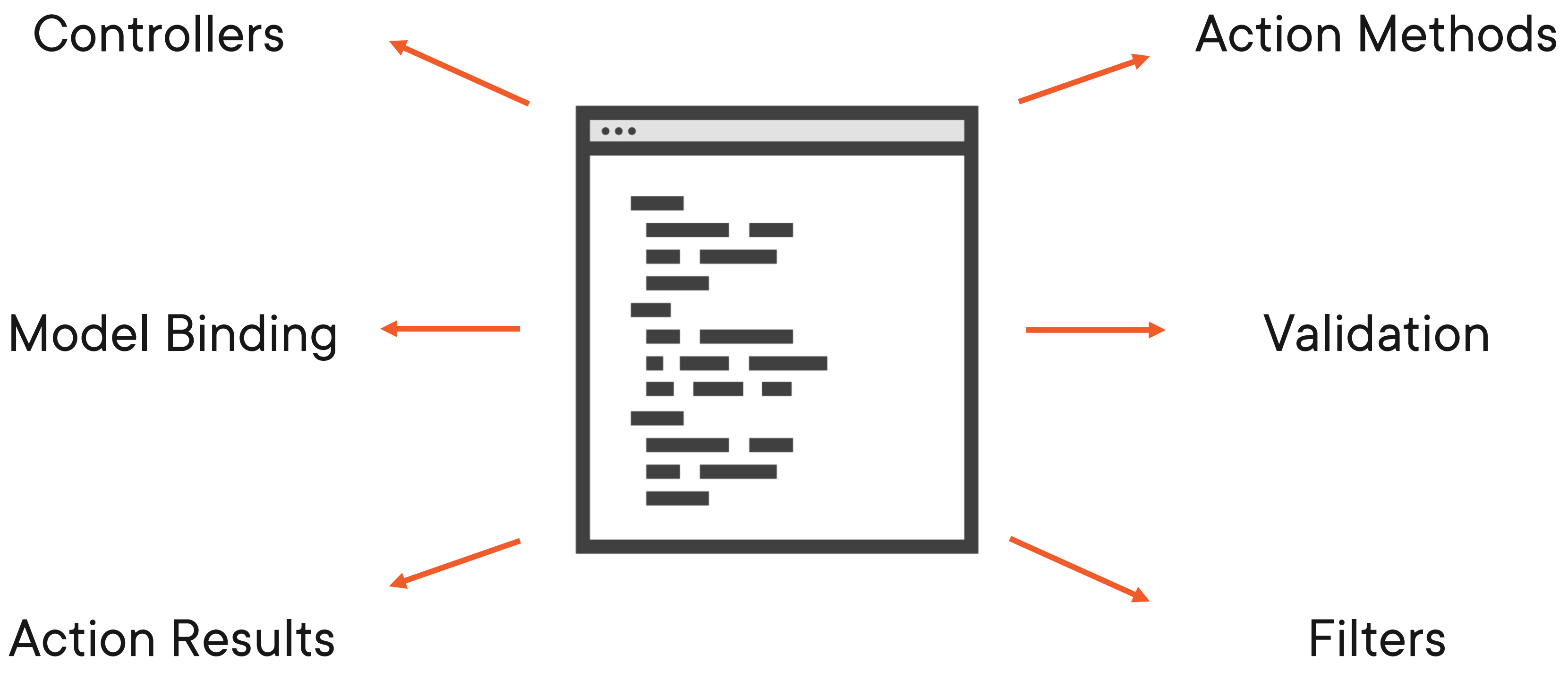
Introducing Minimal APIs

Alex Wolf
.NET Developer

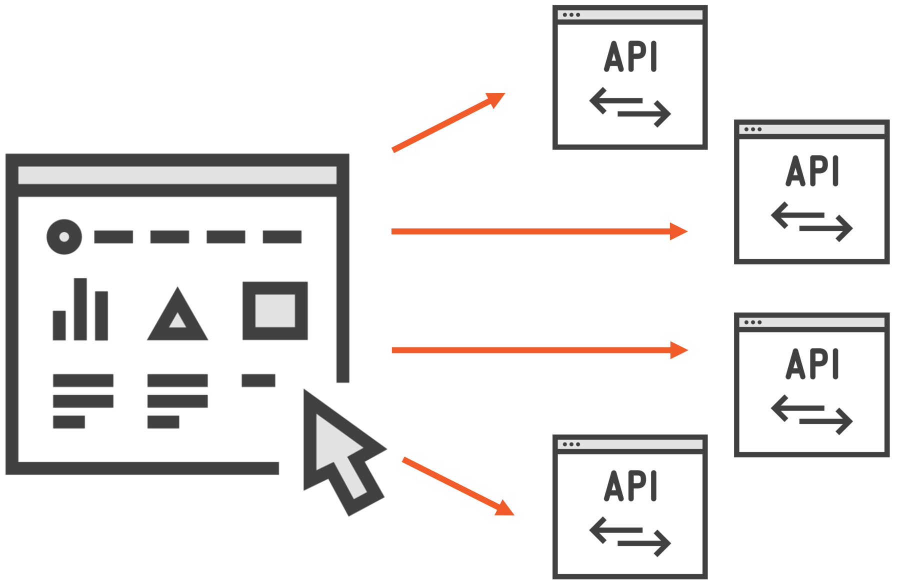
www.thecodewolf.com

Revisiting ASP.NET Web APIs

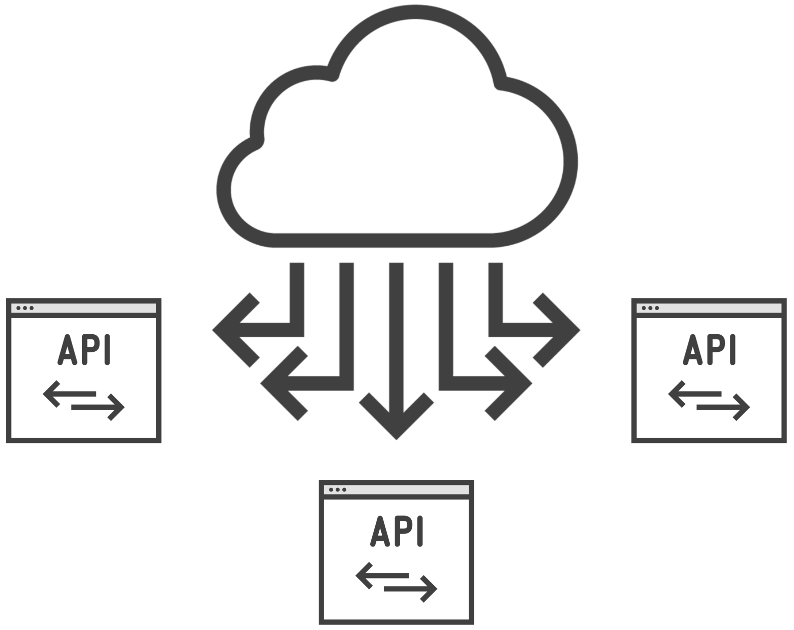


The Evolution of Web Services

Microservice architecture



Cloud expansion



Options for Building Web Services

JavaScript

JS

Python

PY

ASP.NET

C#

Minimal APIs

A pattern for building ASP.NET web services with minimalistic dependencies, syntax, and project structures.

```
using Microsoft.AspNetCore.Mvc;

namespace WiredBrainCoffee.Controllers
{
    private IMenuService service { get; set; }

    [ApiController]
    [Route("[controller]")]
    public class MenuController()
    {
     public MenuController(IMenuService
     service)
     {
          this.service = service;
     }

        [HttpGet(Name = “GetMenu")]
        public IActionResult Get()
        {
            return Ok(new MenuItem());
        }

    }

}
```

◀ Using statements

◀ Namespace

◀ Controller

◀ Constructor

◀ Action method

◀ Action result

◀ Lots of curly braces

```
app.MapGet("/menuItem/{id}", (int id,
MenuService service) =>
{
    return service.GetMenuItemById(id);
})
```

- Minimal API method syntax
- Delegate or lambda expression

Minimal API Features

- Middleware
- Dependency injection
- Parameter binding
- Swagger
- Result abstractions
- Cors

Minimal APIs Limitations

Limited binding features

No parameter validation

No view templates

No filters

No API versioning

Limited file structure

Prefer traditional Web APIs?

Keep at it! (but give Minimal APIs a try)

Demo

Creating a simple endpoint

Demo

Implementing dependency injection

Demo

Building CRUD operations to manage data

Demo

Improving the response objects

Demo

Working with HTTP and async requests

Overview/ Summary

- Minimal APIs provide a lightweight alternative to traditional .NET Web APIs
- They minimize boilerplate code by using modern language and framework features
- Methods like MapGet and MapPost bind incoming requests to handler methods
- Minimal APIs can bind values from the request to populate parameters
- They support Middleware pipelines and most related .NET features
- Dependency injection is fully supported in Minimal APIs
- Minimal APIs also support response abstractions to streamline request handling

Thank you for watching…
and good luck!