

# Adding Navigation to the Site

---



**Gill Cleeren**

CTO XPIRIT BELGIUM

@gillcleeren [www.snowball.be](http://www.snowball.be)



# Overview



Understanding routing in ASP.NET Core MVC

Configuring the routing system

Navigating with tag helpers

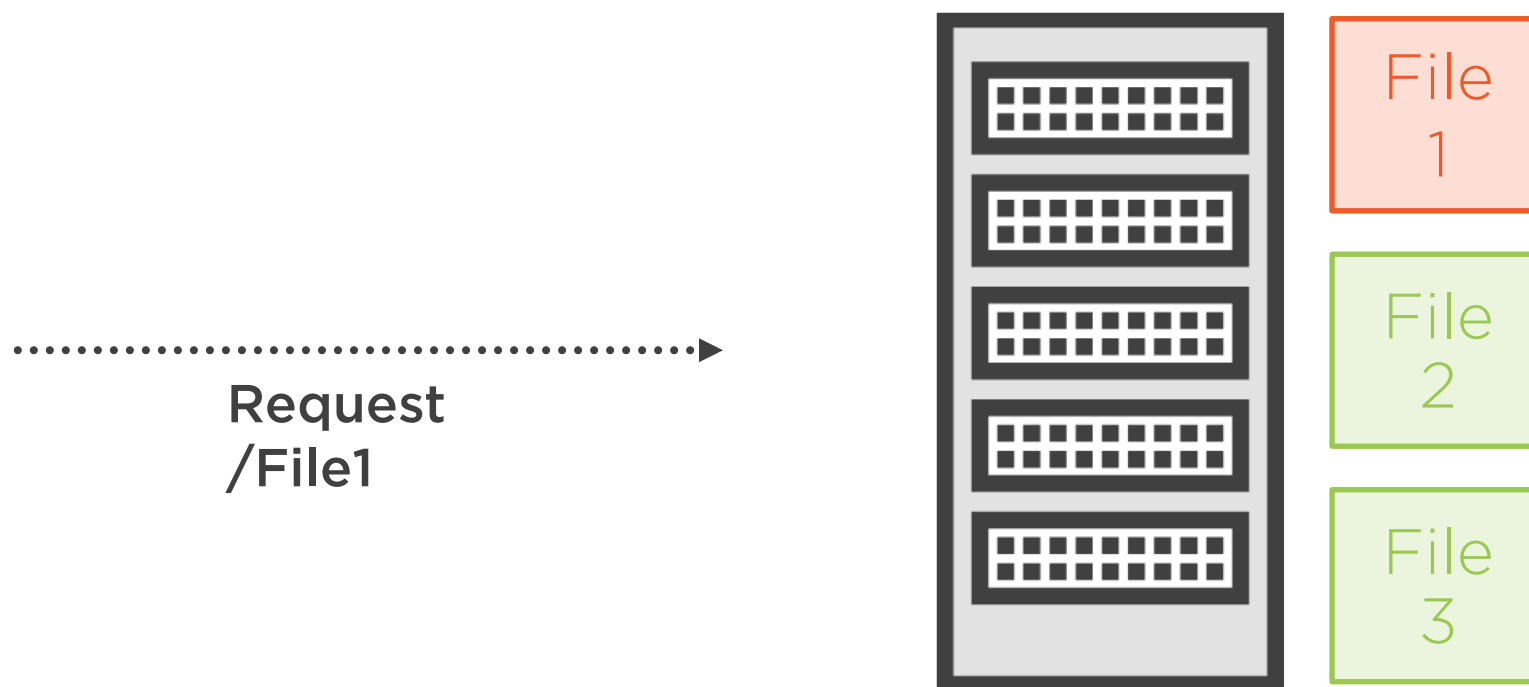


# Understanding Routing in ASP.NET Core MVC

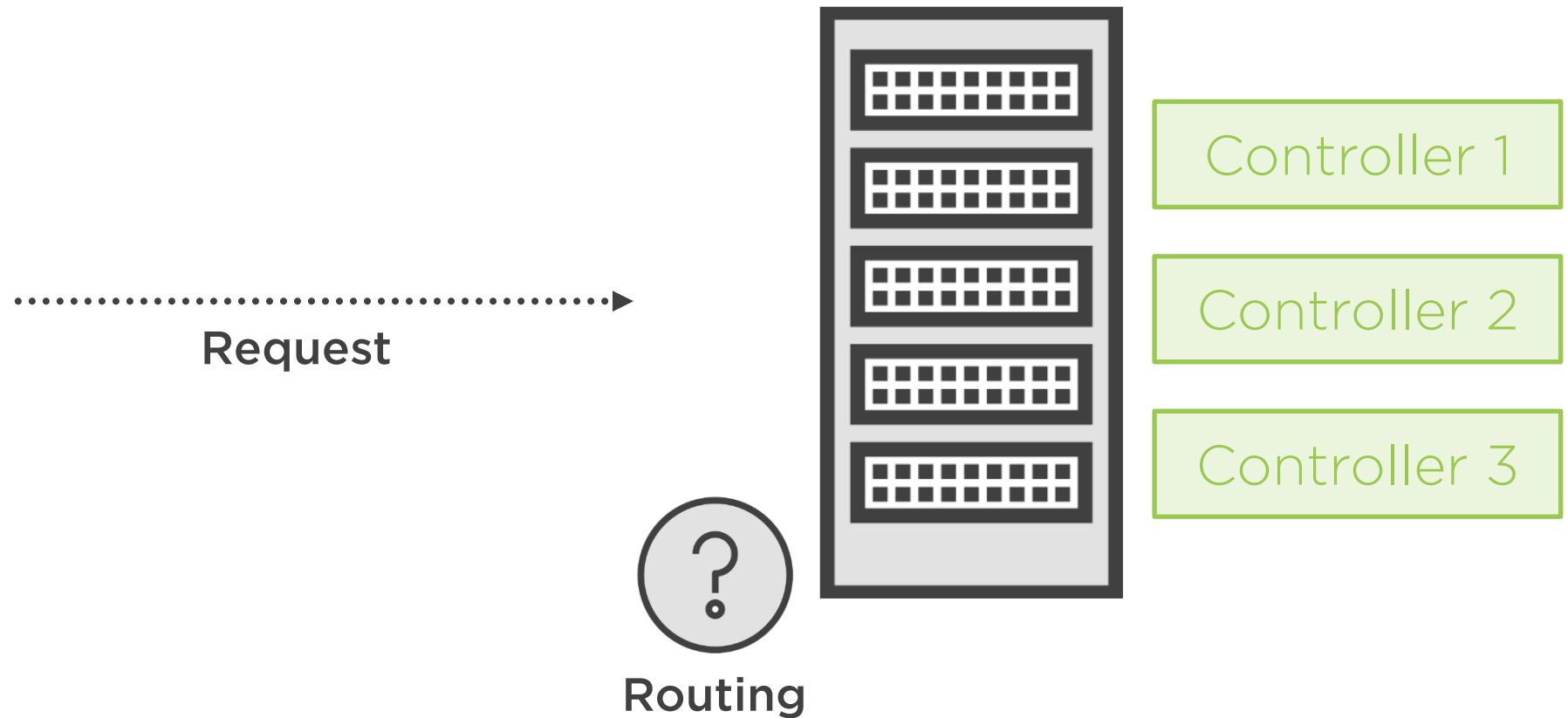
---



# Serving Files



# Handling Requests in ASP.NET Core MVC



# Routing

Mapping URI to endpoint

Generate links

Different types

Changed with .NET Core 3



# Patterns and Routes

`http://www.bethanyspieshop.com/Pie/List`

↑  
Controller ↑  
Action

`{Controller}/{Action}`



# Routing to the Action Method

```
public class PieController : Controller
{
    public ActionResult List()
    {
        return View();
    }
}
```





# Working with Segments

`http://www.bethanyspieshop.com/Pie/Details/1`

Controller

Action

Value

`{Controller}/{Action}/{id}`



# Passing a Value

```
public class PieController : Controller
{
    public ViewResult Details(int id)
    {
        //Do something
    }
}
```



# Configuring the Routing System

---



```
app.UseRouting();  
app.UseEndpoints(endpoints =>  
{  
    endpoints.MapControllerRoute(  
        name: "default",  
        pattern: "{controller}/{action}");  
});
```

## Configuration in Startup



```
app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller}/{action}");
});
```

## Configuration in Startup (ASP.NET Core 2.1)



```
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}");
});
```

## Route Defaults

- Will match [www.bethanyspieshop.com](http://www.bethanyspieshop.com)



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

## Passing Values



# Receiving the Value

```
public class PieController : Controller
{
    public ViewResult Details(int id)
    {
        //Do something
    }
}
```





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

## Optional Segments

- Will match
  - [www.bethanyspieshop.com/Pie/List](http://www.bethanyspieshop.com/Pie/List)
  - [www.bethanyspieshop.com/Pie/Details/1](http://www.bethanyspieshop.com/Pie/Details/1)



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

## Adding Constraints



# Navigating with Tag Helpers

---





We need to generate correct links in the resulting HTML



```
@Html.ActionLink  
("View Pie List","List",  
"Pie")
```

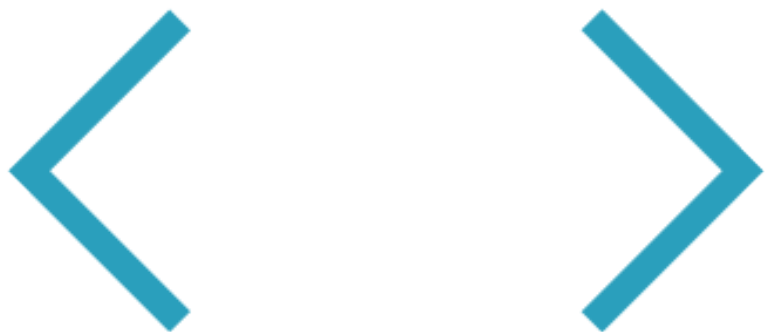
```
<a asp-controller="Pie"  
asp-action="List">  
View Pie List  
</a>
```

◀ HTML Helpers

◀ Tag Helpers



# Tag Helpers



**Server-side**

**Trigger code execution**

**Built-in or custom**

**Replace HTML Helpers**



```
<a asp-controller="Pie" asp-action="List">  
  View Pie List  
</a>
```

## Tag Helpers



```
<a href="/Pie/List">View Pie List</a>
```

Resulting HTML





# Registering Your Tag Helpers

```
@using BethanysPieShop.Models
```

```
@using BethanysPieShop.ViewModels
```

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```



# Tag Helpers

**asp-controller**

**asp-action**

**asp-route-\***

**asp-route**



# Demo



## Adding Navigation



# Summary



**Routing matches incoming request URI with pattern**

**Tag Helpers help to create clean code**

- Also for links





**Up next:**  
Adding more View features

