



THE DEVELOPER'S CONFERENCE

.NET – Novidades do ASP.NET Core 2.1

Renato Groffe
Microsoft MVP, MTAC

Contatos



/renatogroffe



/in/renatogroffe



/renatogroff



/renatogroffe



/canaldotnet



/canaldotnet

<https://medium.com/@renato.groffe/>

Agenda



- ASP.NET Core 2.1: visão geral das novidades
- Alguns exemplos práticos

ASP.NET Core 2.1



- Lançamento oficial no final de **Mairo/2018**, juntamente com o **.NET Core 2.1** e o **Entity Framework Core 2.1**
- Suporte do **Visual Studio 2017 Update 15.7.x**, **Visual Studio Code** e **Visual Studio for Mac**



Microsoft.AspNetCore.App



- Novo metapackage default
- Recomendável o seu uso ao invés de **Microsoft.AspNetCore.All**
- Reduz o número de dependências de terceiros em relação ao metapackage anterior



Compatibility Version



- Possibilidade de manter os comportamentos de versões anteriores, por mais que se empregue um release mais atual
- Acionado no método **ConfigureServices** da classe **Startup**



Compatibility Version



```
services.AddMvc().SetCompatibilityVersion(  
    CompatibilityVersion.Version_2_1);
```

Suporte a HTTPS



- Habilitado por default com o **ASP.NET Core 2.1**
- Uso de um **certificado de desenvolvimento**
- Redirecionamento automático de **HTTP** para **HTTPS**
- Uso do protocolo **HSTS (HTTP Strict Transport Security)**, que determina o acesso a uma aplicação em produção somente via **HTTPS**



Suporte a HTTPS



THE
DEVELOPER'S
CONFERENCE

```
public void Configure(IApplicationBuilder app,  
    IHostingEnvironment env)  
{  
    if (env.IsDevelopment())  
    {  
        app.UseDeveloperExceptionPage();  
    }  
    else  
    {  
        app.UseHsts();  
    }  
  
    app.UseHttpsRedirection();  
    app.UseMvc();  
}
```



Suporte a GDPR



➤ EU General Data Protection Regulation (GDPR)

➤ Regulamentação da **União Europeia** sobre proteção de dados e privacidade de seus cidadãos

➤ Templates de projetos **MVC** e **Razor Pages** já contemplam ajustes



Suporte a GDPR



THE
DEVELOPER'S
CONFERENCE

```
public void ConfigureServices(IServiceCollection services)
{
    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user
        // consent for non-essential cookies is needed for
        // a given request.
        options.CheckConsentNeeded = context => true;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });
}
```



Suporte a GDPR



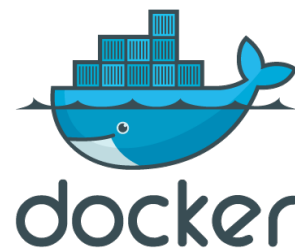
```
public void Configure(IApplicationBuilder app,  
    IHostingEnvironment env)  
{  
    if (env.IsDevelopment()) ...  
    else ...  
  
    app.UseHttpsRedirection();  
    app.UseStaticFiles();  
    app.UseCookiePolicy();  
  
    app.UseMvc();  
}
```



Docker + ASP.NET Core 2.1



- Geração de imagens menores
- Suporte a **Alpine Linux**



Docker + ASP.NET Core 2.1



THE
DEVELOPER'S
CONFERENCE

➤ Antes com o **ASP.NET Core 2.0:**

```
Windows PowerShell
PS C:\Users\renat> docker images apicontagem:latest
REPOSITORY          TAG                IMAGE ID           CREATED            SIZE
apicontagem          latest             3da7563501e8       7 weeks ago       305MB
PS C:\Users\renat> _
```

Docker + ASP.NET Core 2.1



THE
DEVELOPER'S
CONFERENCE

➤ ASP.NET Core 2.1:

```
Windows PowerShell
PS C:\Users\renat> docker images apicontagemv2-normal
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
apicontagemv2-normal latest             689c1fc99ead       2 weeks ago       255MB
PS C:\Users\renat>
```

➤ ASP.NET Core 2.1 com Alpine:

```
Windows PowerShell
PS C:\Users\renat> docker images apicontagemv2-alpine
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
apicontagemv2-alpine latest             47db75495a3c       2 weeks ago       163MB
PS C:\Users\renat>
```

IHttpClientFactory



- Configuração de **HttpClient**s de forma centralizada
- Acesso aos clients via **injeção de dependências**
- **Named e Typed** clientes



IHttpClientFactory - Exemplo



➤ GitHub:
github.com/renatogroffe/ASPNETCore2.1_IHttpClientFactory



O atributo ApiController



- **Atributo associado a um Controller**
- APIs com um **código mais limpo**
- Geração de **erro automático do tipo 400** ao encontrar problemas de validação
- Capacidade de inferir tipos complexos sem o uso de atributos como **[FromBody]**



Antes de ApiController



THE
DEVELOPER'S
CONFERENCE

```
[HttpPost]
0 references
public IActionResult Post(
    [FromBody]Comentario comentario,
    [FromServices]ApplicationDbContext context)
{
    if (!ModelState.IsValid)
        return BadRequest(ModelState);

    comentario.Id = Guid.NewGuid();
    comentario.Data = DateTime.Now;
    context.Comentarios.Add(comentario);
    context.SaveChanges();

    return Ok();
}
```



Com ApiController



THE
DEVELOPER'S
CONFERENCE

```
[Route("api/[controller]")]
[ApiController]
0 references
public class ComentariosController : ControllerBase
{
    [HttpPost]
    0 references
    public IActionResult Post(
        Comentario comentario,
        [FromServices]ApplicationDbContext context)
    {
        comentario.Id = Guid.NewGuid();
        comentario.Data = DateTime.Now;
        context.Comentarios.Add(comentario);
        context.SaveChanges();

        return Ok();
    }
}
```



ApiController - Exemplo



➤ GitHub:
github.com/renatogroffe/ASPNETCore2.1_ApiController



O tipo ActionResult<T>



➤ Simplifica a codificação de instruções de retorno em APIs



Antes de ActionResult<T>



THE
DEVELOPER'S
CONFERENCE

```
[HttpGet("produtos/{codigo}")]
public IActionResult GetProduto(string codigo)
{
    Produto prod = null;
    if (codigo.StartsWith("PROD"))
        prod = _contexto.ObterItem<Produto>(codigo);

    if (prod != null)
        return new ObjectResult(prod);
    else
        return NotFound();
}
```



Com ActionResult<T>



THE
DEVELOPER'S
CONFERENCE

```
[HttpGet("produtos/{codigo}")]
public ActionResult<Produto> GetProduto(string codigo)
{
    Produto prod = null;
    if (codigo.StartsWith("PROD"))
        prod = _contexto.ObterItem<Produto>(codigo);

    if (prod != null)
        return prod;
    else
        return NotFound();
}
```



ActionResult<T> - Exemplo



➤ GitHub:
github.com/renatogroffe/MongoDB-DotNetCore2.1



Razor Class Libraries

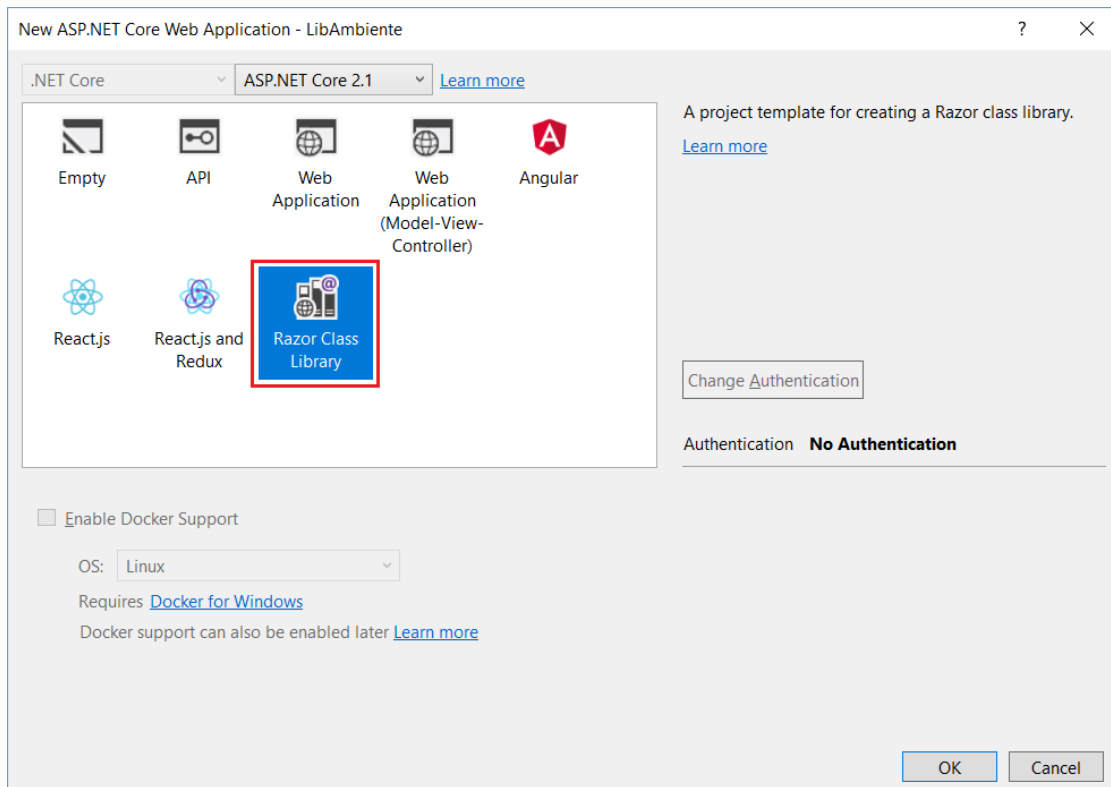


THE
DEVELOPER'S
CONFERENCE

- Elementos de UI disponibilizados como **biblioteca a ser consumida por outros projetos**
- Capacidade de encapsular **Razor Pages, Views e Controllers**
- Possibilidade de publicação como um pacote do **NuGet**
- Conteúdos de **arquivos .cshtml podem ser customizados** no projetos que consomem uma biblioteca



Razor Class Libraries



Razor Class Libraries - Exemplo



➤ GitHub:

github.com/renatogroffe/ASPNETCore2.1_RazorClassLibrary



Outras novidades



➤ Identity UI library & scaffolding

➤ ASP.NET Core Identity fornecido como uma **Razor Class Library**

➤ Kestrel com transporte default baseado em **managed sockets**

➤ ASP.NET Core Generic Host → infraestrutura desacoplada do pipeline HTTP

➤ Templates SPA

➤ Atualizações nos templates para **Angular**, **React** e **React + Redux**

➤ Suporte a **Angular CLI** e **create-react-app**



Outras novidades



- Suporte do **Azure App Service**
- Maior compressão com o suporte ao algoritmo **Brotli**
- Infraestrutura para **testes de integração/funcionais** com um servidor de testes em memória
- **ASP.NET Core SignalR**
 - Possibilidade de desenvolvimento aplicações real-time com ASP.NET Core
 - Integração com **Azure SignalR Service**





THE
DEVELOPER'S
CONFERENCE

Obrigado!!!



THE DEVELOPER'S CONFERENCE