

Қосымшаны конфигурациялау

Listing 14-1. Жүктеу жолағын жүктеу файлына қосу.JSON ConfiguringApps қалтасында

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

Содан кейін мен Controllers қалтасын жасап, HomeController деп аталатын сынып файлын қостым.мен 14-2 листингінде көрсетілген контроллерді анықтау үшін қолданған cs.

Listing14-2. HomeController файлының мазмұны.контроллер қалтасындағы cs

```
using System.Collections.Generic;  
using Microsoft.AspNetCore.Mvc;  
  
namespace ConfiguringApps.Controllers  
{  
  public class HomeController : Controller  
  {  
    public IActionResult Index() => View(new Dictionary<string, string>  
    {  
      ["Message"] = "This is the Index action"  
    });  
  }  
}
```

```
}  
}
```

Мен Views / Home қалтасын жасадым және Index деп аталатын көрініс файлын қостым.мазмұны 14-3 Листингте көрсетілген cshtml.

Listing 14-3. Индекс файлының мазмұны.көрініс / үй қалтасындағы cshtml

```
@model Dictionary<string, string>  
  
@{  
    Layout = null;  
}  
  
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<meta name="viewport" content="width=device-width" />  
<link asp-href-include="lib/bootstrap/dist/css/*.min.css" rel="stylesheet"/>  
<title>Result</title>  
  
</head>  
  
<body class="p-1">  
  
<table class="table table-condensed table-bordered table-striped">  
    @foreach (var kvp in Model)  
    {  
        <tr>  
            <th>@kvp.Key</th>  
            <td>@kvp.Value</td>  
        </tr>  
    }  
</table>
```

```
</body>
```

```
</html>
```

Көріністері сілтеме элементі Bootstrap CSS файлдарын таңдау үшін кірістірілген тег көмекшісін пайдаланады. Кірістірілген тег көмекшілерін қосу үшін мен `_viewimports` құру үшін `MVCViewImportsPage` элементінің шаблонын қолдандым. `Cshtml` файлы `Views` қалтасында және 14-4 Листингте көрсетілген өрнекті қосты.

Listing 14-4. `_ViewImports` файлының мазмұны.көрініс қалтасындағы `cshtml`

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

Бағдарламаны іске қосыңыз, сонда сіз 14-1 суретте көрсетілген хабарламаны көресіз.

Жобаны орнату

Ең маңызды конфигурация файлы- `< жоба_ атауы >` .жоба файлы алмастыратын `csproj`.алдыңғы нұсқаларда қолданылатын `json` ASP.NET Core. Бұл файл `ConfiguringApps` деп аталады.`Csproj` жоба мысалында `Visual Studio` жасырылған және оған кіру үшін шешім шолғышының терезесіндегі жоба элементін тінтуірдің оң жақ түймесімен нұқып, " `ConfiguringApps` өңдеу.`csproj` " қалқымалы мәзірде. 14-5 листингі `ConfiguringApps` файлының бастапқы мазмұнын көрсетеді.`Visual Studio` бос жоба үлгісінің бөлігі ретінде жасаған `csproj`.

Listing 14-5. `ConfiguringApps` файлының мазмұны.`Csproj` `ConfiguringApps` қалтасында

```
<Project Sdk="Microsoft.NET.Sdk.Web">
```

```
<PropertyGroup>
```

```
<TargetFramework>netcoreapp2.0</TargetFramework>
```

```
</PropertyGroup>
```

```
<ItemGroup>
```

```
<Folder Include="wwwroot\" />
```

```
</ItemGroup>
```

```
<ItemGroup>
```

```
<PackageReference Include="Microsoft.AspNetCore.All" Version="2.0.0" /></ItemGroup>
```

```
</Project>
```

Csproj файлы .NET жобаларын жасау үшін қолданылатын MSBuild құралын конфигурациялау үшін қолданылады. Конфигурация XML элементтерін қолдана отырып жасалады және 14-3-кестеде әдепкі конфигурация файлындағы элементтер сипатталған. Кейінгі мысалдарда мен басқа конфигурация элементтерін қолданамын, бірақ кестедегі элементтер ASP жобасында дамуды бастау үшін жеткілікті.NETCoreMVC.

Кесте 14-3. Әдепкі csproj файл элементінің сипаттамасындағы XML конфигурация элементтері

Элемент Сипаттамасы

Жоба бұл MSBuild конфигурация файлы екенін білдіретін түбірлік элемент. Sdk төлсипаты Microsoft.NET.Sdk.Web-ке жобаны құру үшін қажет пакеттердің жасырын импорт жиынтығын ұсынады.

PropertyGroup бұл элемент файлға құрылым қосу үшін байланысты конфигурация қасиеттерін топтайды.

TargetFramework бұл элементті көрсетеді .NETFramework, ол құрастыру процесіне арналған және PropertyGroup элементінде анықталуы керек. Әдепкі мән netcoreapp2.0 болып табылады, ол арналған .NETCore 2.0.

ItemGroup бұл элемент файлға құрылым қосу үшін байланысты конфигурация элементтерін топтайды.

Folder бұл элемент MSBuild-ге жобадағы қалтаны қалай басқаруға болатындығын хабарлайды. Тізімдегі Элемент көрсетеді MSBuild қосымшаны жариялау кезінде wwwroot қалтасын қосу үшін.

PackageReference бұл элемент Nuget пакетіне тәуелділікті көрсету үшін қолданылады, ол Include және Version атрибуттары арқылы анықталады. Microsoft.AspNetCore. Бүкіл пакет ASP функционалдығын қамтамасыз ететін барлық жеке пакеттерге қол жетімділікті қамтамасыз ету үшін қолданылады.NETCore және MVCFramework.

Жобаға пакеттерді қосу

Csproj файлы үшін ең маңызды рөл - бұл жоба байланысты пакеттерді тізімдеу. VisualStudio csproj файлындағы өзгерісті анықтаған кезде, ол пакеттер тізімін тексереді, барлық жаңа қондырмаларды жүктейді және қажет емес барлық пакеттерді жояды. ASP шығарумен.NETCore 2 ASP үшін қажетті барлық негізгі мүмкіндіктер.NETCoreMVC, MVCFramework және EntityFrameworkCore Microsoft метапакетіне кіреді.AspNetCore.Барлығы, бұл жобаға NuGet пакеттерінің ұзақ тізімін қосу арқылы жаңа дамуды бастауға жол бермейтін ыңғайлы мүмкіндік. Алайда, сізге үшінші тарап немесе жетілдірілген мүмкіндіктер үшін NuGet пакеттерін қосу қажет болады. Жобаға пакеттерді қосудың үш әдісі бар. Алдымен құралдарды таңдаңыз менеджер

пакет менеджері Nuget ➤ басқару Nuget пакеттерін қарапайым интерфейс арқылы басқаруға мүмкіндік беретін шешім үшін. Егер сіз. NET-ті дамыту үшін жаңадан болсаңыз, онда бұл ең жақсы тәсіл, өйткені ол пакетті таңдау кезінде қате пайда болу мүмкіндігін азайтады. Мысалы, HTTP сұрауларын орындауға (алуға емес) қолдау көрсететін System.Net.Http пакетін қосу үшін Сіз пакет менеджерінің шолу бөліміне өтіп, аты бойынша іздеп, нұсқалардың толық тізімін көре аласыз. 14-2 суретте көрсетілгендей кез-келген алдын-ала нұсқаларды қоса алғанда, қол жетімді.

Қажетті пакет пен нұсқаны таңдаңыз, пакетті қажет ететін жобаларды таңдап, "Орнату" түймесін басыңыз. Visual Studio пакетті жүктейді және csproj файлына жаңартады. Сондай-ақ, пакеттерді пәрмен жолымен қосуға болады, дегенмен сізге қажет пакеттің атын (және ең дұрысы нұсқасын) білу керек. 14-6-Листингте жобаға System.Net.Http пакетін қосу үшін ConfiguringApps қалтасында жұмыс істейтін команда көрсетілген.

Listing 14-6. Жобаға пакет қосу

```
dotnet add package System.Net.Http --version 4.3.2
```

NuGet пакет менеджері және dotnet бумасын қосу командасы csproj файлына PackageReference элементтерін қосады. Қажет болса, packagereference элементін қолмен қосу үшін конфигурация файлына өңдеу арқылы пакеттерді қосуға болады. Бұл ең тікелей тәсіл, бірақ пакет атауын дұрыс енгізу немесе жоқ нұсқа нөмірін көрсетпеу үшін абай болу керек. 14-7 Листингте Сіз system.Net.Http пакетіне арналған csproj файлына қосымшаны көре аласыз.

Listing 14-7. Пакетті ConfiguringApps файлына қосу.Csproj ConfiguringApps қалтасында

```
<Project Sdk="Microsoft.NET.Sdk.Web">
<PropertyGroup>
<TargetFramework>netcoreapp2.0</TargetFramework>
</PropertyGroup>
<ItemGroup>
<Folder Include="wwwroot\" />
</ItemGroup>
<ItemGroup>
<PackageReference Include="Microsoft.AspNetCore.All" Version="2.0.0" /> <PackageReference
Include="System.Net.Http" Version="4.3.2" />
```

</ItemGroup>

PackageReference элементінде пакеттің атауын көрсететін include төлсіпаты және нұсқа нөмірін көрсететін Version төлсіпаты бар.