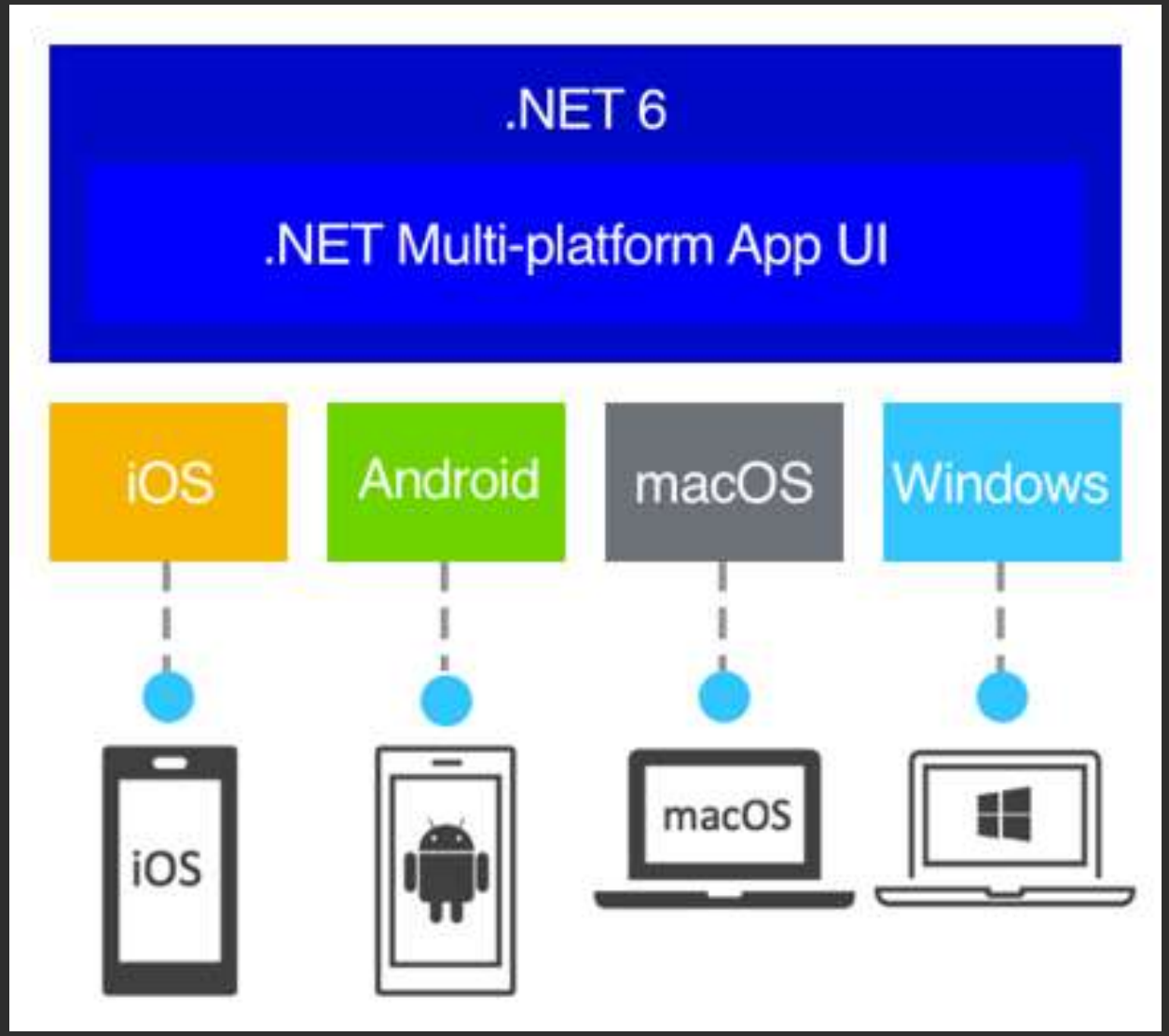
- String
- List patterns
- Generic attribute
- Generic math support
- Auto-default struct
- Extended nameof scope
- File-Scoped types
- Required members

.NET 7 new features



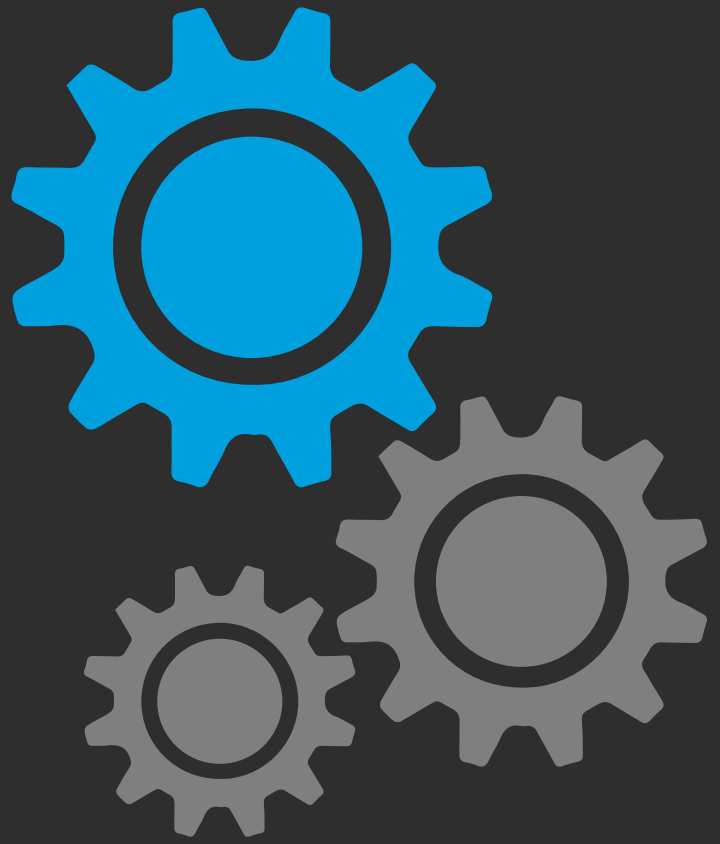
.NET MAUI

- 新增支援 Tizen
- 更多的控制項
- 更好的效能




1
2
3
App Code
.NET MAUI
.NET for Android
.NET for iOS
.NET for Mac
WinUI 3
.NET 6 BCL
Mono Runtime
.NET CoreCLR
Android
iOS
macOS
Windows

ASP.NET Core Migrations



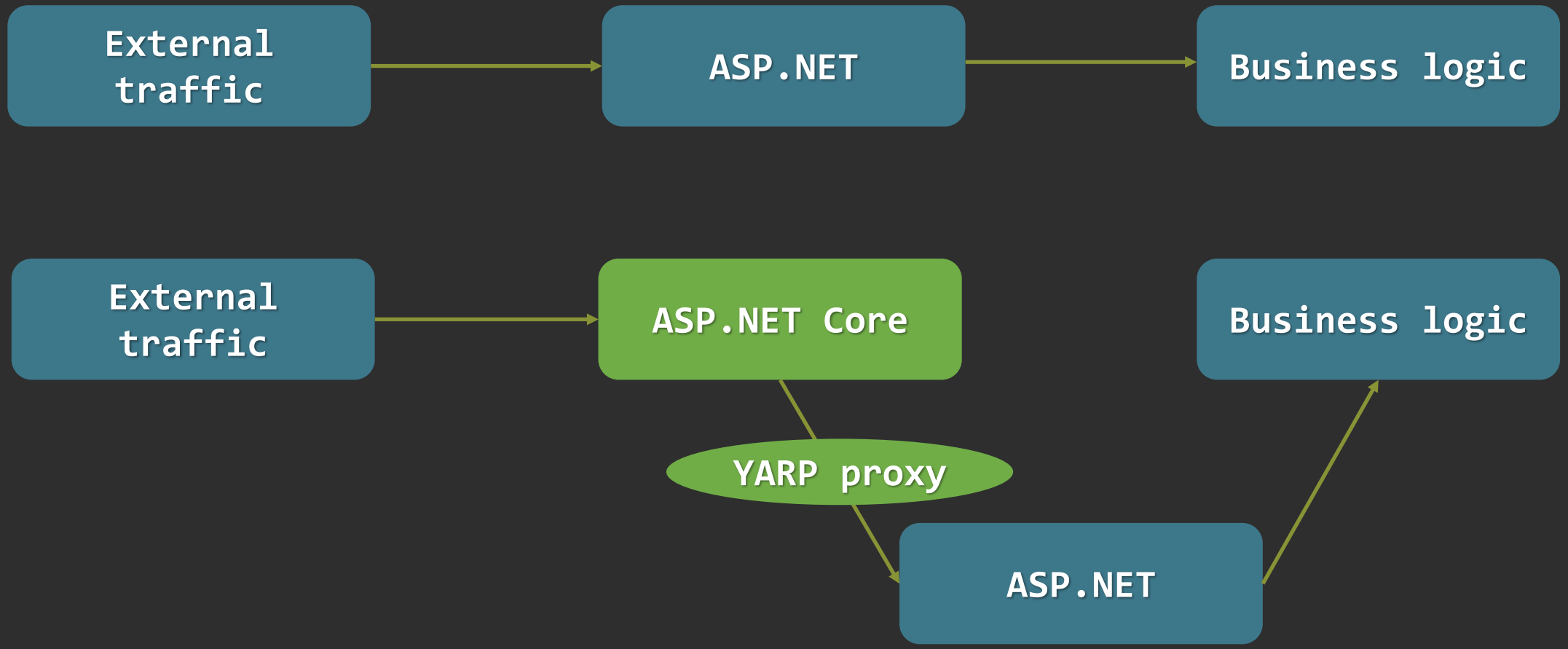
Migrate ASP.NET to ASP.NET Core

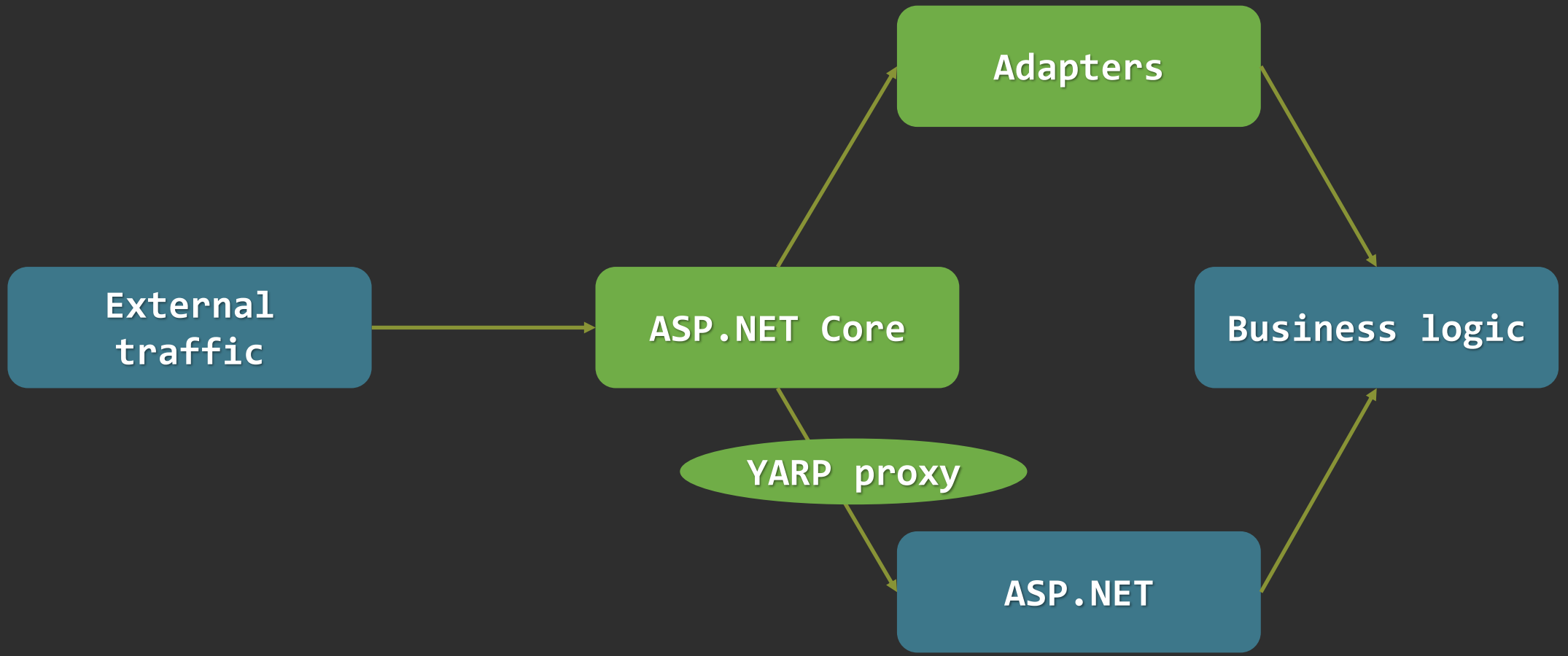
瀏覽
https://aka.ms/AspNetCoreMigration

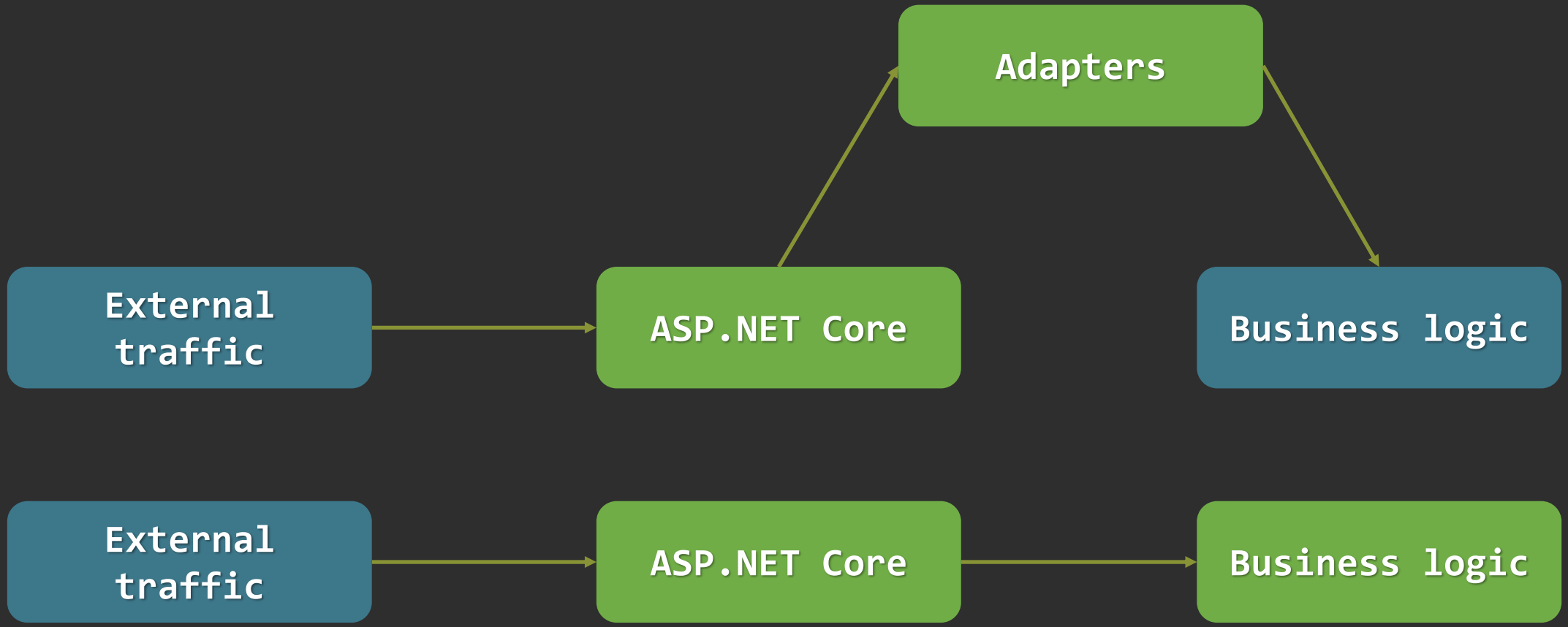
安裝
Microsoft Project Migrations (Experimental)

Migration steps

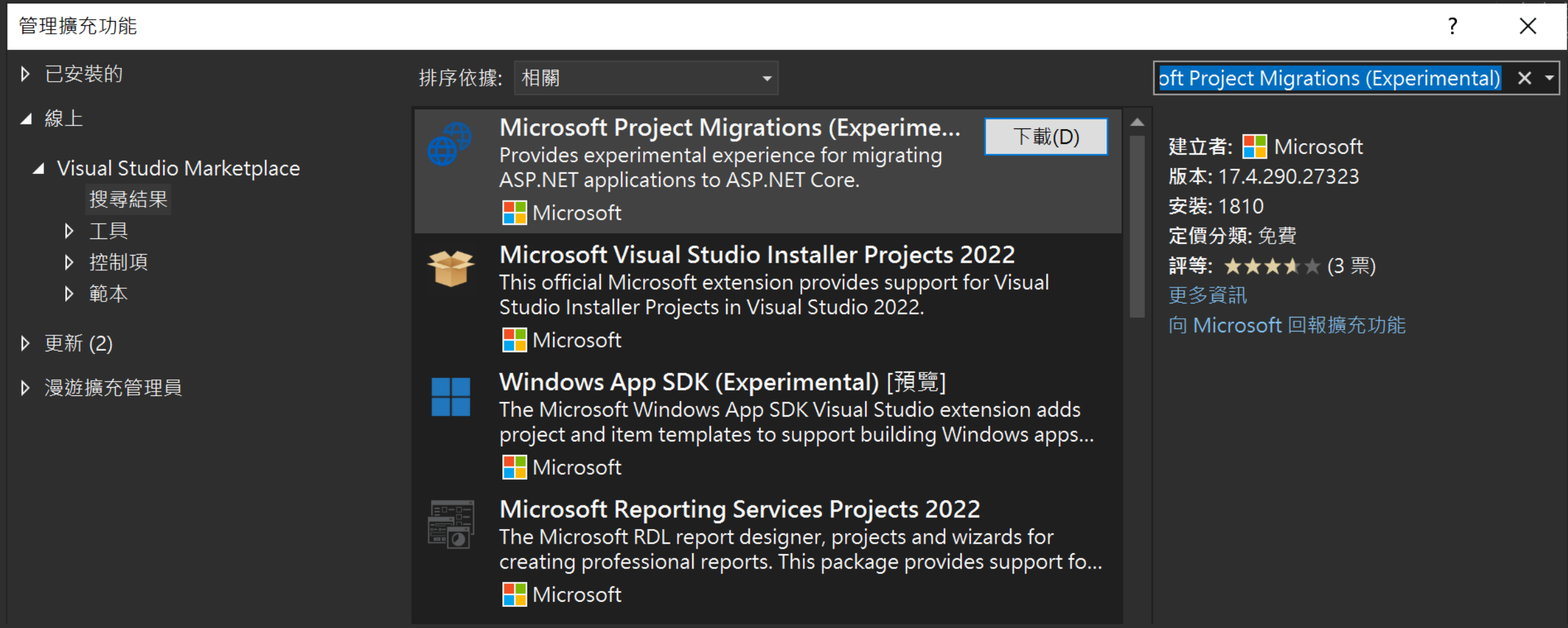



External traffic
ASP.NET
Business logic
External traffic
ASP.NET Core
YARP proxy
ASP.NET
Business logic


Adapters
External traffic
ASP.NET Core
Business logic
YARP proxy
ASP.NET


Adapters
External traffic
ASP.NET Core
Business logic
External traffic
ASP.NET Core
Business logic

Installation



Cloud Native

- gRPC JSON transcoding for .NET
- Built-in container support for the .NET SDK
- Azure v4 support .NET 7



gRPC JSON transcoding for .NET

- Microsoft.AspNetCore.Grpc.JsonTranscoding

```
syntax = "proto3";
import "google/api/annotations.proto";

package greet;
service Greeter {
  rpc SayHello (HelloRequest) returns (HelloReply) {
    option (google.api.http) = {
      get: "/v1/greeter/{name}"
    };
  }
}
```

built-in container support for the .NET SDK

- support Linux container
- just **dotnet publish**

Performance

- JIT
- GC
- Native AOT
- Mono
- Reflection
- Interop
- Threading
- Primitive Types and Numerics
- Arrays, Strings, and Spans
- Regex
- Collections
- Regex
- LINQ
- File I/O
- Compression
- Networking
- JSON
- XML
- Cryptography
- Diagnostics
- Exceptions
- Registry
- Analyzers

WebSockets over HTTP/2

```
class ClientWebSocket : WebSocket
{   // EXISTING
    public Task ConnectAsync(Uri uri, CancellationToken cancellationToken);
    // NEW
    public Task ConnectAsync(Uri uri, HttpMessageInvoker invoker, CancellationToken cancellationToken);
}

class ClientWebSocketOptions
{
    // NEW
    public System.Version HttpVersion
    {
        get { throw null; }
        [System.Runtime.Versioning.UnsupportedOSPlatformAttribute("browser")]
        set { }
    };
    public System.Net.Http.HttpVersionPolicy HttpVersionPolicy
    {
        get { throw null; }
        [System.Runtime.Versioning.UnsupportedOSPlatformAttribute("browser")]
        set { }
    };
}
```

```
class HttpMethod : IEquatable<HttpMethod>
{
    // EXISTING
    // internal static HttpMethod Connect { get { throw null; } }
    // NEW
    public static HttpMethod Connect { get { throw null; } }
}
class HttpRequestHeaders : HttpHeaders
{
    public string? Protocol { get { } set { } };
}
```

Usage

```
ClientWebSocket ws = new();
ws.Options.HttpVersionPolicy = HttpVersionPolicy.RequestVersionOrHigher;
ws.Options.HttpVersion = HttpVersion.Version20;
ws.ConnectAsync(uri, new HttpMessageInvoker(handler), cancellationToken);
HttpRequestMessage request = new(HttpMethod.Connect, server.Address);
request.Headers.Protocol = "websocket";
```

APIs

- INumberBase<T> …
- Added new Tar APIs
- Adding Microseconds and Nanoseconds to TimeStamp, DateTime, DateTimeOffset, and TimeOnly

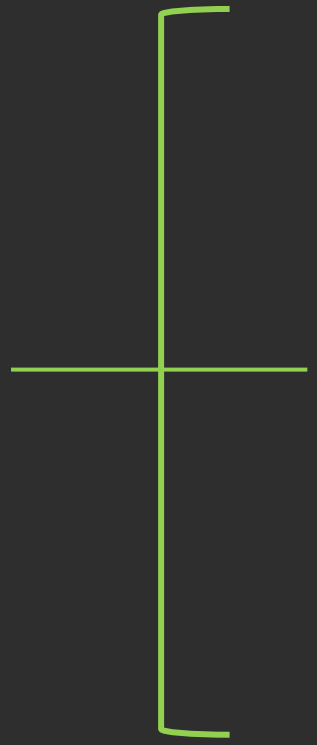
INumberBase<T>

```
public interface INumberBase<TSelf>
    : IAdditionOperators<TSelf, TSelf, TSelf>,
      IAdditiveIdentity<TSelf, TSelf>,
      IDecrementOperators<TSelf>,
      IDivisionOperators<TSelf, TSelf, TSelf>,
      IEquatable<TSelf>,
      IEqualityOperators<TSelf, TSelf, bool>,
      IIncrementOperators<TSelf>,
      IMultiplicativeIdentity<TSelf, TSelf>,
      IMultiplyOperators<TSelf, TSelf, TSelf>,
      ISpanFormattable,
      ISpanParsable<TSelf>,
      ISubtractionOperators<TSelf, TSelf, TSelf>,
      IUnaryPlusOperators<TSelf, TSelf>,
      IUnaryNegationOperators<TSelf, TSelf>
    where TSelf : INumberBase<TSelf>?
```

.C# 11 new features



String



- Raw string literals
- UTF-8 string literals
- Pattern match Span<char> or ReadOnlySpan<char> on a constant string

Raw string literals

- 新的字串常值表示方式，使用三個雙引號 (成對)

```
var s = "以前你要這麼寫 \"雙引號\".";
var s1 = """現在你可以這麼寫 "雙引號".""";
```

UTF-8 string literals

```
string s1 = "魯夫";
var b1 = Encoding.Unicode.GetBytes(s1);
ReadOnlySpan<byte> b2 = "魯夫"u8;
Console.WriteLine(string.Join("-", b1));
Console.WriteLine(string.Join("-", b2.ToArray ()));
var b3 = "魯夫"u8.ToArray();
```

Pattern match Span

- 允許 Span<char> 和 ReadOnlySpan<char> 與字串常值比對

```
static bool Is123(ReadOnlySpan<char> s)
{
    return s is "123";
}

static bool IsABC(Span<char> s)
{
    return s switch { "ABC" => true, _ => false };
}
```

List Patterns

- 清單比對模式，可以比較集合或陣列內容

```
int[] array = { 1,9,8,7,6,5,3};
var result = array switch
             {
                 [1, .. var s, 3] => string.Join("-", s),
                 [1, 2] => "A",
                 [2, 5] => "B",
                 [1, _] => "C",
                 [..] => "D"
             };
```

Generic Attribute

```
public class TypeAttribute : Attribute
{
    public TypeAttribute(Type t) => ParamType = t;
    public Type ParamType { get; }
}

public class TypeAttribute<T> : Attribute { }
```

Possible future

```
[ServiceFilter(typeof(ResponseLoggerFilter))]
```

```
[ServiceFilter<ResponseLoggerFilter>]
```

Generic math support

static virtual members in interfaces

checked operator

static virtual members in interfaces

```
public interface IMyAreaAddOperator<T> where T : IMyAreaAddOperator<T>
{
    static abstract int operator + (T source ,T other);
}
public class MyRectangle : IMyAreaAddOperator<MyRectangle>
{
    public int Width { get; set; }
    public int Height { get; set; }

    public int Area { get => Width * Height; }

    public static int operator +(MyRectangle source, MyRectangle other)
    {
        return source.Area + other.Area;
    }
}
```

Generic math

```
public class Rectangle : IAdditionOperators<Rectangle, Rectangle, int>
{
    public int Width { get; set; }
    public int Height { get; set; }
    public int Area { get => Width * Height; }

    public static int operator +(Rectangle left, Rectangle right)
    {
        return left.Area + right.Area;
    }

    public static int operator checked+(Rectangle left, Rectangle right)
    {
        return left.Area + right.Area;
    }
}
```

Auto-default struct

```
public readonly struct Measurement
{
    public Measurement(double value) { Value = value ;}

    public Measurement(double value, string description)
    {
        Value = value;
        Description = description;
    }

    public Measurement(string description)
    {
        // 會補上
        //this.<Value>k__BackingField = 0.0;
        //this.<Description>k__BackingField = "Ordinary measurement";
        Description = description;
    }

    public double Value { get; init; }
    public string Description { get; init; } = "Ordinary measurement";

    public override string ToString() => $"{Value} ({Description})";
}
```

Extended nameof scope

- nameof 敘述可用於 method 與其 parameters 的 attribute 上

```
[Description(nameof(Main))]
static void Main([Description(nameof(args))]string[] args)
```

File-scoped types

- 存取修飾，限制在同一個檔案內使用的型別

```
file class Class1
{
    public int a;
}

file class Class2
{
    public Class1? c;
}
```

Required members

- 強迫在建立執行體時，成員必須初始化

```
public class Person
{
    [SetsRequiredMembers]
    public Person(string firstName, string lastName) =>
    (FirstName, LastName) = (firstName, lastName);


    public required string FirstName { get; init; }
    public required string LastName { get; init; }
}
```

Blog 是記錄知識的最佳平台



https://dotblogs.com.tw

SkillTree 為了確保內容與實務不會脫節，我們都是聘請企業顧問等級並且目前依然在職場的業界講師，我們不把時間浪費在述說歷史與沿革，我們並不是教您考取證照，而是教您如何上場殺敵，拳拳到肉的內容才是您花錢想要聽到的，而這也剛好是我們擅長的。



https://skilltree.my

天瓏資訊圖書

