

Қосымшада фильтрді пайдалану

Бұл тарауда мен соңғы тараулардағыдай мысал қосымшасын құрудың бірдей тәсілін қолдандым. Мен ASP шаблонын қолдандым.NETCoreWebApplication (.Netcore) Filters деп аталатын жаңа бос жобаны жасау үшін. 19-1 листингінде MVC инфрақұрылымын және дамуға қажет басқа аралық бағдарламалық жасақтаманы қосу үшін Startup сыныбына енгізген өзгерістерім көрсетілген.

Листинг 19-1. Startup файлының мазмұны.сүзгілер 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;

namespace Filters {

public class Startup {

public void ConfigureServices(IServiceCollection services) {

services.AddMvc();

}

public void Configure(IApplicationBuilder app, IHostingEnvironment env) {

app.UseStatusCodePages();

app.UseDeveloperExceptionPage();

app.UseStaticFiles();

app.UseMvcWithDefaultRoute();

}

}

}
```

SSL қосу

Осы тараудағы кейбір мысалдар әдепкі бойынша өшірілген SSL қолдануды талап етеді. SSL қосу үшін Visual Studio жобасы мәзірінен сүзгі сипаттарын таңдап, 19-1 суретте көрсетілгендей күйін келтіру қойындысындағы SSL қосу құсбелгісін қойыңыз. Әр жоба үшін әр түрлі болатын тағайындалған портты жазыңыз.

Сурет 19-1. SSL қосу

Контроллер мен көріністі жасау

Бұл тараудағы контроллерлер қарапайым, өйткені басты назар логиканы қосымшаның басқа жеріне орналастыруға бағытталған. Мен Controllers қалтасын жасадым, HomeController деп аталатын сынып файлын қостым.cs және оны 19-2 листингінде көрсетілген контроллерді анықтау үшін қолданды.

Листинг 19-2. HomeController файлының мазмұны.контроллер қалтасындағы cs

```
using Microsoft.AspNetCore.Mvc;

namespace Filters.Controllers {

public class HomeController : Controller {

public IActionResult Index() => View("Message",

"This is the Index action on the Home controller");

}

}
```

Әрекет әдісі message деп аталатын көріністі визуализациялайды және жолды көрініс деректері ретінде жібереді. Мен Views / Shared қалтасын жасадым және хабарлама деп аталатын Razor көрініс файлын қостым.19-3 әсерін көрсететін таңбамен cshtml.

Листинг 19-3. Message файлының мазмұны.көрініс қалтасындағы cshtml / ортақ

```
@{ Layout = null; }

<!DOCTYPE html>

<html>

<head>

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

<title>Filters</title>

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

</head>
```

```

<body class="m-1 p-1">
  @if (Model is string) {
    @Model
  } else if (Model is IDictionary<string, string>) {
    var dict = Model as IDictionary<string, string>;
    <table class="table table-sm table-striped table-bordered">
      <thead><tr><th>Name</th><th>Value</th></tr></thead>
      <tbody>
        @foreach (var kvp in dict) {
          <tr><td>@kvp.Key</td><td>@kvp.Value</td></tr>
        }
      </tbody>
    </table>
  }
</body>
</html>

```

Бұл көрініс нашар терілген және жолды немесе сөздікті <жол, жол> көрсетеді, бұл жағдайда кесте көрсетіледі. Көрініс HTML элементтерін сәндеуге арналған bootstrap CSS пакетіне байланысты. Жобаға Bootstrap қосу үшін мен bower файлын құру үшін Bower конфигурация файлы элементінің шаблонын қолдандым.json жобаның түбірлік қалтасында және 19-4 листингінде көрсетілгендей тәуелділік бөліміне Bootstrap пакетін қосты.

Листинг 19-4. Bootstrap бумасын bower файлына қосу.JSON "сүзгілер" қалтасында»

```

{
  "name": "asp.net",
  "private": true,
  "dependencies": {
    "bootstrap": "4.0.0-alpha.6"
  }
}

```

Соңғы дайындық `_ViewImports` файлын жасау болып табылады. 19-5 Листингте көрсетілгендей, Razor көріністерінде пайдалану үшін кірістірілген тег көмекшілерін орнататын Views қалтасындағы `cshtml`.

Листинг 19-5. `_ViewImports` файлының мазмұны. көрініс қалтасындағы `cshtml`

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

Егер сіз қосымшаны іске қоссаңыз, 19-2 суретте көрсетілген нәтижені көресіз.

Сурет 19-2. Қолданба мысалын іске қосу

■ ■ Кеңес. Сізден Visual Studio жасаған сертификатқа сену сұралуы мүмкін. SSL пайдаланатын осы тараудағы мысалдарға қатысты осы опцияны қабылдаңыз.

Сүзгілерді пайдалану

Сүзгілер әрекет әдісінде басқаша қолданылатын логиканы Контроллерден алып тастауға және қайта пайдалану класында анықтауға мүмкіндік береді. Мысал ретінде мен әдеттегі шифрланбаған HTTP емес, тек HTTPS көмегімен әрекет ету әдістеріне қол жеткізгім келді деп елестетіп көріңіз. `HttpRequest` мәтінмен нысаны HTTPS 19-6 листингінде көрсетілгендей пайдаланылатындығын анықтау үшін қажетті ақпаратты ұсынады.

Листинг 19-6. `HomeController` файлында HTTPS тестілеу. контроллер қалтасындағы `cs`

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Http;
namespace Filters.Controllers {
public class HomeController : Controller {
public IActionResult Index() {
if (!Request.IsHttps) {
return new StatusCodeResult(StatusCode.Status403Forbidden);
} else {
return View("Message",
"This is the Index action on the Home controller");
}
}
}
}
```

Міне, сіз https проблемасына сүзгілерсіз қалай жүгінесіз. Егер сіз қосымшаны іске қоссаңыз, сіздің шолғышыңыз Index әрекет ету әдісі жұмыс істейтін жоба үшін HTTP-ден басқа әдепкі URL мекенжайын сұрайды, жауап ретінде HTTP 403 күй кодын жіберетін StatusCodeResult қайтарады (17-тарауда сипатталғандай). Егер сіз әдепкі HTTPS URL мекенжайын сұрасаңыз (мен үшін бұл `https://localhost:44318`), Index әрекет ету әдісі хабарламаны көрсету арқылы жауап береді (шолғыш нәтижені көрсетпес бұрын қауіпсіздік ескертуін растау қажет болуы мүмкін). 19-3 сурет екі нәтижені де көрсетеді.

Сурет 19-3. HTTPS сұрауларына кіруді шектеу

■■ Кеңес. Егер сіз осы бөлімдегі мысалдардан күтілетін нәтиже алмаған болсаңыз, шолғыш тарихын тазалаңыз.

Браузерлер көбінесе SSL қателерін жасаған серверлерге сұраулар жіберуден бас тартады, бұл жақсы қауіпсіздік тәжірибесі, бірақ оны бұзуы мүмкін бас әзірлеу кезінде.

19-6 тізіміндегі Код жұмыс істейді, бірақ проблемалары бар. Бірінші мәселе-әрекет әдісі сұранысты өңдеуге, модельді жаңартуға және жауапты таңдауға қарағанда қауіпсіздік саясатын жүзеге асыруға көбірек қатысты кодты қамтиды. Неғұрлым маңызды мәселе-HTTP анықтау кодын әрекет әдісіне қосу өте жақсы масштабталмайды және 19-7 Листингте көрсетілгендей контроллердегі әр әрекет әдісінде қайталануы керек.

Листинг 19-7. HomeController файлына әрекет әдісін қосу. контроллер қалтасындағы cs

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
namespace Filters.Controllers {
public class HomeController : Controller {
public IActionResult Index() {
if (!Request.IsHttps) {
return new StatusCodeResult(StatusCode.Status403Forbidden);
} else {
return View("Message",
"This is the Index action on the Home controller");
}
}
public IActionResult SecondAction() {
if (!Request.IsHttps) {
```

```

return new StatusCodeResult(StatusCode.Status403Forbidden);

} else {

return View("Message",

"This is the SecondAction action on the Home controller");

}

}

}

}

```

Мен HTTPS сұрағым келетін әр контроллердегі әр әрекет әдісінде бірдей тексеруді жүзеге асыруды ұмытпауым керек. Қауіпсіздік саясатын іске асыру коды-бұл қарапайым контроллердің маңызды бөлігі, ол контроллерді түсінуді қиындатады,және мен оны жаңа әрекет әдісіне қосуды ұмытып, тесік жасауды ұмытып кетемін. Менің қауіпсіздік саясатым. Бұл 19-8 листингінде көрсетілгендей сүзгілер жұмыс істей алатын мәселелер.

Листинг 19-8. HomeController файлында сүзгіні қолдану. контроллер қалтасындағы cs

```

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

namespace Filters.Controllers {

public class HomeController : Controller {

[RequireHttps]

public IActionResult Index() => View("Message",

"This is the Index action on the Home controller");

[RequireHttps]

public IActionResult SecondAction() => View("Message",

"This is the SecondAction action on the Home controller");

}

}

```

Requirehttps атрибуты HomeController класына кіріктірілген сүзгілердің бірін қолданады. Бұл тек HTTPS сұрауларына қолдау көрсету үшін әрекет әдістеріне қол жетімділікті шектейді және маған әр әдіс бойынша қауіпсіздік кодын алып тастауға және сәтті сұрауларды өңдеуге назар аударуға мүмкіндік береді.