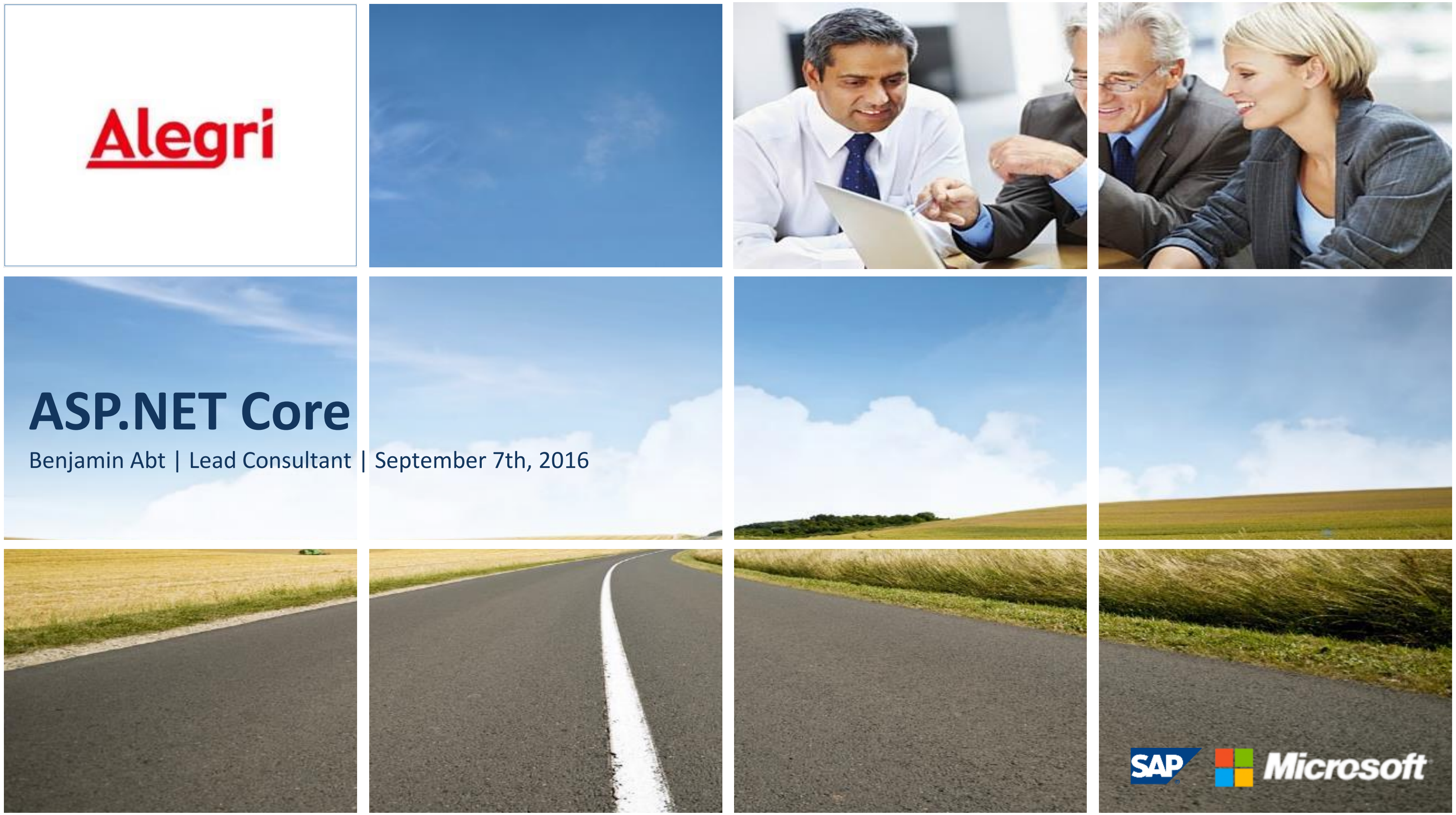


ASP.NET Core

Benjamin Abt | Lead Consultant | September 7th, 2016

Agenda

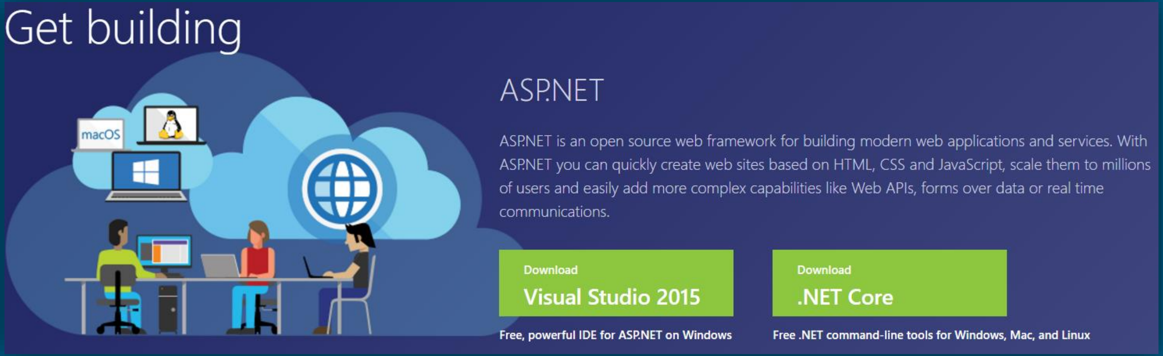
- What is ASP.NET Core?
- Anatomy
  - Startup (aka Bootstraper)
  - Built-in Dependency Injection
  - Middleware
  - Web Server
- Demos

Alegri

ASP.NET Core is the latest version of the ASP.NET family

www.asp.net

Get building



ASP.NET

ASP.NET is an open source web framework for building modern web applications and services. With ASP.NET you can quickly create web sites based on HTML, CSS and JavaScript, scale them to millions of users and easily add more complex capabilities like Web APIs, forms over data or real time communications.

Download
Visual Studio 2015
Free, powerful IDE for ASP.NET on Windows

Download
.NET Core
Free .NET command-line tools for Windows, Mac, and Linux

Based on .NET Core. Run it on..
- ✓ Windows
- ✓ macOS
- ✓ Linux (incl. Docker!)

.. can also run on the full .NET Framework.

Develop with
- ✓ Visual Studio
- ✓ Visual Studio Code
- ✓ Atom
- ✓ Any other editor and CLI

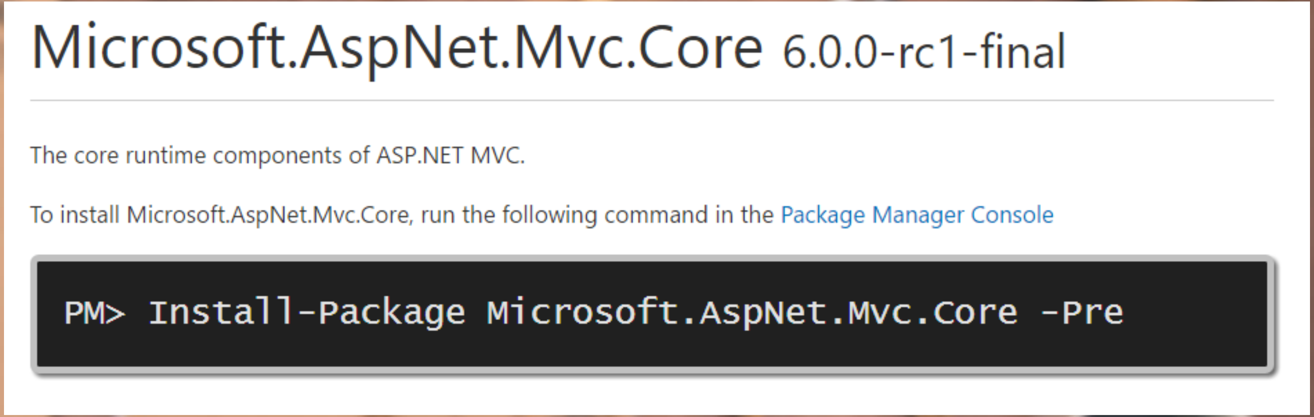
It is open source (MIT License) on GitHub!

Contributes are welcome ☺

Alegri

Anatomy facts

ASP.NET Core dependencies are shipped with NuGet



Microsoft.AspNet.Mvc.Core 6.0.0-rc1-final

The core runtime components of ASP.NET MVC.

To install Microsoft.AspNet.Mvc.Core, run the following command in the Package Manager Console

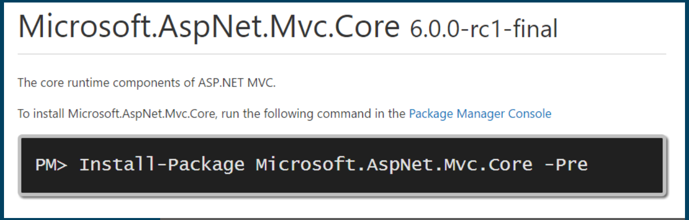
```
PM> Install-Package Microsoft.AspNet.Mvc.Core -Pre
```

https://www.nuget.org/packages/Microsoft.AspNet.Mvc.Core/

Alegri

Anatomy facts

ASP.NET Core dependencies are shipped with NuGet



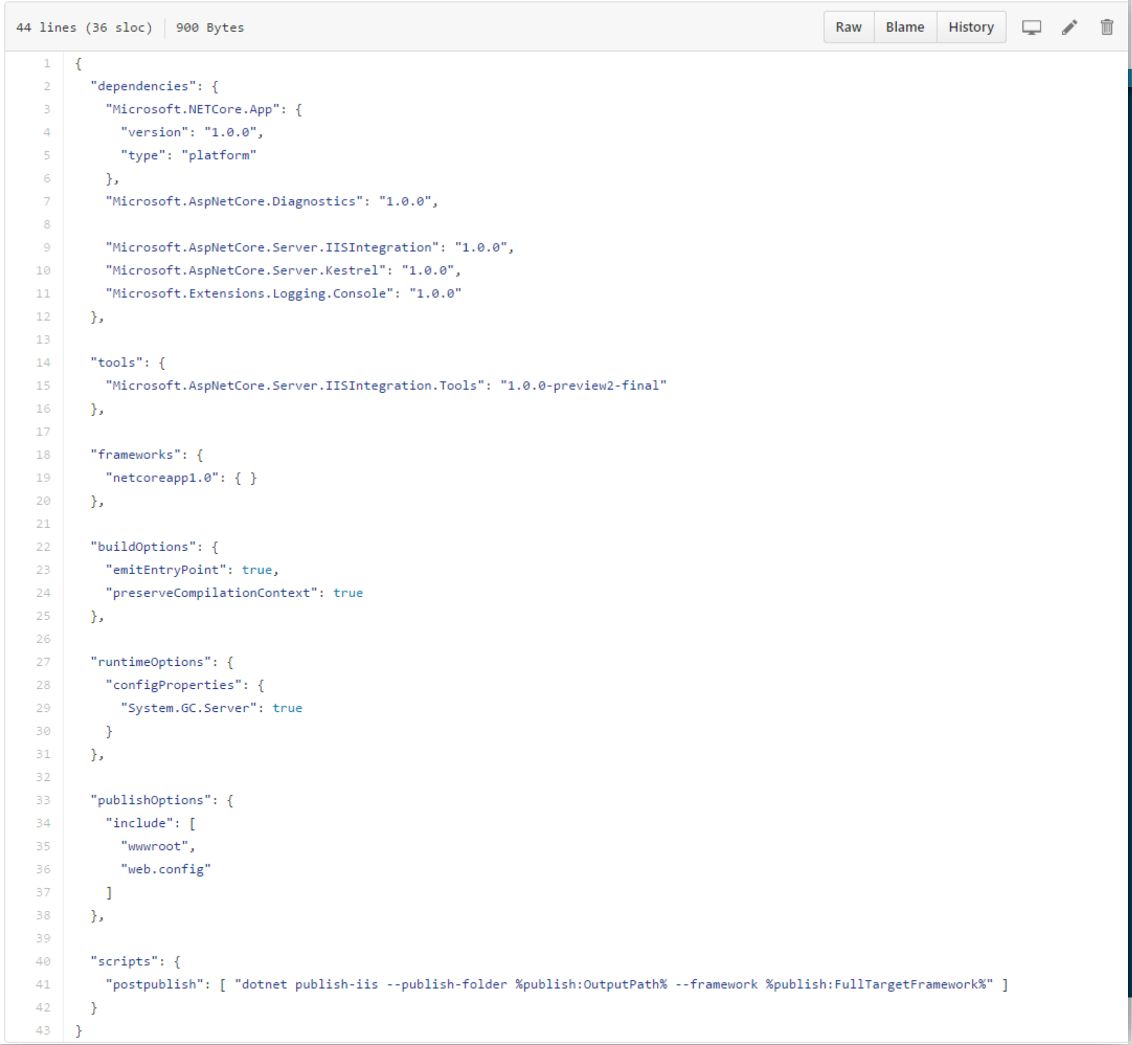
Microsoft.AspNet.Mvc.Core 6.0.0-rc1-final

The core runtime components of ASP.NET MVC.

To install Microsoft.AspNet.Mvc.Core, run the following command in the Package Manager Console

```
PM> Install-Package Microsoft.AspNet.Mvc.Core -Pre
```

https://www.nuget.org/packages/Microsoft.AspNet.Mvc.Core/



```
{
  "dependencies": {
    "Microsoft.NETCore.App": {
      "version": "1.0.0",
      "type": "platform"
    },
    "Microsoft.AspNetCore.Diagnostics": "1.0.0",

    "Microsoft.AspNetCore.Server.IISIntegration": "1.0.0",
    "Microsoft.AspNetCore.Server.Kestrel": "1.0.0",
    "Microsoft.Extensions.Logging.Console": "1.0.0"
  },

  "tools": {
    "Microsoft.AspNetCore.Server.IISIntegration.Tools": "1.0.0-preview2-final"
  },

  "frameworks": {
    "netcoreapp1.0": { }
  },

  "buildOptions": {
    "emitEntryPoint": true,
    "preserveCompilationContext": true
  },

  "runtimeOptions": {
    "configProperties": {
      "System.GC.Server": true
    }
  },

  "publishOptions": {
    "include": [
      "wwwroot",
      "web.config"
    ]
  },

  "scripts": {
    "postpublish": [ "dotnet publish-iis --publish-folder %publish:OutputPath% --framework %publish:FullTargetFramework%" ]
  }
}
```

Anatomy facts

Builds on a console application

```
class Program
{
    public static void Main(string[] args)
    {
        var host = new WebHostBuilder()
               .UseKestrel()
               .UseStartup<Startup>()
               .Build();

        host.Run();
     }
}
```

Services (DI/IoC)

```
class Startup
{
  public void ConfigureServices( IServiceCollection services )
  {

  }

  public void Configure( IApplicationBuilder app )
  {
      app.Run(async (context) =>
      {
          await context.Response.WriteAsync("Hello World!");
      });
   }
}
```

Middleware

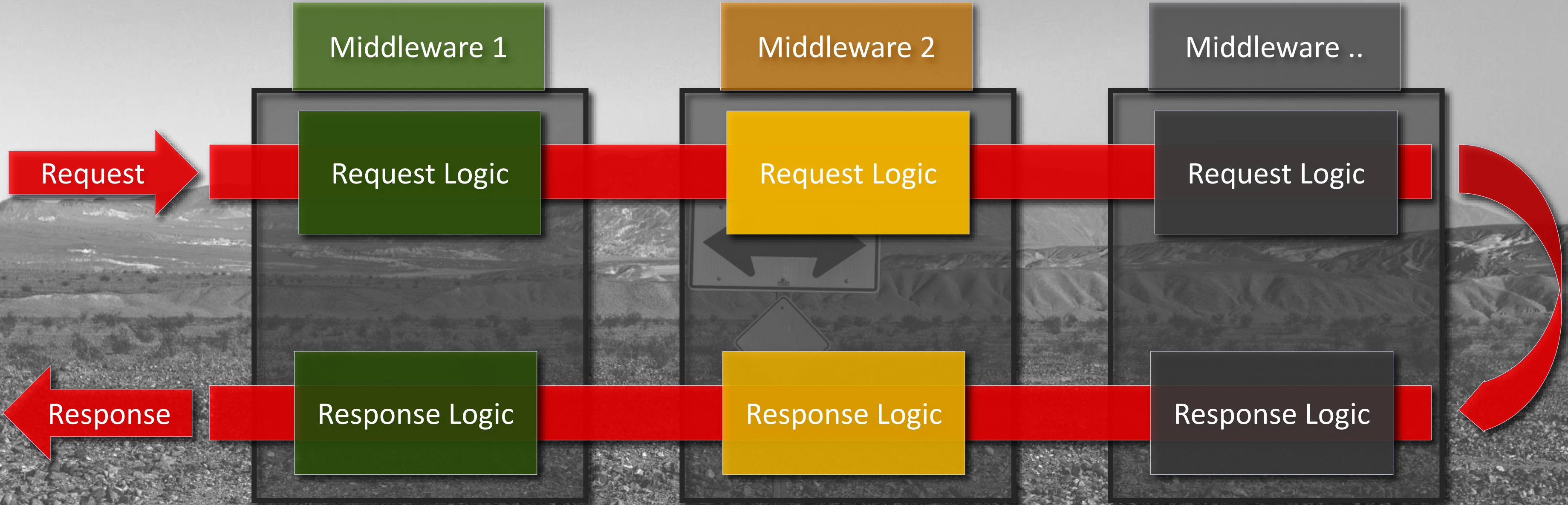
Anatomy facts

ASP.NET Core comes with a built-in Dependency Injection

```
public void ConfigureServices(IServiceCollection services)
{

    // Add framework services.
    var conString = Configuration.GetConnectionString("DefaultConnection");

    services.AddDbContext<ApplicationDbContext>(options =>
                                  options.UseSqlServer( conString );

    services.AddIdentity<ApplicationUser, IdentityRole>()
        .AddEntityFrameworkStores<ApplicationDbContext>()
        .AddDefaultTokenProviders();

    services.AddMvc();

    // Add application services.
    services.AddTransient<IEmailSender, EMailSenderService();
    services.AddTransient<ISmsSender, SMSSenderService();
}
```

Alegri

Middleware



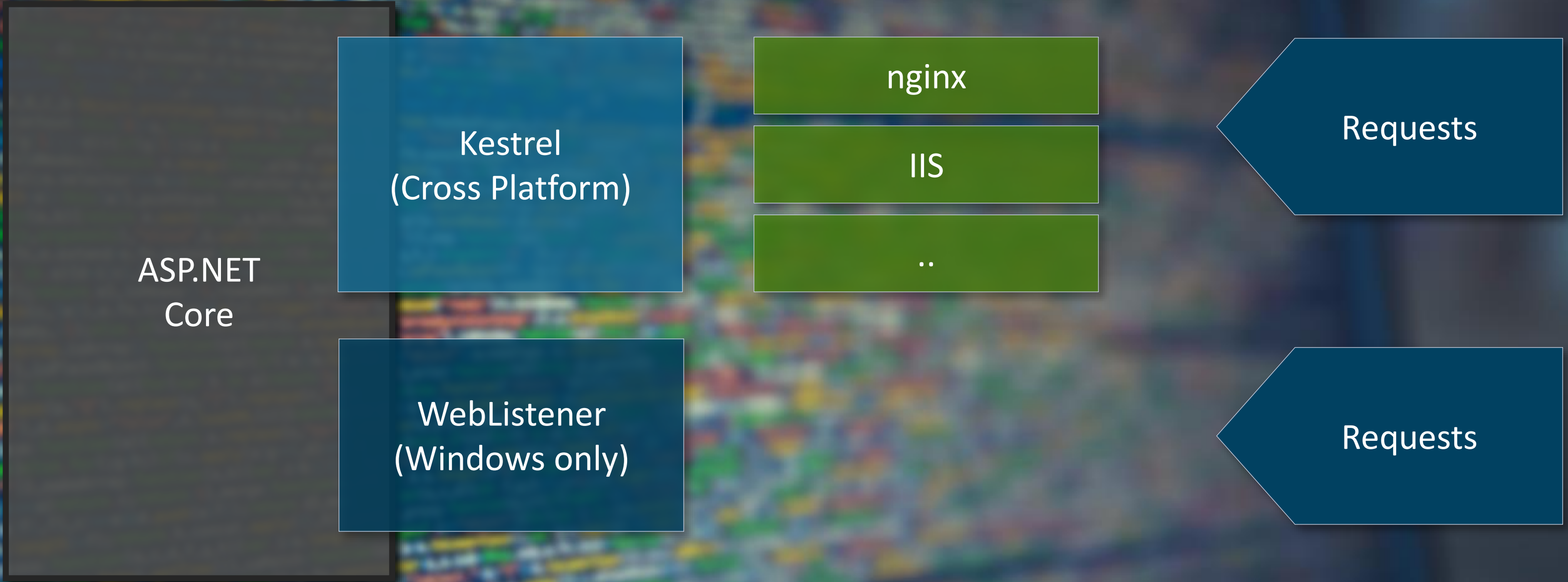
Prebuilt Middlewares ex: Static File serve, Routing, Authentication

Middleware Sample

```
public class LogMiddleware
{
   private readonly RequestDelegate _next;
   private readonly ILogger _logger;

   public LogMiddleware( RequestDelegate next,
                                ILoggerFactory lf )
   {
      _next = next;
      _logger = lf.CreateLogger< LogMiddleware >();
   }

   public async Task Invoke(HttpContext context)
   {
      _logger.Log($"Request: {context.Request.Path}");

       await _next.Invoke(context);

      _logger.Log("Done.");
   }
}
```

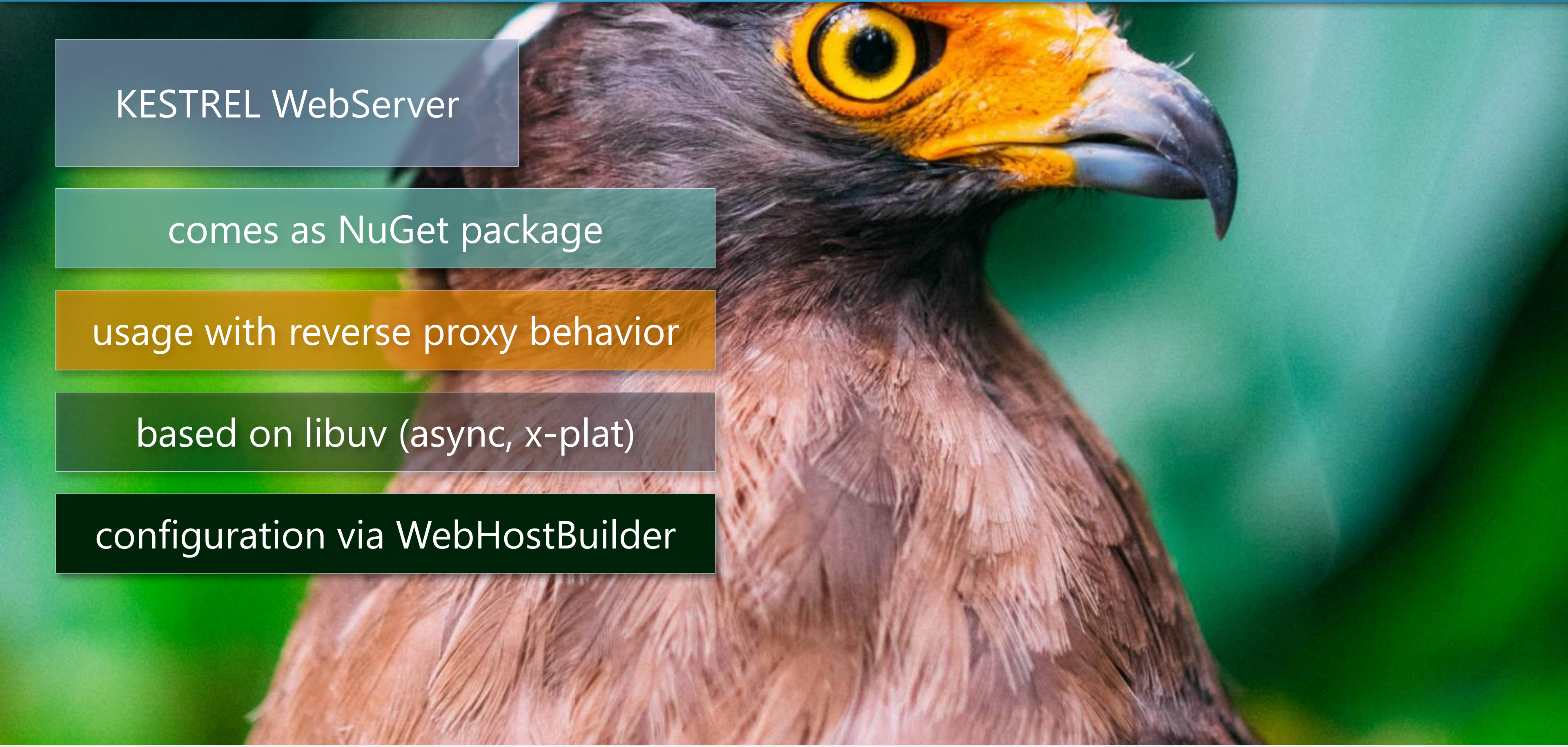
```
class Startup
{
  public void ConfigureServices( IServiceCollection services )
  {

  }

  public void Configure( IApplicationBuilder app )
  {
      app.UseMiddleware< LogMiddleware >

      app.Run(async (context) =>
      {
          await context.Response.WriteAsync("Hello World!");
      });
   }
}
```

WebServers


ASP.NET Core
Kestrel (Cross Platform)
WebListener (Windows only)
nginx
IIS
..
Requests
Requests

Alegri

Kestrel fundamentals



KESTREL WebServer

comes as NuGet package

usage with reverse proxy behavior

based on libuv (async, x-plat)

configuration via WebHostBuilder

Benjamin Abt

Lead Consultant Development

Alegri International Service GmbH
http://alegri.eu

Benjamin Abt is
Microsoft MVP for ASP.NET/IIS

This talk and the code samples were published on https://github.com/AlegriGroup/AspNetCore-RTM-1.0-Demos
Images are from https://unsplash.com/license