

# Get the maximum out of Minimal APIs with ASP.NET Core

Maarten Balliauw  
@maartenballiauw

**JET  
BRAINS**



# Agenda

- What are Minimal APIs?
- Mapping requests
- Validation & model binding
- Link building / named routes
- Responses
- OpenAPI
- Testing
- Authentication/Authorization
- Everything together

What are Minimal APIs?

# Minimal APIs

**"What would it mean to try and simplify starting with ASP.NET core? "**

- Easy to learn & get started
- Reduced complexity, minimal project layout
- Fast development & fast feedback loop

# Hello, World

Demo

# Minimal APIs

- New in .NET 6
- Minimal project layout, minimal code needed to get started
- Minimal != Simple
- Nothing is deprecated!
  - `IServiceCollection`
    - Routing
    - MVC
    - Razor pages
    - Web API
    - ...

# Mapping requests

Demo

# Mapping requests

- Use the `Map*` methods on `WebApplication`
- Delegate logic to
  - Anonymous lambda handler
  - Request delegate
    - Inject `HttpContext`, `HttpRequest`, `HttpResponse`, `ClaimsPrincipal`, `CancellationToken`
  - Local or static function
  - Instance function (register it in DI container)
- Support for dependency injection, route parameters, ...



# Validation & model binding

Demo

# Validation

- No `ModelState.IsValid...`
- You're closer to HTTP, return 400 Bad Request / 200 OK yourself
- Validation frameworks
  - Roll your own
  - Use [FluentValidation](#)
  - Use [MiniValidation](#)

# Model binding

- Out of the box:
  - Route values/query string
  - Header
  - Body (as JSON)
  - Services provided by dependency injection
- Custom:
  - When context is needed:

```
static ValueTask<T?> BindAsync<T>(
    HttpContext httpContext, ParameterInfo parameter)
```
  - When value is a string and you need an object from it

```
static bool TryParse<T>(string? value,
    IFormatProvider? provider, out T parameter)
```

# Link building / named routes

Demo

# Link building / named routes

- Endpoint names:  
app.`MapGet(...)`  
    `.WithName("endpointName")` (or convention based)
- Generate URL to endpoint name:
  - Inject `HttpContext` and `LinkGenerator` into handler method
  - Call `GetUriByName(http, "endpointName", new { ... })`

# Responses

Demo

# Responses

- Writes `string` as text/plain
- Writes `object` as application/json (using default serializer)
- Implement custom response type with `ActionResult`
  - Built-in:
    - `Text, Json, File, Bytes`
    - Status codes
    - Redirects
    - `Problem, ValidationProblem`
    - `SignIn/SignOut`
  - Extension methods go on `ActionResultExtensions`

# OpenAPI

Demo



# OpenAPI

- Add OpenAPI specification using [Swashbuckle](#)
  - `builder.Services.AddEndpointsApiExplorer();`
  - `builder.Services.AddSwaggerGen();`
  - `app.UseSwagger();`
  - `app.UseSwaggerUI();`
- Add details about operations
  - `.WithTags("sometag")`
  - `.Produces<SomeClass>(200)`
  - `.Produces(404)`
  - `.Accepts("application/json")`
  - `.ExcludeFromDescription()`

# Testing

Demo

# Testing

- Project SDK: `<Project Sdk="Microsoft.NET.Sdk.Web">`
- Reference NuGet package: [Microsoft.AspNetCore.Mvc.Testing](#)
- Use `WebApplicationFactory<T>`
  - Overrides for injecting services, customizing client, ...

# Authentication/Authorization

Demo

# Authentication/Authorization

- Same as in any other ASP.NET Core application
- `[Authorize]`
- `[Authorize(policy)]`
- `.RequireAuthorization()`
- `.RequireAuthorization(policy)`

# Everything together!

Demo

# Closing thoughts

- Easy to learn & get started
- Minimal project layout
  - If needed, structure with language features (extension methods etc.)
  - Wish list: top-level functions to remove the `static class`
- Be aware `ActionResult` and `HttpActionResult` are not compatible
  - Snapshot testing to the rescue - <https://github.com/VerifyTests/Verify>
- Yet another ASP.NET option

# Thank you!

<https://blog.maartenballiau.be>  
@maartenballiau

