

Modul 02: Erstellen eine Request-Culture Middleware

Ziel: Middleware aufbauen und in die Pipeline einbinden

Tools: Visual Studio 2022

Dauer: ~20min

1 Theorie

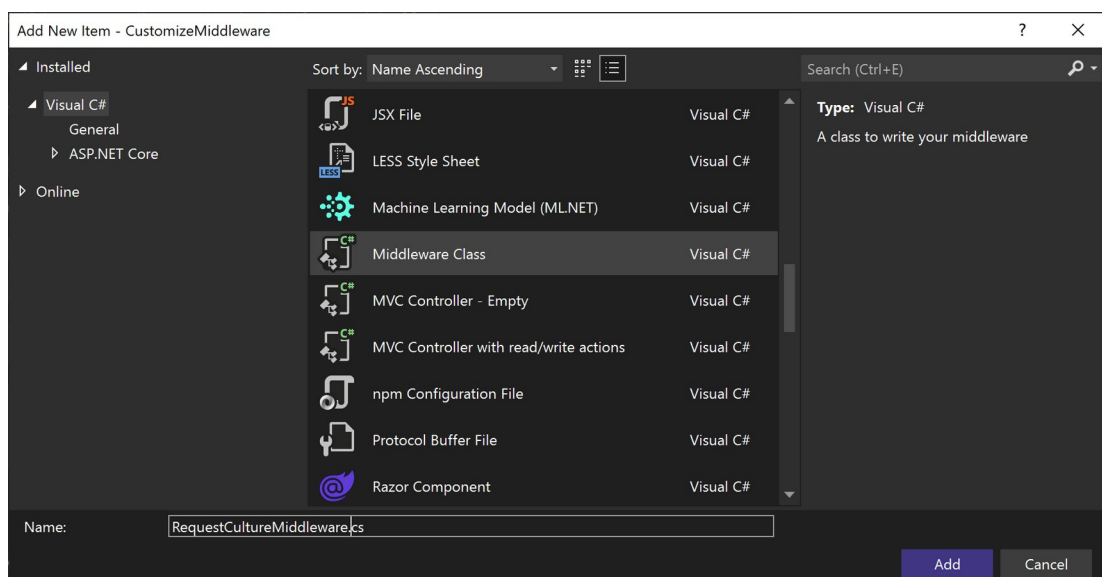
Middleware ist Service, die zu einer Anwendungspipeline zusammengesetzt wird, um Requests und Responses zu verarbeiten. ASP.NET Core stellt in umfangreichem Maß integrierte Middleware-komponenten zur Verfügung. In manchen Szenarios möchten Sie aber vielleicht selbst eine benutzerdefinierte Middleware schreiben.

2 Vorbereitung

Middleware werden in jedem ASP.NET Core Projekt verwendet. Die im Beispiel erstellte Middleware kann unter jedem ASP.NET Core Template laufen. Die Wahl ist daher frei.

3 Erstellen der Middleware

1. Erstellen sie in ihrem Projekt ein Verzeichnis mit dem Namen *Middlewares*
2. Rechte Maustaste auf das Verzeichnis *Middlewares* klicken und im Popup-Menü-
2.1. > Add – New Item auswählen
3. Selektiere die Vorlage Middleware aus und benenne die Datei nach *RequestCultureMiddleware.cs* um.



4. Implementiere die Klasse aus:

4.1. Klassen / Members / Konstruktor:

```
private readonly RequestDelegate _next;

public RequestCultureMiddleware(RequestDelegate next)
{
    _next = next;
}
```

4.2. Invoke – Async Methode und Klassenende

```
public Task Invoke(HttpContext httpContext)
{
    var cultureCode = httpContext.Request.Query["culture"].ToString();
    if (!string.IsNullOrEmpty(cultureCode))
    {
        var culture = new CultureInfo(cultureCode);

        CultureInfo.CurrentCulture = culture;
        CultureInfo.CurrentUICulture = culture;
    }

    return _next(httpContext);
}
```

4.3. Unterhalb der Request-Culture Middleware – Klasse, wird folgende Klasse implementiert

```
public static class RequestCultureMiddlewareExtensions
{
    public static IApplicationBuilder UseRequestCultureMiddleware(this
    IApplicationBuilder builder)
    {
        return builder.UseMiddleware<RequestCultureMiddleware>();
    }
}
```

Die Middlewareklasse Kriterien erfüllen

- Einen *public* Konstruktor mit einem Parameter des Typs [RequestDelegate](#).
- Eine *public* Methode mit dem Namen *Invoke* oder *InvokeAsync* und dem ersten Parameter des Typs [HttpContext](#)

Zusätzliche Parameter für den Konstruktor und Invoke/InvokeAsync werden mittels [Abhängigkeitsinjektion](#) aufgefüllt.

5. Testen Sie die Middleware, indem Sie die Kultur übergeben. Fordern Sie beispielsweise `https://localhost:5001/?culture=no` an.