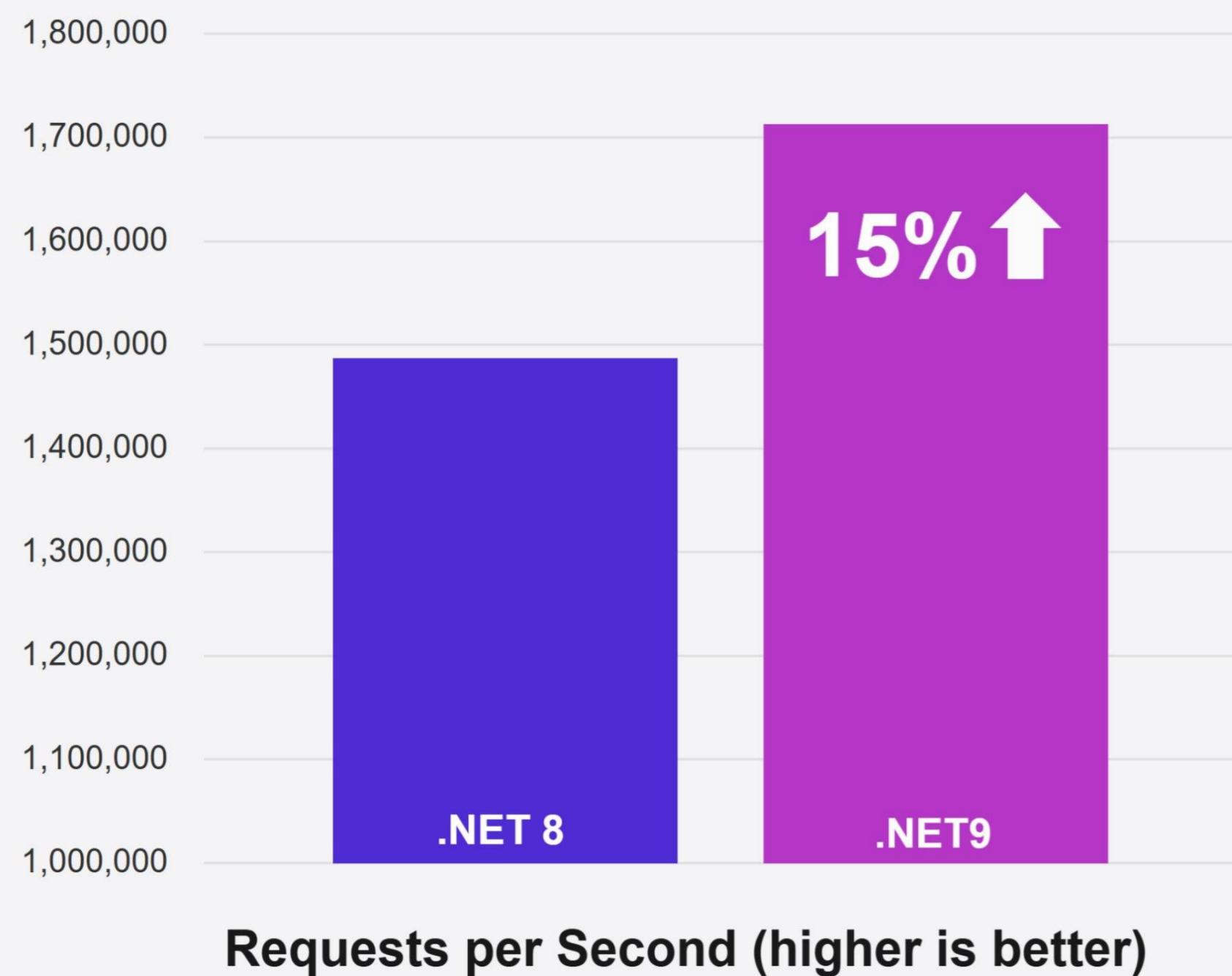


10 Reasons To Upgrade to .NET 9

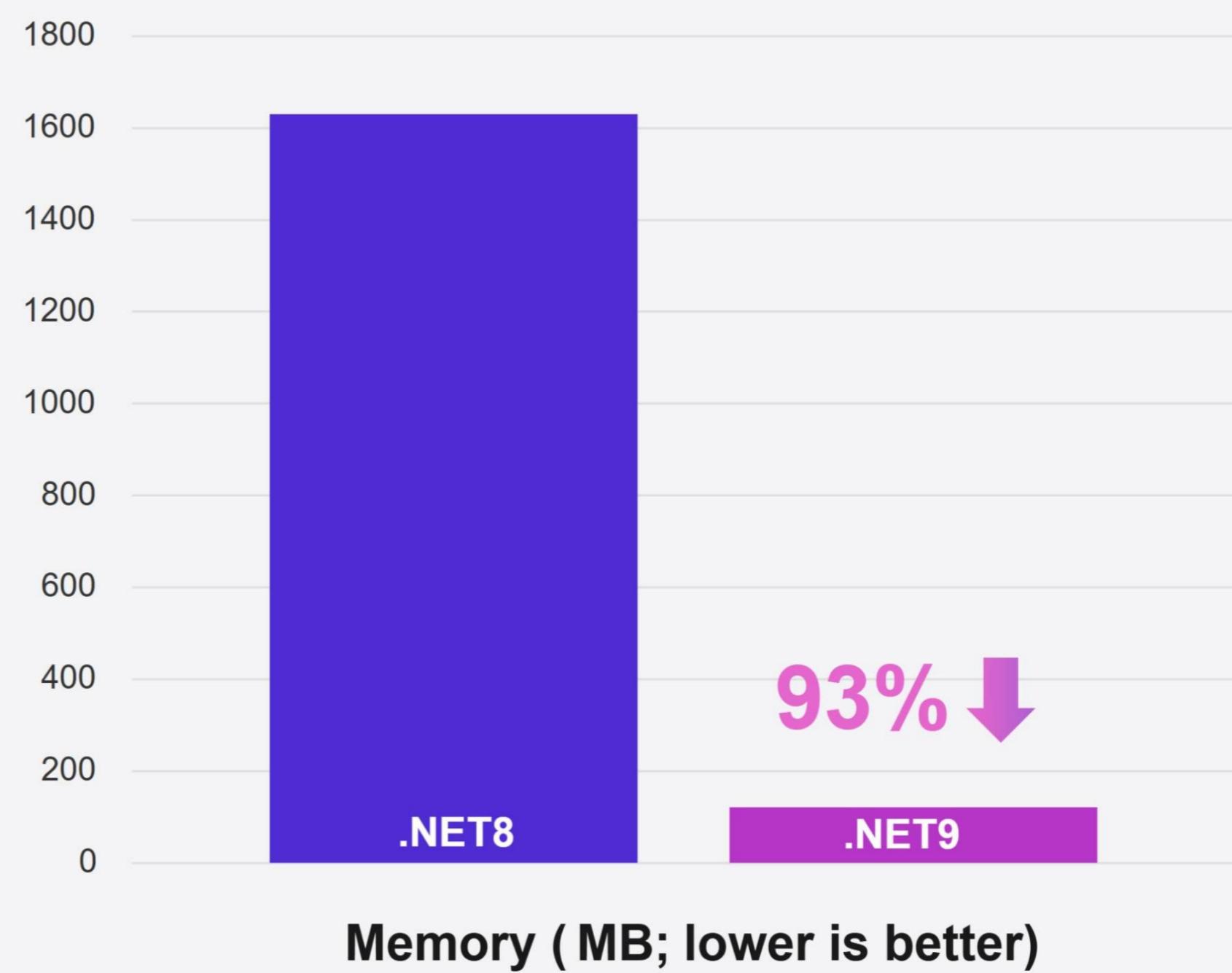
#1: Performance improvements of Minimal APIs

.NET 9 Minimal API Performance

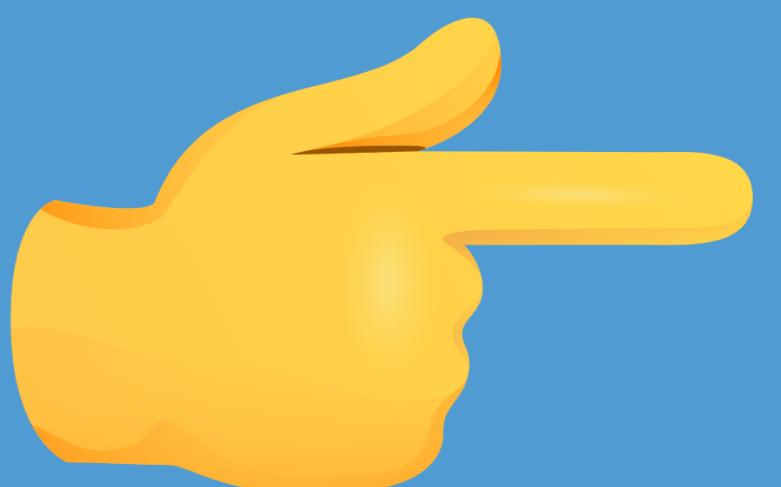
JSON Benchmarks



Intel Gold 56 cores (logical) Linux
Source: aka.ms/aspnet/benchmarks



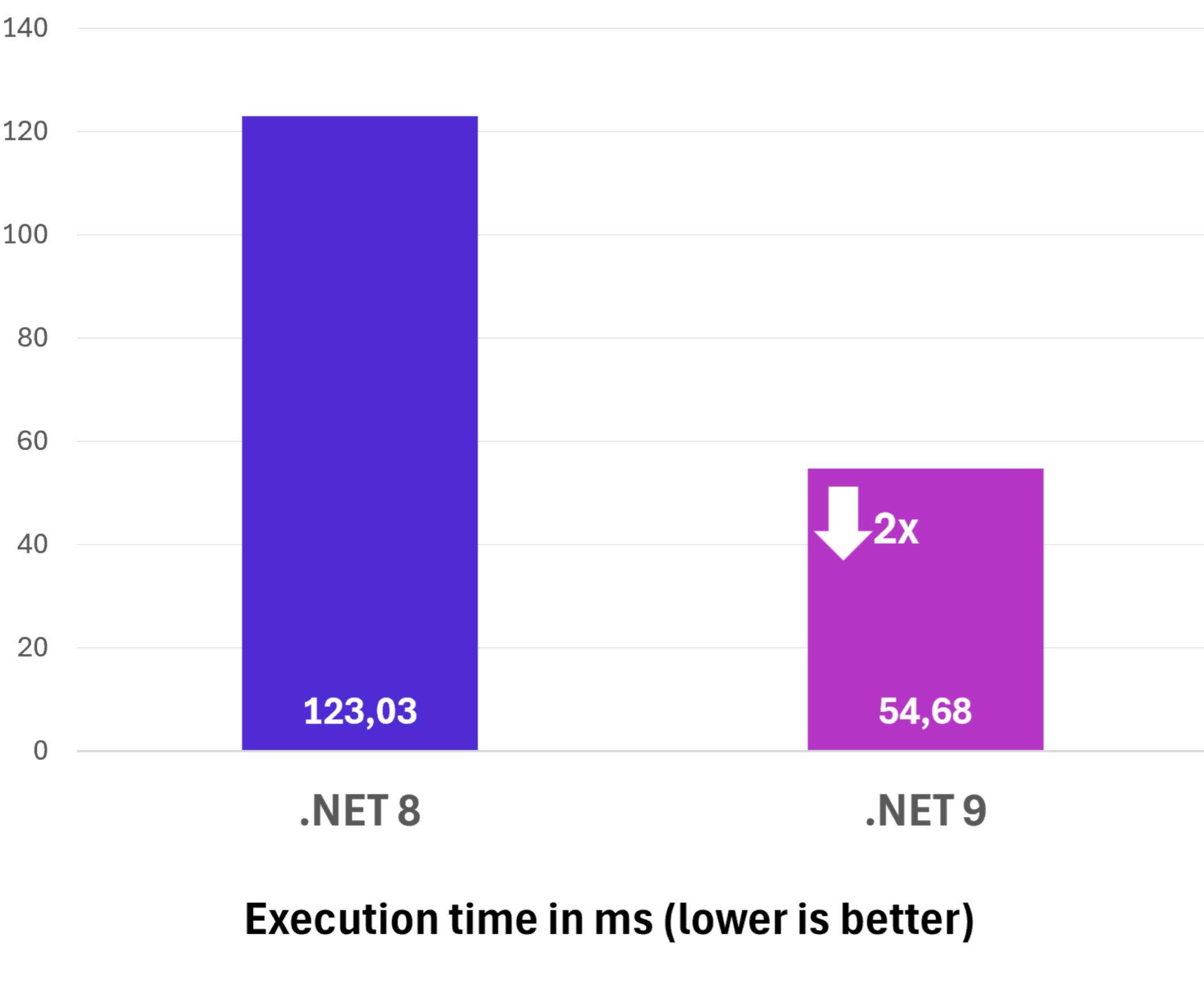
Anton Martyniuk
antondevtips.com



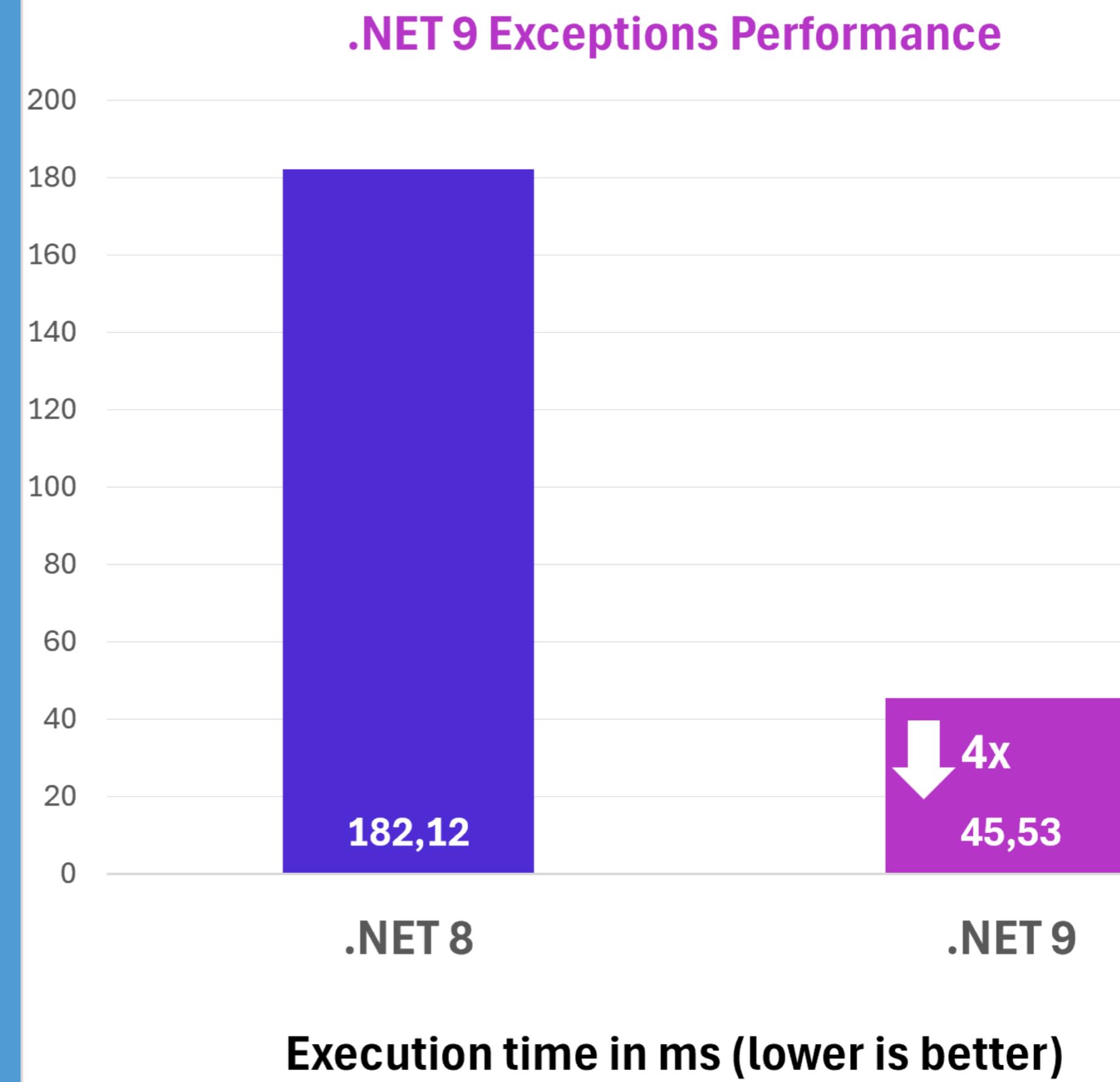
10 Reasons To Upgrade to .NET 9

#2: Performance improvements of Exceptions

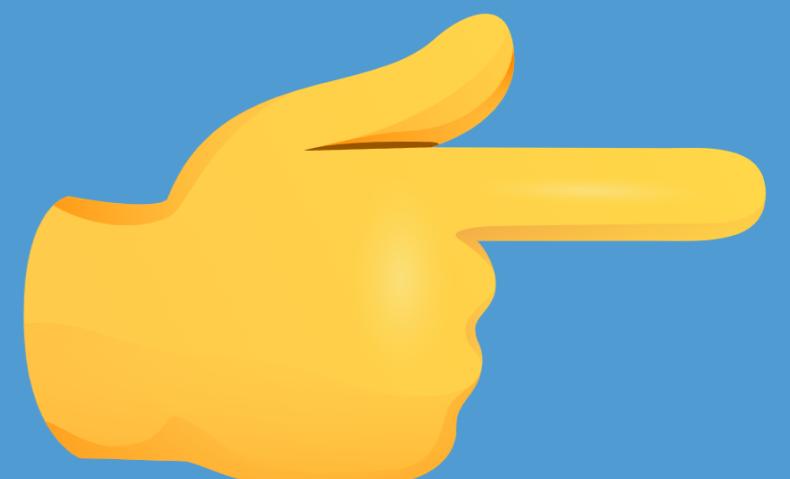
.NET 9 Exceptions Performance



.NET 9 Exceptions Performance



Anton Martyniuk
antondevtips.com



10 Reasons To Upgrade to .NET 9

#3: New Hybrid Cache

```
builder.Services.AddHybridCache();

[HttpGet("orders/{id}")]
public async Task<IActionResult> GetOrderAsync(int id,
    [FromServices] IHybridCache cache)
{
    string cacheKey = $"Order_{id}";
    var order = await cache.GetOrCreateAsync(cacheKey, async entry =>
    {
        entry.AbsoluteExpirationRelativeToNow = TimeSpan.FromMinutes(10);
        using var context = new AppDbContext();
        return await context.Orders.FindAsync(id);
    });

    if (order == null)
        return NotFound();

    return Ok(order);
}
```



Anton Martyniuk
antondevtips.com



10 Reasons To Upgrade to .NET 9

#4: New LINQ Methods

```
// 1. AggregateBy
var aggregateBy = orders.AggreegateBy(x => x.Price,
    x => 0, (x, y) => x + y.Price);

// 2. CountBy
foreach (var order in orders.CountBy(p => p.Name))
{
    Console.WriteLine($"Name: {order.Key}, Count: {order.Value}");
}

// 3. Index
foreach (var (index, item) in orders.Index())
{
    Console.WriteLine($"Order #{index}: {item}");
}
```



Anton Martyniuk
antondevtips.com



10 Reasons To Upgrade to .NET 9

#5: EF Core New Seeding method



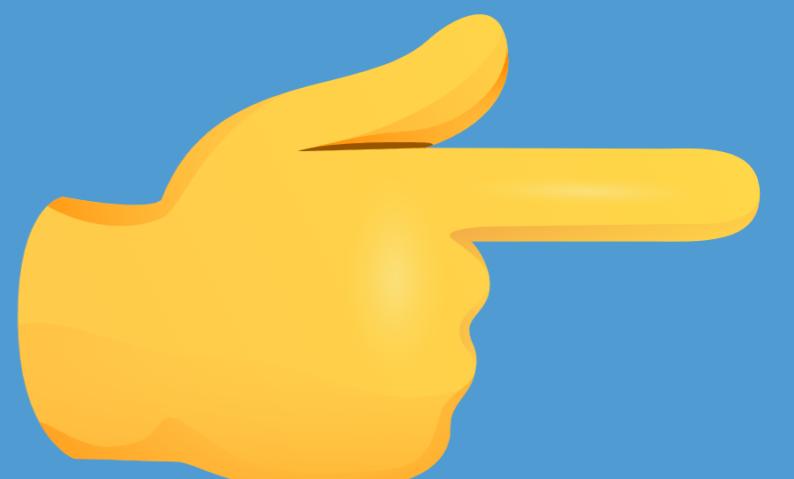
```
builder.Services.AddDbContext<ApplicationContext>((provider, options) =>
{
    options
        .UseNpgsql(connectionString)
        .UseAsyncSeeding(async (dbContext, _, cancellationToken) =>
    {
        var authors = GetAuthors(3);
        var books = GetBooks(20, authors);

        await dbContext.Set<Author>().AddRangeAsync(authors);
        await dbContext.Set<Book>().AddRangeAsync(books);

        await dbContext.SaveChangesAsync();
    });
});
```



Anton Martyniuk
antondevtips.com



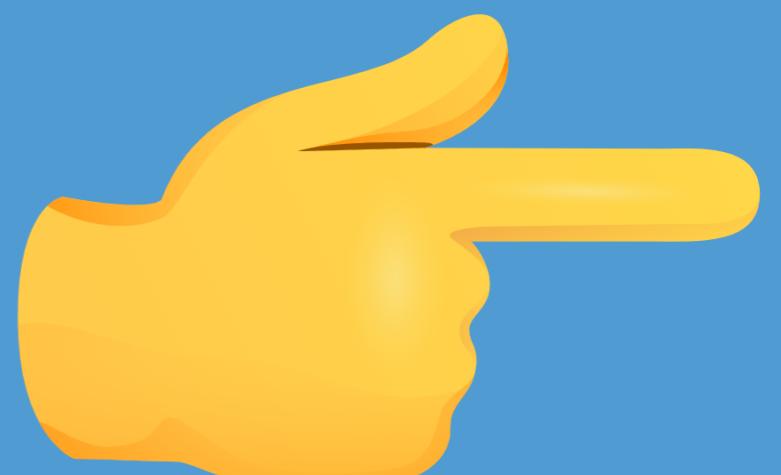
10 Reasons To Upgrade to .NET 9

#6: EF Core Better LINQ and SQL translation

```
// EF9 supports grouping by a complex type instance.  
var groupedAddresses = await context.Stores  
    .GroupBy(b => b.StoreAddress)  
    .Select(g => new { g.Key, Count = g.Count() })  
    .ToListAsync();  
  
// EF9 ExecuteUpdate has also been improved to accept complex type properties.  
var newAddress = new Address("Gressenhall Farm Shop", null, "Beetley");  
  
await context.Stores  
    .Where(e => e.Region == "Germany")  
    .ExecuteUpdateAsync(s => s SetProperty(b => b.StoreAddress, newAddress));  
  
// AND a LOT MORE
```



Anton Martyniuk
antondevtips.com



10 Reasons To Upgrade to .NET 9

#7: C# 13 Params Collection



Old Way with using only array in params

```
PrintNumbers(1, 2, 3, 4, 5);
```

```
public void PrintNumbers(params int[] numbers)
{
    ...
}
```



A new way, you can specify a list

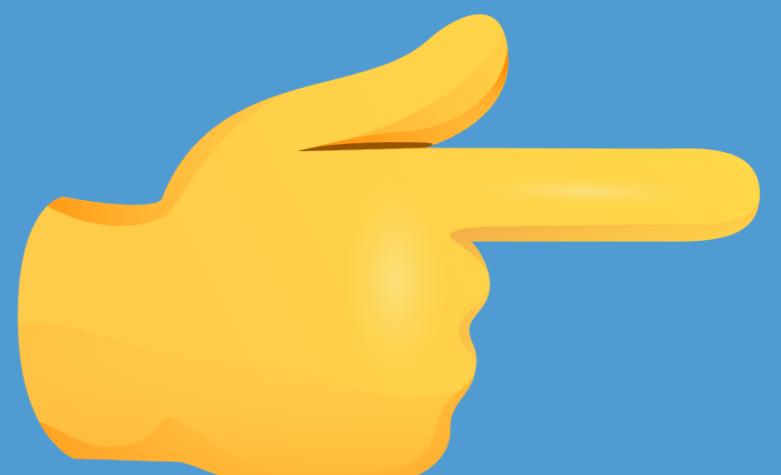
```
List<int>numbers = [ 1, 2, 3, 4, 5 ];
PrintNumbers(numbers);
```

```
public void PrintNumbers(params List<int> numbers)
{
    ...
}
```

A new way to use params

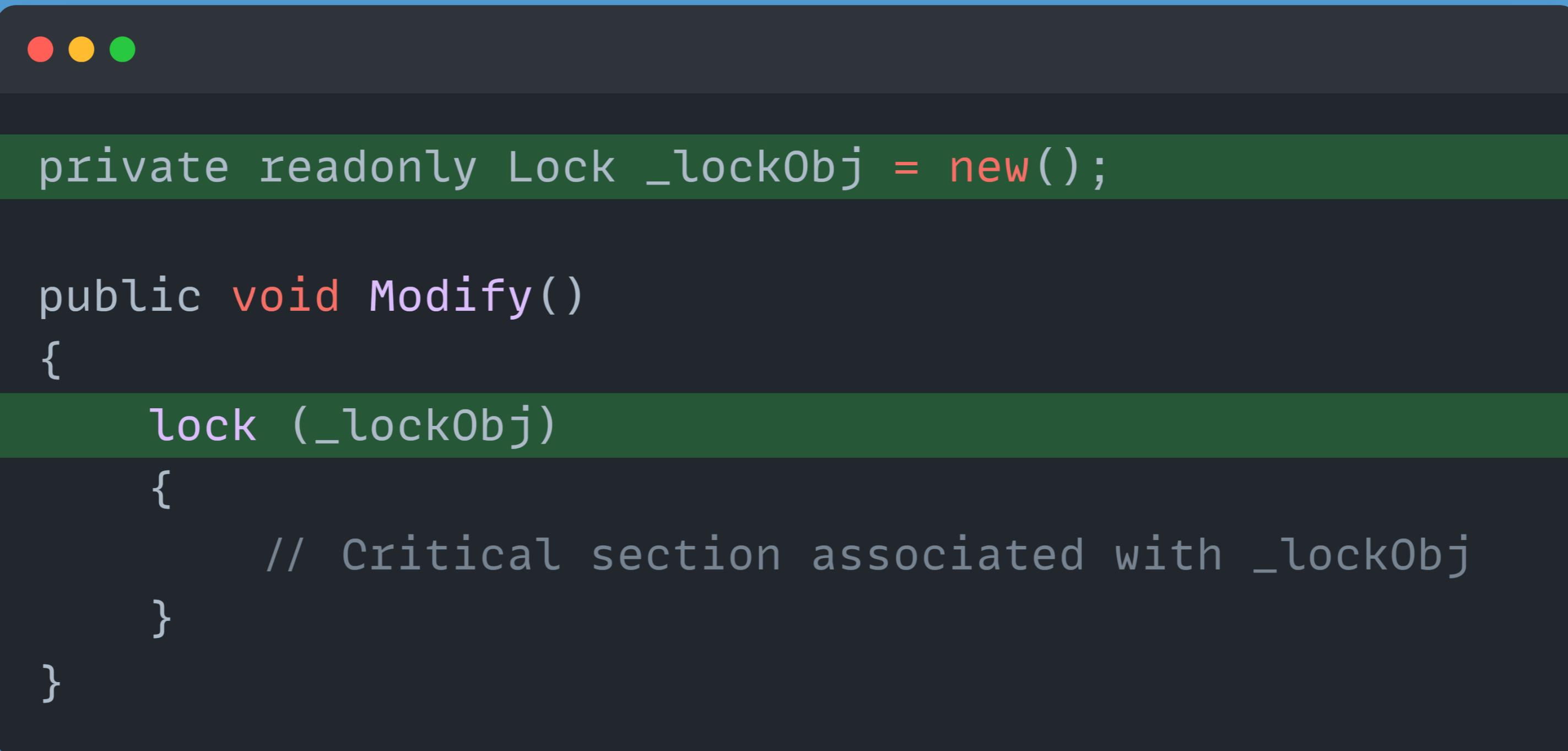


Anton Martyniuk
antondevtips.com



10 Reasons To Upgrade to .NET 9

#8: C# 13 New Lock Object



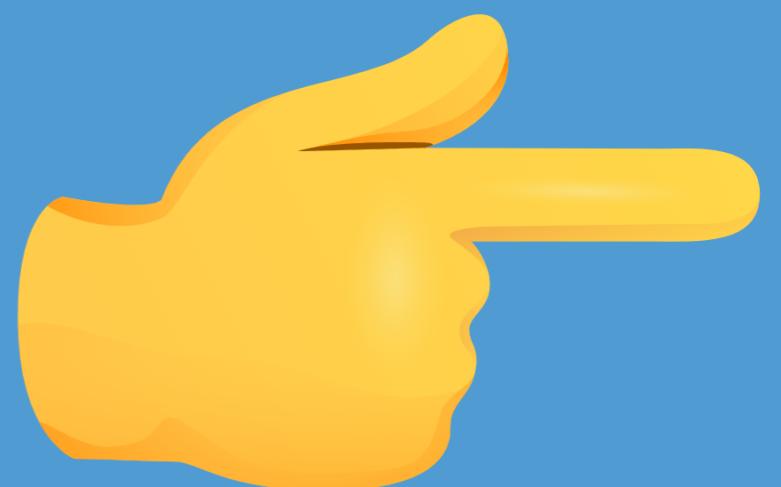
```
private readonly Lock _lockObj = new();

public void Modify()
{
    lock (_lockObj)
    {
        // Critical section associated with _lockObj
    }
}
```

Use new Lock class instead of object



Anton Martyniuk
antondevtips.com



#9: ASP.NET Core: Static asset delivery optimization

[expand] Expand table

File	Original	Compressed	% Reduction
bootstrap.min.css	163	17.5	89.26%
jquery.js	89.6	28	68.75%
bootstrap.min.js	78.5	20	74.52%
Total	331.1	65.5	80.20%

The following table shows the original and compressed sizes using the [Fluent UI Blazor components library](#):

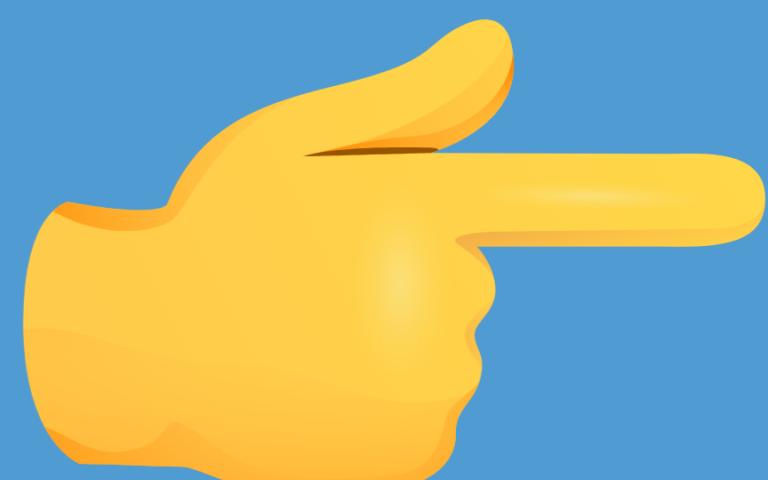
[expand] Expand table

File	Original	Compressed	% Reduction
fluent.js	384	73	80.99%
fluent.css	94	11	88.30%
Total	478	84	82.43%

For a total of 478 KB uncompressed to 84 KB compressed.



Anton Martyniuk
antondevtips.com





10 Reasons To Upgrade to .NET 9

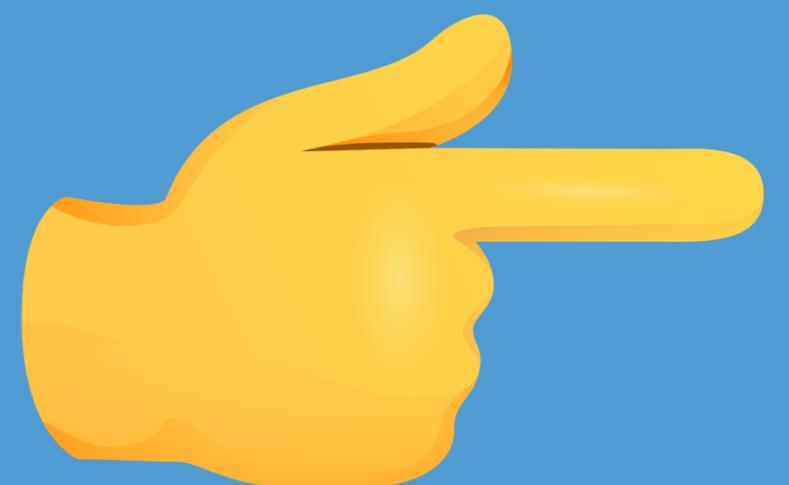
#10: Blazor Improvements



1. .NET MAUI Blazor Hybrid and Web App solution template
2. Detect rendering location, interactivity, and assigned render mode at runtime
3. Improved server-side reconnection experience
4. Simplified authentication state serialization for Blazor Web Apps
5. Add static server-side rendering (SSR) pages to a globally-interactive Blazor Web App
6. Constructor injection
7. Websocket compression for Interactive Server components
8. Handle keyboard composition events in Blazor



Anton Martyniuk
antondevtips.com





THANKS FO READING

LETS CONNECT ON LINKED-IN

REPOST TO HELP ME SHARE
USEFUL CONTENT FOR OTHERS

SUBSCRIBE TO ANTONDEVTIPS
TO GET MORE .NET TIPS



Anton Martyniuk
antondevtips.com

