Jon Hilton

From layers to vertical slices

Simplify your code and focus on your features

Remote

ABOUT US SERVICES CASE STUDIES INSIGHTS CAREERS CONTACT US

We build custom online software for purpose-driven teams

LEARN MORE

@jonhilt

Pinned Tweet

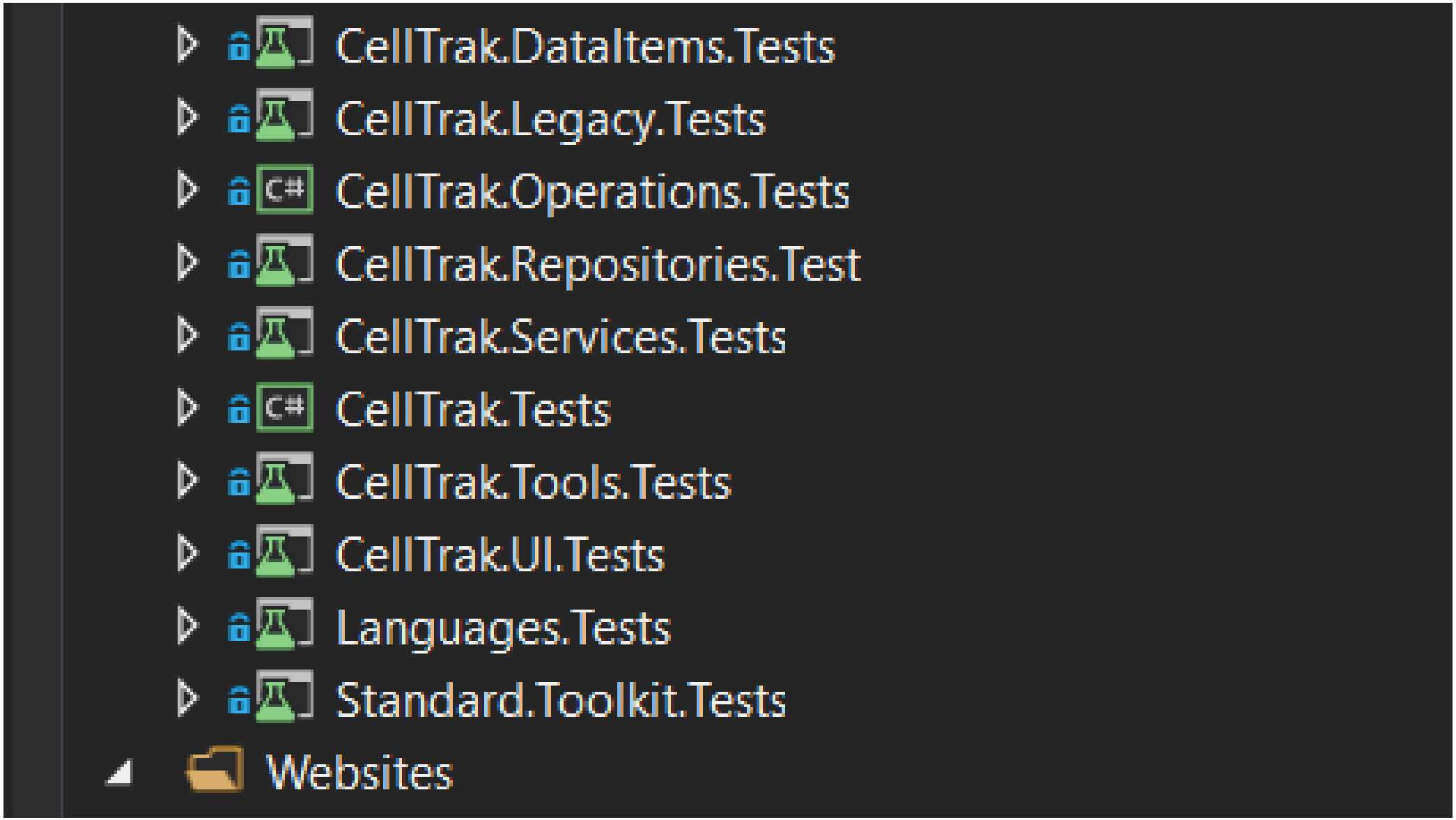
Stroke Active @StrokeActive · 19h

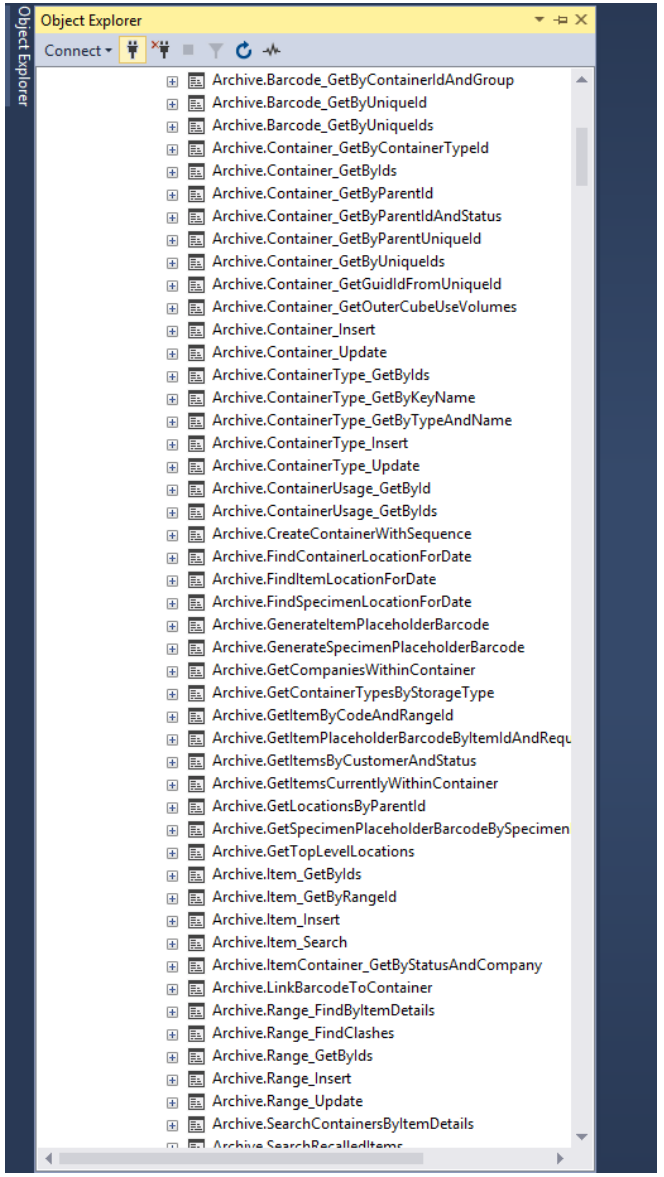
'Innovation of The Year' Thank you to all that voted. We are absolutely delighted to receive this award @NeuroConvention 2019. #winner #strokeactive #innovation

1 7

Could you just… ?


MIND THE GAP


CellTrak.DataItems.Tests
CellTrak.Legacy.Tests
CellTrak.Operations.Tests
CellTrak.Repositories.Test
CellTrak.Services.Tests
CellTrak.Tests
CellTrak.Tools.Tests
CellTrak.UI.Tests
Languages.Tests
Standard.Toolkit.Tests
Websites


Object Explorer
Object Explorer
Connect
Archive.Barcode_GetByContainerIdAndGroup
Archive.Barcode_GetByUniqueId
Archive.Barcode_GetByUniqueIds
Archive.Container_GetByContainerTypeId
Archive.Container_GetByIds
Archive.Container_GetByParentId
Archive.Container_GetByParentIdAndStatus
Archive.Container_GetByParentUniqueId
Archive.Container_GetByUniqueIds
Archive.Container_GetGuidIdFromUniqueId
Archive.Container_GetOuterCubeUseVolumes
Archive.Container_Insert
Archive.Container_Update
Archive.ContainerType_GetByIds
Archive.ContainerType_GetByKeyName
Archive.ContainerType_GetByTypeAndName
Archive.ContainerType_Insert
Archive.ContainerType_Update
Archive.ContainerUsage_GetById
Archive.ContainerUsage_GetByIds
Archive.CreateContainerWithSequence
Archive.FindContainerLocationForDate
Archive.FindItemLocationForDate
Archive.FindSpecimenLocationForDate
Archive.GenerateItemPlaceholderBarcode
Archive.GenerateSpecimenPlaceholderBarcode
Archive.GetCompaniesWithinContainer
Archive.GetContainerTypesByStorageType
Archive.GetItemByCodeAndRangeId
Archive.GetItemPlaceholderBarcodeByItemIdAndRequ
Archive.GetItemsByCustomerAndStatus
Archive.GetItemsCurrentlyWithinContainer
Archive.GetLocationsByParentId
Archive.GetSpecimenPlaceholderBarcodeBySpecimen
Archive.GetTopLevelLocations
Archive.Item_GetByIds
Archive.Item_GetByRangeId
Archive.Item_Insert
Archive.Item_Search
Archive.ItemContainer_GetByStatusAndCompany
Archive.LinkBarcodeToContainer
Archive.Range_FindByItemDetails
Archive.Range_FindClashes
Archive.Range_GetByIds
Archive.Range_Insert
Archive.Range_Update
Archive.SearchContainersByItemDetails

```
1 reference
public class PersonService
{
    private readonly IPersonRepository _personRepository;
    private readonly IAdminRepository _adminRepository;
    private readonly IRoleRepository _roleRepository;
    private readonly IRoleMapper _roleMapper;
    private readonly IUserManager _userManager;

    0 references | 0 exceptions
    public PersonService(IPersonRepository personRepository,
        IAdminRepository adminRepository,
        IRoleRepository roleRepository,
        IRoleMapper roleMapper,
        IUserManager userManager)
    {
        _personRepository = personRepository;
        _adminRepository = adminRepository;
        _roleRepository = roleRepository;
        _roleMapper = roleMapper;
        _userManager = userManager;
    }
```

Cohesion - the degree to which the elements inside a module belong together

Modules with **high cohesion** tend to be preferable, because high cohesion is associated with several desirable traits of software including robustness, reliability, reusability, and **understandability**

Wikipedia

Coupling is the degree of interdependence between software modules; a measure of how closely connected two routines or modules are.

Wikipedia

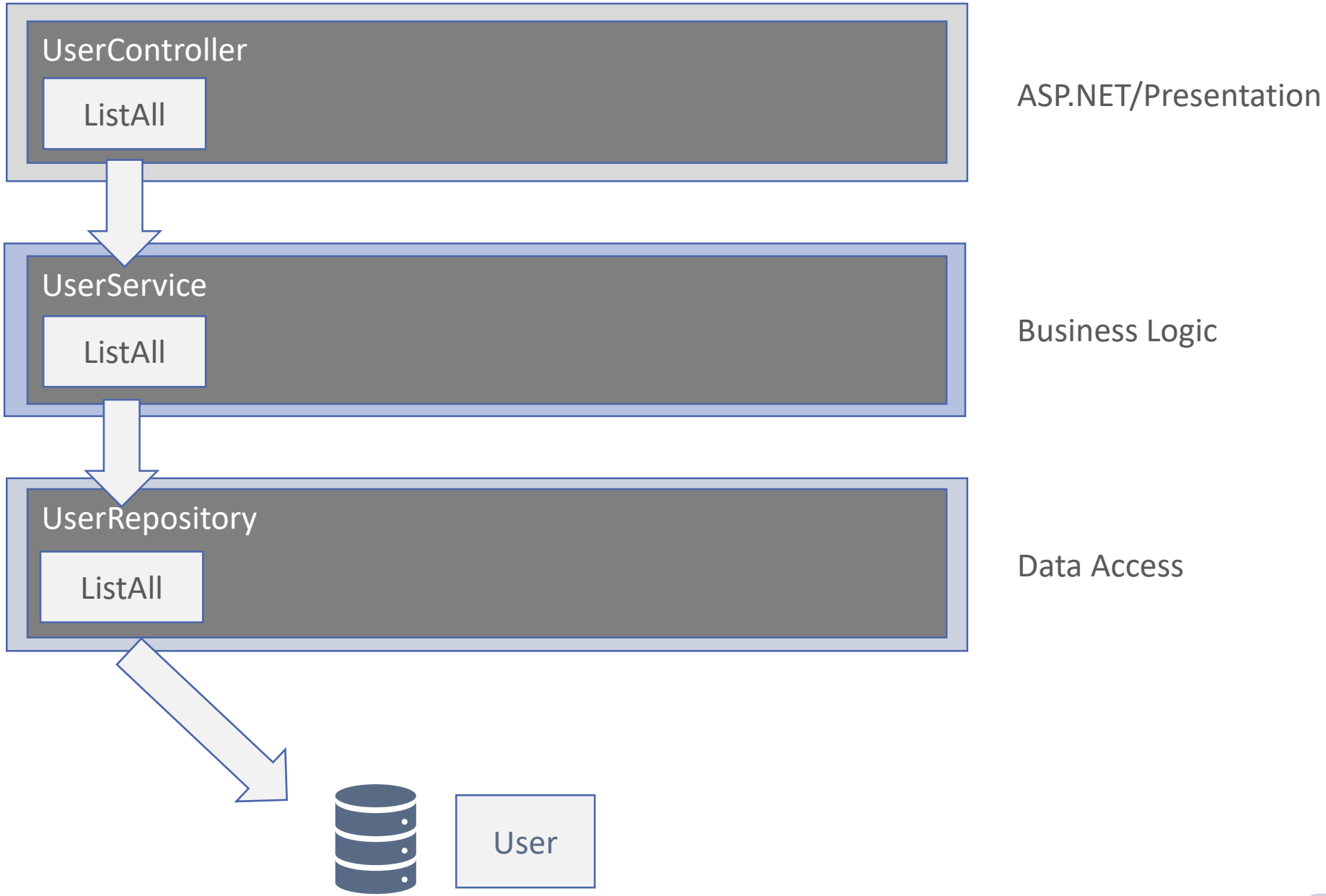
All Users

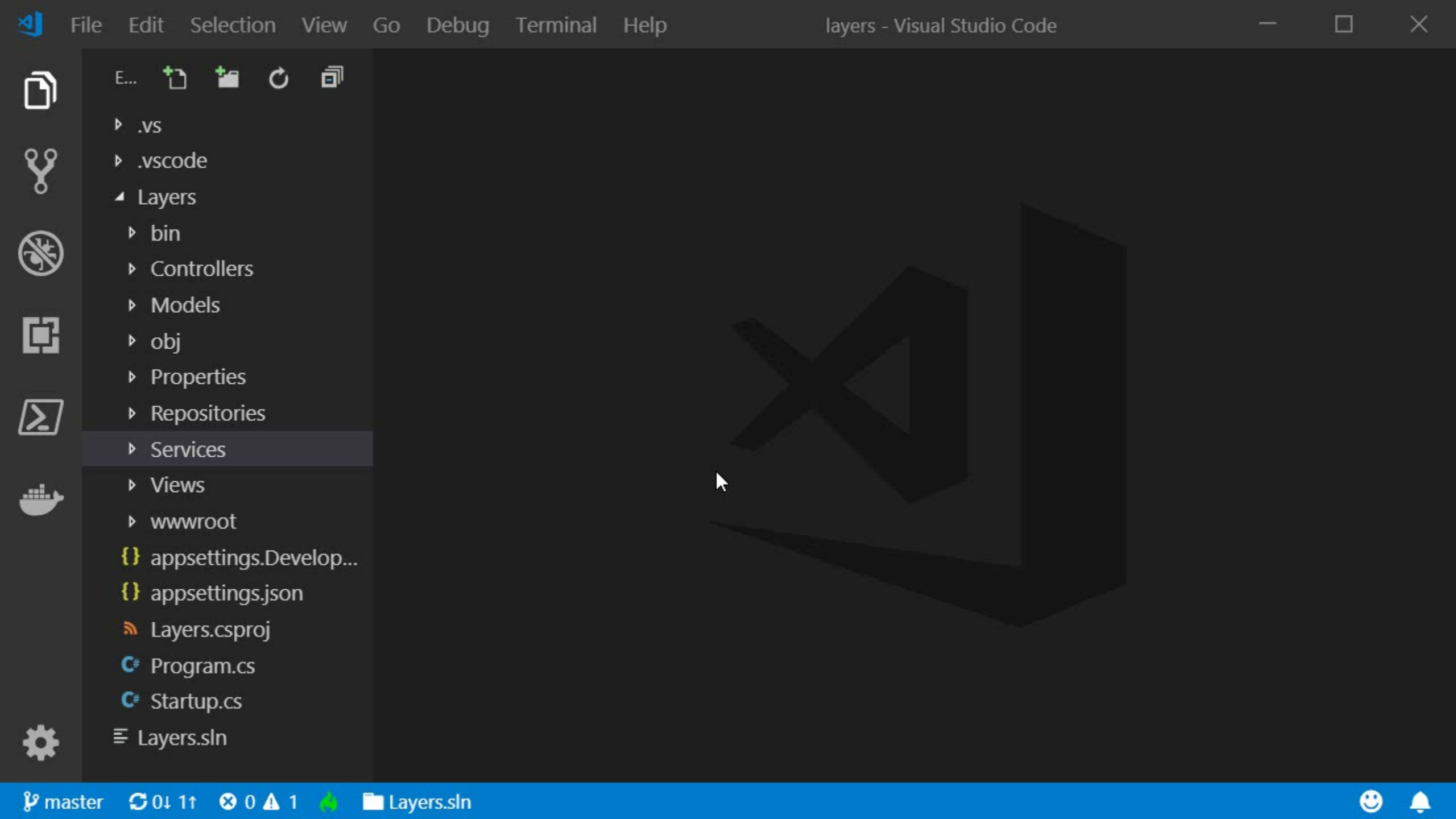
First Name	Last Name	Last Login	Active
Bob	Smith	01/02/2018	☑
James	Barrow	03/01/2019	☐
Janet	Jones	03/02/2018	☐
Ollie	Norburn	01/01/2018	☑
Jane	Tomley	05/01/~~208~~	☑

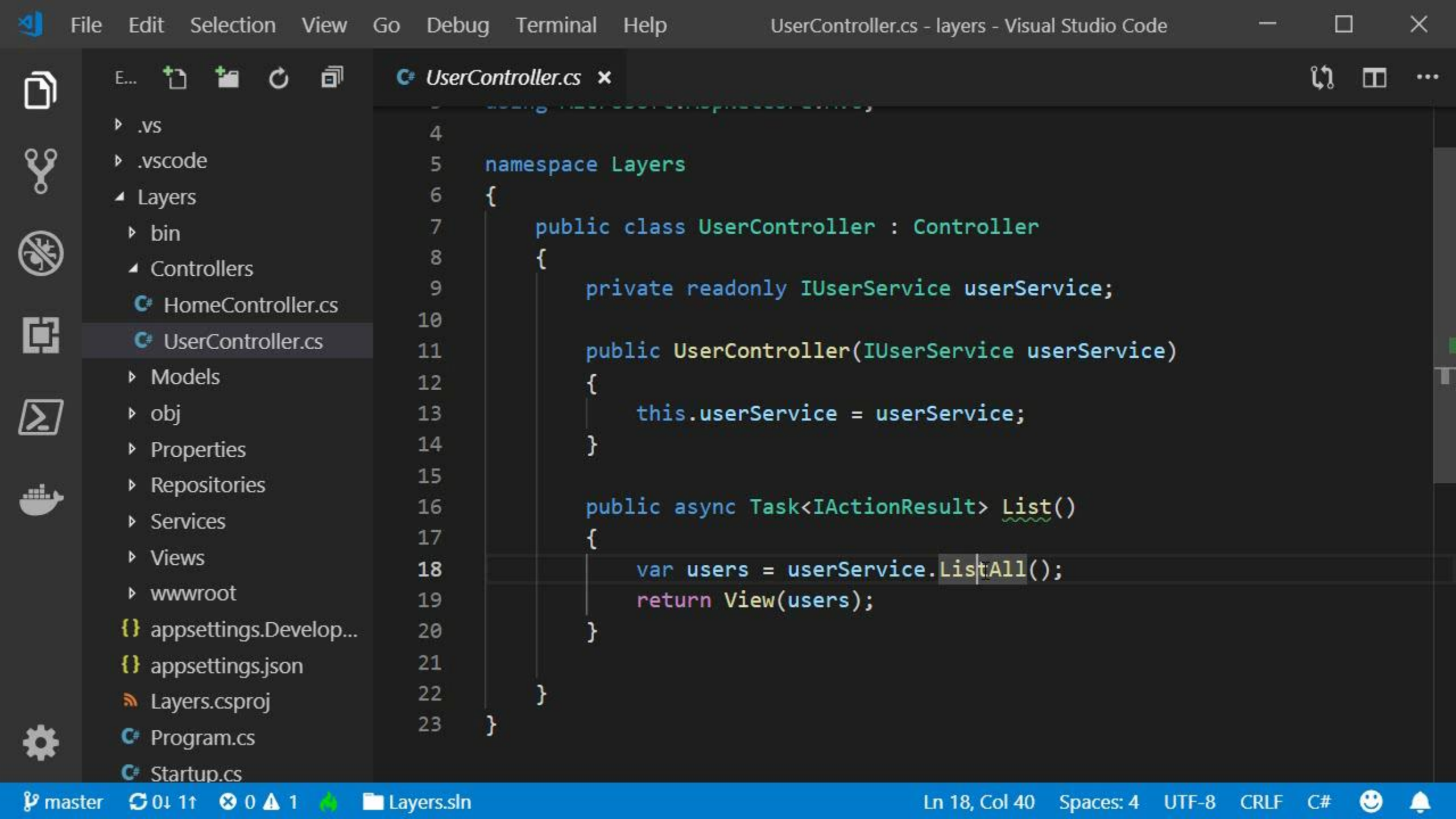
2018

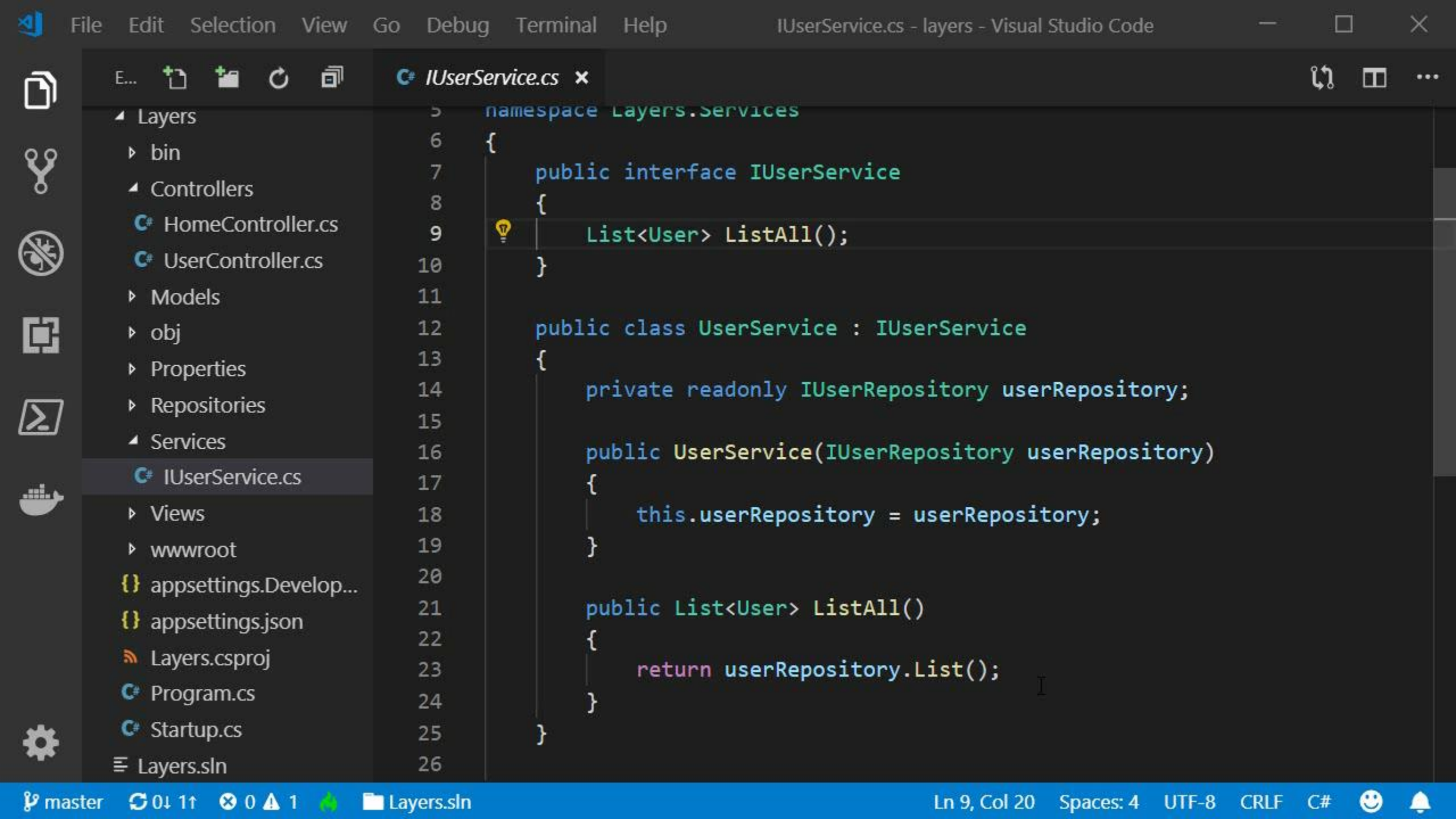
ASP.NET/Presentation

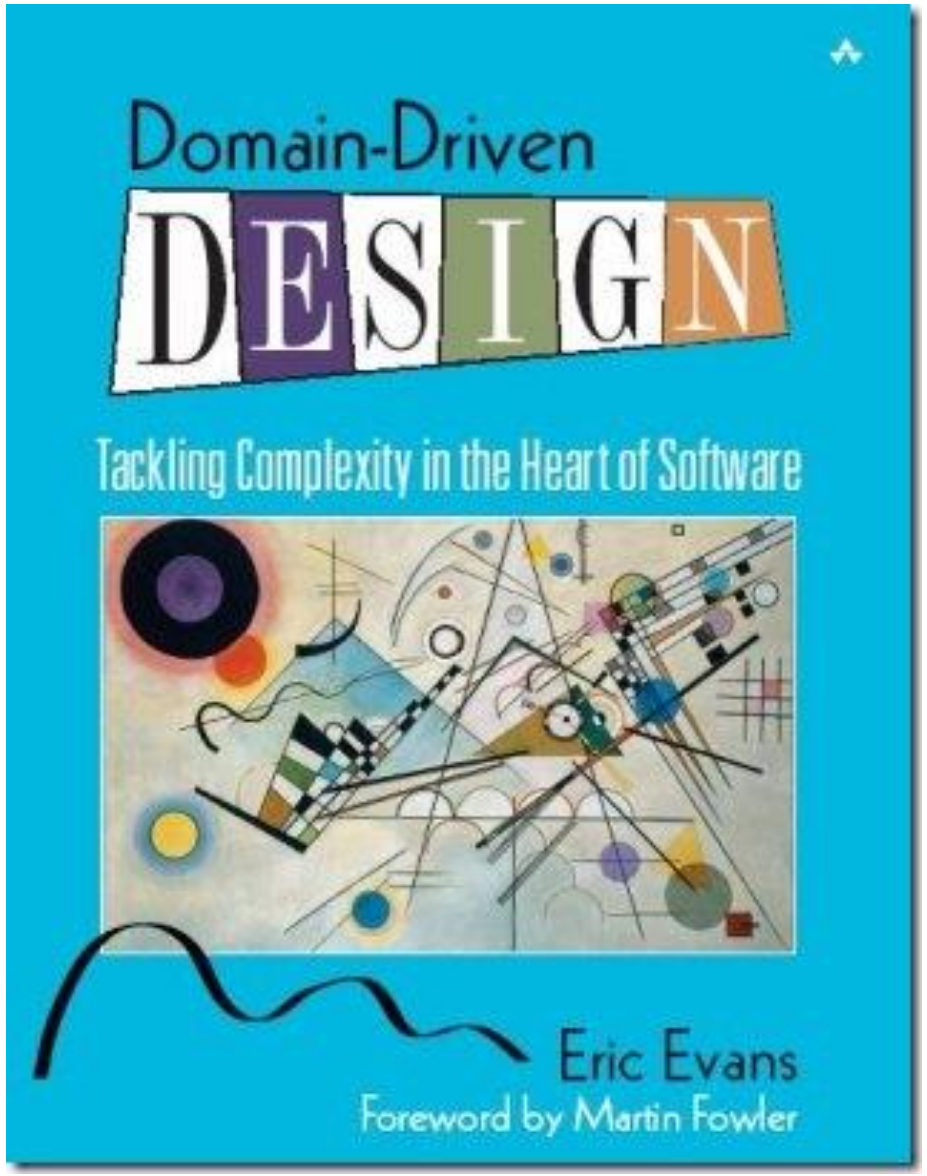
?

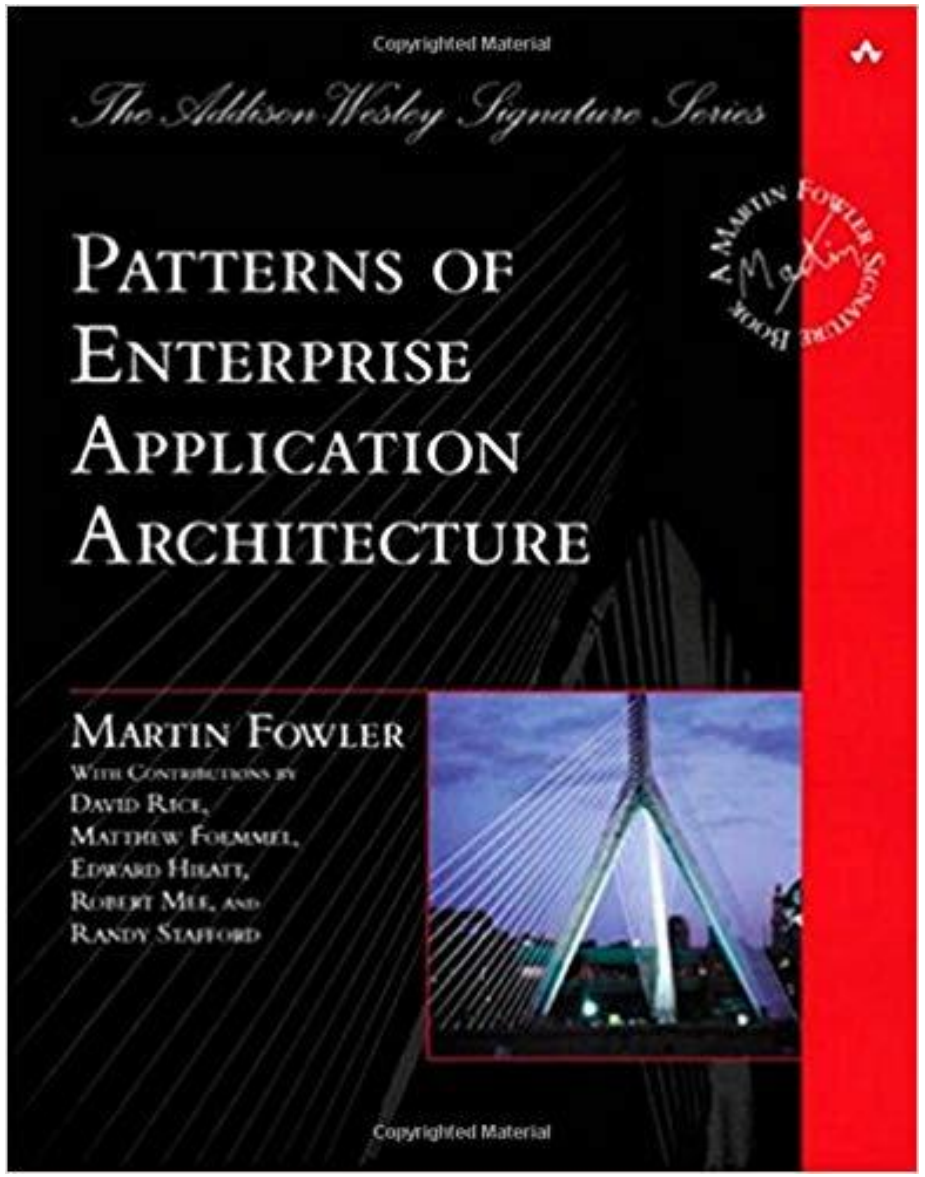

UserController
ListAll
ASP.NET/Presentation
UserService
ListAll
Business Logic
UserRepository
ListAll
Data Access
User


File Edit Selection View Go Debug Terminal Help
layers - Visual Studio Code
E...
.vs
.vscode
Layers
bin
Controllers
Models
obj
Properties
Repositories
Services
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
Layers.sln
master
0↓ 1↑
0 1
Layers.sln


File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - Visual Studio Code
E...
.vs
.vscode
Layers
bin
Controllers
HomeController.cs
UserController.cs
Models
obj
Properties
Repositories
Services
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
UserController.cs
4
5 namespace Layers
6 {
7 public class UserController : Controller
8 {
9 private readonly IUserService userService;
10
11 public UserController(IUserService userService)
12 {
13 this.userService = userService;
14 }
15
16 public async Task<IActionResult> List()
17 {
18 var users = userService.ListAll();
19 return View(users);
20 }
21
22 }
23 }
master
0↓ 1↑
0 1
Layers.sln
Ln 18, Col 40
Spaces: 4
UTF-8
CRLF
C#


File Edit Selection View Go Debug Terminal Help
IUserService.cs - layers - Visual Studio Code
E...
IUserService.cs
Layers
bin
Controllers
HomeController.cs
UserController.cs
Models
obj
Properties
Repositories
Services
IUserService.cs
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
Layers.sln
namespace Layers.Services
{
public interface IUserService
{
List<User> ListAll();
}
public class UserService : IUserService
{
private readonly IUserRepository userRepository;
public UserService(IUserRepository userRepository)
{
this.userRepository = userRepository;
}
public List<User> ListAll()
{
return userRepository.List();
}
}
master
0↓ 1↑
0 1
Layers.sln
Ln 9, Col 20
Spaces: 4
UTF-8
CRLF
C#


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


Copyrighted Material
The Addison Wesley Signature Series
A Martin Fowler Signature Book
PATTERNS OF ENTERPRISE APPLICATION ARCHITECTURE
MARTIN FOWLER
WITH CONTRIBUTIONS BY
DAVID RICE,
MATTHEW FOEMMEL,
EDWARD HIEATT,
ROBERT MEE, AND
RANDY STAFFORD
Copyrighted Material

"Conceptually, a Repository encapsulates the set of objects persisted in a **data store** and the operations performed over them, providing a more **object-oriented view** of the persistence layer. Repository also supports the objective of achieving a clean **separation** and one-way dependency between the domain and **data mapping** layers."

Patterns of Enterprise Application Architecture by Martin Fowler et al

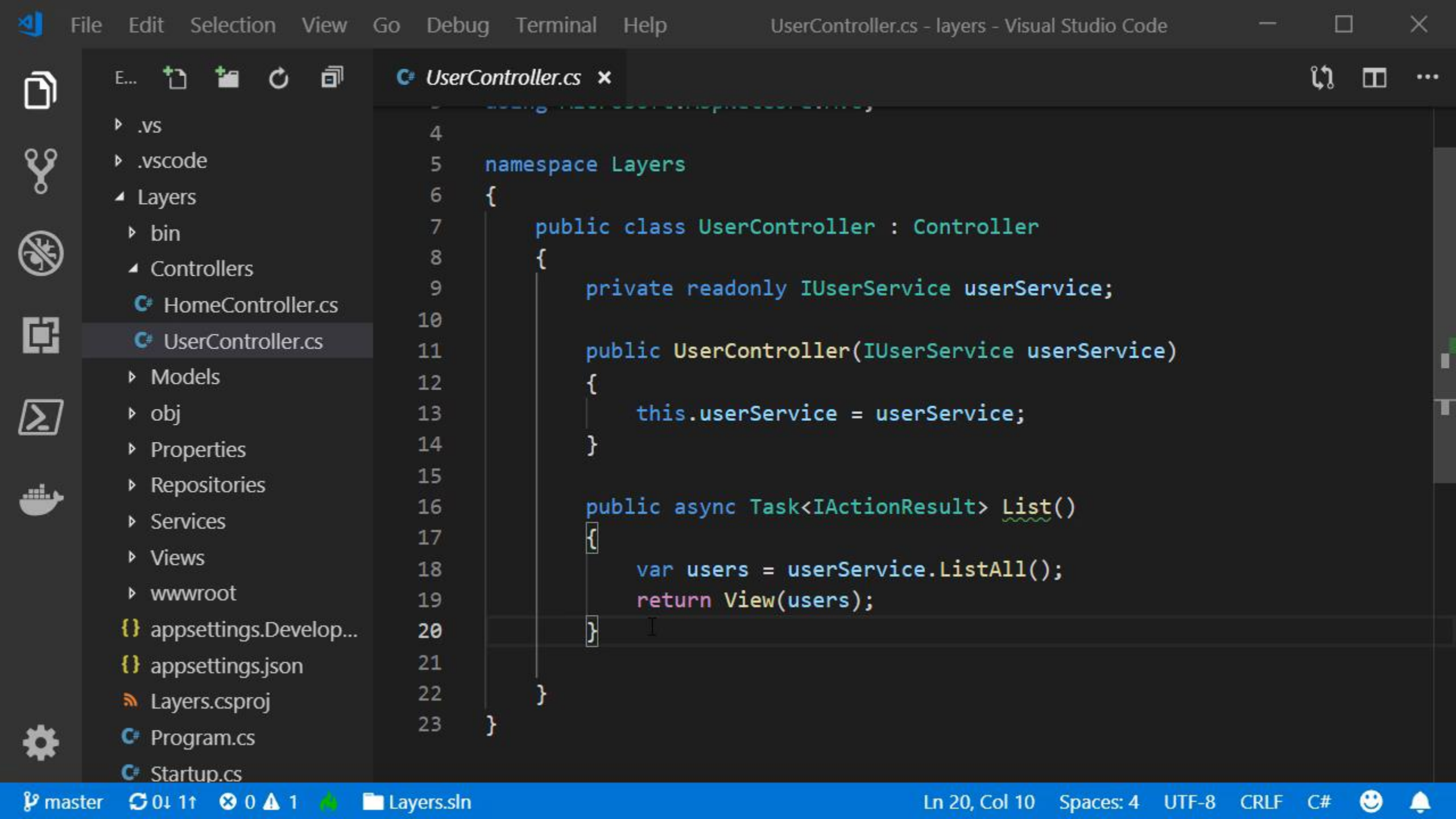
"adding this layer helps **minimise duplicate query logic**..."

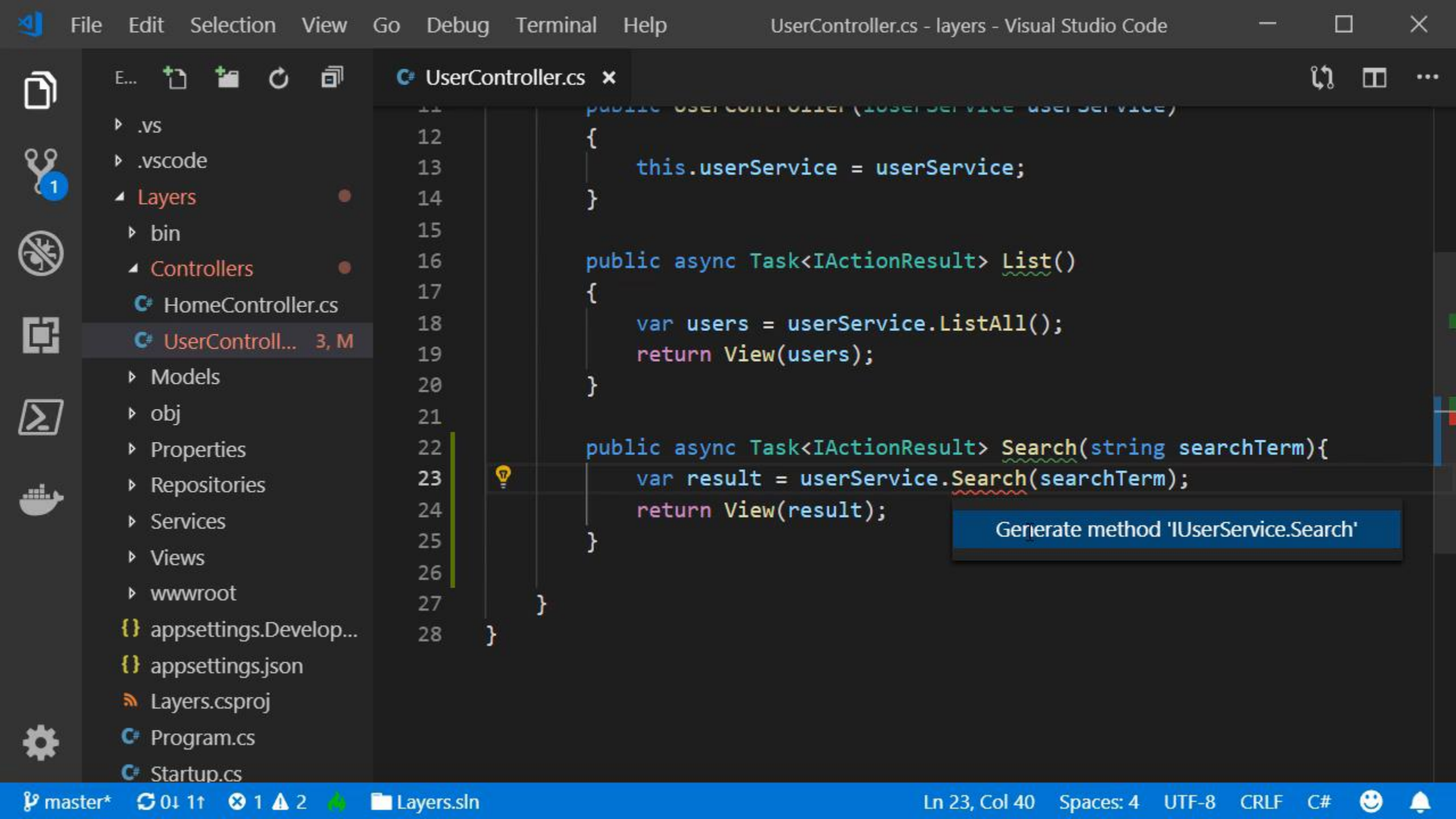
Patterns of Enterprise Application Architecture by Martin Fowler et al

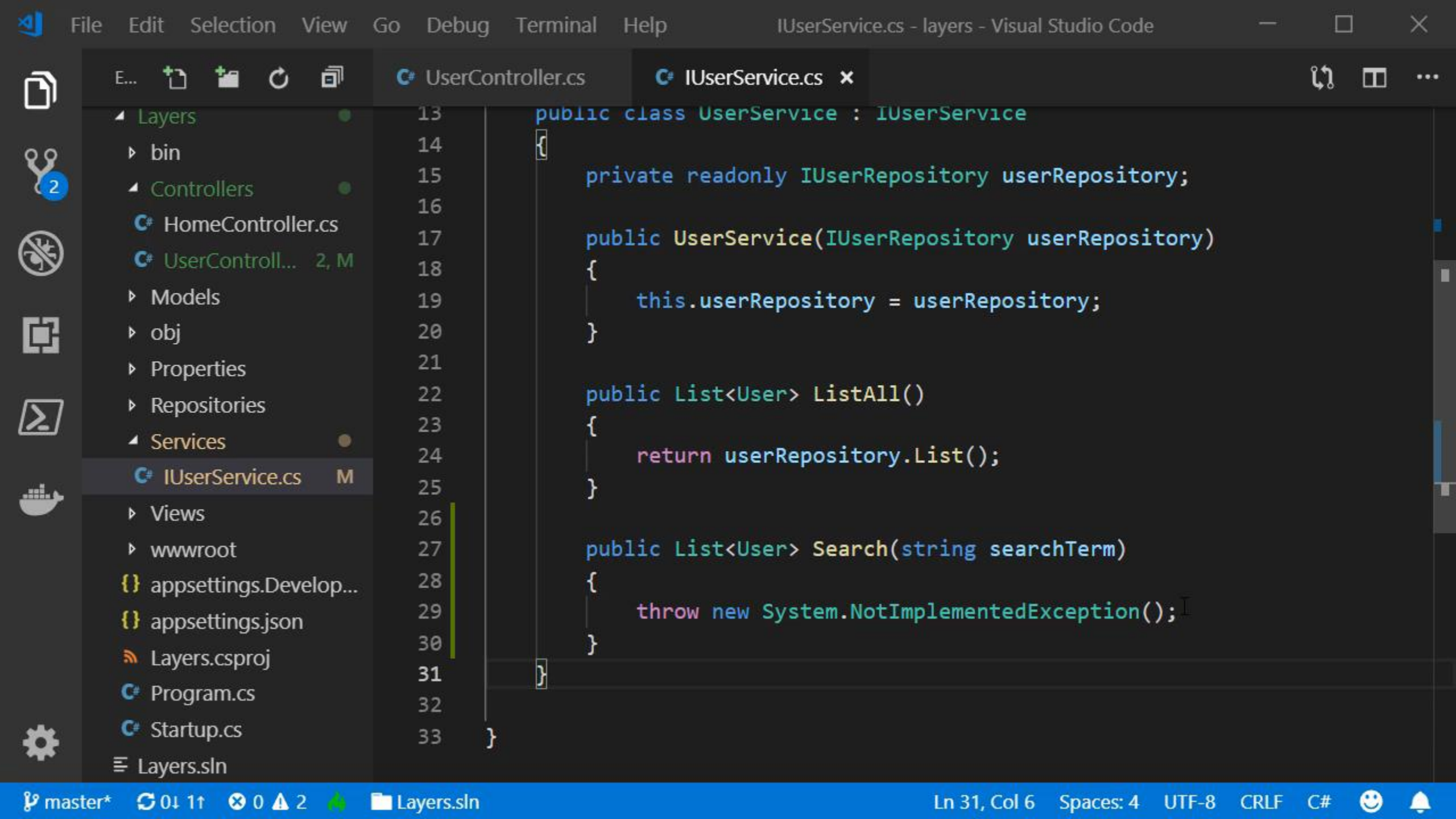
User Search

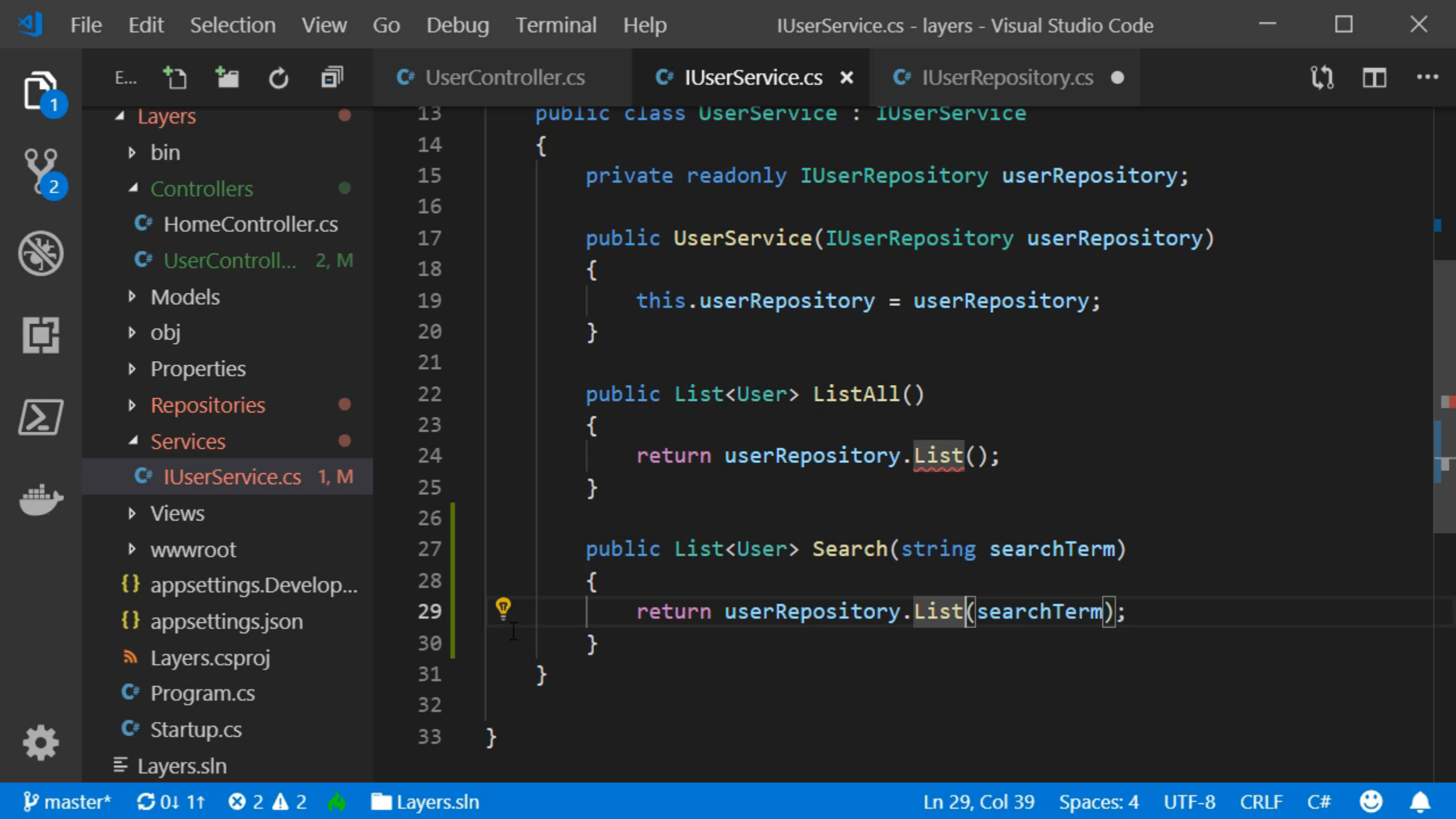
bob

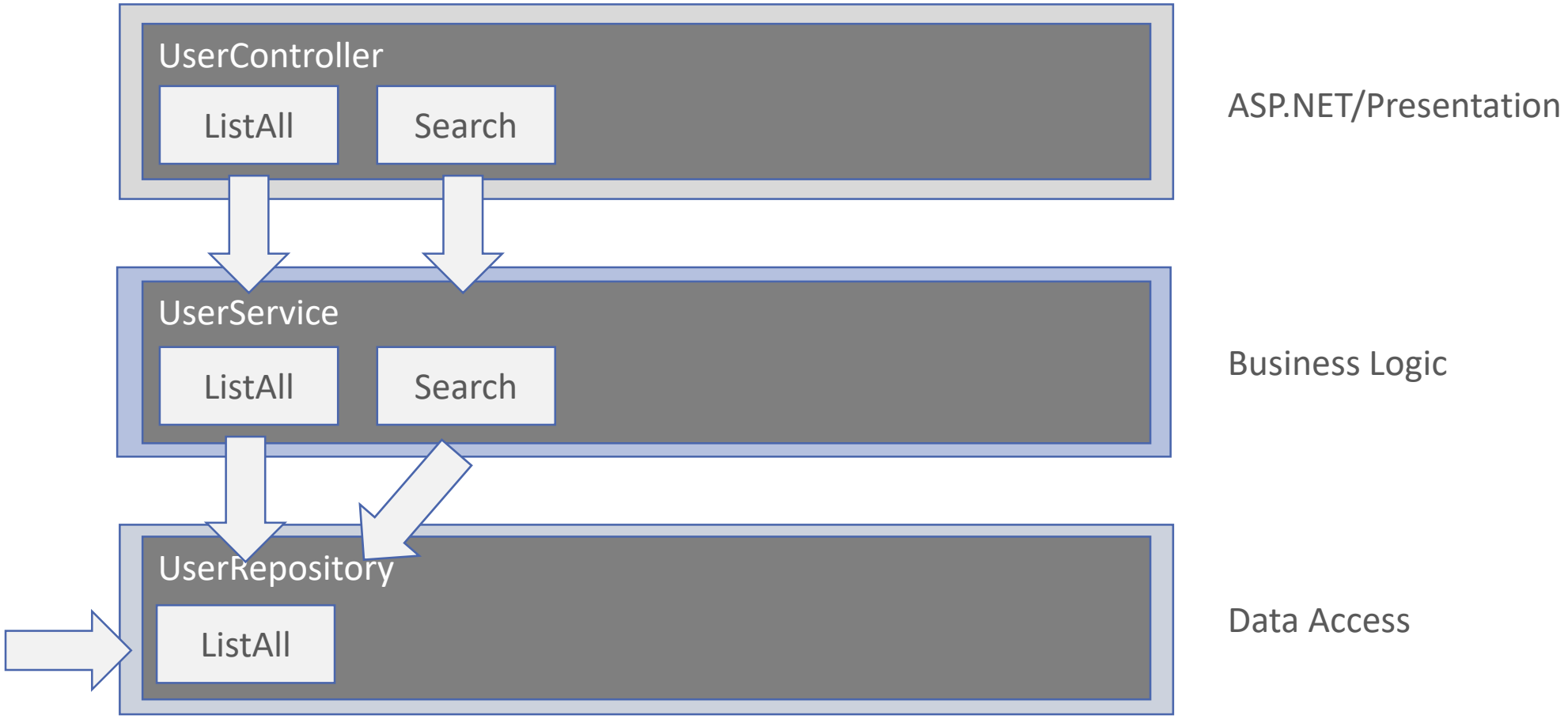
First Name	Last Name	Last Login
Bob	Smith	01/02/2018
Bob	Bennet	03/01/2019
Bobina	Bobalot	03/02/2018
Bobby	Boat	01/01/2018

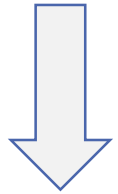

File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - Visual Studio Code
UserController.cs
.vs
.vscode
Layers
bin
Controllers
HomeController.cs
UserController.cs
Models
obj
Properties
Repositories
Services
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
4
namespace Layers
{
public class UserController : Controller
{
private readonly IUserService userService;
public UserController(IUserService userService)
{
this.userService = userService;
}
public async Task<IActionResult> List()
{
var users = userService.ListAll();
return View(users);
}
}
}
master
0↓ 1↑
0 1
Layers.sln
Ln 20, Col 10
Spaces: 4
UTF-8
CRLF
C#


File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - Visual Studio Code
E...
.vs
.vscode
Layers
bin
Controllers
HomeController.cs
UserControll... 3, M
Models
obj
Properties
Repositories
Services
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
UserController.cs
{
this.userService = userService;
}
public async Task<IActionResult> List()
{
var users = userService.ListAll();
return View(users);
}
public async Task<IActionResult> Search(string searchTerm){
var result = userService.Search(searchTerm);
return View(result);
}
}
}
Generate method 'IUserService.Search'
master*
0↓ 1↑
1 2
Layers.sln
Ln 23, Col 40
Spaces: 4
UTF-8
CRLF
C#


File Edit Selection View Go Debug Terminal Help
IUserService.cs - layers - Visual Studio Code
UserController.cs
IUserService.cs
Layers
bin
Controllers
HomeController.cs
UserControll... 2, M
Models
obj
Properties
Repositories
Services
IUserService.cs M
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
Layers.sln
public class UserService : IUserService
{
private readonly IUserRepository userRepository;
public UserService(IUserRepository userRepository)
{
this.userRepository = userRepository;
}
public List<User> ListAll()
{
return userRepository.List();
}
public List<User> Search(string searchTerm)
{
throw new System.NotImplementedException();
}
}
}
master* 0↓ 1↑ 0 2 Layers.sln
Ln 31, Col 6 Spaces: 4 UTF-8 CRLF C#

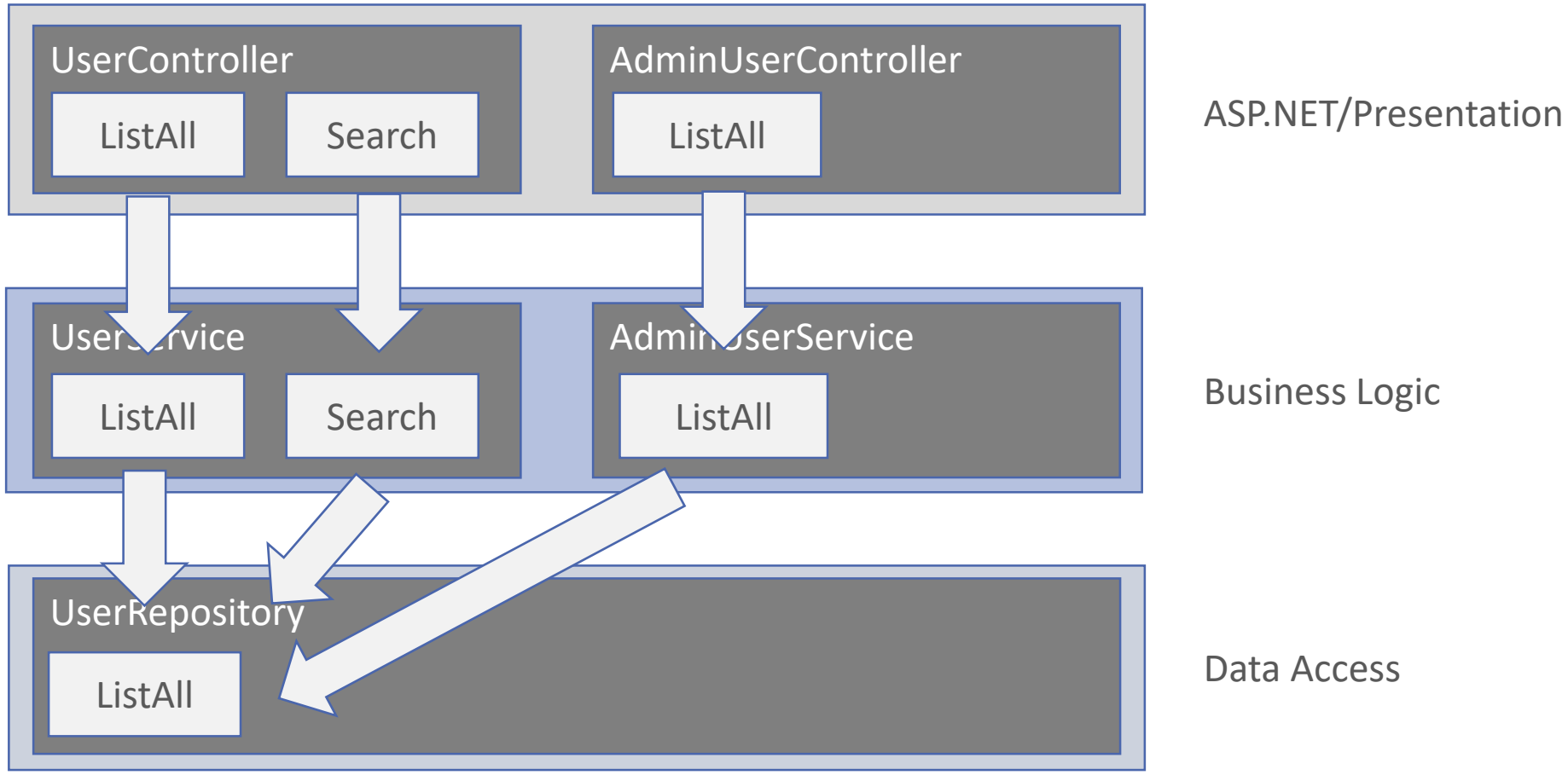

File Edit Selection View Go Debug Terminal Help
IUserService.cs - layers - Visual Studio Code
UserController.cs
IUserService.cs
IUserRepository.cs
Layers
bin
Controllers
HomeController.cs
UserControll... 2, M
Models
obj
Properties
Repositories
Services
IUserService.cs 1, M
Views
wwwroot
appsettings.Develop...
appsettings.json
Layers.csproj
Program.cs
Startup.cs
Layers.sln
public class UserService : IUserService
{
private readonly IUserRepository userRepository;
public UserService(IUserRepository userRepository)
{
this.userRepository = userRepository;
}
public List<User> ListAll()
{
return userRepository.List();
}
public List<User> Search(string searchTerm)
{
return userRepository.List(searchTerm);
}
}
}
master* 0↓ 1↑ 2 2 Layers.sln
Ln 29, Col 39 Spaces: 4 UTF-8 CRLF C#

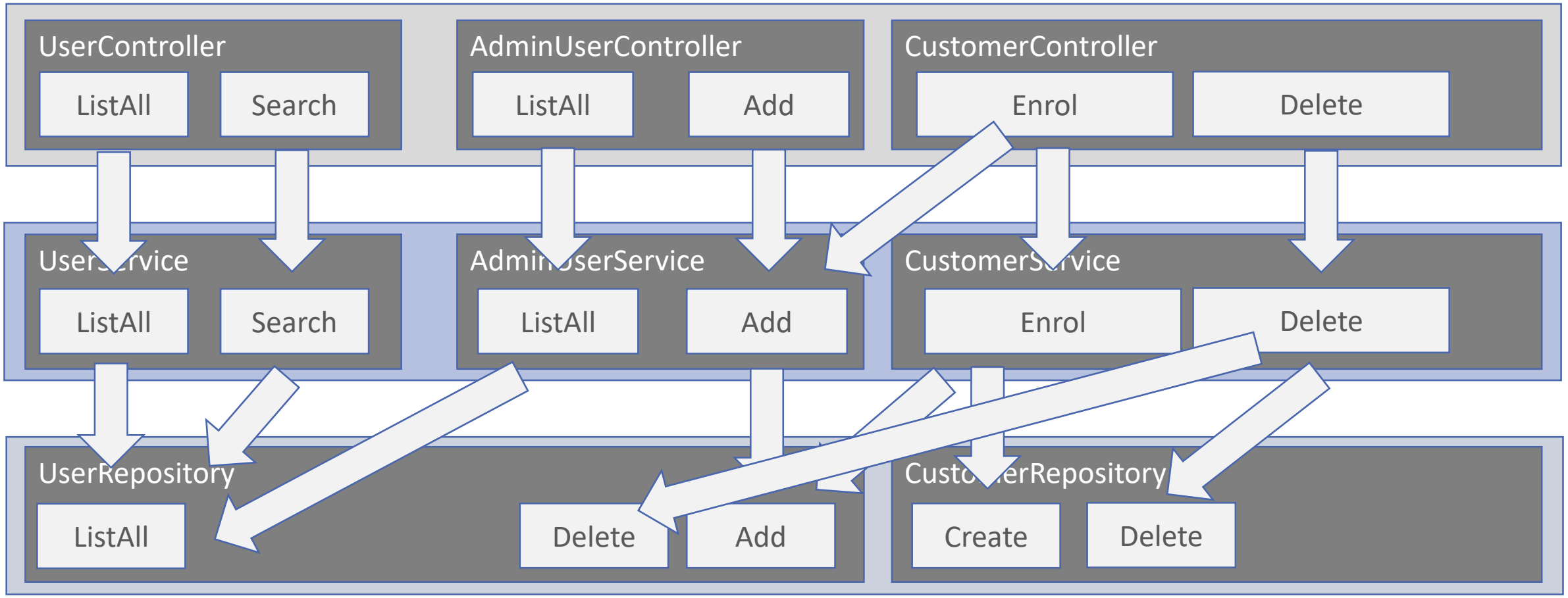

UserController
ListAll
Search
UserService
ListAll
Search
UserRepository
ListAll
ASP.NET/Presentation
Business Logic
Data Access



All Users

First Name	Last Name	Last Login	Roles
Bob	Smith	01/02/2018	Admin
Bob	Bennet	03/01/2019	Super User
Bobina	Bobalot	03/02/2018	User
Bobby	Boat	01/01/2018	User


UserController
ListAll
Search
AdminUserController
ListAll
ListAll
Search
AdminUserService
ListAll
UserRepository
ListAll
ASP.NET/Presentation
Business Logic
Data Access


UserController
ListAll
Search
AdminUserController
ListAll
Add
CustomerController
Enrol
Delete
UserService
ListAll
Search
AdminUserService
ListAll
Add
CustomerService
Enrol
Delete
UserRepository
ListAll
Delete
Add
CustomerRepository
Create
Delete

Helpers

Mappers

Managers

Factories

ServiceFactories

Functions

Requests

Responses

Models

ViewModels

DALs

Readers

Repositories

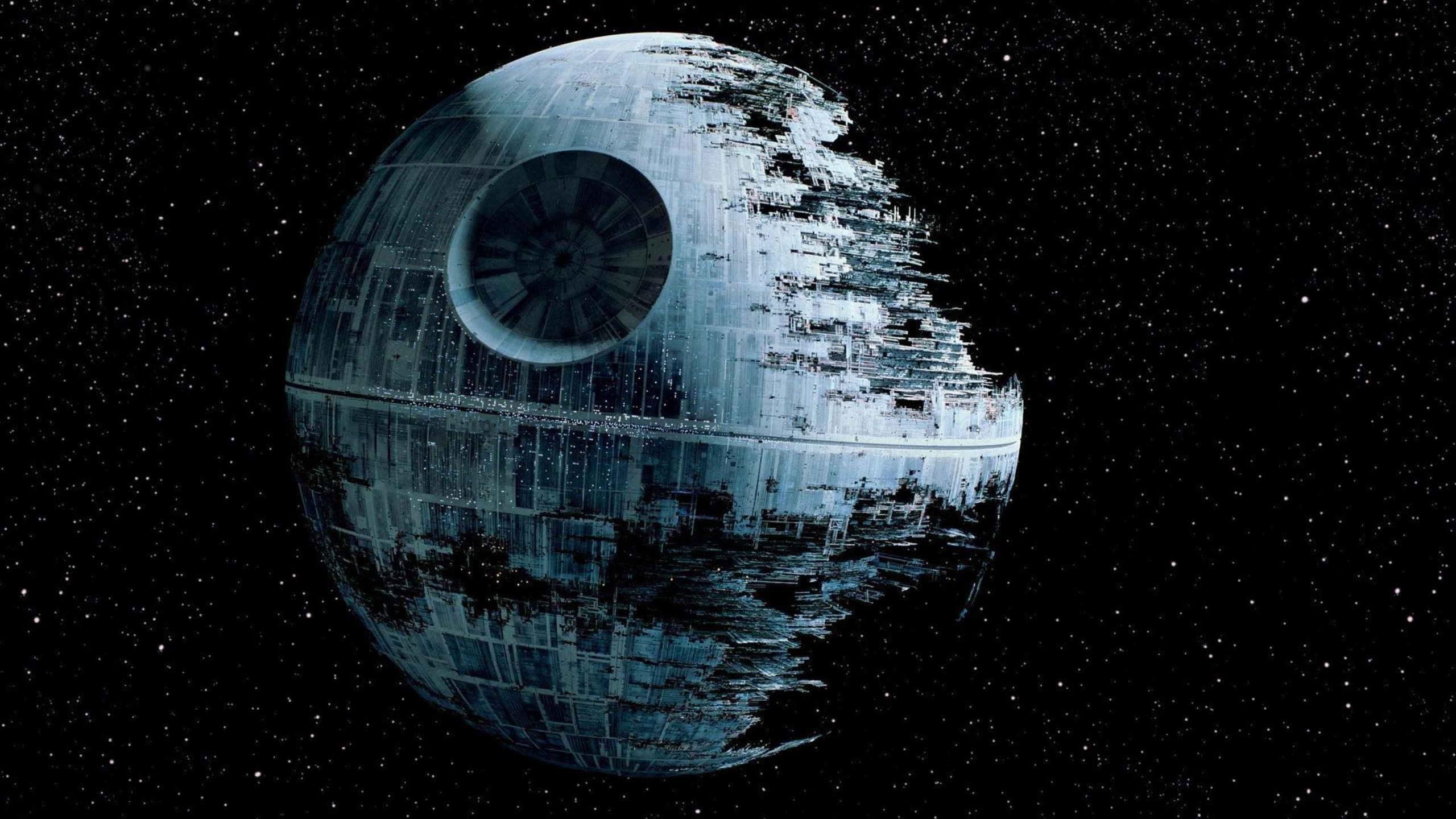
ReadOnlyRepositories

Search Twitter

Bernard Vander Beken liked

Marc von Renteln
@mvonrenteln

When your Micro Service Architecture looks like this, it's split the wrong way. This is a #microlith: The worst union of Microservices and a big ball of mud. A lot of accidental complexity. You should invest in finding the proper boundaries. #boundedContext #DDDesign

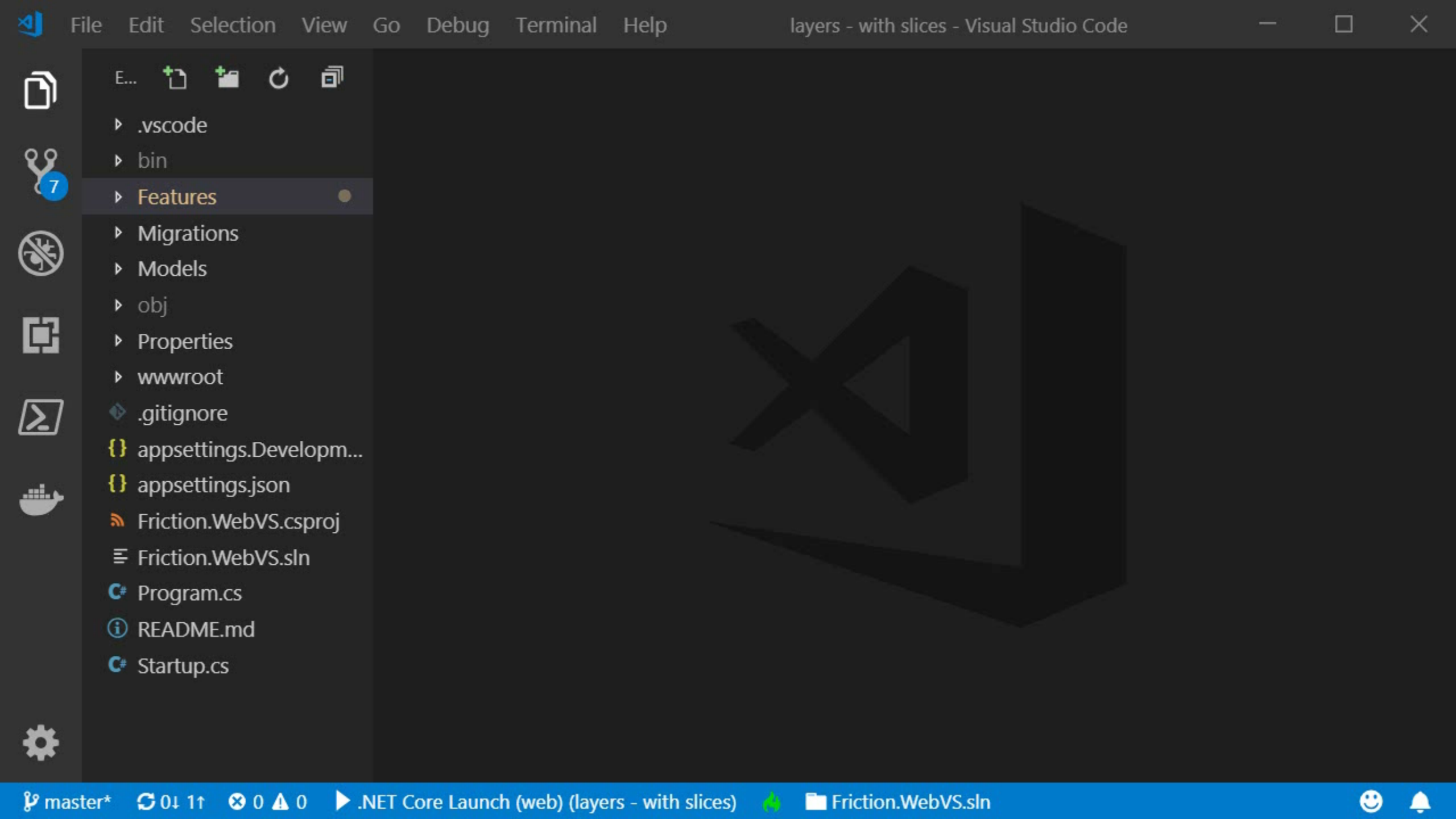


What if we focus on features?

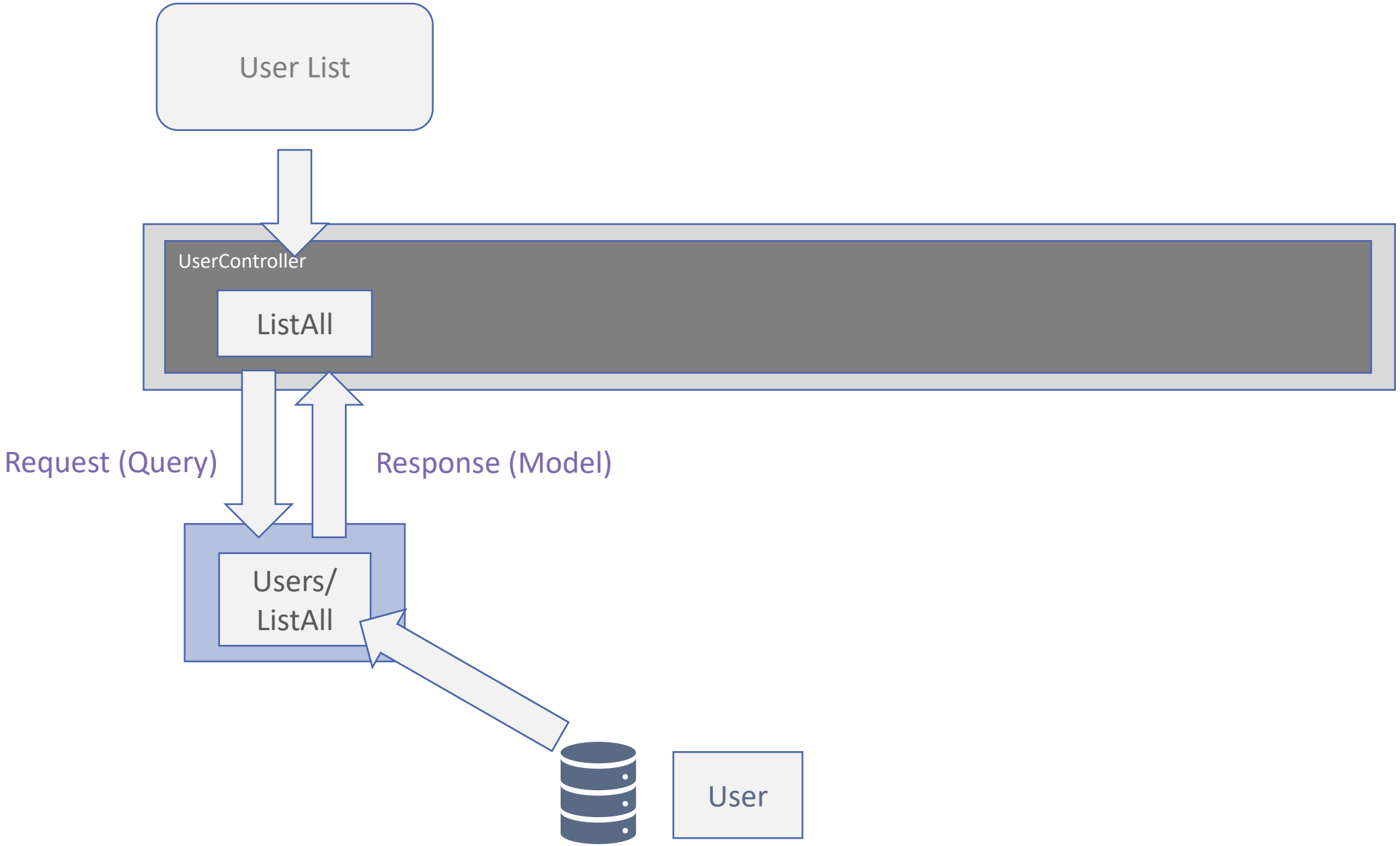
All Users

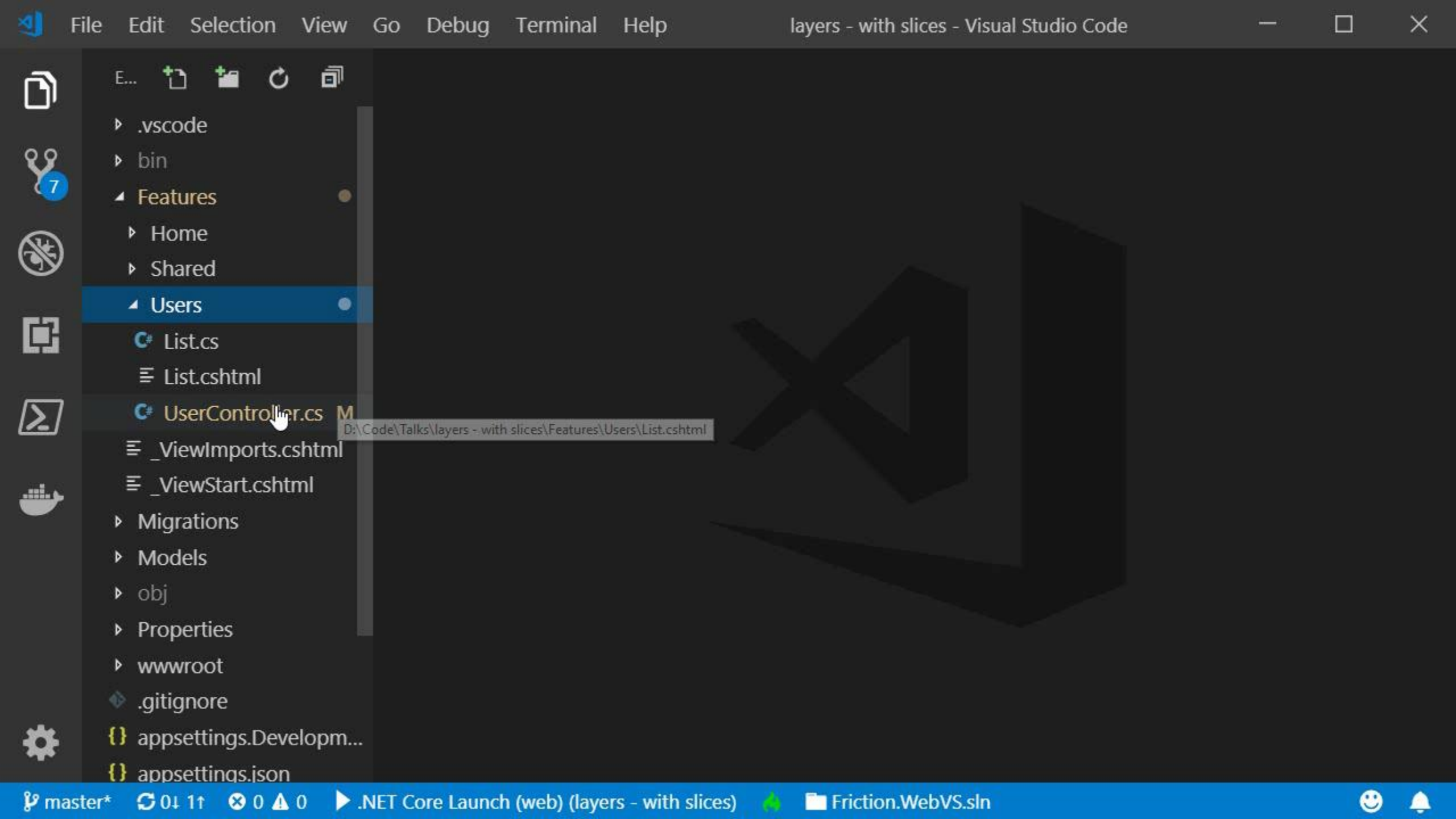
First Name	Last Name	Last Login	Active
Bob	Smith	01/02/2018	☑
James	Barrow	03/01/2019	☐
Janet	Jones	03/02/2018	☐
Ollie	Norburn	01/01/2018	☑
Jane	Tomley	05/01/~~208~~	☑

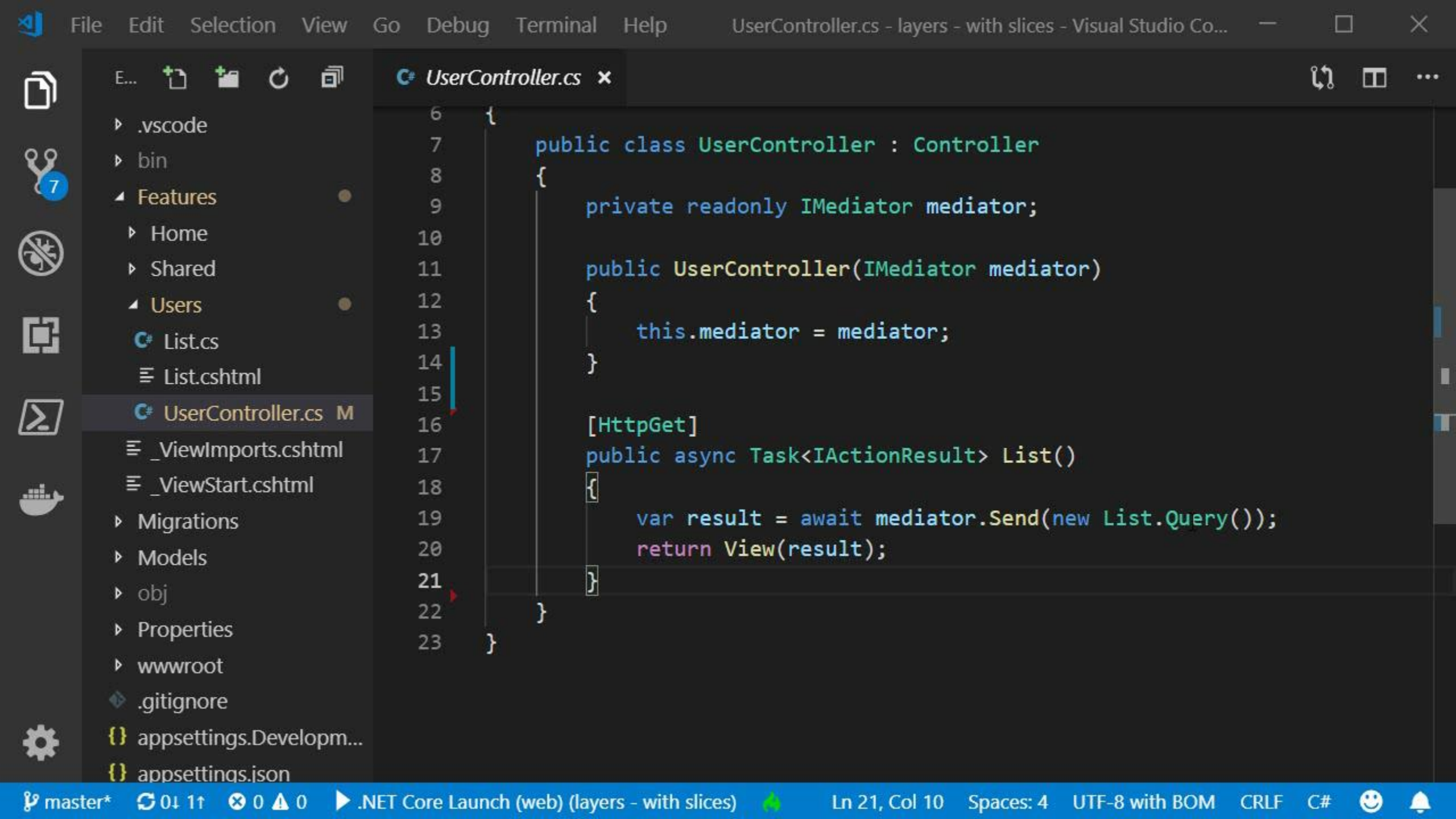
2018

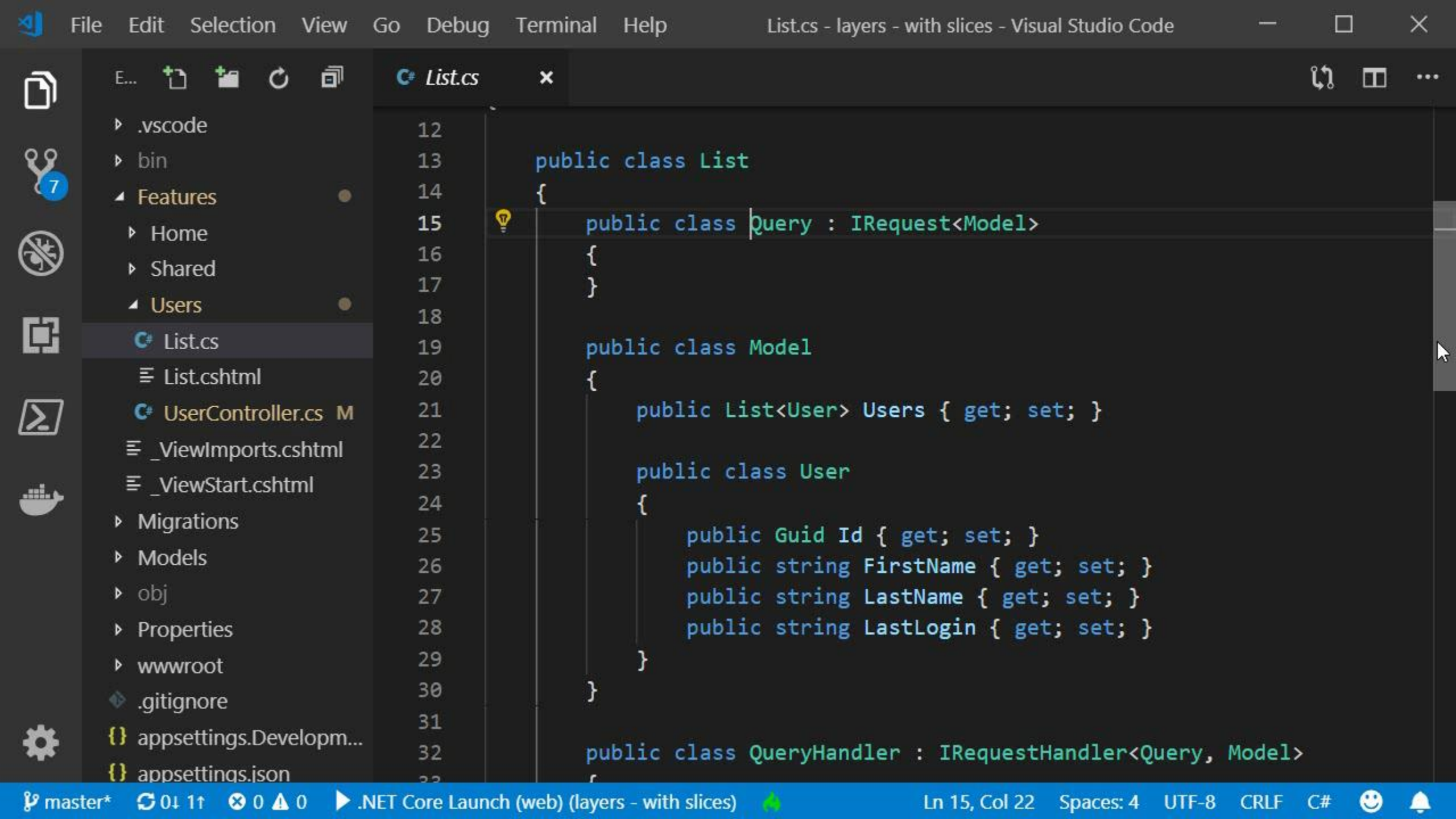

File Edit Selection View Go Debug Terminal Help
layers - with slices - Visual Studio Code
E...
.vscode
bin
Features
Migrations
Models
obj
Properties
wwwroot
.gitignore
appsettings.Developm...
appsettings.json
Friction.WebVS.csproj
Friction.WebVS.sln
Program.cs
README.md
Startup.cs
master*
0↓ 1↑
0 0
.NET Core Launch (web) (layers - with slices)
Friction.WebVS.sln


User List
UserController
ListAll


User List
UserController
ListAll
Request (Query)
Response (Model)
Users/
ListAll
User


File Edit Selection View Go Debug Terminal Help
layers - with slices - Visual Studio Code
E...
.vscode
bin
Features
Home
Shared
Users
List.cs
List.cshtml
UserController.cs M
D:\Code\Talks\layers - with slices\Features\Users\List.cshtml
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
wwwroot
.gitignore
appsettings.Developm...
appsettings.json
master*
0↓ 1↑
0 0
.NET Core Launch (web) (layers - with slices)
Friction.WebVS.sln


File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - with slices - Visual Studio Co...
E...
.vscode
bin
Features
Home
Shared
Users
List.cs
List.cshtml
UserController.cs M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
wwwroot
.gitignore
appsettings.Developm...
appsettings.json
UserController.cs
6 {
7 public class UserController : Controller
8 {
9 private readonly IMediator mediator;
10
11 public UserController(IMediator mediator)
12 {
13 this.mediator = mediator;
14 }
15
16 [HttpGet]
17 public async Task<IActionResult> List()
18 {
19 var result = await mediator.Send(new List.Query());
20 return View(result);
21 }
22 }
23 }
master* 0↓ 1↑ 0 0 .NET Core Launch (web) (layers - with slices)
Ln 21, Col 10 Spaces: 4 UTF-8 with BOM CRLF C#

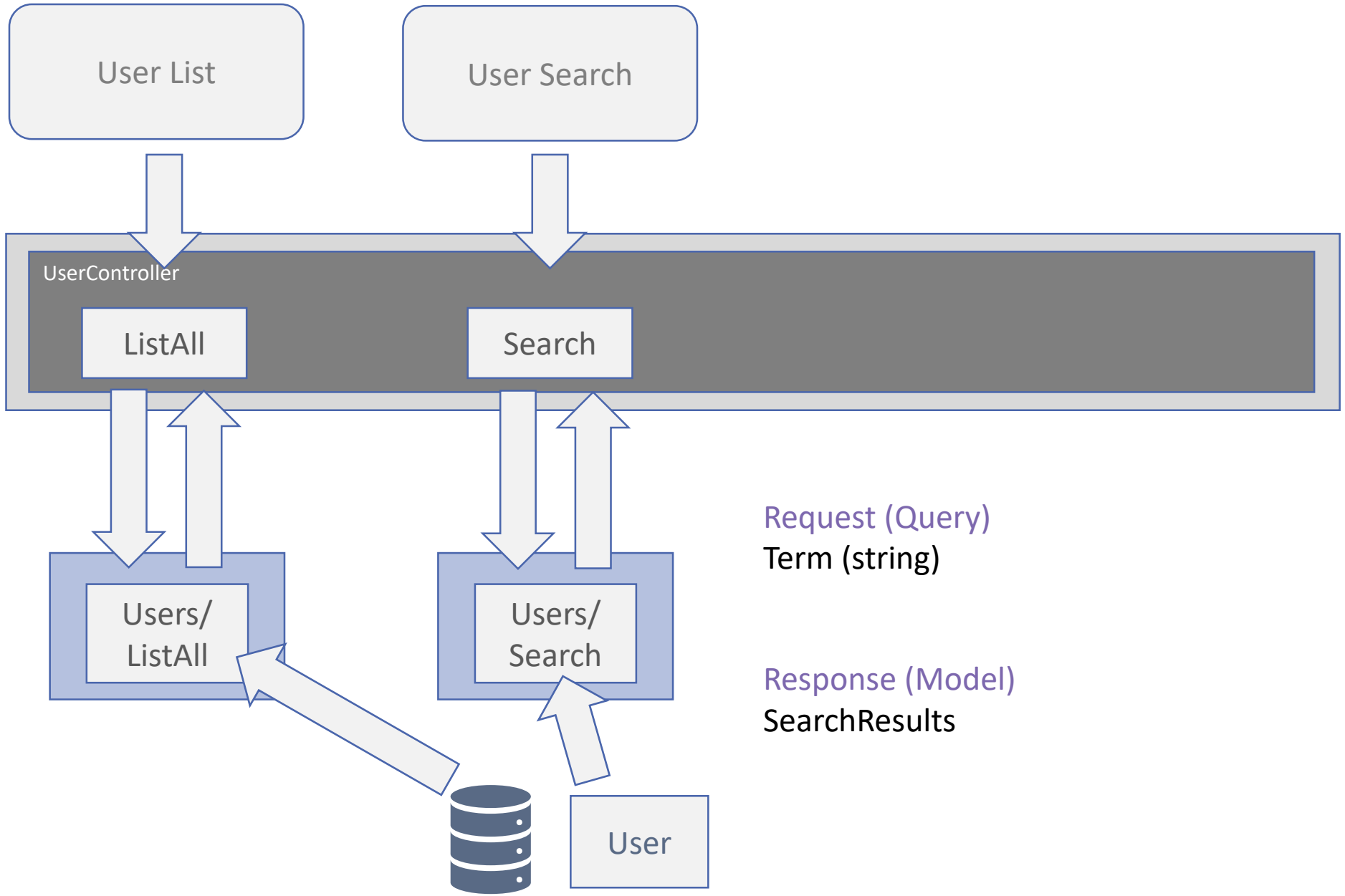

File Edit Selection View Go Debug Terminal Help
List.cs - layers - with slices - Visual Studio Code
List.cs
.vscode
bin
Features
Home
Shared
Users
List.cs
List.cshtml
UserController.cs M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
wwwroot
.gitignore
appsettings.Developm...
appsettings.json
12
13 public class List
14 {
15 public class Query : IRequest<Model>
16 {
17 }
18
19 public class Model
20 {
21 public List<User> Users { get; set; }
22
23 public class User
24 {
25 public Guid Id { get; set; }
26 public string FirstName { get; set; }
27 public string LastName { get; set; }
28 public string LastLogin { get; set; }
29 }
30 }
31
32 public class QueryHandler : IRequestHandler<Query, Model>
master* 0↓ 1↑ 0 0 .NET Core Launch (web) (layers - with slices)
Ln 15, Col 22 Spaces: 4 UTF-8 CRLF C#

User Search

bob

4 users found...

First Name	Last Name	Last Login
Bob	Smith	01/02/2018
Bob	Bennet	03/01/2019
Bobina	Bobalot	03/02/2018
Bobby	Boat	01/01/2018

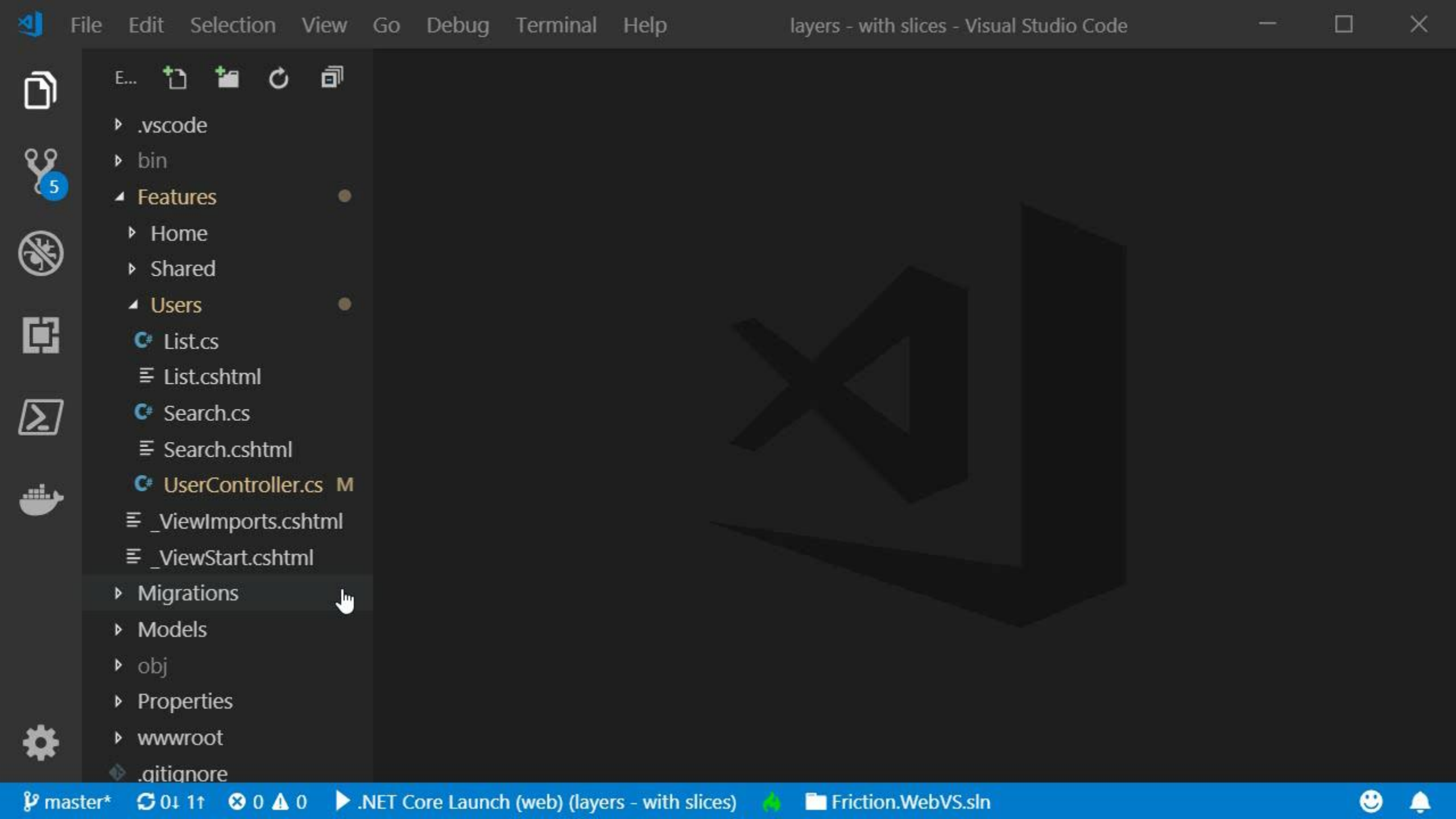


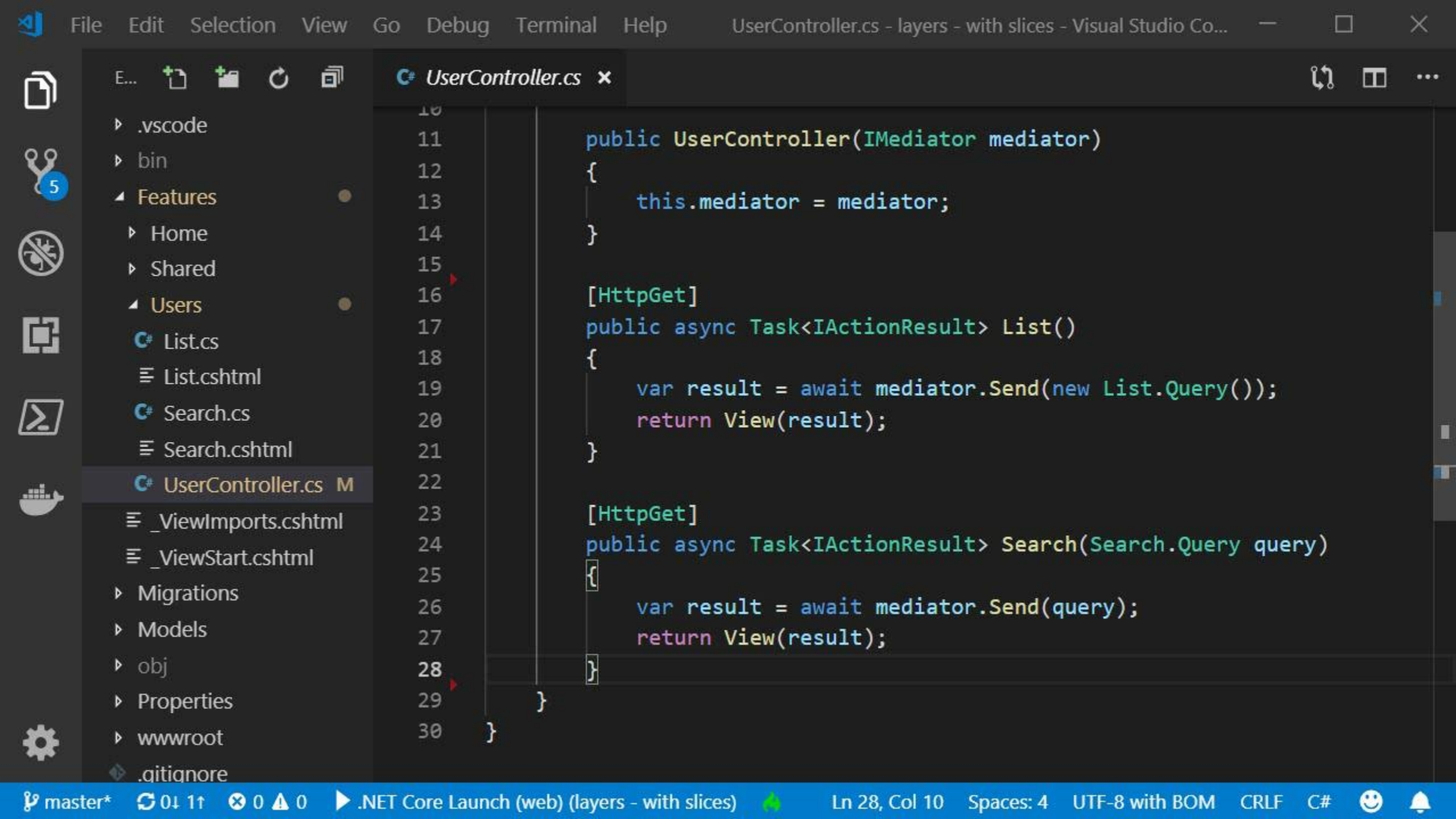
Request (Query)

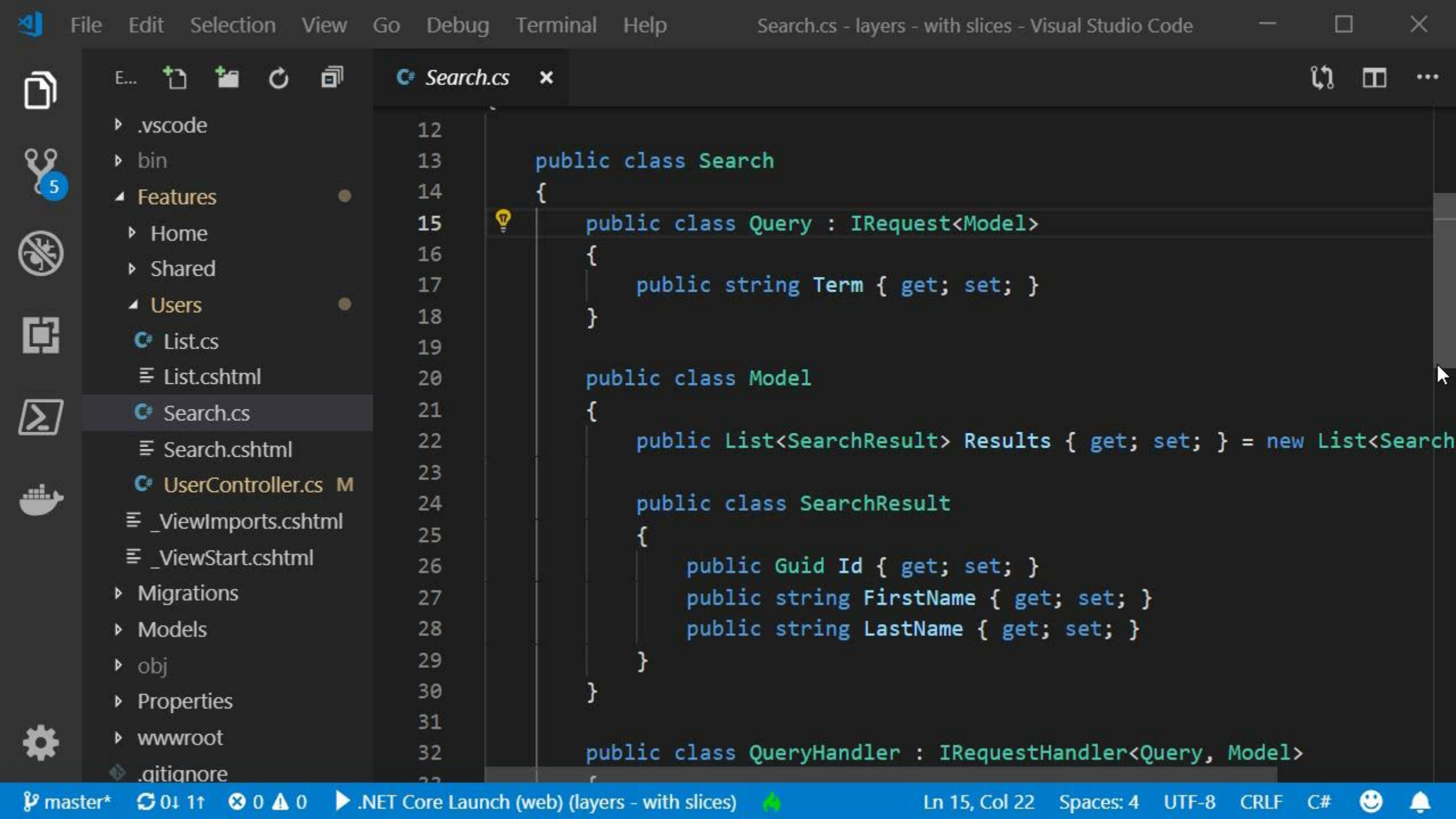
Term (string)

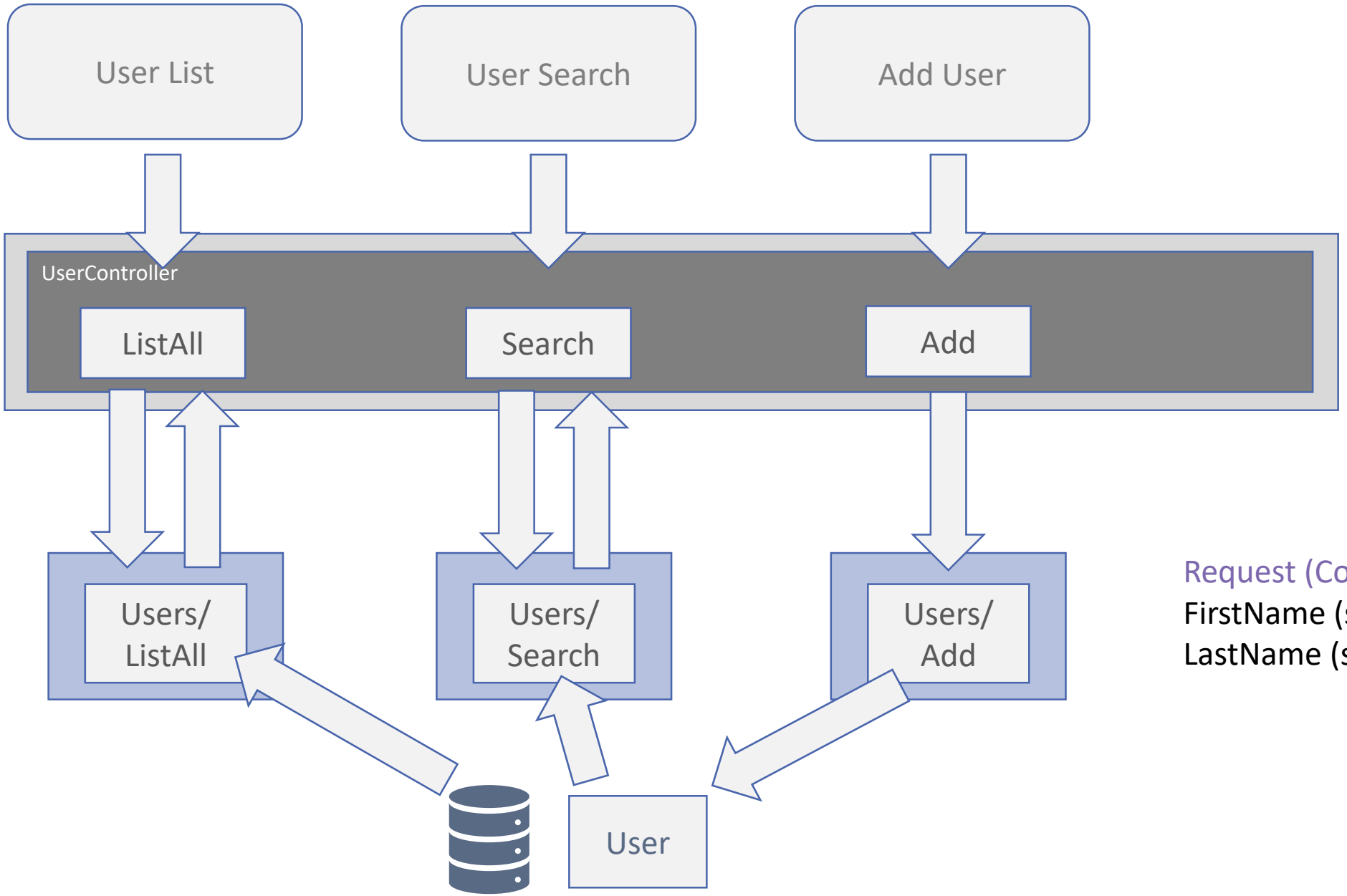
Response (Model)

SearchResults

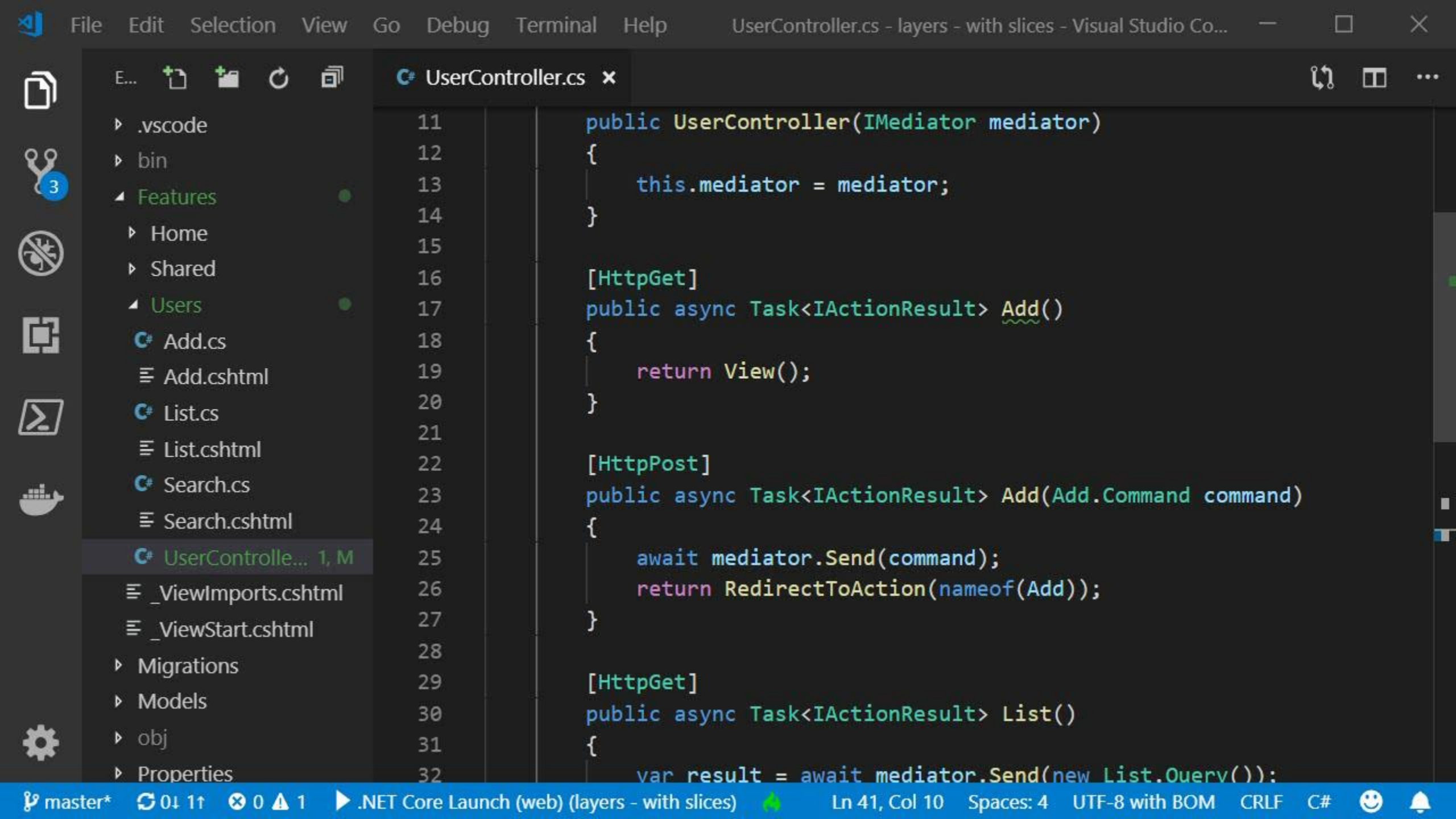

File Edit Selection View Go Debug Terminal Help
layers - with slices - Visual Studio Code
E...
.vscode
bin
Features
Home
Shared
Users
List.cs
List.cshtml
Search.cs
Search.cshtml
UserController.cs M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
wwwroot
.gitignore
master*
0↓ 1↑
0 0
.NET Core Launch (web) (layers - with slices)
Friction.WebVS.sln

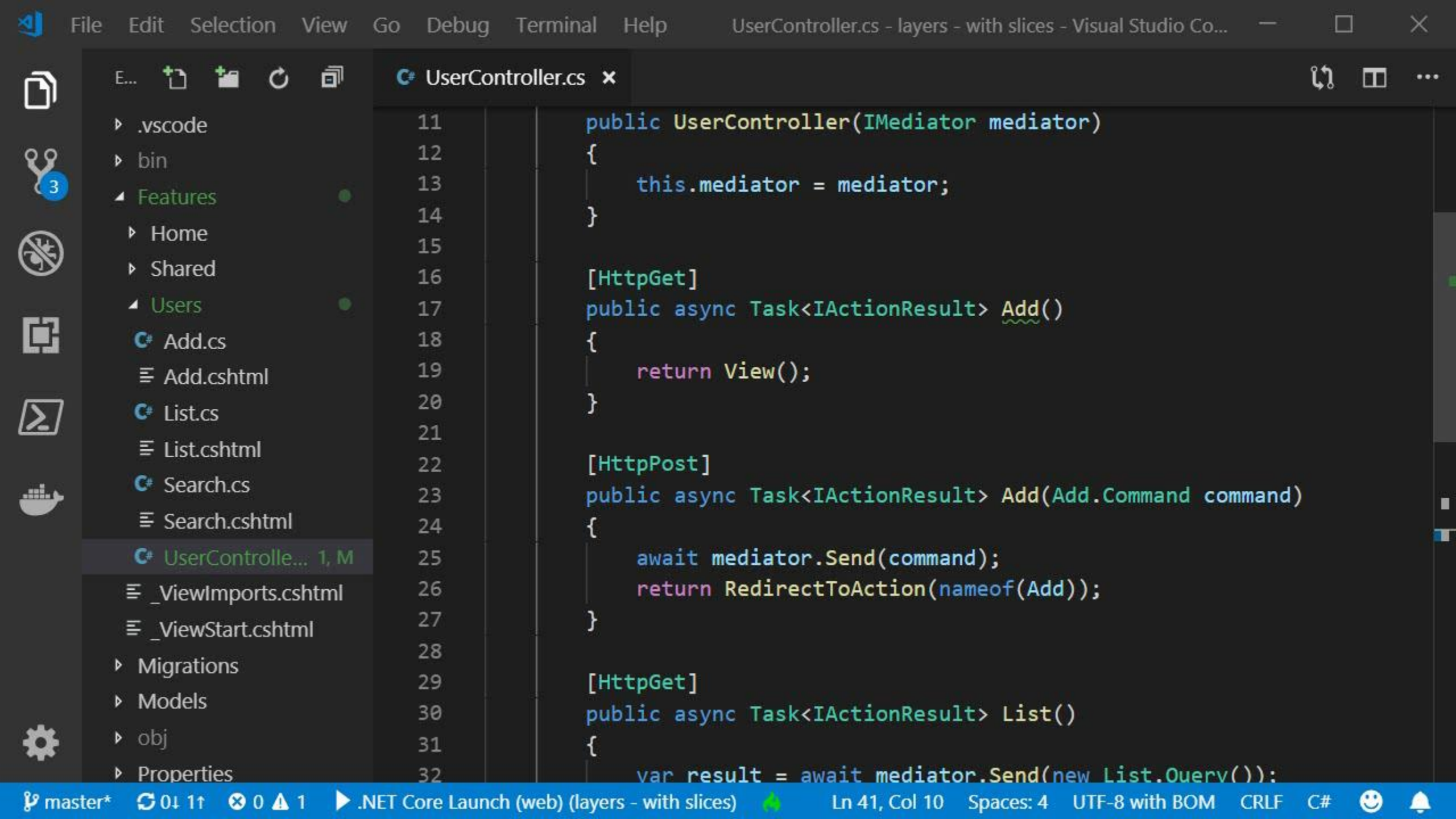

File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - with slices - Visual Studio Co...
E...
.vscode
bin
Features
Home
Shared
Users
List.cs
List.cshtml
Search.cs
Search.cshtml
UserController.cs M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
wwwroot
.gitignore
UserController.cs
public UserController(IMediator mediator)
{
this.mediator = mediator;
}
[HttpGet]
public async Task<IActionResult> List()
{
var result = await mediator.Send(new List.Query());
return View(result);
}
[HttpGet]
public async Task<IActionResult> Search(Search.Query query)
{
var result = await mediator.Send(query);
return View(result);
}
}
}
master* 0↓ 1↑ 0 0 .NET Core Launch (web) (layers - with slices)
Ln 28, Col 10 Spaces: 4 UTF-8 with BOM CRLF C#


File Edit Selection View Go Debug Terminal Help
Search.cs - layers - with slices - Visual Studio Code
Search.cs
.vscode
bin
Features
Home
Shared
Users
List.cs
List.cshtml
Search.cs
Search.cshtml
UserController.cs M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
wwwroot
.gitignore
12
13 public class Search
14 {
15 public class Query : IRequest<Model>
16 {
17 public string Term { get; set; }
18 }
19
20 public class Model
21 {
22 public List<SearchResult> Results { get; set; } = new List<Search
23
24 public class SearchResult
25 {
26 public Guid Id { get; set; }
27 public string FirstName { get; set; }
28 public string LastName { get; set; }
29 }
30 }
31
32 public class QueryHandler : IRequestHandler<Query, Model>
master* 0↓ 1↑ 0 0 .NET Core Launch (web) (layers - with slices)
Ln 15, Col 22 Spaces: 4 UTF-8 CRLF C#

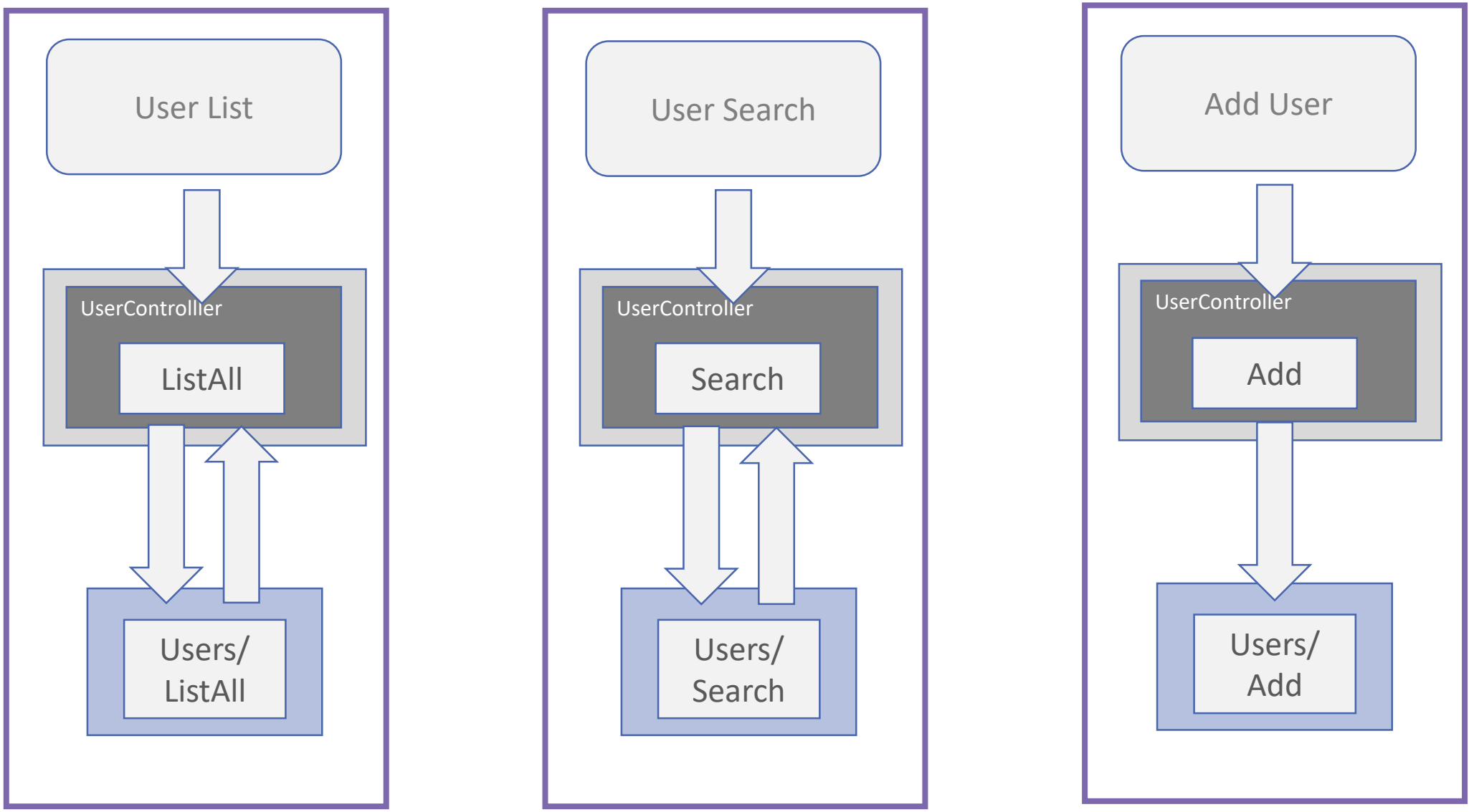


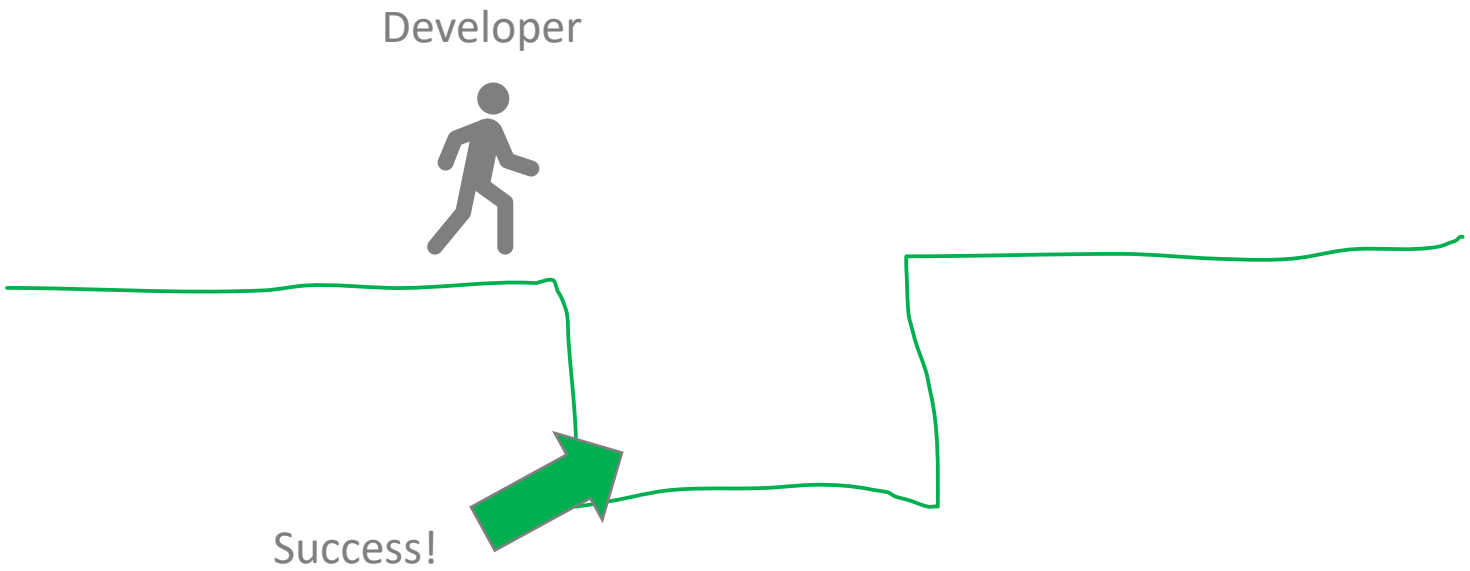
Request (Command)
FirstName (string)
LastName (string)


File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - with slices - Visual Studio Co...
E...
.vscode
bin
Features
Home
Shared
Users
Add.cs
Add.cshtml
List.cs
List.cshtml
Search.cs
Search.cshtml
UserControlle... 1, M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
UserController.cs
11 public UserController(IMediator mediator)
12 {
13 this.mediator = mediator;
14 }
15
16 [HttpGet]
17 public async Task<IActionResult> Add()
18 {
19 return View();
20 }
21
22 [HttpPost]
23 public async Task<IActionResult> Add(Add.Command command)
24 {
25 await mediator.Send(command);
26 return RedirectToAction(nameof(Add));
27 }
28
29 [HttpGet]
30 public async Task<IActionResult> List()
31 {
32 var result = await mediator.Send(new List.Query());
master* 0↓ 1↑ 0 1 .NET Core Launch (web) (layers - with slices)
Ln 41, Col 10 Spaces: 4 UTF-8 with BOM CRLF C#

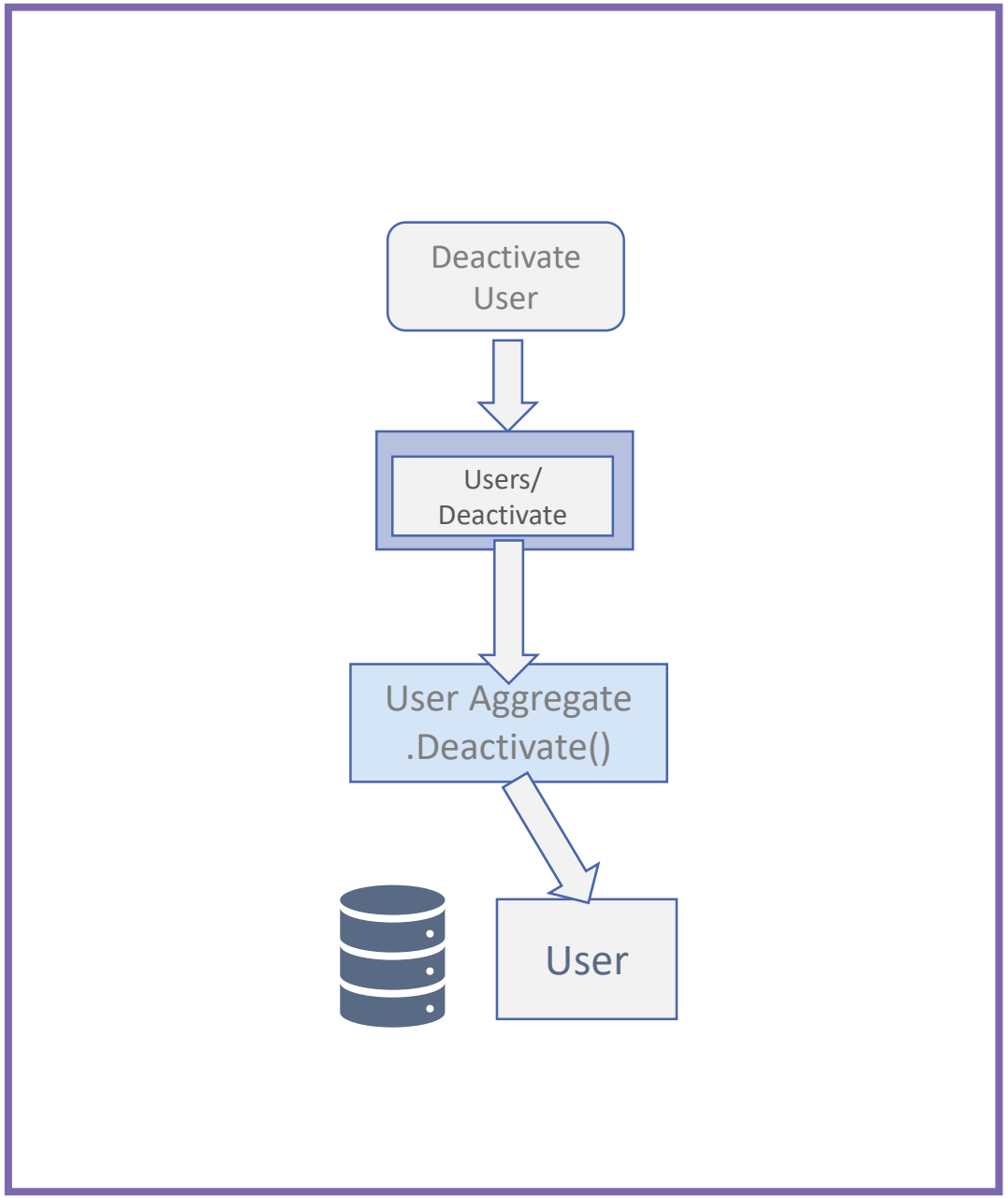

File Edit Selection View Go Debug Terminal Help
UserController.cs - layers - with slices - Visual Studio Co...
E...
.vscode
bin
Features
Home
Shared
Users
Add.cs
Add.cshtml
List.cs
List.cshtml
Search.cs
Search.cshtml
UserControlle... 1, M
_ViewImports.cshtml
_ViewStart.cshtml
Migrations
Models
obj
Properties
UserController.cs
11 public UserController(IMediator mediator)
12 {
13 this.mediator = mediator;
14 }
15
16 [HttpGet]
17 public async Task<IActionResult> Add()
18 {
19 return View();
20 }
21
22 [HttpPost]
23 public async Task<IActionResult> Add(Add.Command command)
24 {
25 await mediator.Send(command);
26 return RedirectToAction(nameof(Add));
27 }
28
29 [HttpGet]
30 public async Task<IActionResult> List()
31 {
32 var result = await mediator.Send(new List.Query());
master* 0↓ 1↑ 0 1 .NET Core Launch (web) (layers - with slices)
Ln 41, Col 10 Spaces: 4 UTF-8 with BOM CRLF C#

Encapsulate by feature


User List
UserControllier
ListAll
Users/
ListAll
User Search
UserController
Search
Users/
Search
Add User
UserController
Add
Users/
Add


Developer
Success!

Refactor by feature


Deactivate User
Users/ Deactivate
User Aggregate .Deactivate()
User

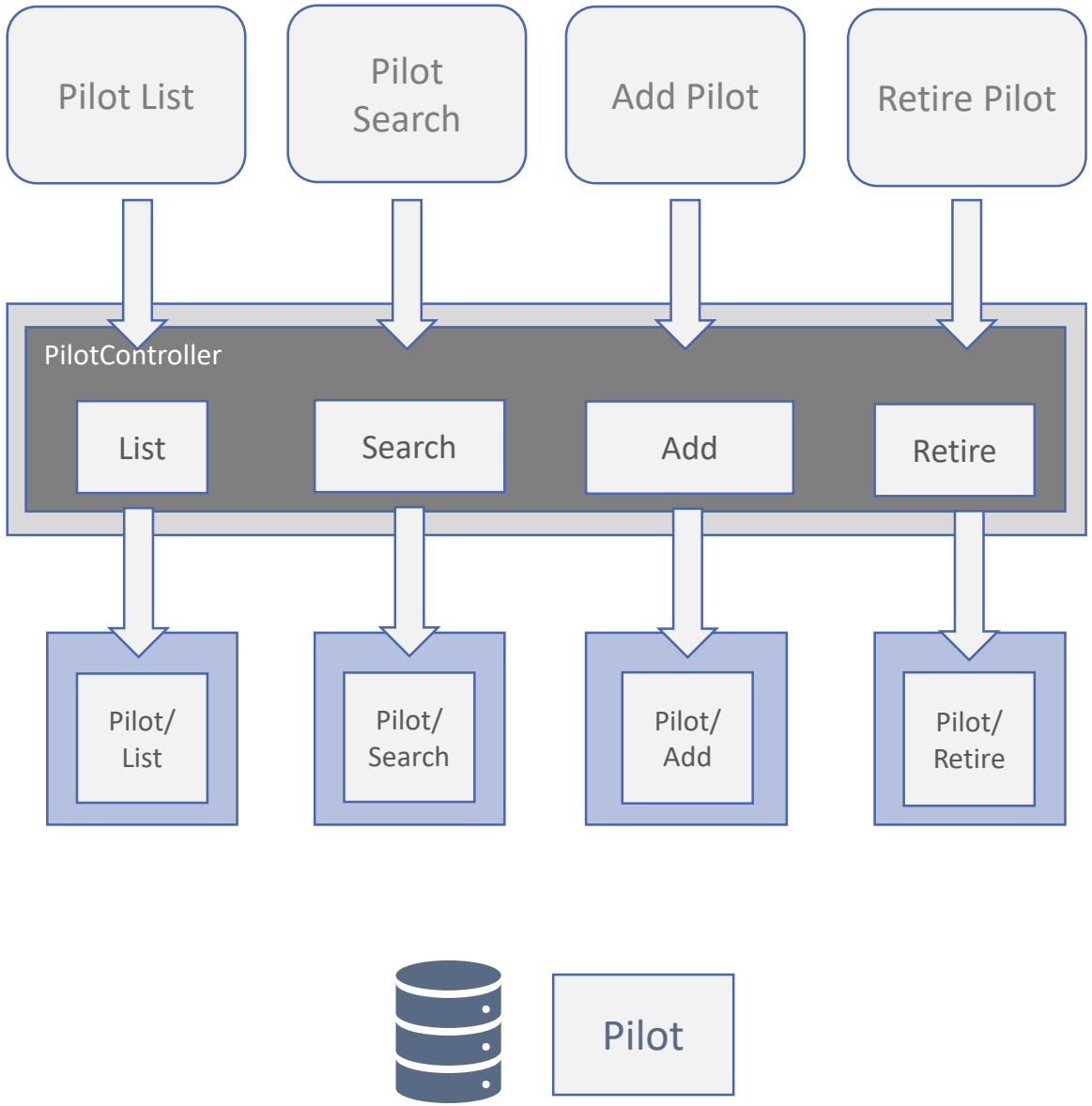
UpdateTaskDescription.cs

```
9
10  namespace Stroke_Active.WebAPI.Features.HCP.Programme
11  {
12      public class UpdateTaskDescription
13      {
14          public class Command : AuthenticatedRequest<CommandResult>
15          {
16              public string Description { get; set; }
17              public Guid PatientId { get; set; }
18              public Guid TaskId { get; set; }
19          }
20
21          public class CommandHandler : IRequestHandler<Command, CommandResult>
22          {
23              private readonly IRehabProgrammeRepository _rehabProgrammeRepository;
24              private readonly StrokeActiveContext _context;
25
26              public CommandHandler(IRehabProgrammeRepository programmeRepository, StrokeActiveContext context)
27              {
28                  _context = context;
29                  _rehabProgrammeRepository = programmeRepository;
30              }
31
32              public async Task<CommandResult> Handle(Command request, CancellationToken cancellationToken)
33              {
34                  var professionalId = await _context.HealthCareProfessionals
35                      .Where(x => x.Identifier == request.Identifier)
36                      .Select(x=>x.Id)
37                      .SingleOrDefaultAsync(cancellationToken);
38
```

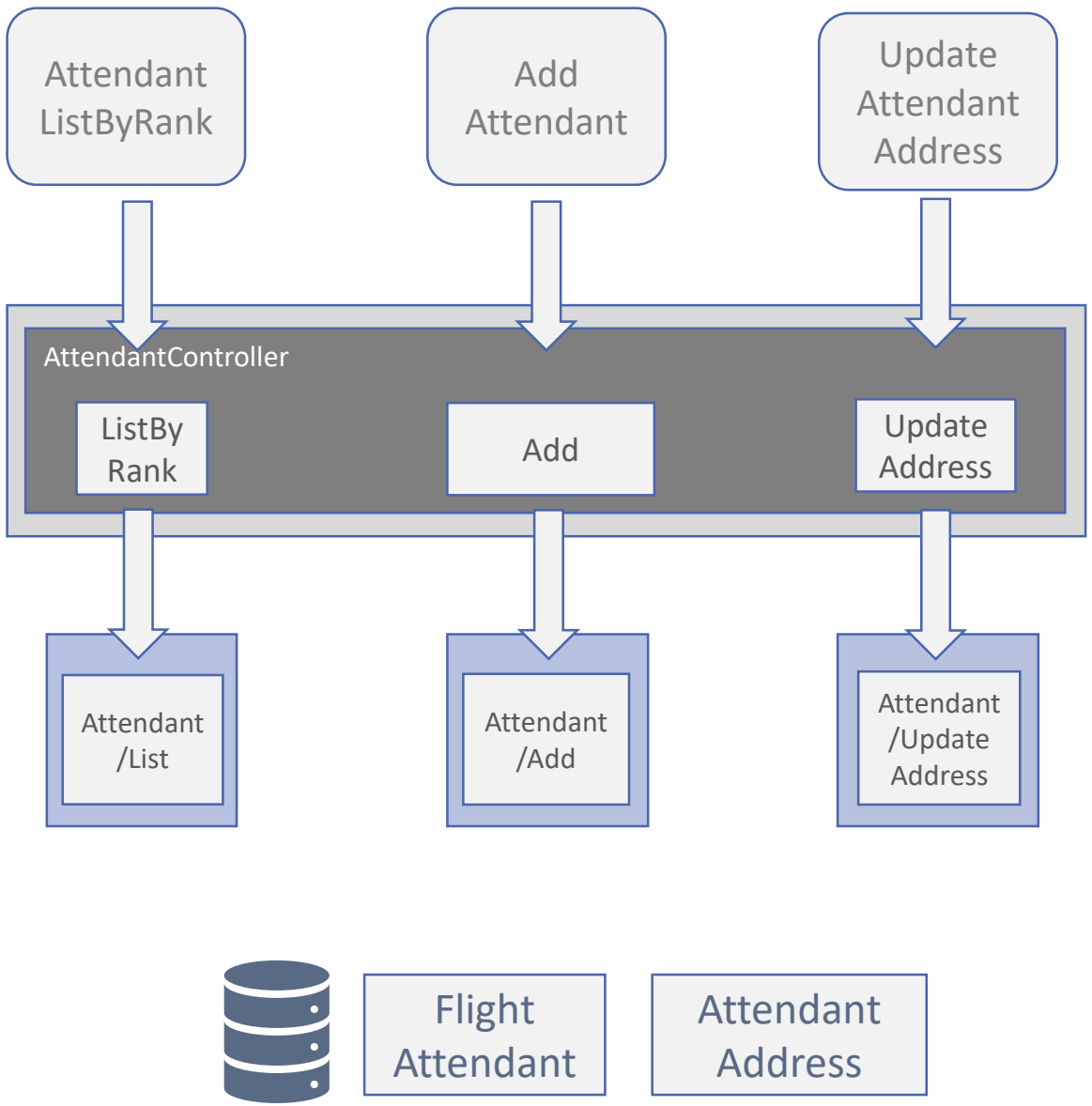
UpdateTaskDescription.cs

```
20
21          public class CommandHandler : IRequestHandler<Command, CommandResult>
22          {
23              private readonly IRehabProgrammeRepository _rehabProgrammeRepository;
24              private readonly StrokeActiveContext _context;
25
26              public CommandHandler(IRehabProgrammeRepository programmeRepository, StrokeActiveContext context)
27              {
28                  _context = context;
29                  _rehabProgrammeRepository = programmeRepository;
30              }
31
32              public async Task<CommandResult> Handle(Command request, CancellationToken cancellationToken)
33              {
34                  var professionalId = await _context.HealthCareProfessionals
35                      .Where(x => x.Identifier == request.Identifier)
36                      .Select(x=>x.Id)
37                      .SingleOrDefaultAsync(cancellationToken);
38
39                  var programme = await _rehabProgrammeRepository.ForPatient(request.PatientId);
40                  if (programme == null)
41                      return CommandResult.WithError("Could not find programme for patient");
42
43                  programme.ChangeTaskDescription(request.TaskId, request.Description, professionalId);
44
45                  return CommandResult.Success();
46              }
47          }
48      }
49  }
```

Avoid links between features (specific, business logic)



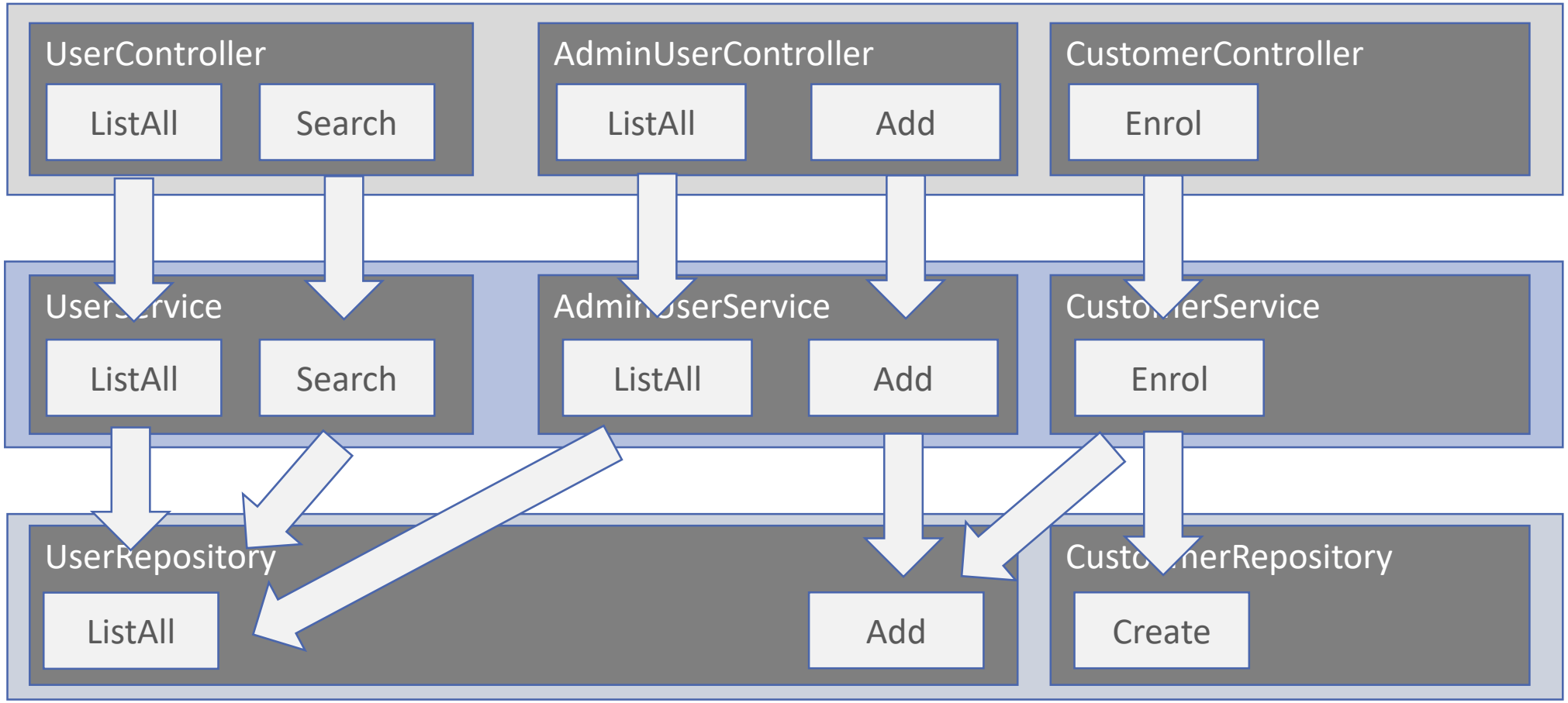
Pilot



Flight Attendant

Attendant Address

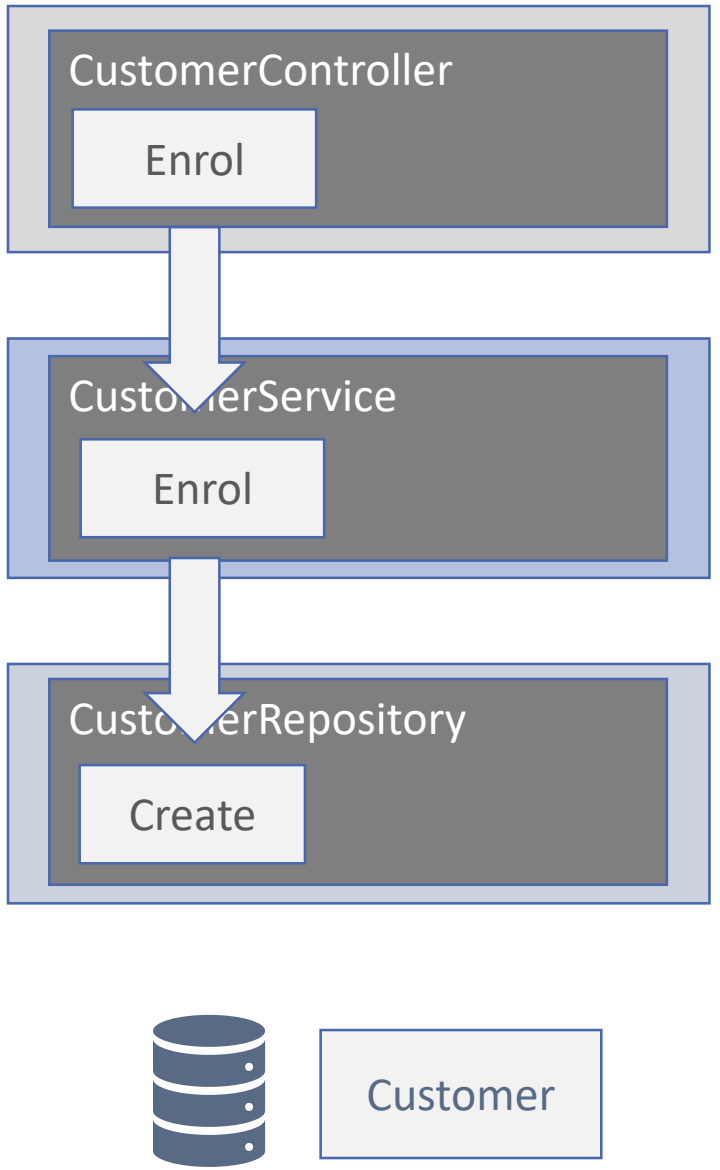
What about tests?


UserController
ListAll
Search
AdminUserController
ListAll
Add
CustomerController
Enrol
ListAll
Search
ListAll
Add
Enrol
UserRepository
ListAll
Add
Create

Test?

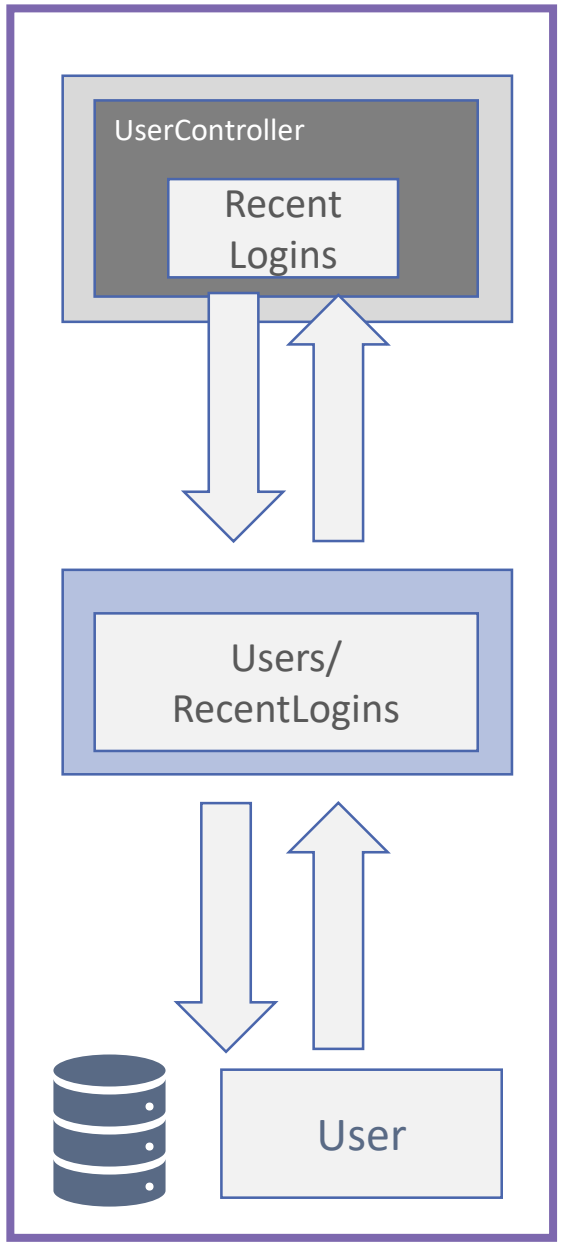
Test in isolation

Test?

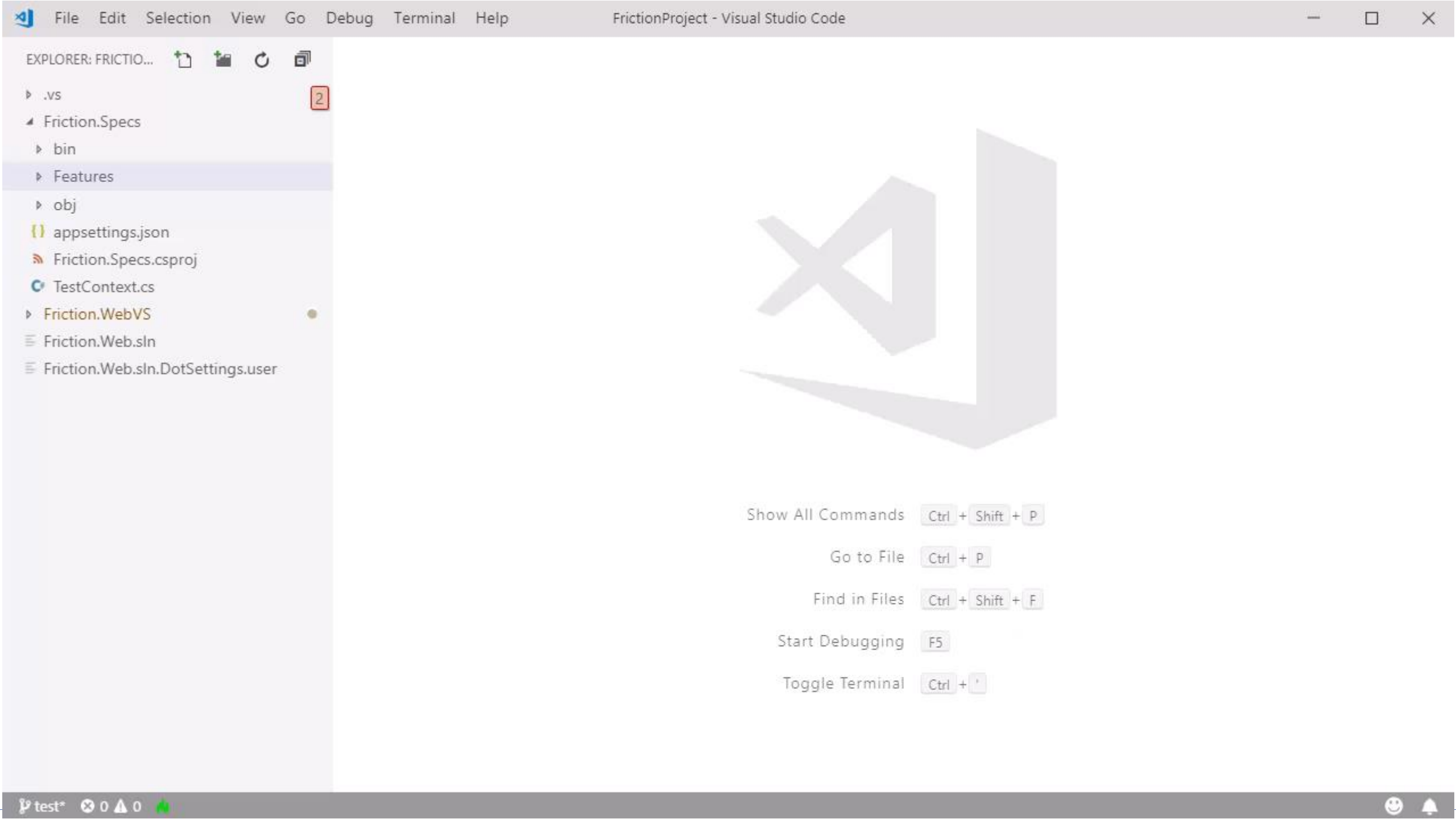


When viewing the **most recent user logins**

I should see the last x users who logged in (in descending order by their last login date/time)



The names used are those of the business domain (feature)


File Edit Selection View Go Debug Terminal Help
FrictionProject - Visual Studio Code
EXPLORER: FRICTIO...
.vs
2
Friction.Specs
bin
Features
obj
appsettings.json
Friction.Specs.csproj
TestContext.cs
Friction.WebVS
Friction.Web.sln
Friction.Web.sln.DotSettings.user
Show All Commands Ctrl + Shift + P
Go to File Ctrl + P
Find in Files Ctrl + Shift + F
Start Debugging F5
Toggle Terminal Ctrl + '
test*
0
0

File Edit Selection View Go Debug Terminal Help RecentLogins.cs - FrictionProject - Visual Studio Code

RecentLogins.cs

```
using Friction.WebVS.Features.Users;
using Friction.WebVS.Models;
using Shouldly;
using Xunit;

namespace Friction.Specs.Features.Users
{
    public class RecentLogins : IntegrationTestBase
    {
        [Fact]
        public async Task ShouldReturnTop2MostRecentlyActiveUsers()
        {
            var bob = new User { FirstName = "Bob", LastName = "Smith", LastLogin = new DateTime(2012,1,1), Id = Guid.NewGuid()  };
            var alice = new User { FirstName = "Alice", LastName = "Smith", LastLogin = new DateTime(2018, 2, 1), Id = Guid.NewGuid() };
            var cooper = new User { FirstName = "Cooper", LastName = "Barrow", LastLogin = new DateTime(2018, 2, 2), Id = Guid.NewGuid() };

            await TestContext.InsertAsync(bob, alice, cooper);

            var result = await TestContext.SendAsync(new ListMostActive.Query());

            result.Users.Count.ShouldBe(2);
            result.Users.ShouldContain(x => x.FirstName == "Alice");
            result.Users.ShouldContain(x => x.FirstName == "Cooper");
        }
    }
}
```

test* 0 0 Ln 30, Col 1 Spaces: 4 UTF-8 with BOM CRLF C# [off]

Examples in the wild

Humanitarian Toolbox – allready
https://github.com/HTBox/allReady

Jimmy Bogard – Contoso University
https://github.com/jbogard/ContosoUniversityDotNetCore

Tackling Business Complexity in a Microservice with DDD and CQRS Patterns
https://bit.ly/msmediatr

HTBox / allReady

Watch 143 | Star 792 | Fork 573

Code | Issues 233 | Pull requests 12 | Projects 1 | Wiki | Insights

This repo contains the code for allReady, an open-source solution focused on increasing awareness, efficiency and impact of preparedness campaigns as they are delivered by humanitarian and disaster response organizations in local communities. http://www.htbox.org/projects/allready

allready disaster-response preparedness-campaigns volunteers c-sharp asp-net-core

3,626 commits | 5 branches | 0 releases | 164 contributors | MIT

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

MisterJames Just a sample scenario for a campaign — Latest commit b3d35a1 7 days ago

AllReadyApp	Updating cdn sources for validation to match fallback	a year ago
SampleData	Update data import to use postalcode	2 years ago
docs	Just a sample scenario for a campaign	7 days ago
meetings	Update 2017-08-14 Team Meeting.md	2 years ago
.deployment	move deployment script to the root of the repo	3 years ago

allReady / AllReadyApp / Web-App / AllReady / Features /

Create new file | Upload files | Find file | History

stevejgordon Converting HomeController pages/action to Razor pages ... Latest commit e0acef4 on Nov 11, 2017

..		
Admin	Removing ConfigureAwait(false) usage	2 years ago
Campaigns	Converting HomeController pages/action to Razor pages	a year ago
ClosestLocation	RTM Changes builds	3 years ago
Events	A lot of test based changes	2 years ago
Home	Converting HomeController pages/action to Razor pages	a year ago
Login	Removing ConfigureAwait(false) usage	2 years ago
Manage	Fixed to display the URL in the confirmation email	2 years ago
Notifications	Fixing some missing task > volunteer task renaming	2 years ago
Organizations	Removing ConfigureAwait(false) usage	2 years ago
Requests	P.M-A - Issue 1585	2 years ago
Resource	a bunch of renamings/code cleanup for mostly MediatR handlers.	2 years ago
Sms	Move phone number validation check into foreach loop along with the a...	2 years ago
Tasks	Casing changes for enums (neglected to commit these in previous push).	2 years ago
Volunteers	Volunteers Dashboard enahnsements	2 years ago

```
pace AllReady.Features.Organizations
{
    public class OrganizationDetailsQueryHandler : IAsyncRequestHandler<OrganizationDetailsQuery, OrganizationViewModel>
    {
        private readonly AllReadyContext _context;

        public OrganizationDetailsQueryHandler(AllReadyContext context)
        {
            _context = context;
        }

        public async Task<OrganizationViewModel> Handle(OrganizationDetailsQuery message)
        {
            var result = await _context.Organizations.AsNoTracking()
                .Include(t => t.Campaigns)
                .Include(t => t.Location)
                .SingleOrDefaultAsync(t => t.Id == message.Id);

            if (result == null)
                return null;

            result.Campaigns.RemoveAll(c => c.Locked);

            return new OrganizationViewModel(result);
        }
    }
}
```

There is no “best” design.

There is only the “best design given our current understanding”

Jimmy Bogard – “Strengthening your domain” https://bit.ly/NoBestDesign

jonhilton.net/slices

@jonhilt

Thank You

Jon Hilton

jon@jonhilton.net

jonhilton.net/slices