Қосымшада URL маршруттауды орындау

Бұл жұмыста мен веб-қосымшаның шаблонын қолдандым ASP.NET UrlsAndRoutes деп аталатын жаңа бос жобаны жасау үшін Core (. NET Core). MVC Framework қолдауын, әзірлеуші қате беттерін және статикалық файлдарды қосу үшін мен Startup класына 15-1 листингінде көрсетілген мәлімдемелерді қосамын.

Listing 15-1. Бағдарламаны Startup файлында теңшеу.UrlsAndRoutes қалтасындағы cs

```
usingSystem;

using System.Collections.Generic;

using System.Linq;

usingSystem.Threading.Tasks;

usingMicrosoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

usingMicrosoft.Extensions.DependencyInjection;


namespace UrlsAndRoutes

{

public class Startup

{

public void ConfigureServices(IServiceCollection services)

{

services.AddMvc();

}

public void Configure(IApplicationBuilder app, IHostingEnvironment env)

{

app.UseDeveloperExceptionPage();

app.UseStatusCodePages();

app.UseStaticFiles();
```

```
app.UseMvc();

}

}

}
```

Модель класын құру

Осы тараудағы барлық күш-жігер сұрау URL мекен-жайларын әрекеттермен салыстыруға бағытталған. Маған қажет жалғыз модель класы сұранысты өңдеу үшін таңдалған контроллер мен әрекет әдісі туралы егжей-тегжейлі мәлімет береді. Мен Models қалтасын жасап, Result деп аталатын сынып файлын қостым.мен 15-2 Листингте көрсетілген сыныпты анықтау үшін қолданған cs.

Listing 15-2. Result файлының мазмұны.модельдер қалтасындағы cs

```
using System.Collections.Generic;

namespace UrlsAndRoutes.Models

{

public class Result

{

public string Controller { get; set; }

public string Action { get; set; }

public IDictionary<string, object> Data { get; } = new Dictionary<string, object>();

}

}
```

Controller және Action сипаттары сұраудың қалай өңделгенін көрсету үшін қолданылады, ал деректер сөздігі бағыттау жүйесі берген сұрау туралы басқа ақпаратты сақтау үшін қолданылады.

Контроллер мысалдарын жасау

Маршруттаудың қалай жұмыс істейтінін көрсету үшін маған бірнеше қарапайым контроллер қажет. Мен Controllers қалтасын жасадым және HomeController деп аталатын сынып файлын қостым.мазмұны 15-3 Листингте көрсетілген cs.

Listing 15-3. HomeController файлының мазмұны.контроллер қалтасындағы cs

```
using Microsoft.AspNetCore.Mvc;

using UrlsAndRoutes.Models;

namespace UrlsAndRoutes.Controllers

{

public class HomeController : Controller

{

public ViewResult Index() => View("Result",

new Result {

Controller = nameof(HomeController),

Action = nameof(Index)

});

}

}
```

Home контроллері анықтаған index әрекет әдісі нәтиже деп аталатын көріністі визуализациялау үшін Көрініс әдісін шақырады (Мен оны келесі бөлімде анықтаймын) және нәтиже нысанын модель нысаны ретінде ұсынады. Модель объектісінің қасиеттері nameof функциясы арқылы орнатылады және сұрау үшін қандай контроллер мен әрекет әдісі қолданылғанын көрсету үшін қолданылады.

Мен CustomerController файлын қосу арқылы сол үлгіні ұстандым.CS controllers қалтасында және оны 15-4 листингінде көрсетілген Customer контроллерін анықтау үшін пайдаланыңыз.

Listing 15-4. CustomerController файлының мазмұны.контроллер қалтасындағы cs

```
using Microsoft.AspNetCore.Mvc;

using UrlsAndRoutes.Models;

namespace UrlsAndRoutes.Controllers

{
```

```
public class CustomerController : Controller
{
public ViewResult Index() => View("Result",
new Result {
Controller = nameof(CustomerController),
Action = nameof(Index)
});
public ViewResult List() => View("Result",
new Result {
Controller = nameof(CustomerController),
Action = nameof(List)
});
}
}
```

Үшінші және соңғы контроллер AdminController деп аталатын файлда анықталған.мен 15-5 Листингте көрсетілгендей Controllers қалтасына қосқан cs. Ол басқа контроллерлермен бірдей схеманы ұстанады.

Listing 15-5. AdminController файлының мазмұны.контроллер қалтасындағы cs

```
using Microsoft.AspNetCore.Mvc; using UrlsAndRoutes.Models;
namespace UrlsAndRoutes.Controllers
{
public class AdminController : Controller {
public ViewResult Index() => View("Result",
new Result {
Controller = nameof(AdminController),
Action = nameof(Index)
});
```

```
}
}
```

Көріністі жасау

Мен барлық контроллерлерге қол жетімді болатын бір көріністі жасауға мүмкіндік беретін алдыңғы бөлімде анықталған барлық әрекет әдістерінде нәтиже көрінісін ұсындым. Мен Views / Shared қалтасын жасап, оған жаңа Result түрін қостым.мазмұны 15-6 Листингте көрсетілген cshtml.

Listing 15-6. Result файлының мазмұны.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>

</body>

</html>
```

Көріністе Bootstrap көмегімен стильдендірілген кестеде модель объектісінің қасиеттерін көрсететін кесте бар. Жобаға Bootstrap қосу үшін мен bower файлын құру үшін Bower конфигурация файлы элементінің шаблонын қолдандым.json және Bootstrap пакетін 15-7 Листингте көрсетілгендей тәуелділік бөліміне қосты.

Listing 15-7. Жүктеу бумасын bower файлына қосу.UrlsAndRoutes қалтасындағы json

```
{

"name": "asp.net",

"private": true,

"dependencies":

{

"bootstrap": "4.0.0-alpha.6"

}

}
```

Соңғы дайындық _ViewImports файлын жасау болып табылады.views қалтасындағы cshtml, ол Razor көріністерінде пайдалану үшін кірістірілген тег көмекшілерін орнатады және 15-8 Листингте көрсетілгендей модельдік аттар кеңістігін импорттайды.

Listing 15-8. _ViewImports файлының мазмұны.көрініс қалтасындағы cshtml

```
@using UrlsAndRoutes.Models

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

Startup класындағы конфигурацияда MVC HTTP сұрауларын контроллерлер мен әрекеттермен қалай сәйкестендіретіні туралы нұсқаулар жоқ. Бағдарламаны іске қосқан кезде сіз сұраған кез-келген URL мекен - жайы 15-1 суретте көрсетілгендей 404-NotFound жауабына әкеледі.