

Маршруттаудың қосымша мүмкіндіктері

Алдыңғы тараудан `UrlsAndRoutes` жобасын қолдануды жалғастырамын. Жалғыз талап етілетін өзгеріс - бұл `Startup` класында, мен `UseMvcWithDefaultRoute` әдісін 16-1 листингінде көрсетілгендей әсер ететін айқын бағытпен алмастырдым.

Listing 16-1. `Startup` файлындағы бағыттау конфигурациясын өзгерту. `UrlsAndRoutes` қалтасындағы `cs`

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.AspNetCore.Routing.Constraints;

using Microsoft.AspNetCore.Routing;

using UrlsAndRoutes.Infrastructure;

namespace UrlsAndRoutes
{
    public class Startup
    {
        public void ConfigureServices(IServiceCollection services)
        {
            services.Configure<RouteOptions>(options => options.ConstraintMap.Add("weekday",
            typeof(WeekDayConstraint))); services.AddMvc();
        }
    }
}
```

```

public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    app.UseDeveloperExceptionPage();

    app.UseStatusCodePages();

    app.UseStaticFiles();

    app.UseMvc(routes => {

        routes.MapRoute(

            name: "default",

            template: "{controller=Home}/{action=Index}/{id?}");

        });

    }

}
}
}

```

Егер сіз қосымшаны іске қоссаңыз, браузер әдепкі URL мекенжайын сұрайды, ол 16-1 суретте көрсетілгендей home контроллеріндегі Index әрекетіне жіберіледі.

Көріністерде Шығыс URL мекен-жайларын құру

Барлық дерлік MVC қосымшаларында сіз қолданушыға бір көріністен екіншісіне ауысуға рұқсат бергіңіз келеді, ол әдетте екінші көріністі құратын әрекет әдісіне арналған бірінші көріністегі сілтемені қамтиды. Href атрибуты әрекет әдісіне бағытталған статикалық элементті (якорь элементі ретінде белгілі) қосу өте қызықты:

```
<a href="/Home / CustomVariable " >бұл URL адресі < / a>
```

Бағдарлама әдепкі бағыттау конфигурациясын қолданады деп есептесеңіз, бұл HTML элементі үй контроллеріндегі CustomVariable әрекет әдісіне бағытталған сілтеме жасайды. Қолмен берілген URL мекенжайлары тез және оңай жасалады. Олар сондай-ақ өте қауіпті және сіз өзіңіздің ҚОСЫМШАҢЫЗДЫҢ URL схемасын өзгерткен кезде қатты кодталған барлық URL мекен-жайларын бұзасыз. Содан кейін сіз өзіңіздің бағдарламаңыздағы барлық көріністерді қарап, контроллерлер мен әрекет әдістеріне сілтемелерді жаңартуыңыз керек - бұл қиын, қателікке бейім және қиын процесс. Ең жақсы балама-Шығыс URL мекен-жайларын құру үшін бағыттау жүйесін пайдалану, бұл URL мекен-жайларының динамикалық URL-мекен-жайларын құру үшін пайдаланылатындығына және қосымшаның URL-мекен-жай схемасын көрсетуге кепілдік береді.

Шығыс сілтемелерді құру

Көріністе Шығыс URL мекен-жайын құрудың ең оңай жолы - HTML а элементі үшін href тәлсіпатын құратын сілтеме тегінің көмекшісін пайдалану, 16-2 Листингте көрсетілгендей, мен көрініске/көріністерге/ортақ/нәтижеге қосқан қосымшаны көрсетеді.cshtml.

Кеңес. Мен 23-тарауда тегтер көмекшілерінің қалай жұмыс істейтінін егжей-тегжейлі түсіндіремін.

Listing 16-2. Result файлында якорь тег көмекшісін пайдалану.көрініс қалтасындағы cshtml / ортақ

```
@model Result
```

```
@{ Layout = null; }
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta name="viewport" content="width=device-width" />
```

```
<title>Routing</title>
```

```
<link rel="stylesheet" asp-href-include="lib/bootstrap/dist/css/*.min.css" />
```

```
</head>
```

```
<body class="m-1 p-1">
```

```
<table class="table table-bordered table-striped table-sm">
```

```
<tr>
```

```
<th>Controller:</th>
```

```
<td>@Model.Controller</td>
```

```
</tr>
```

```
<tr>
```

```
<th>Action:</th>
```

```
<td>
```

```
@Model.Action
```

```
</td>
```

```
</tr>
```

```
@foreach (string key in Model.Data.Keys) {
```

```
<tr>
```

```
<th>
```

```
@key :  
</th>  
<td>@Model.Data[key]</td>  
</tr>  
}  
</table>  
<a asp-action="CustomVariable">This is an outgoing URL</a>  
</body>  
</html>
```

Asp-action тәлсипаты href тәлсипатындағы URL мекенжайын көрсететін әрекет әдісінің атауын көрсету үшін қолданылады. Нәтижені 16-2 суретте көрсетілгендей қосымшаны іске қосу арқылы көруге болады.

Тег көмекшісі ағымдағы бағыттау конфигурациясын қолдана отырып, а элементі үшін href тәлсипатын орнатады. Егер сіз шолғышқа жіберілген HTML-ді тексерсеңіз, оның құрамында келесі элемент бар екенін көресіз:

```
<a href="/Home / CustomVariable " >бұл URL адресі < / a>
```

Бұл сізге бұрын көрсеткен қолмен берілген URL мекенжайын қалпына келтіру үшін қосымша күш сияқты көрінуі мүмкін, бірақ бұл тәсілдің артықшылығы-бағыттау конфигурациясының өзгеруіне автоматты түрде жауап береді. Көрсету үшін мен Startup файлына жаңа маршрут қостым.CS, 16-3 Листингте көрсетілгендей.

Listing 16-3. Startup файлына маршрут қосу.UrlAndRoutes қалтасындағы cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using Microsoft.AspNetCore.Builder;
```

```
using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.Extensions.DependencyInjection; using Microsoft.AspNetCore.Routing.Constraints;

using Microsoft.AspNetCore.Routing;

using UrlsAndRoutes.Infrastructure;

namespace UrlsAndRoutes

{

public class Startup

{

public void ConfigureServices(IServiceCollection services)

{

services.Configure<RouteOptions>(options => options.ConstraintMap.Add("weekday",
typeof(WeekDayConstraint))); services.AddMvc();

}

public void Configure(IApplicationBuilder app, IHostingEnvironment env)

{

app.UseDeveloperExceptionPage();

app.UseStatusCodePages();

app.UseStaticFiles();

app.UseMvc(routes => {

routes.MapRoute(

name: "NewRoute",

template: "App/Do{action}",

defaults: new { controller = "Home" });

routes.MapRoute(

name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

});
```

}

}

}

Жаңа маршрут home контроллеріне бағытталған сұраулар үшін URL схемасын өзгертеді. Бағдарламаны іске қосу арқылы Сіз бұл өзгеріс ActionLinkHTML көмекші әдісі жасаған HTML кодында келесідей көрінетінін көресіз:

` бұл шығыс URL < / a>`

Тег көмекшісімен сілтемелер құру маңызды қызмет мәселесін шешеді. Мен маршруттау схемасын өзгерте аламын және көріністердегі Шығыс сілтемелер қолданбадағы көріністерді қолмен өңдеусіз бұл өзгерісті автоматты түрде көрсетеді. Сілтемені басқан кезде Шығыс URL мекенжайы кіріс HTTP сұрауын жасау үшін қолданылады, содан кейін 16-3 суретте көрсетілгендей сұранысты өңдейтін әрекет әдісі мен контроллерді тағайындау үшін қолданылады.