

Қосымшада контроллерлер мен әрекеттерді қолдану

Веб-қосымшаның осы тарауында мен негізгі үлгіні қолдандым ASP.NET (.ControllersAndActions деп аталатын жаңа бос жобаны құру үшін. 17-1 Листингте мен MVC инфрақұрылымын қосу және басқа аралық бағдарламалық жасақтама компоненттерін қосу үшін Іске қосу класына операторлар қостым.

Ескерту. Бұл тарауда негізгі функциялардың бірлік сынақтары бар. Қысқаша болу үшін мен үлгі жобасын құру нұсқауларына блок-тест жобасын қоспадым. Сіз 7-тарауда сипатталған процесті орындау арқылы сынақ жобасын жасай аласыз немесе осы кітаптың Git Hub репозиторийінен жобаны жүктей аласыз (<https://github.com / apress / Pro-ASP.net-core-mvc-2>).

Листинг 17-1. Startup файлына MVC және басқа аралық бағдарламалық жасақтаманы қосу. CS ControllersAndActions Қалтасы

```
using System;
```

```
using System.Коллекциялар.Жалпы;
```

```
жүйесін пайдалана отырып.LINQ сұрау;
```

```
жүйені пайдалану.Бұранданы кесу.Міндеттері;
```

```
Microsoft пайдалану.AspNetCore.Құрылысшы;
```

```
Microsoft пайдалану.AspNetCore.Хостинг;
```

```
Microsoft пайдалану.AspNetCore.HTTP хаттамасы;
```

```
Microsoft пайдалану.Кеңейту.DependencyInjection;
```

```
ControllersAndActions аттар кеңістігі {
```

```
public class Startup {
```

```
public void ConfigureServices(IServiceCollection services) {
```

```
services.AddMvc();
```

```
қызмет.AddMemoryCache();
```

```
қызмет.AddSession();
```

```
}
```

```
қоғамдық жарамсыз теңшеу (IApplicationBuilder app, IHOSTINGENVIRONMENT OKR) {
```

```
- қосымша.UseStatusCodePages();
```

```
- қосымша.UseDeveloperExceptionPage();
```

```
- қосымша.UseStaticFiles();  
  
- қосымша.UseSession();  
  
- қосымша.UseMvcWithDefaultRoute();  
  
}  
  
}  
  
}
```

AddMemoryCache және AddSession әдістері тәжірибені басқару үшін қажет Қызметтерді жасайды. UseSession әдісі құбырға аралық бағдарламалық жасақтама компонентін қосады, ол сеанс деректерін сұраулармен байланыстырады және болашақ сұраныстарды анықтау үшін жауаптарға куки файлдарын қосады. UseSession әдісі UseMvc әдісінің алдында шақырылуы керек, осылайша сеанс компоненті MVC аралық бағдарламалық жасақтамасына жеткенге дейін сұраныстарды ұстап алады және олар жасалғаннан кейін жауаптарды өзгерте алады. Басқа әдістер мен 14-тарауда сипатталған стандартты пакеттерді орнатады.

Көріністерді дайындау

Бұл тарауда контроллерлер мен олардың әрекет ету әдістеріне баса назар аударылады, мен осы тарауда контроллер кластарын анықтаймын. Бұған дайындалу үшін мен олардың қалай жұмыс істейтінін көрсетуге көмектесетін бірнеше идеяларды анықтаймын. Мен осы бөлімде жасаған көріністер қалтада анықталған көріністер / жалпы, сондықтан мен оларды кейінірек осы тарауда жасаған кез-келген Контроллерден пайдалана аламын. Мен қарау / бөлісу қалтасын жасадым, оған нәтиже деп аталатын Razor көрініс файлын қостым. cshtml және 17-2 листингінде көрсетілген түзетуді қолданды.

Листинг 17-2.Содержимоефайла Result.cshtml впаке қарау / ортақ пайдалану

```
@model жолы
```

```
@{
```

```
Layout = null;
```

```
}
```

```
<!DOCTYPE html>
```

```
< HTML > және
```

```
<басы>
```

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

```
< title>контроллерлер мен әрекеттер< / title>
```

```
<сілтеме отн=" стиль кестесі "АСП-Солар, - қамтиды=" Либ / өзін-өзі жүктеу / қашықтық / CSS немесе*.CSS-те мин." / >
```

```
</head>
```

```
<шанақ класы="m-1 p-1">
```

```
Үлгі деректері: @Model
```

```
</body>
```

```
тег < / HTML >
```

Бұл көріністің Моделі-бұл қарапайым хабарламаларды көрсетуге мүмкіндік беретін жол. Содан кейін мен DictionaryResult файлын жасадым.cshtml бағдарламалық жасақтамасы, сонымен қатар қарау / жалпы қалтада орналасқан және 17-3 Листингте көрсетілген түзетуді қосты. Бұл көрініс үшін бұл модель алдыңғы көрініске қарағанда күрделі деректерді көрсететін сөздік болып табылады.

Листинг 17-3.Мазмұны DictionaryResult.cshtml впаке қарау / ортақ пайдалану

```
@model IDictionary<string, string>
```

```
@{
```

```
Layout = null;
```

```
}
```

```
<!DOCTYPE html>
```

```
< HTML > және
```

```
<басы>
```

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

```
< title>контроллерлер мен әрекеттер< / title>
```

```
<сілтеме отн=" стиль кестесі "АСП-Солар, - қамтиды=" Либ / өзін-өзі жүктеу / қашықтық / CSS немесе*.CSS-те мин." / >
```

```
</head>
```

```
<шанақ класы="m-1 p-1">
```

```
<сынып кестесі= "кесте кесте-разграфленная кесте-см кесте-жолақ">
```

```
<tr>
```

```
<th>аты</th >
```

```
тег <Th > мәні< / Th > болады
```

```
</tr>
```

```
@foreach (модельдегі жол кілті.Кілттер) {
```

```
<tr>
```

```
<td>@key</td>
```

```
<td>@Model[key]</td>
```

```
</tr>
```

```
}
```

```
</table>
```

```
</body>
```

```
тег < / HTML >
```

Содан кейін мен SimpleForm деп аталатын файл жасадым.cshtml бағдарламалық жасақтамасы, сонымен қатар қарау / жалпы қалтада орналасқан және оны 17-4 Листингте көрсетілген көріністі анықтау үшін қолданған. Атауынан көрініп тұрғандай, бұл көріністе қарапайым HTML формасы бар, ол Пайдаланушыдан деректерді жинайды.

Листинг 17-4.Мазмұныфайл SimpleForm.cshtml впапке қарау / ортақ пайдалану

```
@{
```

```
Layout = null;
```

```
}
```

```
<!DOCTYPE html>
```

```
< HTML > және
```

```
<басы>
```

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

```
< title>контроллерлер мен әрекеттер< / title>
```

```
<сілтеме отн=" стиль кестесі "АСП-Солар, - қамтиды=" Либ / өзін-өзі жүктеу / қашықтық / CSS немесе*.CSS-те мин." / >
```

```
</head>
```

```
<шанақ класы="m-1 p-1">
```

```
<form method="post" asp-action="ReceiveForm">
```

```
<div class="form-group">
```

```
<label for="name">аты:< / label>
```

```
<input class="form-control" name="name" />
```

```
</div>
```

```
<div class="form-group">
```

```
<label for= "name" > қала:< / label>
```

```
<input class="form-control" name="city" />
```

```
</div>
```

```
<Класс="BTN BTN-бастапқы Орталық-блок "түрі= "жіберу" > Жіберу< / батырма>
```

```
</form>
```

```
</body>
```

```
тег < / HTML >
```

Көріністер маршруттау жүйесінен URL мекен-жайларын құру үшін кірістірілген тег көмекшілерін пайдаланады. Тег көмекшілерін қосу үшін мен `_viewImports` деп аталатын көріністі импорттау файлын жасадым. Cshtml Views қалтасында және 17-5 Листингте көрсетілген өрнекті қосты.

Листинг 17-5. `_ViewImports` файлының мазмұны. көрініс қалтасындағы cshtml

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

Мен қарау / бөлісу қалтасында жасаған барлық түрлер bootstrap CSS пакетіне байланысты. Жүктеу жобасына қосу үшін мен Бауэр файлын жасау үшін газебо конфигурация файлының шаблонын қолдандым.JSON және 17-6 листингінде көрсетілген пакет қосылды.

Листинг 17-6. Буманы bower файлына қосу.JSON ControllersAndActions қалтасында

```
{
```

```
"аты": "asp.net",
```

```
"private": true,
```

```
"dependencies": {
```

```
"bootstrap": "4.0.0-alpha.6"
```

```
}
```

```
}
```