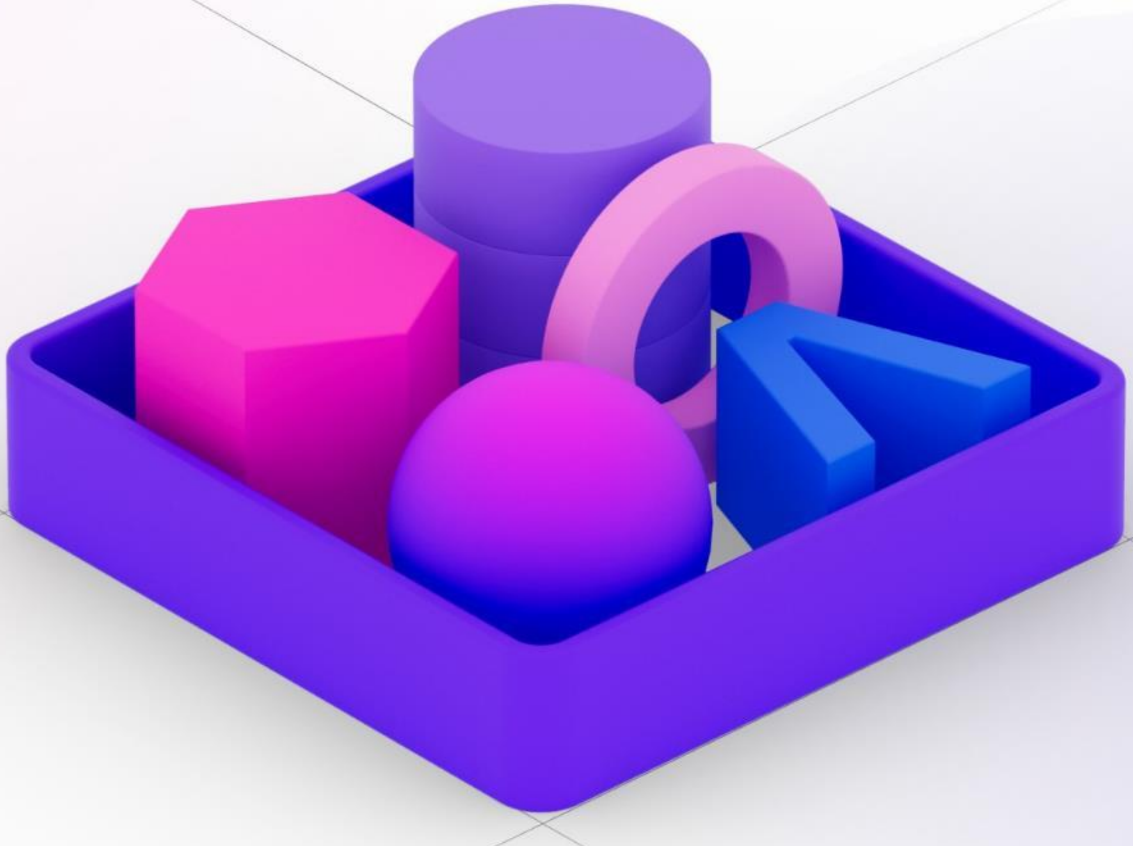
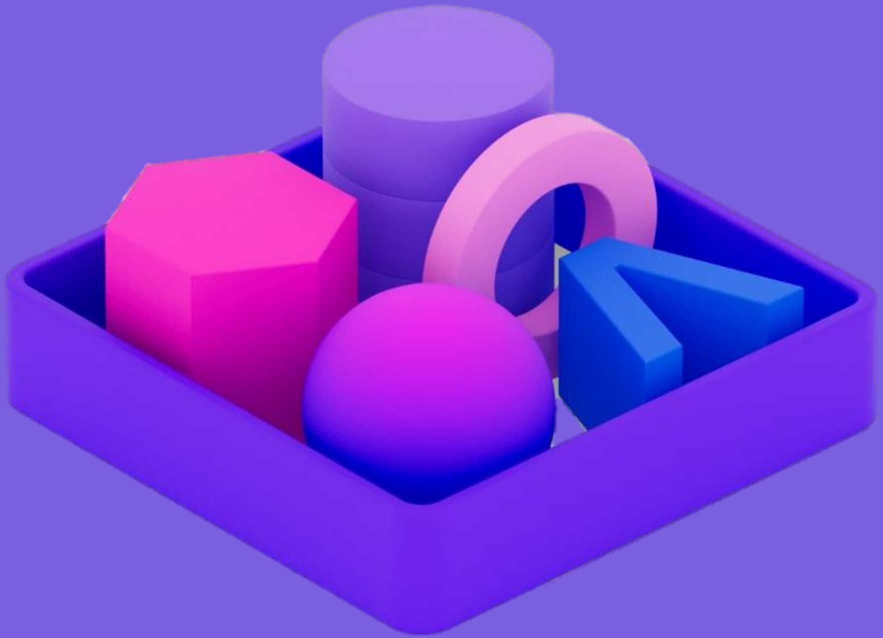
.NET Conf



Boost Your Applications with .NET 8

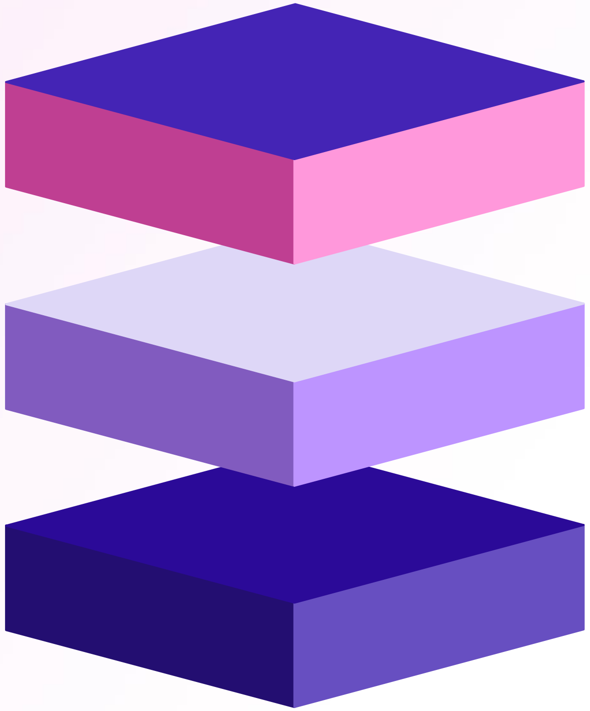


Sharon Gur
Senior Software Engineer
Microsoft

Dynamic PGO

Profile Guided Optimization

Tiered Compilation



Tier 0

Tier 0 + Instrumented

Tier 1

Not optimized, not instrumented

Not optimized, instrumented

Optimized with Dynamic PGO

How does Dynamic PGO Optimize?

Inlining

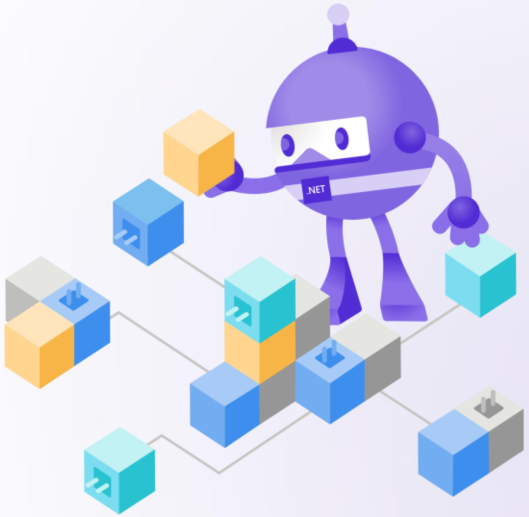
Caching

Constant Propagation

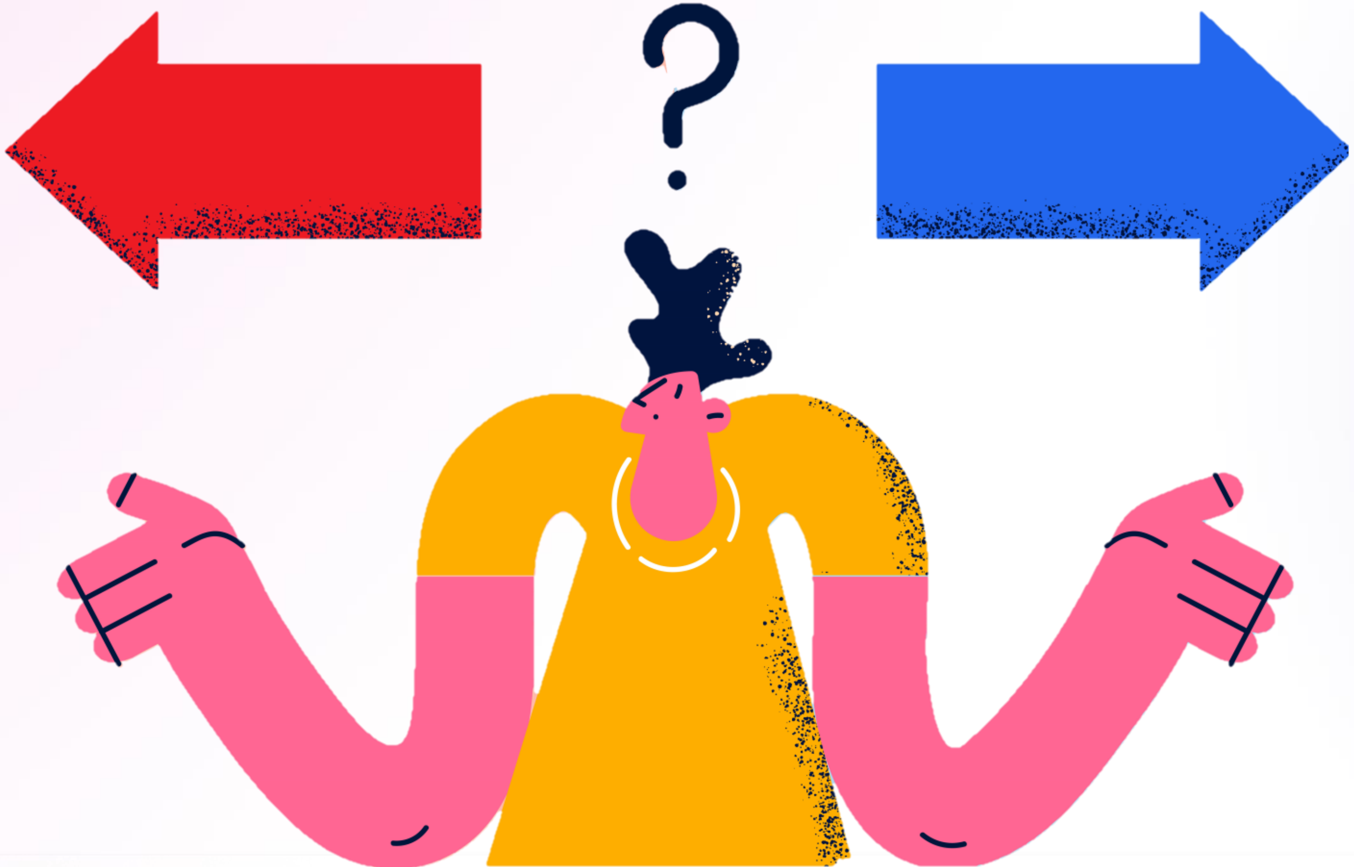
Dead Code Elimination

Code Layout

GDV



Guarded DeVirtualization



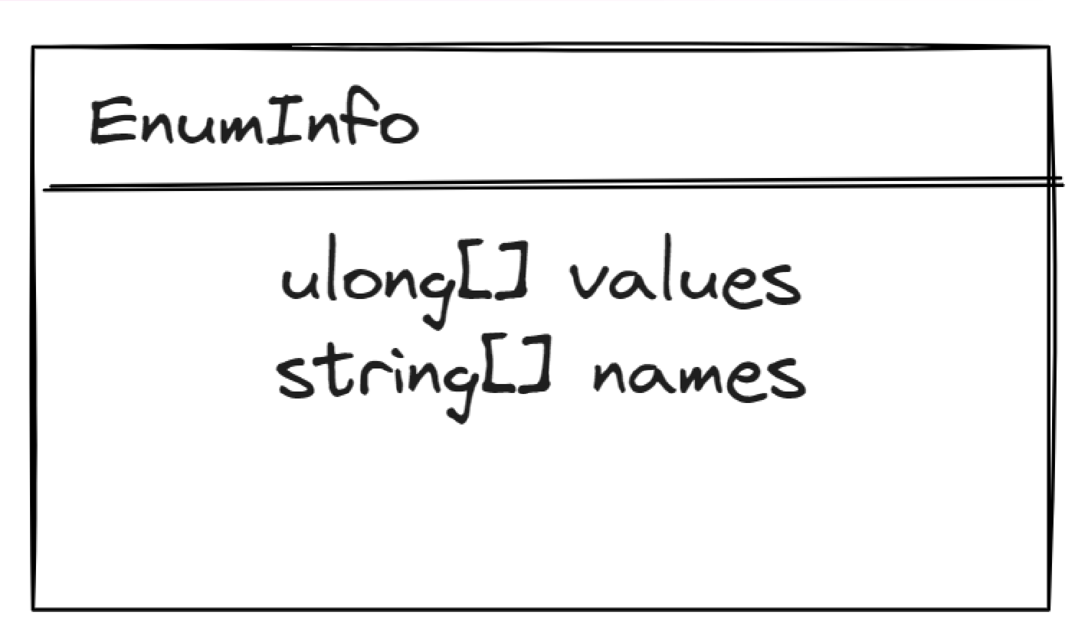
“First, do it,
Then, do it right,
Then, do it fast.”

- Addy Osmani

Dynamic PGO does its job pretty well.

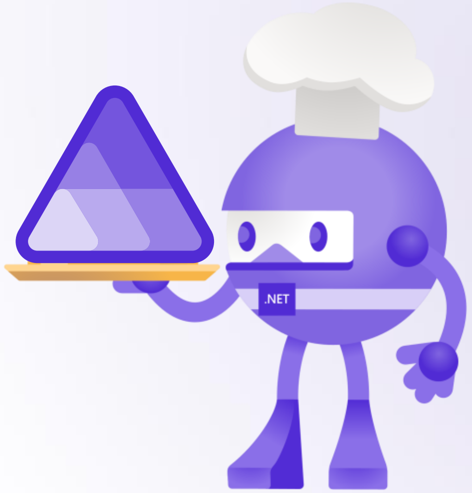
Make sure your benchmarks test what you intended them to test!

Enums

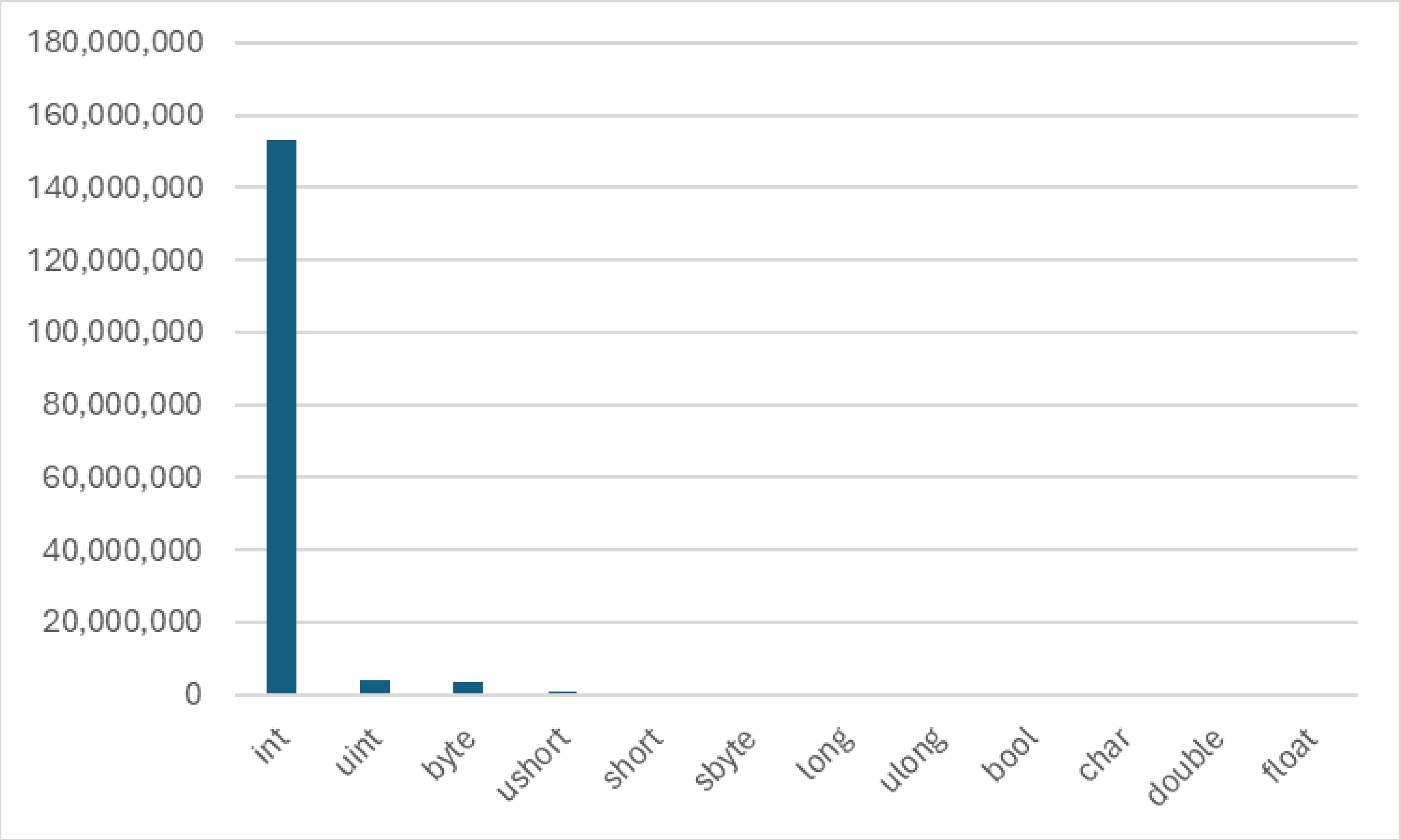


EnumInfo

ulong[] values
string[] names

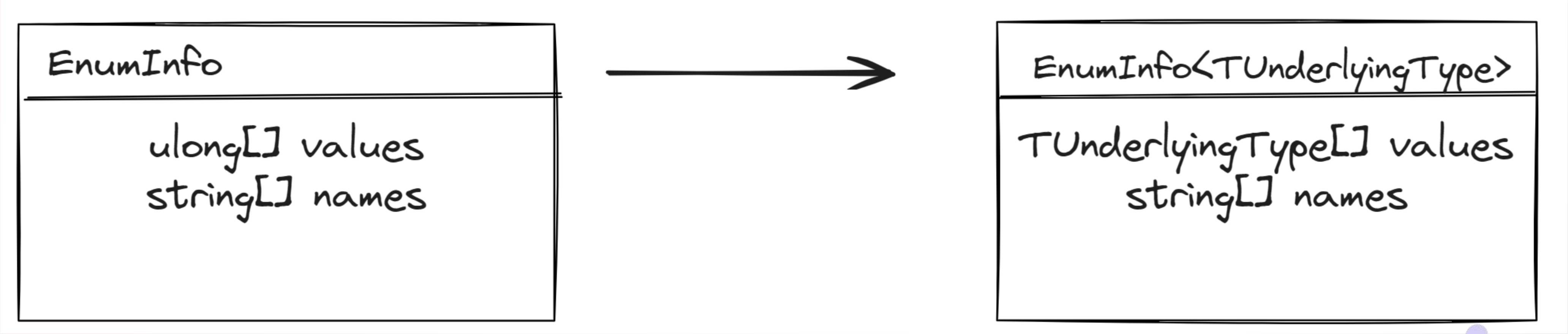


Enums


180,000,000
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
int
uint
byte
ushort
short
sbyte
long
ulong
bool
char
double
float

.NET

Enums

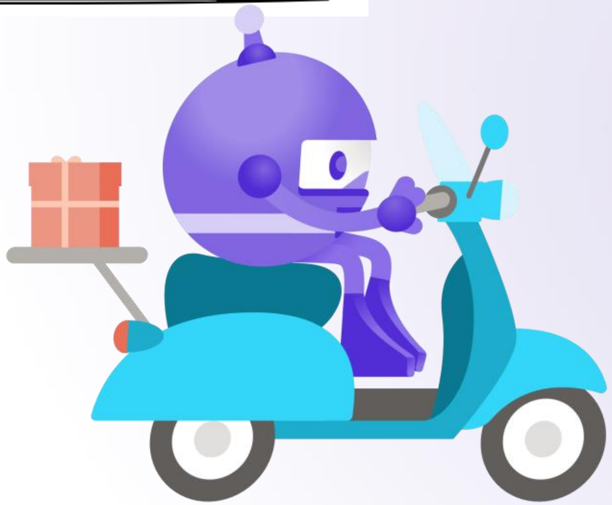


EnumInfo

ulong[] values
string[] names

EnumInfo<TUnderlyingType>

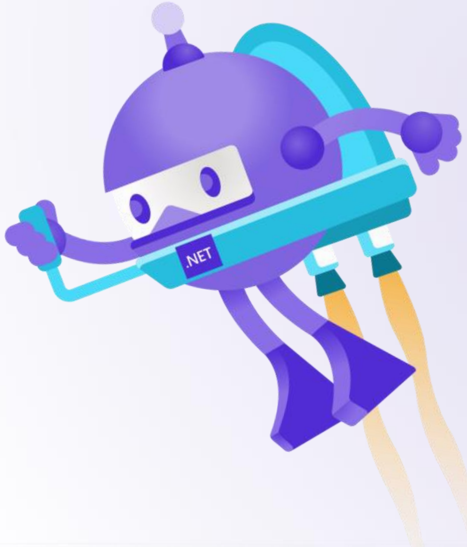
TUnderlyingType[] values
string[] names



Tasks / Async

Task

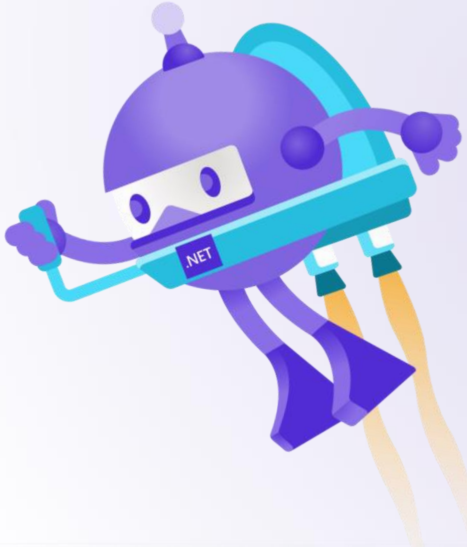
Task <TResult>



Tasks / Async

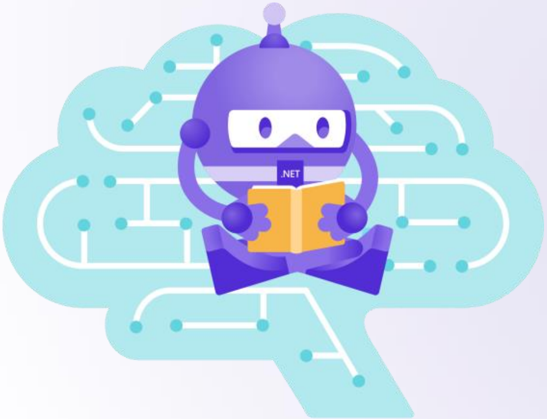
Cache empty default(T)

To be returned for sync completions



Tasks / Async

Reduce Async completion by 16 bytes

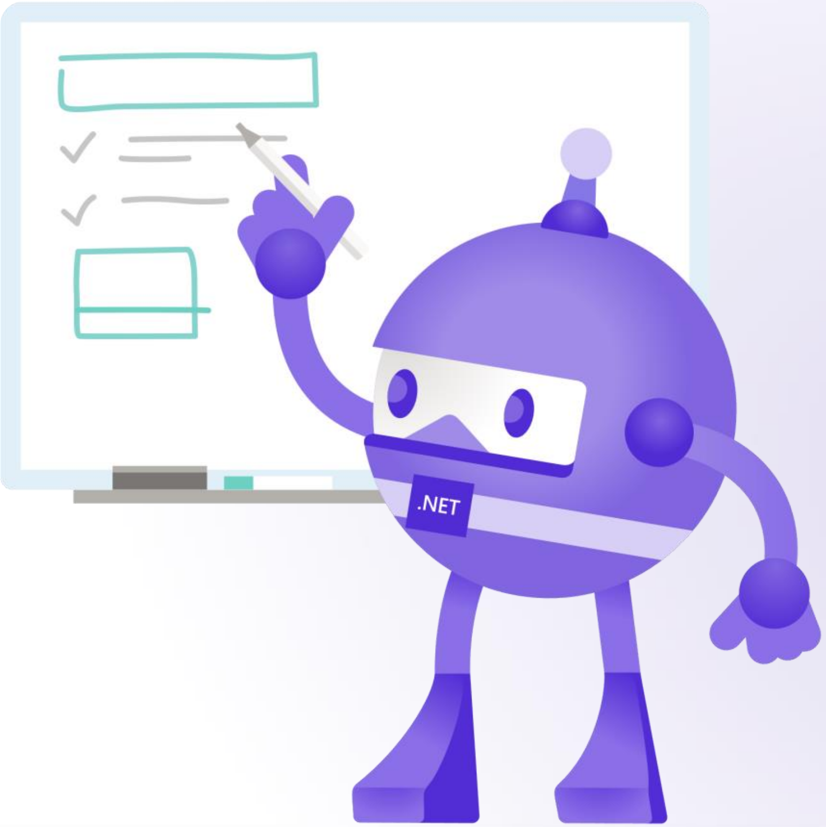


Frozen Collections

- FrozenDictionary<TKey, TValue>
- FrozenSet<Tkey, Tvalue>

JSON

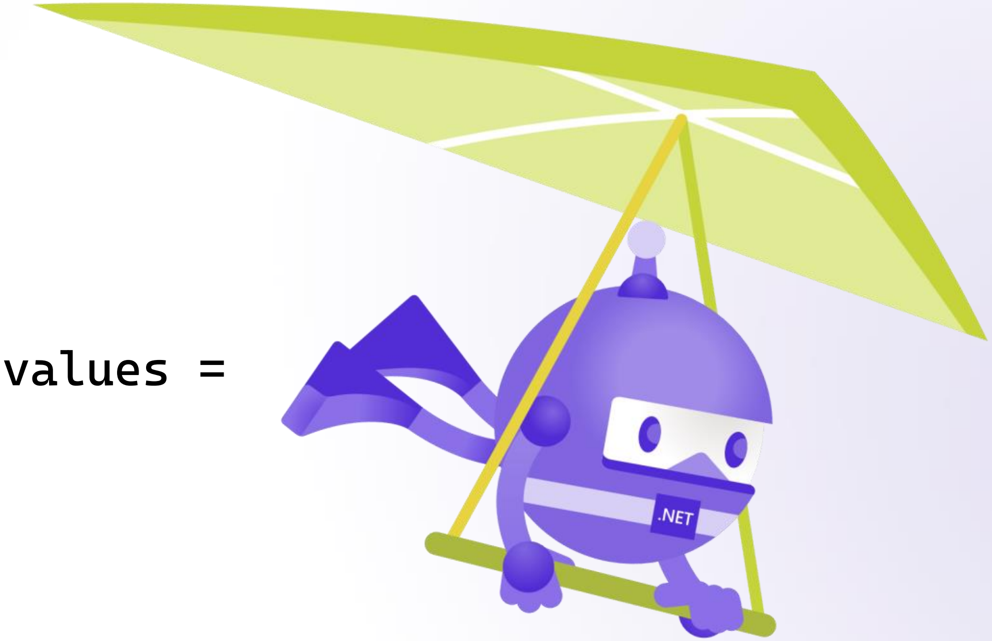
Now supporting Source Generators Fast Paths in Streaming/Async



SearchValues

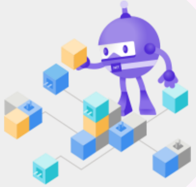
```
private static readonly char[] search_chars = new char[]
      { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' };
int pos = source.IndexOfAny(search_chars);
```

```
private static readonly SearchValues<char> search_values =
        SearchValues.Create("abcdefgh");
int pos = source.IndexOfAny(search_values);
```



We’ve covered:

01 Dynamic PGO



02 Enums

03 Async

04 Frozen Collections

05 JSON

06 SearchValues

Download .NET 8

https://aka.ms/get-dotnet-8