Miroslav Holec

KONZULTANT PRO .NET

Big Minimal APIs v .NET 7

prezentace a dema

www.**odkaz.me/wug**

dotnetnews.cz & miroslavholec.cz

mirek@miroslavholec.cz

Kdo jsem

Miroslav Holec

software architect

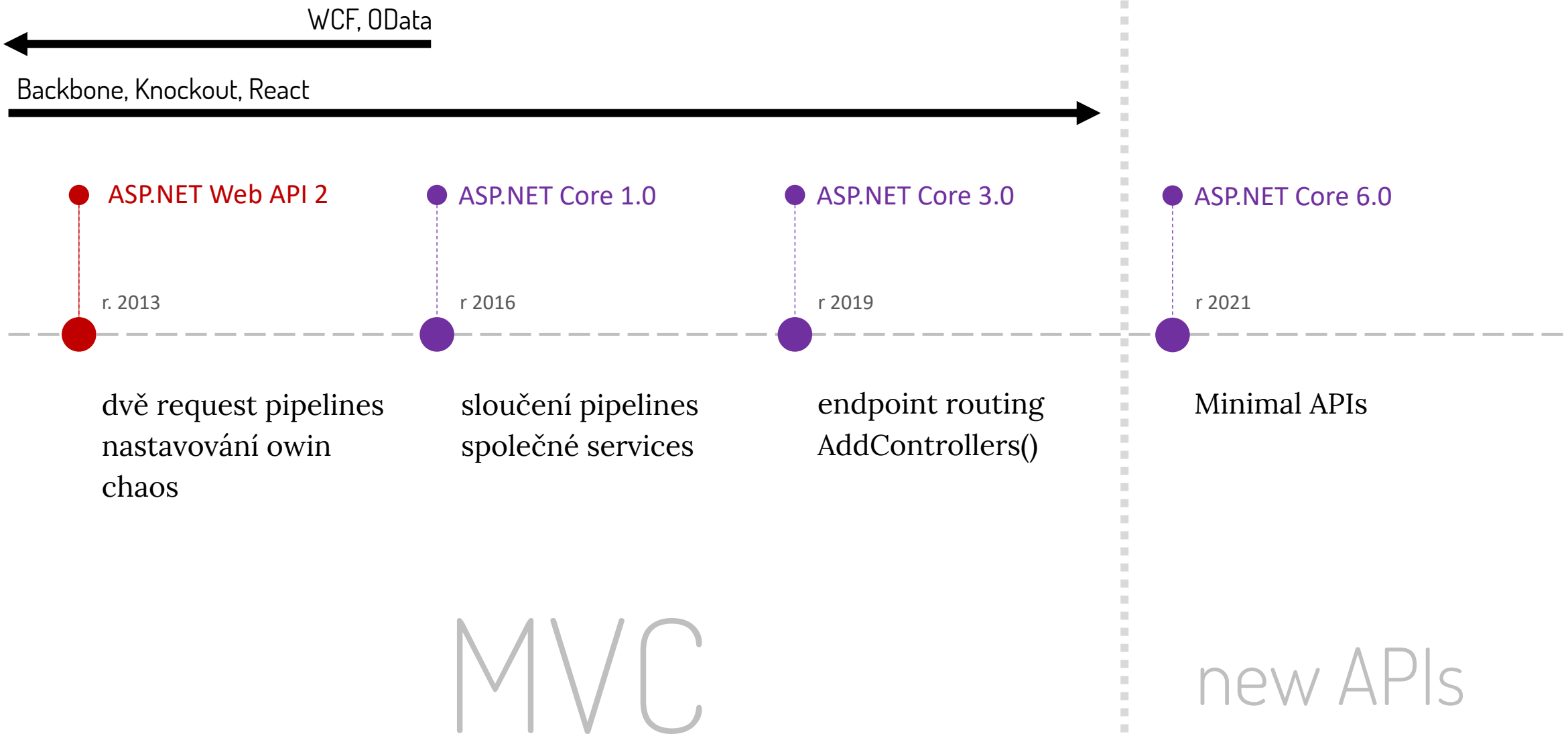
konzultant na volné noze

zaměření na platformu .NET

a návrh REST služeb

restapi.cz	průvodce designem REST API
dotnetnews.cz	zpravodaj pro .NET vývojáře, newsletter
grpc.cz	vývoj gRPC služeb
miroslavholec.cz	články, přednášky, videa
youtube.com/mirekholec	youtube videa

Evoluce REST API v.NETu



Minimal Hosting

- zjednodušená konfigurace webové aplikace (eliminace `Startup.cs`)
- nové třídy `WebApplication` a `WebApplicationBuilder`
- použití **top-level statements** pro „čitelnější“ `Program.cs`

```
WebApplicationBuilder builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();
builder.Logging.AddConsole();
builder.Configuration.AddIniFile("myconfig.ini")

WebApplication app = builder.Build();

app.UseMiddleware<MyMiddleware>();
app.MapControllers();
app.Run();
```

ConfigureServices

Configure

Třída WebApplication

.NET 5

```
public void Configure(IApplicationBuilder app)
{
    app.UseRouting();
    app.UseMiddleware<MyMiddleware>();
    app.UseEndpoints(routes => {
        routes.MapControllers();
    }
}
```

IEndpointRouteBuilder

.NET 6

```
WebApplication app = builder.Build();

app.UseMiddleware<MyMiddleware>();
app.MapControllers();
```

Minimal APIs

- top-level statements (C# 9)
- implicit & global usings (C# 10)
- řada nových API nad různými třídami (.NET 6)

inferred lambda types

```
Func<string, string> func = (string name) => $("Hello {name}");
```



```
var func = (string name) => $("Hello {name}");
```

atributy nad lambda types

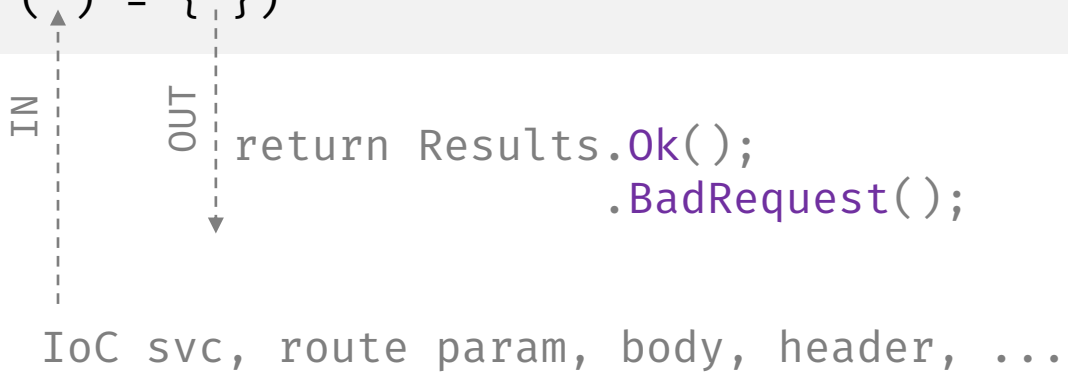
```
var func = ([FromQuery]string name) => $("Hello {name}");
```

Minimal APIs

- rozšíření nad třídou `WebApplication`
- umožňují definovat si API endpointy na jednom místě v `Program.cs`
- nepoužívají MVC pipeline... pouze dílčí části (např.: routing, binding)

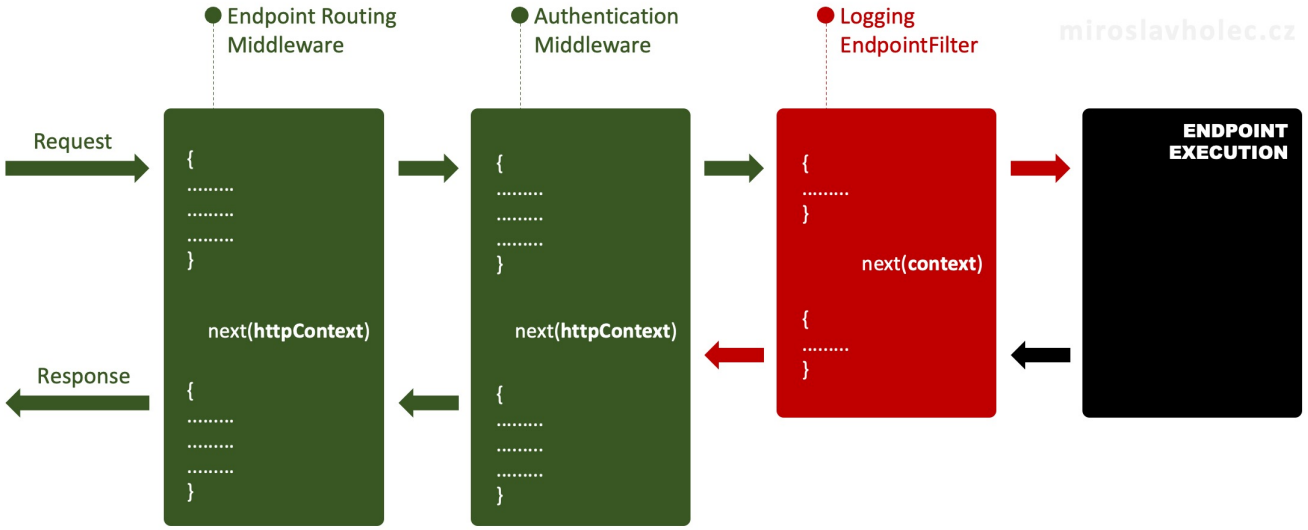
```
WebApplication app = builder.Build();

app.MapGet("path/{id:int}"), async ( ) => { await x.y(); }
app.MapPost
app.MapDelete
app.MapPut
app.MapMethods("path", new []{ "HEAD", ( ) = { })
```



Proč Minimal APIs

- transparentní request pipeline
- čistější návrh aplikace
- lepší výkonnost aplikace
- minimalističnost a rozšiřitelnost
- budoucnost a nové funkce




```
ProductsController
    ctor: Service1 svc1
          Service2 svc2
          Service3 svc3

 GetArchive() { svc1 }
 GetBooks()   { svc2 }
 GetAuthors() { svc3 }
```

```
GetArchive(Service1) {}
GetBooks(Service2) {}
GetAuthors(Service3) {}
```

Novinky v .NET 7

- **groups** – sdružování endpointů do skupin a podskupin
- **filters** - sdílení logiky pro groups nebo endpoints

- binding pole primitivních typů
- speciální atribut AsParameters
- rozšířené extensions pro Open API
- nové přetížení Results.Stream
- injection pro body Stream a PipeReader
- IFormFile a IFormFileCollection

```
tags?q=1&q=2  =>  ([FromQuery]int[] q)
```

```
[AsParameters]MyFilter filter
```

```
WithOpenApi, WithDescription, WithSummary
```

```
return Results.Stream()
```

```
async(HttpRequest req, Stream body)
```

```
("/upload", async (IFormFile file) => {} );
```

Co Minimal APIs neumí

- **content negotiation**
  - řešením je vytvořit vlastní IResult, který vrátí požadovaný typ
- **automatická validace**
  - řešením je použití fluent validation
- **spojení swagger attributes + fluent validation**
  - řešením je přístup Design First

DEMO

Zvažované funkce pro .NET 8

- generování REST klientů pro Minimal APIs (něco ála Refit)
- větší kontrola nad chybovými hláškami při použití TryParse, BindAsync
- zvažuje se podpora [FromQuery] pro objekty, souvislost s [AsParameters] nejasná
- konfigurace autorizačních policies přes IConfiguration (appsettings.json)
- hlubší podpora Open API, automatické nastavení metadat (zřejmě pomocí reflexe)
- podpora pro FormData a komplexní typy (file + parameters)

ŠKOLENÍ

Big Minimal APIs v .NET 7

Naučíte se vytvářet tradiční REST API s pomocí nového deklarativního přístupu Minimal APIs. Vedle základních scénářů se můžete těšit i na tipy pro vytváření infrastruktury velkých API.

- Nastavení prostředí a seznámení s šablonou projektu
- Konfigurace projektu a moduly ASP.NET Core
- Pravidla pro routing a constraints
- Deklarativní zápis pro handlování HTTP požadavků
- Seskupování endpointů a globální filtry
- Návratový typ IResult a vytváření vlastních typů
- Binding v Minimal APIs: routes, query, headers, body...
- Implementace CRUD operací s využitím HTTP metod
- Validační mechanismus Fluent Validation
- Globální správa výjimek pomocí MVC filtrů a middlewares
- Doporučení pro strukturování projektu a kontraktů
- Zapojení balíčku MediatR, Pipeline behavior a async operace
- Generování dokumentace Swashbuckle Swagger
- CORS, Response Caching a další funkcionality

www.**odkaz.me/minimal**

Miroslav Holec

KONZULTANT

mirek@miroslavholec.cz

miroslavholec.cz

DOTAZY