# Dell IT Academy

# Recursos

- Documentação
  - https://docs.microsoft.com/en-us/aspnet/core/
- Fontes
  - https://github.com/aspnet/home

# INTRODUÇÃO AO ASP.NET CORE

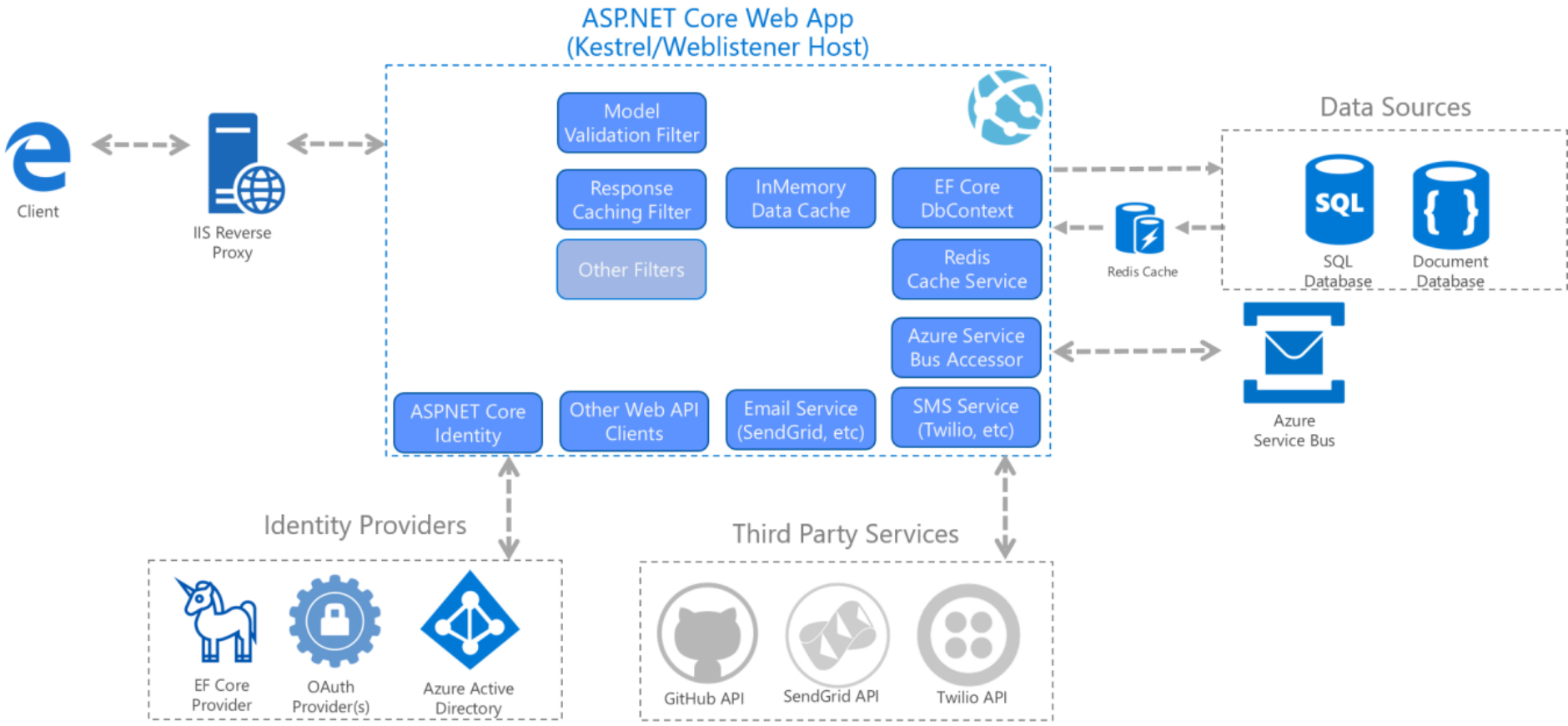
# O que é?

- ASP.NET Core é um dos frameworks .NET para o desenvolvimento de soluções para a Web

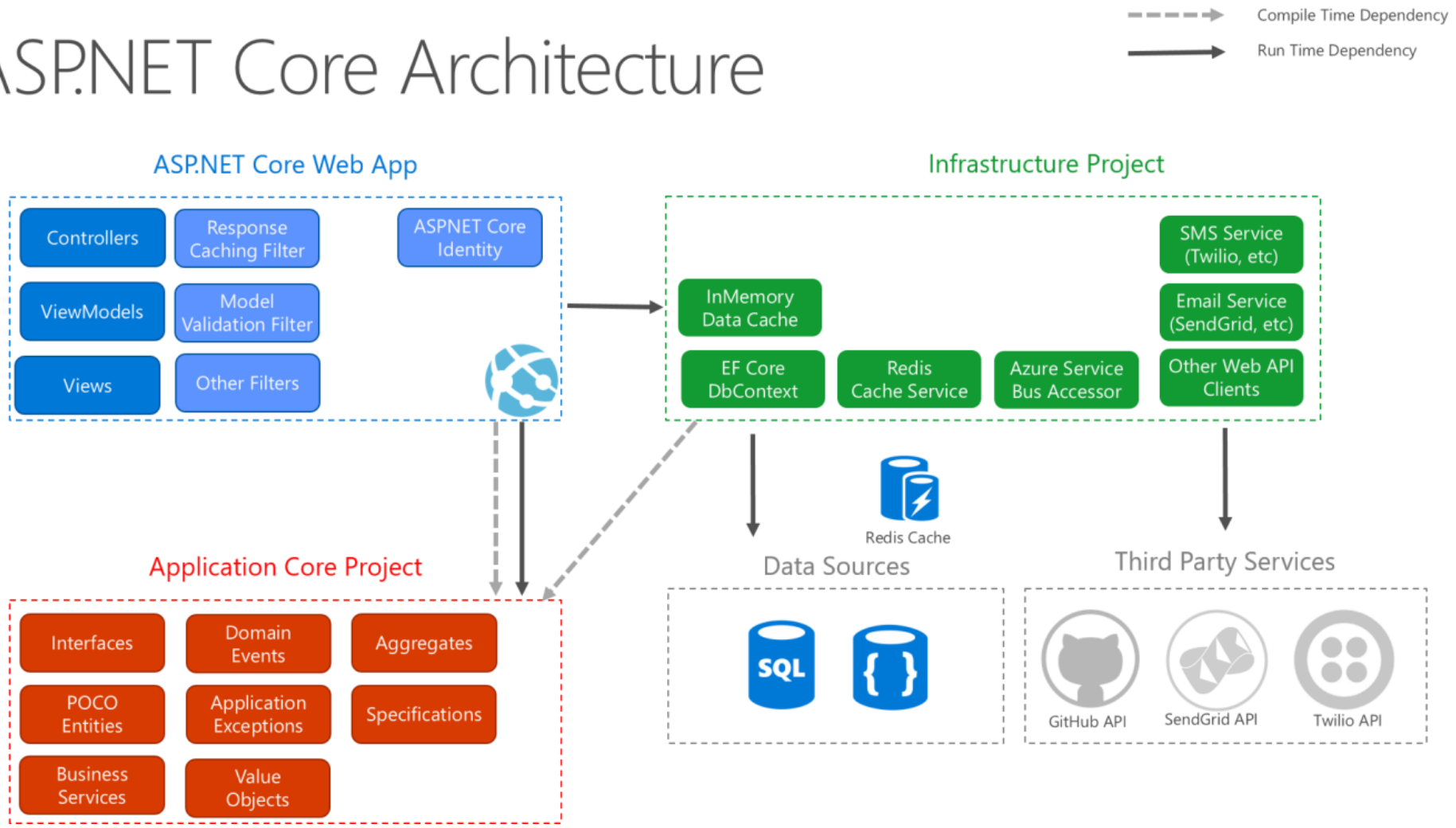
# ASP.NET Core

- Características gerais:
  - Framework unificado para IU Web e Serviços Web
  - Implementação de padrões de Injeção de Dependências
  - Implementação de padrões Model-View-Controller MVC
  - Pipeline configurável para processamento de requisições (Middleware)
  - Múltiplas opções de *hosts* (Kestrel, IIS, Nginx, Apache, Docker, processo do sistema operacional)
  - Suporte a configurações de ambientes
  - Multiplataforma (Windows, Linux, Mac)
  - Distribuído via NuGet

# ASP.NET Core Architecture



# ASP.NET Core Architecture



# ASP.NET Core

- Passos essenciais:
  - Configuração do processo de *bootstrap* (em especial do *host*)
  - Configuração do sistema de injeção de dependências
  - Configuração do *pipeline* de *middleware*

# ASP.NET Core – Bootstrap (.NET6)

- Aplicação de linha de comando
  - Ponto de início é *Program.cs*
- Configuração via código
  - Arquivo *Program.cs*
    - Objetos *WebApplicationBuilder* e *WebApplication* responsáveis pelas configurações
      - *Host*
      - *Middleware* (métodos *Run*, *Map* e *Use*)
      - *Injeção de dependências* (propriedade *Services*)
  - Arquivos JSON/XML
- Ver https://docs.microsoft.com/en-us/aspnet/core/fundamentals/startup

# ASP.NET Core - Servidores

- Servidores suportados
  - IIS – Internet Information Services https://www.iis.net/
  - Servidores integrados
    - Kestrel (multiplataforma, opção padrão dos templates VS)
    - HTTP.sys (Windows)
    - etc
  - Servidores de terceiros
    - Apache http://httpd.apache.org/
    - Nginx http://nginx.org/
    - etc
- Ver https://docs.microsoft.com/en-us/aspnet/core/fundamentals/servers/

# ASP.NET Core – Injeção de Dependências

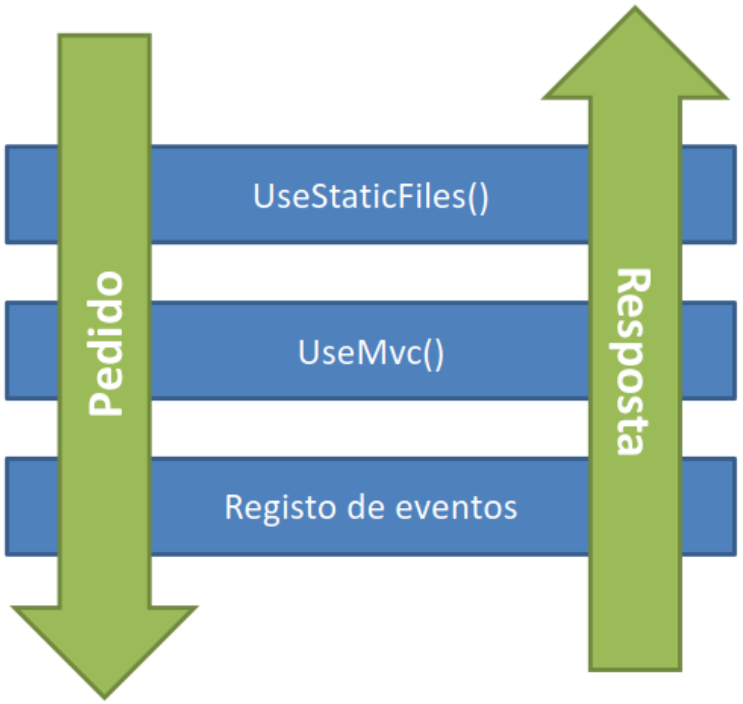
- Um serviço é um componente que oferece uma funcionalidade
  - ASP.NET Core MVC, Entity Framework Core, Identity Framework, Logging, etc
  - Ver https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-6.0#framework-provided-services
- Serviços se tornam disponíveis via Injeção de Dependências
- ASP.NET Core possui um contêiner gerenciador de Inversão de Controle
  - Suporta injeção de dependência via construtor e em ações de controladores
  - Suporta gerenciamento de escopo
    - Transient, Scoped, Singleton
- Ver https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection

# ASP.NET Core – Injeção Dependências (.NET6)

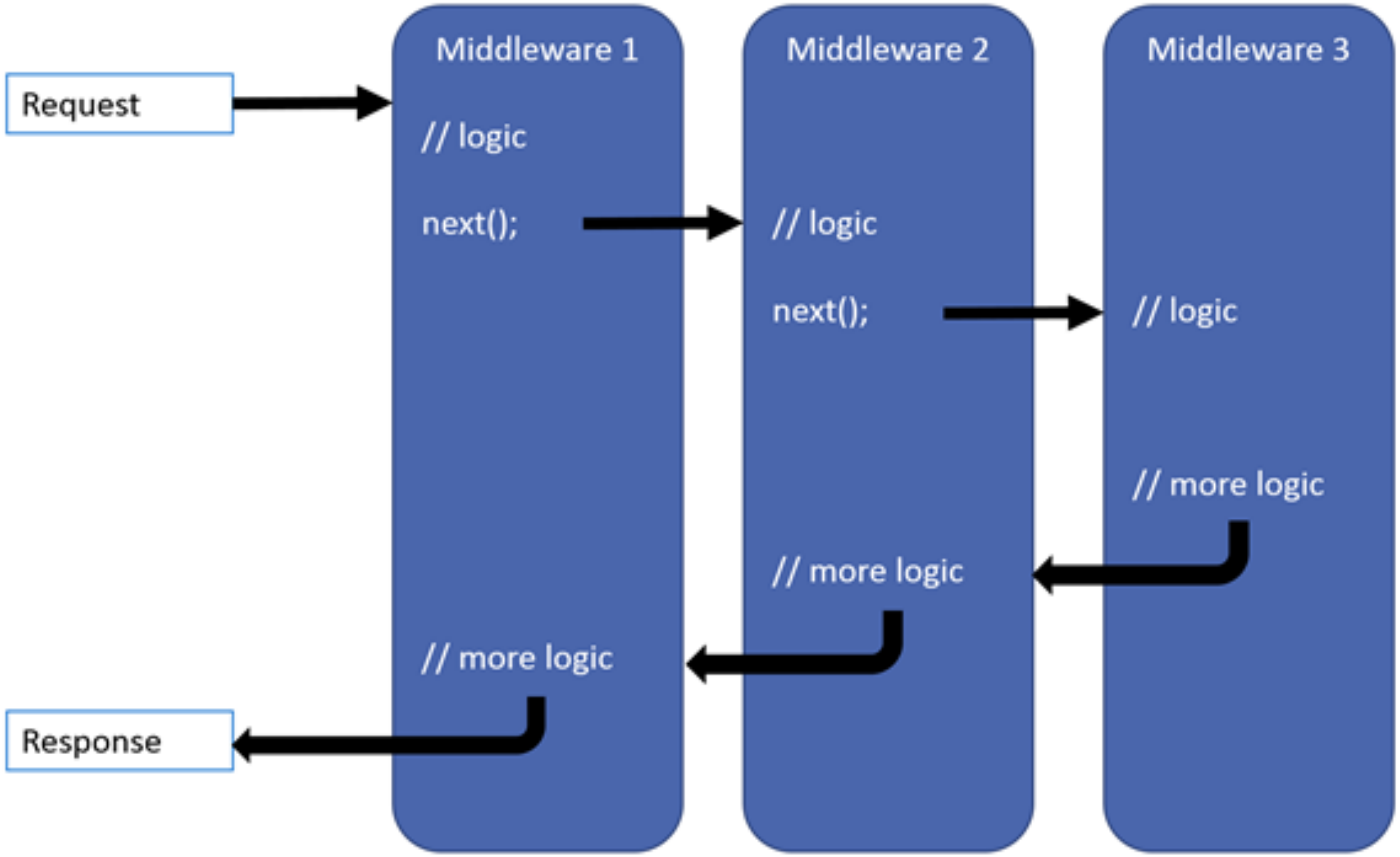
```
var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddScoped<IMyDependency, MyDependency>();
```

# ASP.NET Core - Middleware

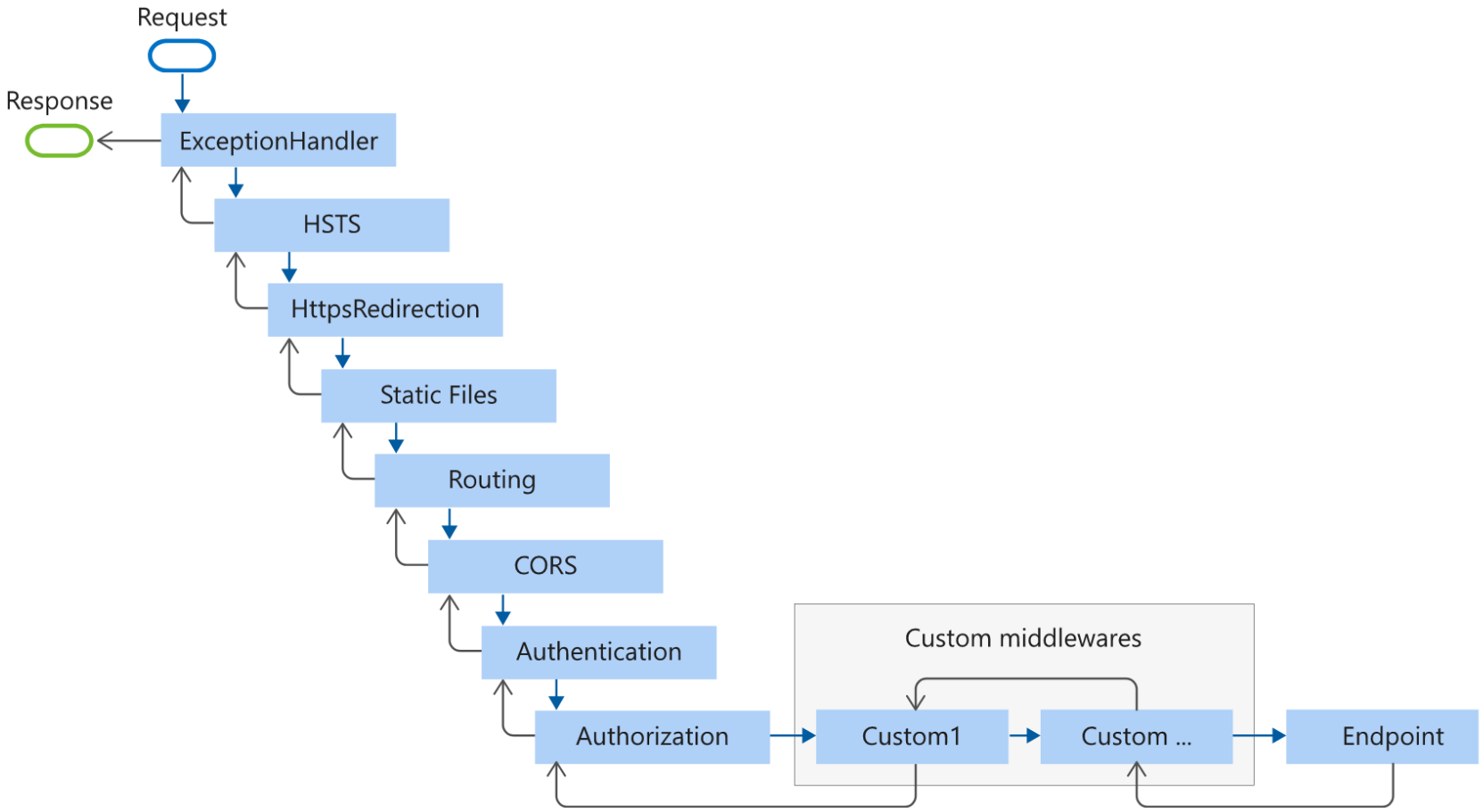
- ASP.NET Core utiliza *pipeline* para o processamento de requisições
  - Arquivos estáticos, roteamento, autenticação, CORS, caching, sessão, etc
  - Ver https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-6.0#built-in-middleware
- *Middleware* lê e escreve diretamente no *pipeline*
- Ver https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/



# ASP.NET Core - Middleware



# ASP.NET Core - Middleware



# ASP.NET Core - Middleware (.NET6)

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

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
app.UseHttpsRedirection();
app.UseAuthentication();
app.UseAuthorization();
```