Endpoint Routing

C# Web 2

**DE HOGESCHOOL
MET HET NETWERK**

Endpoint Routing

- Intro
- Dependency Injection
  - Routing repository

- Endpoint Routing
  - Default
  - Url routing
  - Constraints
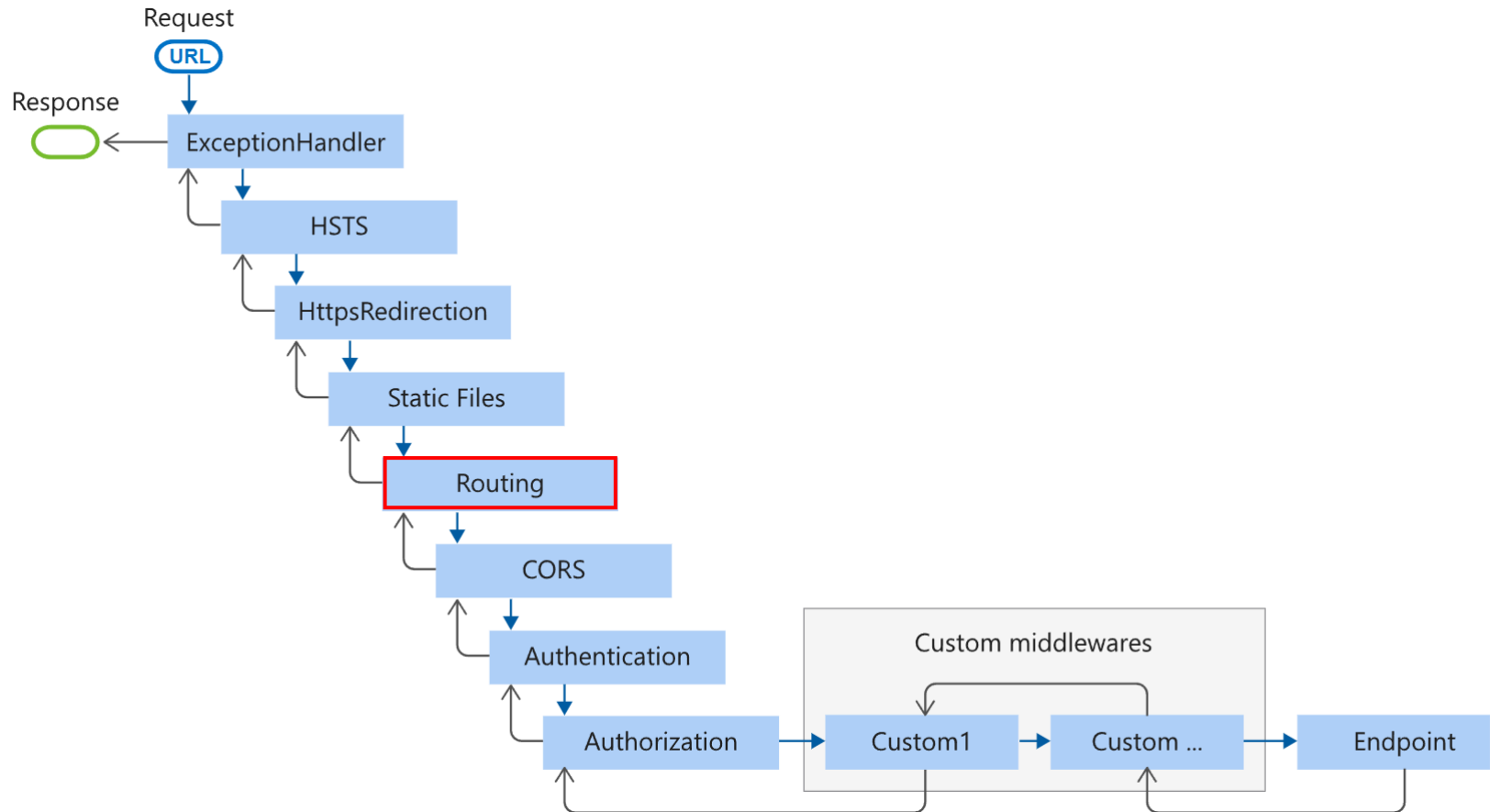   - Custom constraint

Endpoint Routing

Vele middleware (zoals MVC) heeft informatie nodig uit de url van de binnenkomende request om te kunnen beslissen welke actie te moeten ondernemen

- Het analyseren van de url is logica die vaak terug komt

  -> duplicatie in code

- Url routing introduceert middleware die deze logica centraal zet en binnenkomende URLs kan matchen op zogenaamde **endpoints** (bv. een MVC controller).

  De endpoints zorgen voor de nodige http response.

- De mapping tussen URLs en endpoints worden uitgedrukt in **routes**. De URL routing middleware inspecteert deze routes om te bepalen welk endpoint een binnenkomende request gaat afhandelen.

Middleware Pipeline

Endpoint Routing – Program.cs

```
...
app.UseRouting();
app.MapControllerRoute(
    name: "default",
    pattern:"{controller=Home}/{action=Index}/{id?}");
...
```

Endpoint Routing – ViewModels

- Create folder
  - Models/ViewModels
  - Create class
    - RoutingModel

Endpoint Routing – ViewModels

```csharp
namespace MVCEndPointRouting.Models.ViewModels
{
    public class RoutingModel
    {
        public RoutingModel(string controller)
        {
            Controller = controller;
            Date = DateTime.Now;
        }
        public string Controller { get; set; }
        public string ActionName { get; set; }
        public DateTime Date { get; set; }
        public Dictionary<string, string> RoutingParameters =
          new Dictionary<string, string>();
        public string Parameters => string.Join(',',
                RoutingParameters.Select(x =>
                string.Format("{0}:{1}", x.Key, x.Value)));

    }
}
```

Endpoint Routing – Services

```csharp
public interface IRoutingRepository
{
    IEnumerable<RoutingModel> RoutingModels { get; }
    void Add(RoutingModel routingmodel);
}
```

```csharp
public class RoutingRepository : IRoutingRepository
{
    private List<RoutingModel> _routingModels = new List<RoutingModel>();

    public IEnumerable<RoutingModel> RoutingModels => _routingModels;

    public void Add(RoutingModel routingModel)
    {
        _routingModels.Add(routingModel);
    }
}
```

Endpoint Routing -Program.cs DI - Routing repository

```
builder.Services.AddControllersWithViews();
builder.Services.AddSingleton<IRoutingRepository, RoutingRepository>();
```

Endpoint Routing – HomeController

```csharp
public class HomeController : Controller
{
    IRoutingRepository _routingRepo;
    public HomeController(IRoutingRepository routingRepo)
    {
        _routingRepo = routingRepo;
    }
    public IActionResult Index()
    {

        RoutingModel routingModel = new RoutingModel("Home");
        routingModel.ActionName = "Index";
        _routingRepo.Add(routingModel);
        return View();
    }
    public IActionResult RoutingList()
    {

        RoutingModel routingModel = new RoutingModel("Home");
        routingModel.ActionName = "RoutingList";
        _routingRepo.Add(routingModel);
        return View(_routingRepo.RoutingModels);
    }
}
```

Endpoint Routing – _Layout

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Endpoint routing</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
    <div class="bg-dark text-white p-2">
        <span class="navbar-brand ml-2">MVCEndpointRouting</span>
    </div>
    <div class="row m-1 p-1">
        <div id="menu" class="col-3">
            <ul>
                <li>
                    <a asp-controller="Home" asp-action="Index">Home</a>
                </li>
                <li>
                    <a asp-controller="Home" asp-action="RoutingList">Routing list</a>
                </li>
            </ul>
        </div>
        <div class="col-9">
            @RenderBody()
        </div>
    </div>
</body>
</html>
```

Endpoint Routing – RoutingList View

- Views/Home
  - Add razor view
    - Template: List
    - Name: RoutingList
    - Model: RoutingModel

Endpoint Routing

```
<a asp-controller="Home" asp-action="Index">Home</a>
      -> Html href=Home/Index

<a asp-controller="Home" asp-action="RoutingList">Routing list</a>
      -> Html href=Home/RoutingList
```

MVCEndpointRouting

- Home
- Routing list

RoutingList

Controller	ActionName	Date	Parameters
Home	Index	17/03/2021 14:44:20	
Home	RoutingList	17/03/2021 14:44:23	

Endpoint Routing - RoutingController

```csharp
namespace MVCEndPointRouting.Controllers
{
    public class RoutingController : Controller
    {
        IRoutingRepository _routingRepo;
        public RoutingController(IRoutingRepository routingRepo)
        {
            _routingRepo = routingRepo;
        }
        public IActionResult RoutingInt(string id)
        {
            RoutingModel routingModel = new RoutingModel("Routing");
            routingModel.ActionName = "RoutingInt";
            routingModel.RoutingParameters.Add("id", id);
            _routingRepo.Add(routingModel);
            ViewBag.ValueofId = id ?? "Null Value";

            return View();
        }
    }
}
```

Endpoint Routing - _Layout

- Asp-Controller
    - HomeController
        - Asp-Action
            - Index
            - RoutingList
- Asp-Controller
    - RoutingController
        - Asp-Action
            - RoutingInt

```html
<ul>
    <li>
        <a asp-controller="Home" asp-action="Index">Home</a>
    </li>
    <li>
        <a asp-controller="Home" asp-action="RoutingList">Routing list</a>
    </li>
    <li>
        <a asp-controller="Routing" asp-action="RoutingInt">Routing int zonder parameter</a>
    </li>
    <li>
        <a asp-controller="Routing" asp-action="RoutingInt" asp-route-id="10">Routing int - 10</a>
    </li>
</ul>
```

Endpoint Routing – Views/Routing/RoutingInt

- Asp-Controller
    - HomeController
        - Asp-Action
            - Index
            - RoutingList
- Asp-Controller
    - RoutingController
        - Asp-Action
            - RoutingInt

```html
<!DOCTYPE html>
<html>
<head>
    <title>Routing</title>
</head>
<body>
    <h1>'Routing' Controller, 'RoutingInt' View</h1>
    <h2>Id value is: @ViewBag.ValueofId</h2>
</body>
</html>
```

Endpoint Routing - RoutingController

Endpoint Routing - _Layout

```
<li>
  <a asp-controller="Routing" asp-action="RoutingIdAndName" asp-route-id="10"
       asp-route-name="Test">Routing id and name</a>
</li>
```


Bouw zelf de controller en de bijhorende view!

Endpoint Routing - Constraints

```
app.MapControllerRoute(
     name: "intConstraint",
     pattern: "{controller=Home}/{action=Index}/{id:int?}");
```

- Enkel gehele getallen of null zijn toegelaten
  - Home/Index
  - Home/Index/5
  - Home/Index/1
  - Home/Index/abc   -> niet toegelaten

Endpoint Routing - Constraints

```
app.MapControllerRoute(
    name: "rangeConstraint",
    pattern: "{controller=Home}/{action=Index}/{id:range(5,20)?}");
```

- Enkel getallen in de range van 5 t.e.m. 20 of null zijn toegelaten
  - Home/Index
  - Home/Index/5
  - Home/Index/abc   -> niet toegelaten
  - Home/Index/1   -> niet toegelaten

Endpoint Routing - Constraints

```
app.MapControllerRoute(
    name: "alphaConstraint",
    pattern: "{controller=Home}/{action=Index}/{id:alpha?}");
```

- Enkel null of karakters a-z of A-Z zijn toegestaan
  - Home/Index
  - Home/Index/abc
  - Home/Index/1   -> niet toegelaten

Endpoint Routing – Custom Constraint

- Nieuwe Folder in het project
  - CustomConstraint
    - Add class
      - FirstNameConstraint

```csharp
namespace MVCEndPointRouting.CustomConstraint
{
    public class FirstNameConstraint : IRouteConstraint
    {
        private string[] firstNames = new[] { "Kristof", "Wim" };
        public bool Match(HttpContext httpContext, IRouter route,
                string routeKey, RouteValueDictionary values,
                RouteDirection routeDirection)
        {
            return firstNames.Contains(values[routeKey]);
        }
    }
}
```

Endpoint Routing – Custom Constraint
Program.cs

```
builder.Services.Configure<RouteOptions>(options =>
        options.ConstraintMap.Add("allowedFirstNames", typeof(FirstNameConstraint)));
builder.Services.AddControllersWithViews();
builder.Services.AddSingleton<IRoutingRepository, RoutingRepository>();
```

```
...
// Custom constraint
app.MapControllerRoute(
    name: "customConstraint",
    pattern: "{controller=Home}/{action=Index}/{id:allowedFirstNames?}");
```

Endpoint Routing – Custom Constraint

- Enkel null of voornamen uit de lijst zijn toegestaan
  - Home/Index
  - Home/Index/Kristof of Wim
  - Home/Index/abc   -> niet toegelaten
  - Home/Index/1   -> niet toegelaten

Endpoint Routing – Oefening Blackboard