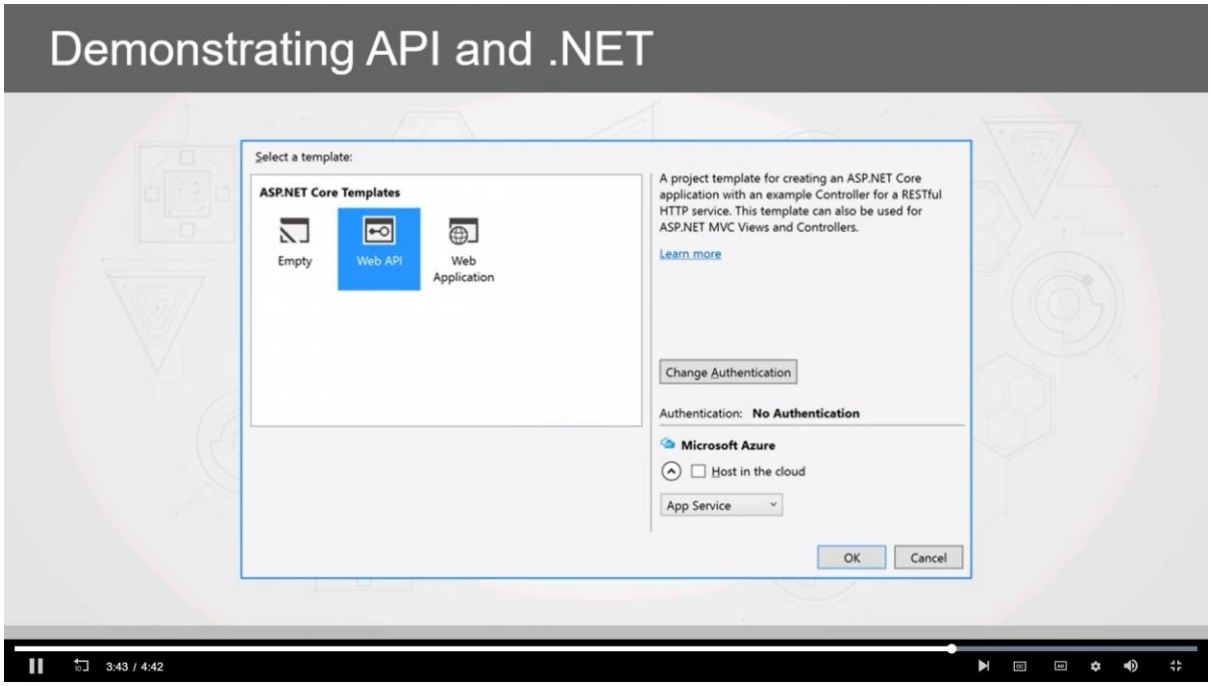
Overview of Design

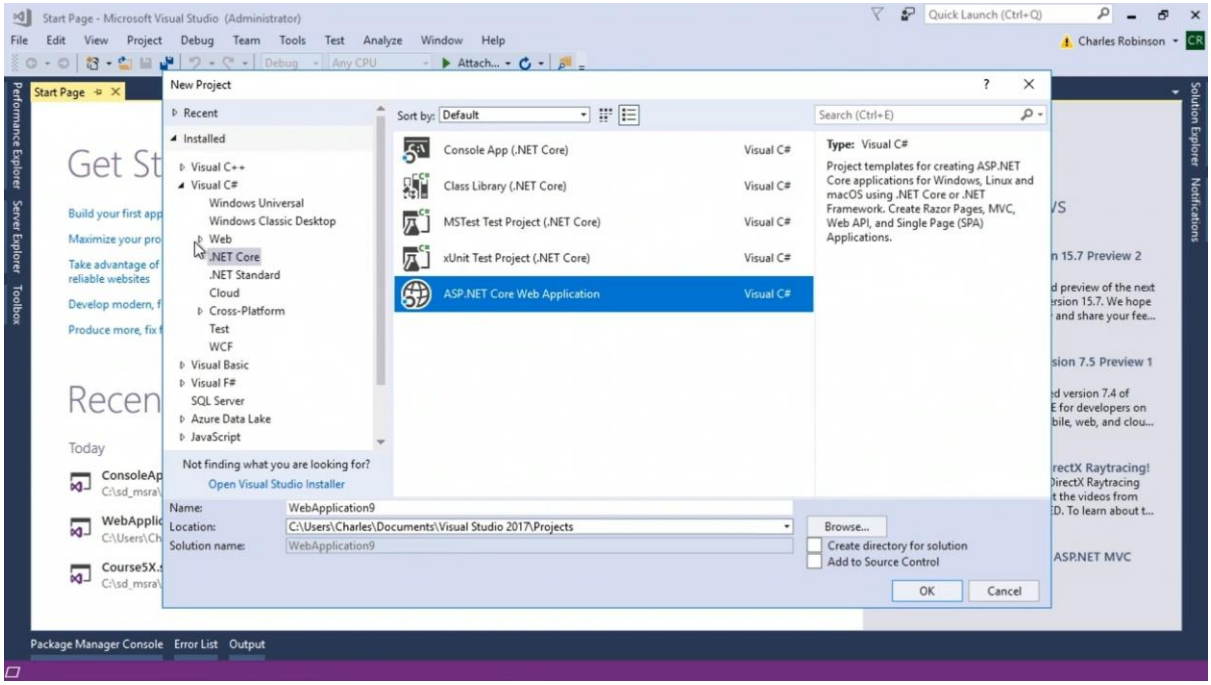
- Well-designed REST API
- Basic steps:
  - Install .NET Core
  - Create a web application with ASP.NET
    - Choose a Web API application
    - Startup.cs
    - Global.asax
  - Create an API controller
    - `GET` method to list items

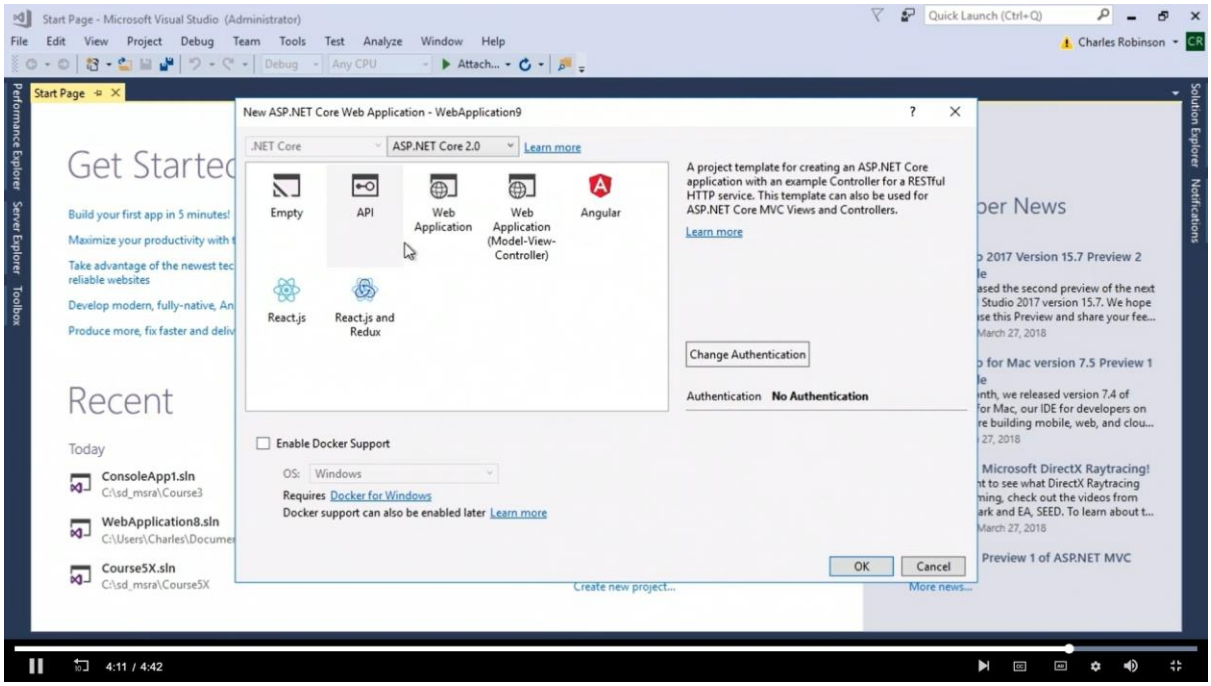
Implementation of Design

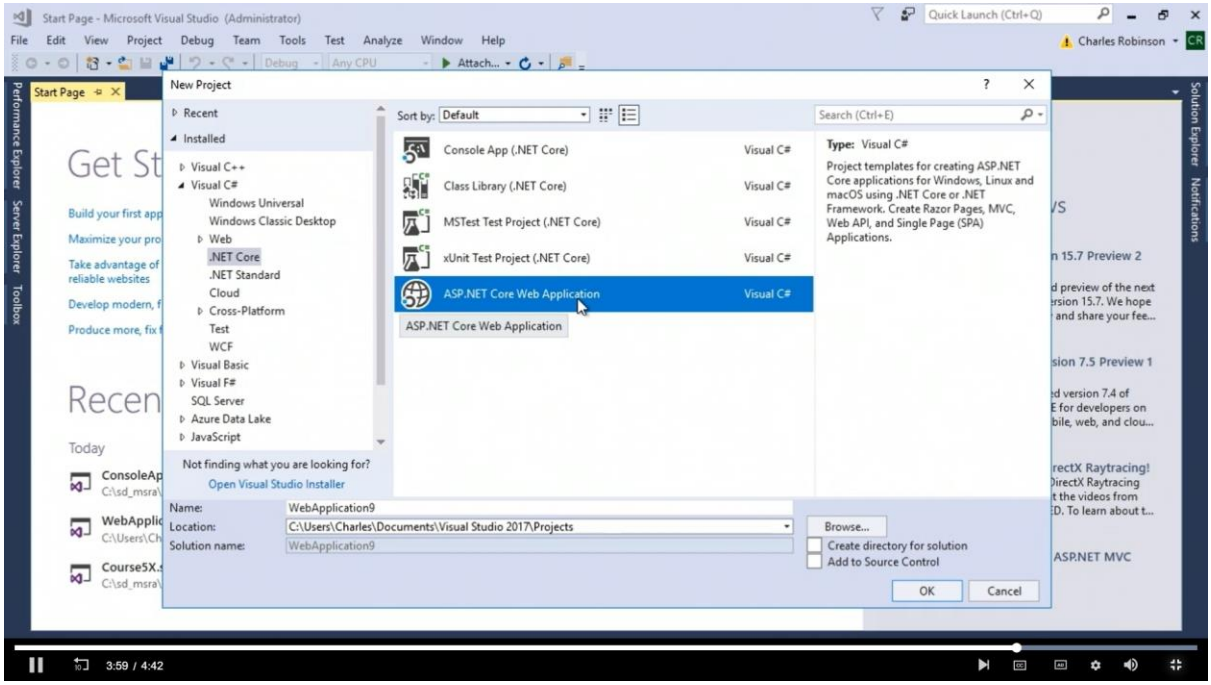
- Data Model
  - Design the model according to the specifications
  - Object representing data in the app
- Define the controllers
  - Define the necessary controllers for the API
  - Object that handles requests
- Configure the controllers

Demonstrating API and .NET

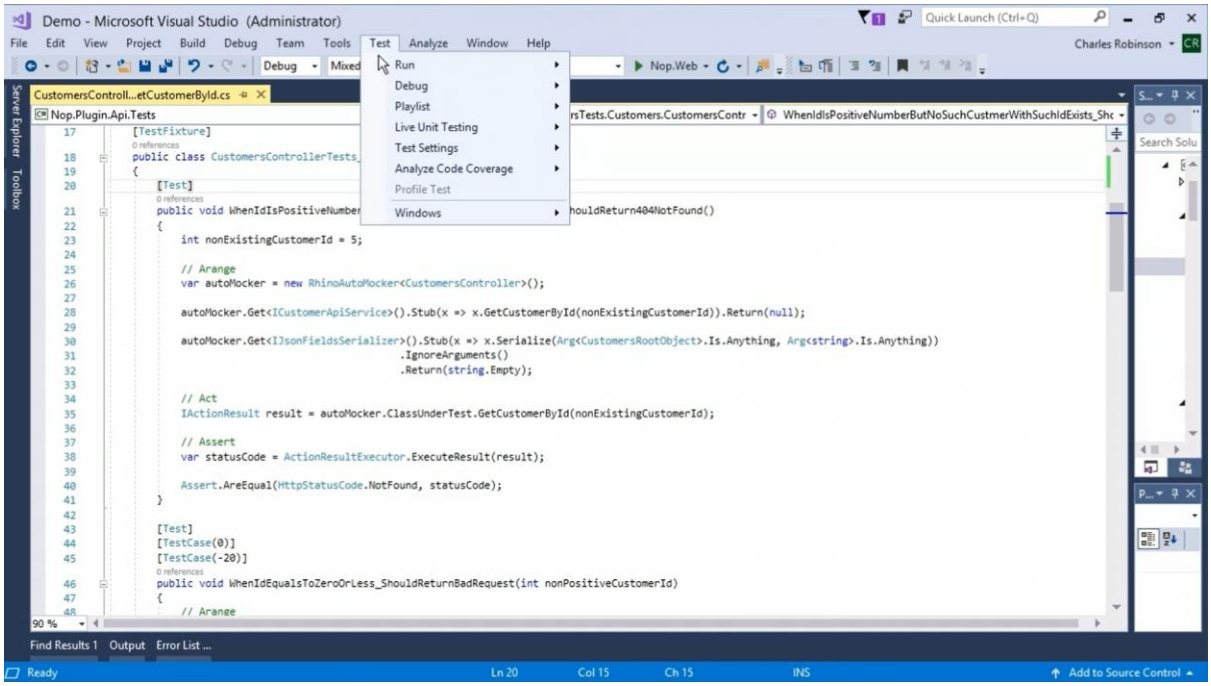


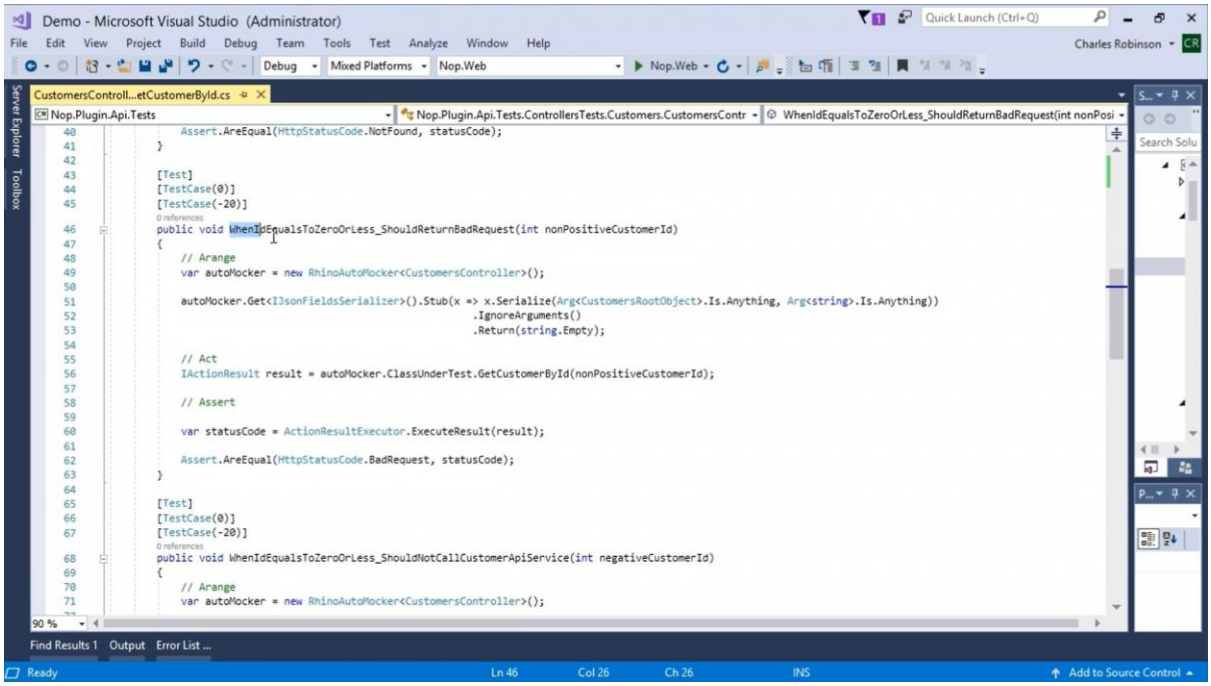


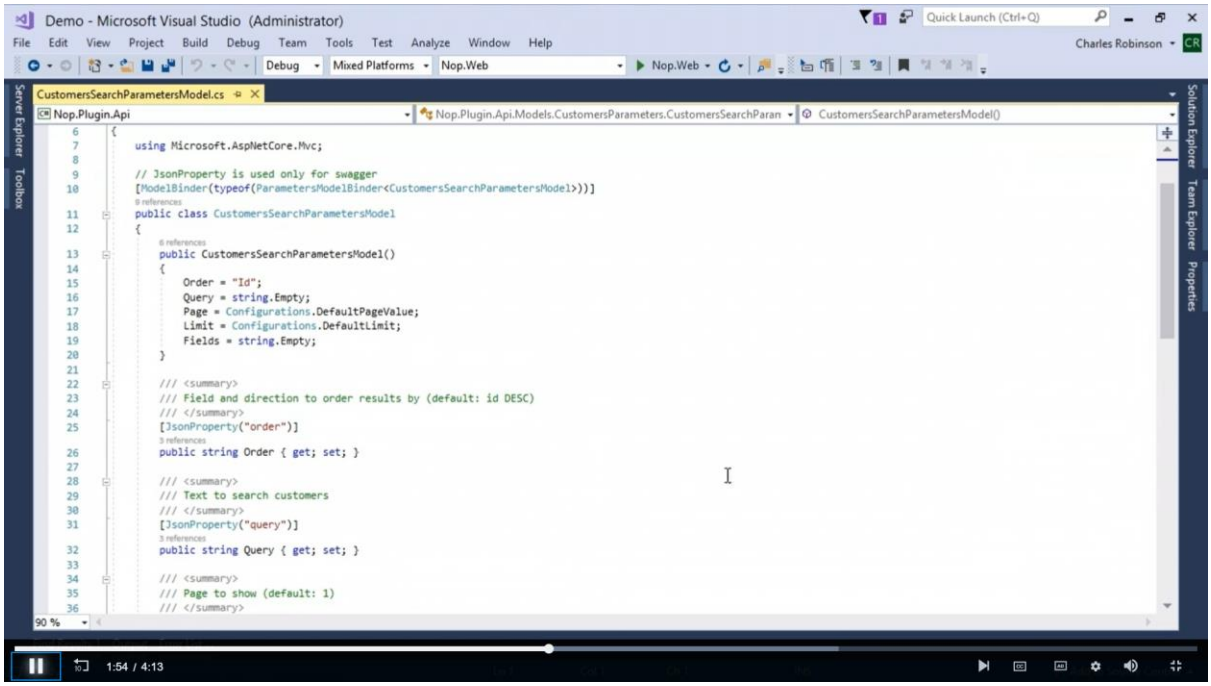


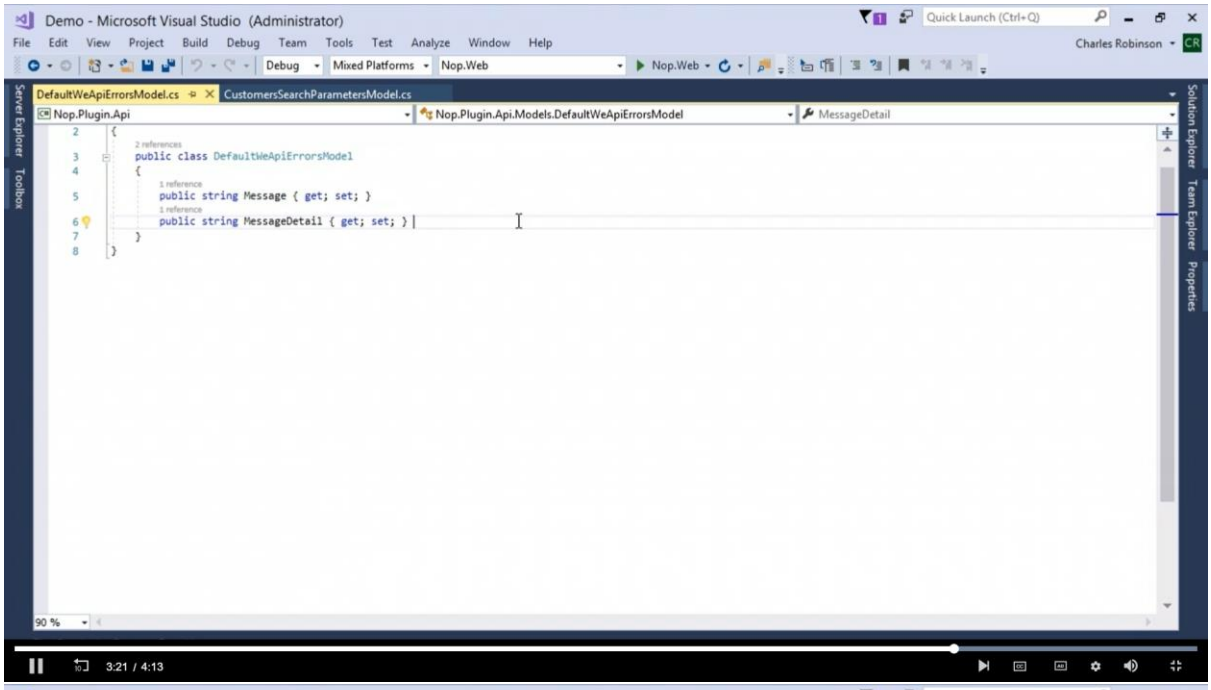


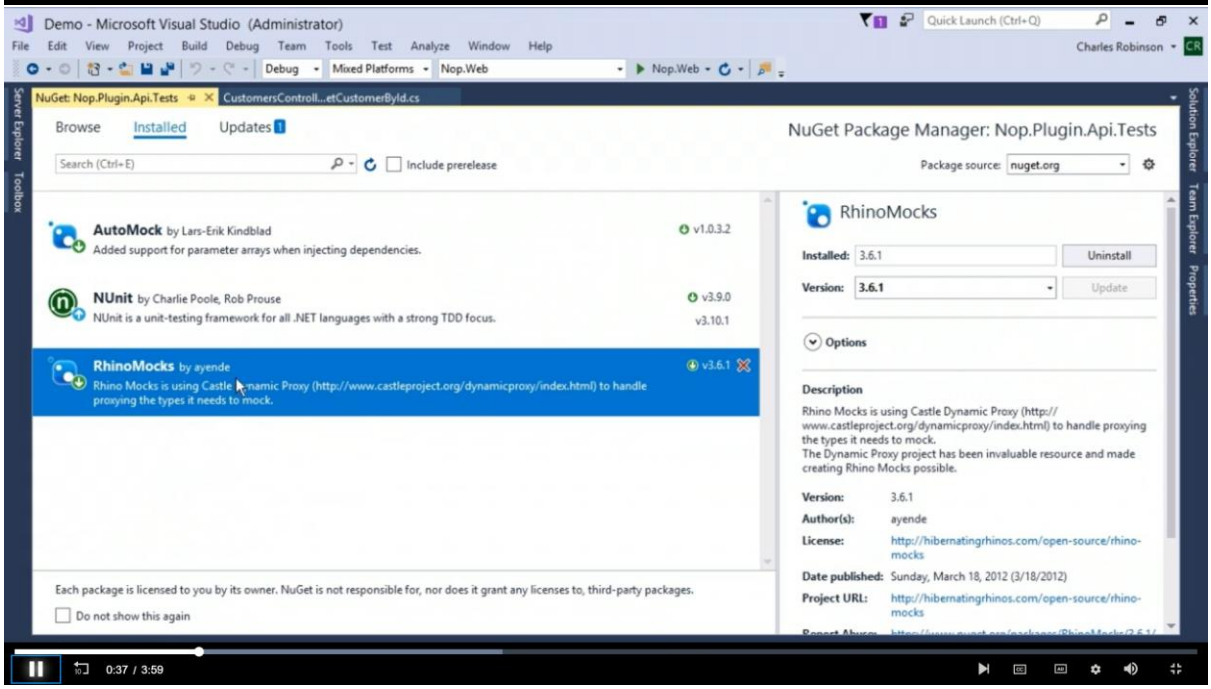
Unit testing

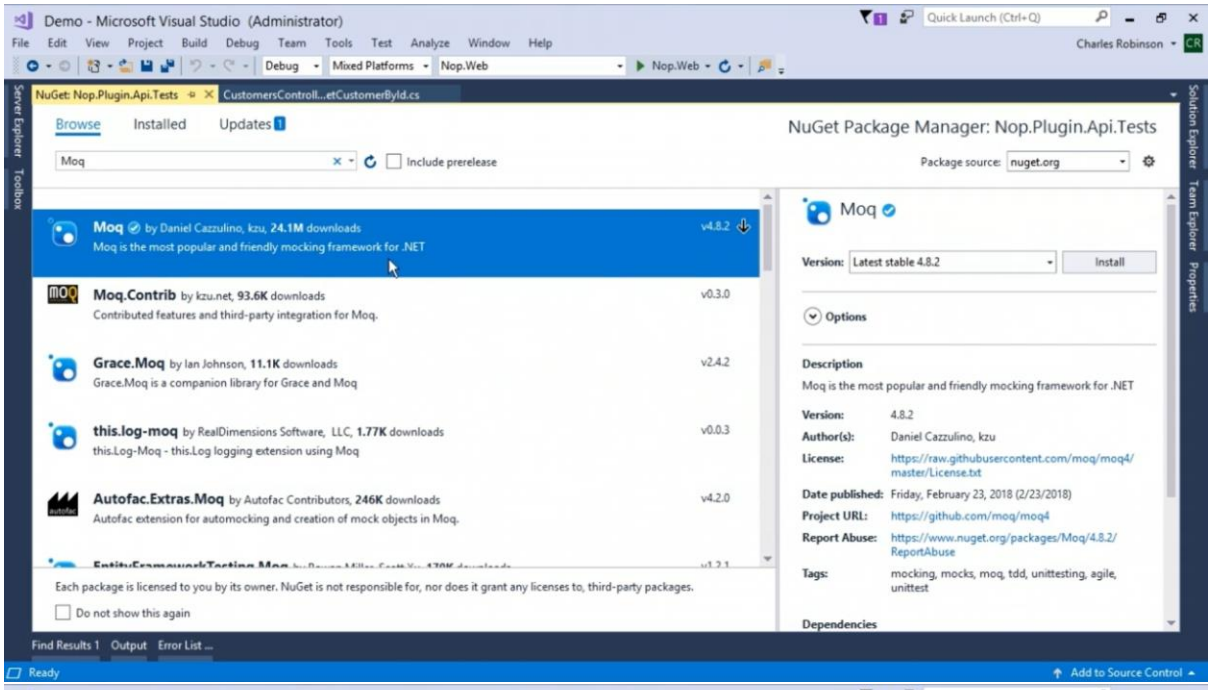

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Run
Debug
Playlist
Live Unit Testing
Test Settings
Analyze Code Coverage
Profile Test
Windows
CustomersControll...etCustomerById.cs
Nop.Plugin.Api.Tests
[TestFixture]
public class CustomersControllerTests_
{
[Test]
public void WhenIdIsPositiveNumber
int nonExistingCustomerId = 5;
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
autoMocker.Get<ICustomerApiService>().Stub(x => x.GetCustomerById(nonExistingCustomerId)).Return(null);
autoMocker.Get<IJsonFieldsSerializer>().Stub(x => x.Serialize(Arg<CustomersRootObject>.Is.Anything, Arg<string>.Is.Anything))
.IgnoreArguments()
.Return(string.Empty);
// Act
IActionResult result = autoMocker.ClassUnderTest.GetCustomerById(nonExistingCustomerId);
// Assert
var statusCode = ActionResultExecutor.ExecuteResult(result);
Assert.AreEqual(HttpStatusCode.NotFound, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
public void WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPositiveCustomerId)
{
// Arange
Find Results 1 Output Error List ...
Ready Ln 20 Col 15 Ch 15 INS Add to Source Control

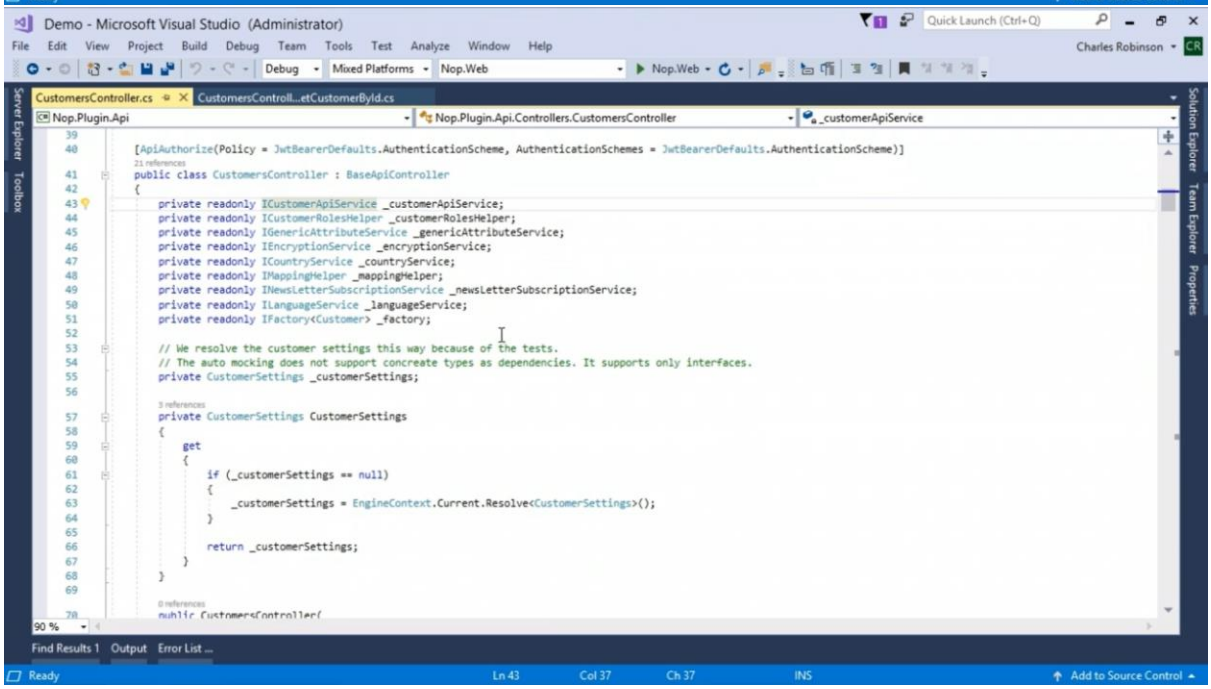

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Mixed Platforms
Nop.Web
CustomersControll...etCustomerById.cs
Nop.Plugin.Api.Tests
Nop.Plugin.Api.Tests.ControllersTests.Customers.CustomersContr
WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPosi
Assert.AreEqual(HttpStatusCode.NotFound, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
public void WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPositiveCustomerId)
{
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
autoMocker.Get<IJsonFieldsSerializer>().Stub(x => x.Serialize(Arg<CustomersRootObject>.Is.Anything, Arg<string>.Is.Anything))
.IgnoreArguments()
.Return(string.Empty);
// Act
IActionResult result = autoMocker.ClassUnderTest.GetCustomerById(nonPositiveCustomerId);
// Assert
var statusCode = ActionResultExecutor.ExecuteResult(result);
Assert.AreEqual(HttpStatusCode.BadRequest, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
public void WhenIdEqualsToZeroOrLess_ShouldNotCallCustomerApiService(int negativeCustomerId)
{
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
Find Results 1 Output Error List ...
Ready Ln 46 Col 26 Ch 26 INS Add to Source Control

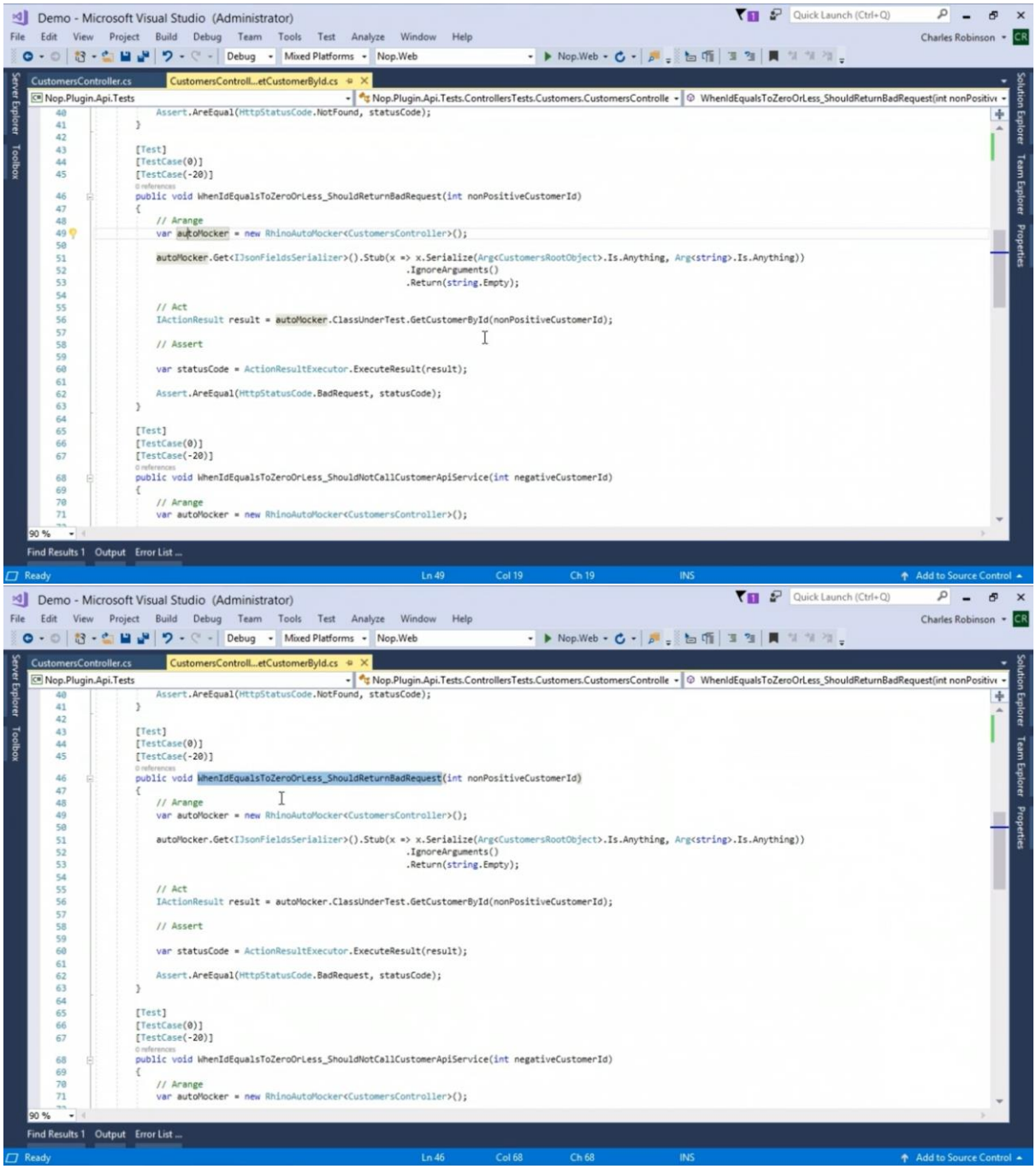

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
Nop.Web
Quick Launch (Ctrl+Q)
CustomersSearchParametersModel.cs
Nop.Plugin.Api
Nop.Plugin.Api.Models.CustomersParameters.CustomersSearchParam
CustomersSearchParametersModel()
Server Explorer
Toolbox
Solution Explorer
Team Explorer
Properties
6 {
7 using Microsoft.AspNetCore.Mvc;
8
9 // JsonProperty is used only for swagger
10 [ModelBinder(typeof(ParametersModelBinder<CustomersSearchParametersModel>))]
9 references
11 public class CustomersSearchParametersModel
12 {
6 references
13 public CustomersSearchParametersModel()
14 {
15 Order = "Id";
16 Query = string.Empty;
17 Page = Configurations.DefaultPageValue;
18 Limit = Configurations.DefaultLimit;
19 Fields = string.Empty;
20 }
21
22 /// <summary>
23 /// Field and direction to order results by (default: id DESC)
24 /// </summary>
25 [JsonProperty("order")]
3 references
26 public string Order { get; set; }
27
28 /// <summary>
29 /// Text to search customers
30 /// </summary>
31 [JsonProperty("query")]
3 references
32 public string Query { get; set; }
33
34 /// <summary>
35 /// Page to show (default: 1)
36 /// </summary>
90 %
1:54 / 4:13


Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
DefaultWeApiErrorsModel.cs
CustomersSearchParametersModel.cs
Nop.Plugin.Api
Nop.Plugin.Api.Models.DefaultWeApiErrorsModel
MessageDetail
2 references
public class DefaultWeApiErrorsModel
{
1 reference
public string Message { get; set; }
1 reference
public string MessageDetail { get; set; }
}
}
90 %
3:21 / 4:13

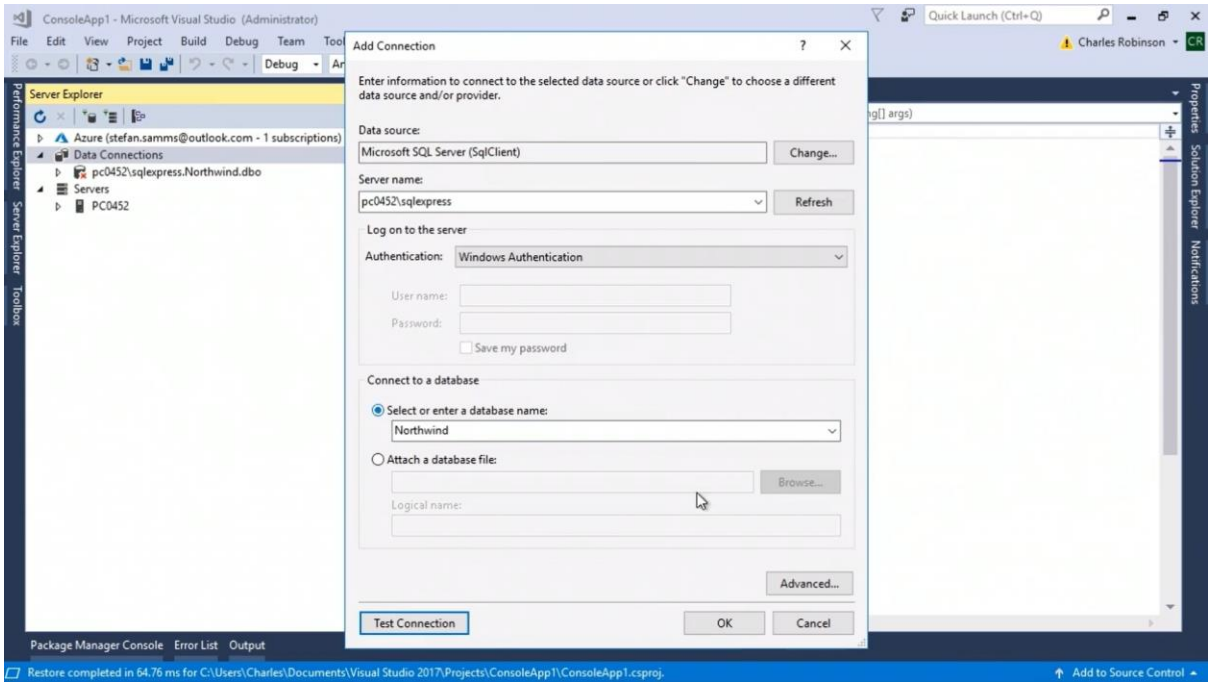

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api.Tests
CustomersControll...etCustomerById.cs
Browse
Installed
Updates 1
NuGet Package Manager: Nop.Plugin.Api.Tests
Search (Ctrl+E)
Include prerelease
Package source: nuget.org
AutoMock by Lars-Erik Kindblad
Added support for parameter arrays when injecting dependencies.
v1.0.3.2
NUnit by Charlie Poole, Rob Prouse
NUnit is a unit-testing framework for all .NET languages with a strong TDD focus.
v3.9.0
v3.10.1
RhinoMocks by ayende
Rhino Mocks is using Castle Dynamic Proxy (http://www.castleproject.org/dynamicproxy/index.html) to handle proxying the types it needs to mock.
v3.6.1
RhinoMocks
Installed: 3.6.1
Uninstall
Version: 3.6.1
Update
Options
Description
Rhino Mocks is using Castle Dynamic Proxy (http://www.castleproject.org/dynamicproxy/index.html) to handle proxying the types it needs to mock.
The Dynamic Proxy project has been invaluable resource and made creating Rhino Mocks possible.
Version: 3.6.1
Author(s): ayende
License: http://hibernatingrhinos.com/open-source/rhino-mocks
Date published: Sunday, March 18, 2012 (3/18/2012)
Project URL: http://hibernatingrhinos.com/open-source/rhino-mocks
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again
0:37 / 3:59

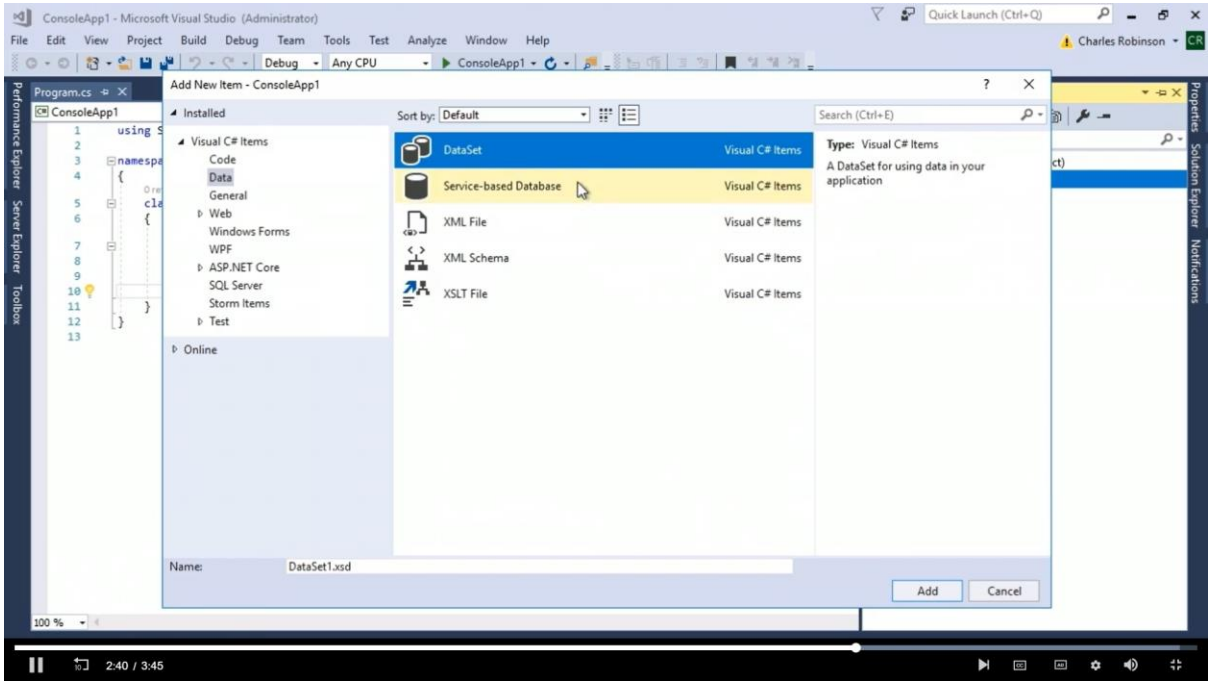

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api.Tests
CustomersControll...etCustomerById.cs
Browse
Installed
Updates 1
NuGet Package Manager: Nop.Plugin.Api.Tests
Moq
Include prerelease
Package source: nuget.org
Moq by Daniel Cazzulino, kzu, 24.1M downloads
Moq is the most popular and friendly mocking framework for .NET
v4.8.2
Moq.Contrib by kzu.net, 93.6K downloads
Contributed features and third-party integration for Moq.
v0.3.0
Grace.Moq by Ian Johnson, 11.1K downloads
Grace.Moq is a companion library for Grace and Moq
v2.4.2
this.log-moq by RealDimensions Software, LLC, 1.77K downloads
this.Log-Moq - this.Log logging extension using Moq
v0.0.3
Autofac.Extras.Moq by Autofac Contributors, 246K downloads
Autofac extension for automocking and creation of mock objects in Moq.
v4.2.0
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again
Moq
Version: Latest stable 4.8.2
Install
Options
Description
Moq is the most popular and friendly mocking framework for .NET
Version: 4.8.2
Author(s): Daniel Cazzulino, kzu
License: https://raw.githubusercontent.com/moq/moq4/master/License.txt
Date published: Friday, February 23, 2018 (2/23/2018)
Project URL: https://github.com/moq/moq4
Report Abuse: https://www.nuget.org/packages/Moq/4.8.2/ReportAbuse
Tags: mocking, mocks, moq, tdd, unittesting, agile, unittest
Dependencies
Find Results 1 Output Error List ...
Ready
Add to Source Control

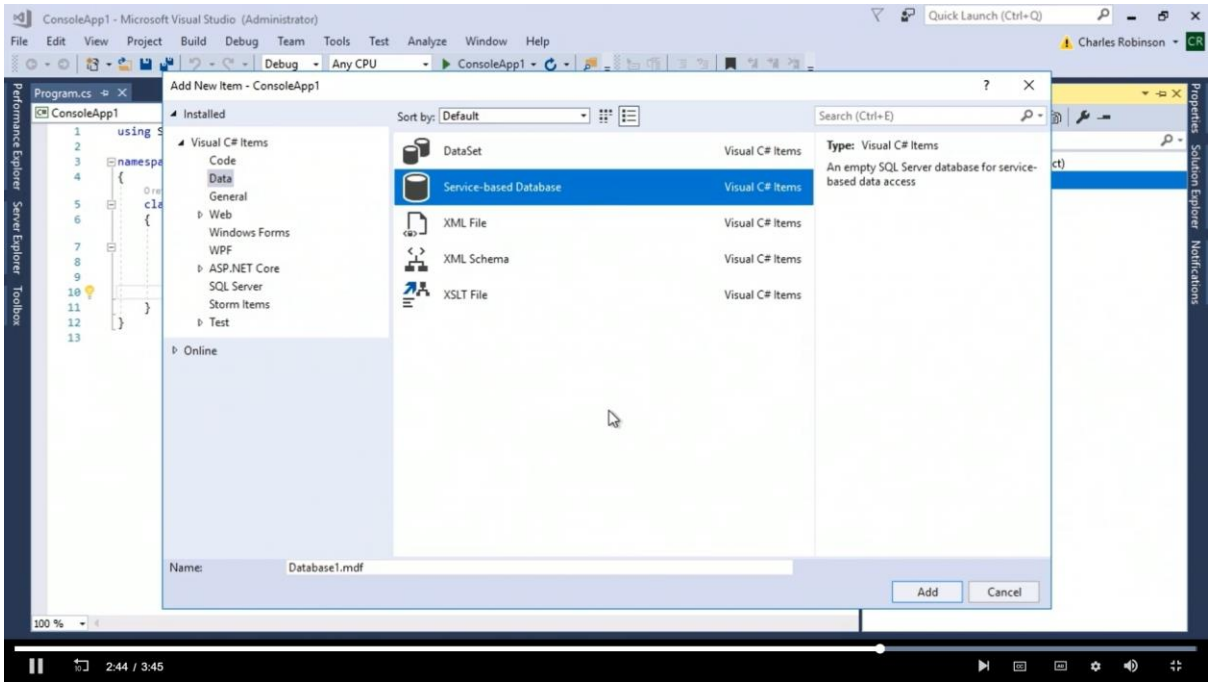

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
CustomersController.cs
CustomersControll...etCustomerById.cs
Nop.Plugin.Api
Nop.Plugin.Api.Controllers.CustomersController
_customerApiService
[ApiAuthorize(Policy = JwtBearerDefaults.AuthenticationScheme, AuthenticationSchemes = JwtBearerDefaults.AuthenticationScheme)]
21 references
public class CustomersController : BaseApiController
{
private readonly ICustomerApiService _customerApiService;
private readonly ICustomerRolesHelper _customerRolesHelper;
private readonly IGenericAttributeService _genericAttributeService;
private readonly IEncryptionService _encryptionService;
private readonly ICountryService _countryService;
private readonly IMappingHelper _mappingHelper;
private readonly INewsLetterSubscriptionService _newsLetterSubscriptionService;
private readonly ILanguageService _languageService;
private readonly IFactory<Customer> _factory;
// We resolve the customer settings this way because of the tests.
// The auto mocking does not support concreate types as dependencies. It supports only interfaces.
private CustomerSettings _customerSettings;
3 references
private CustomerSettings CustomerSettings
{
get
{
if (_customerSettings == null)
{
_customerSettings = EngineContext.Current.Resolve<CustomerSettings>();
}
return _customerSettings;
}
}
0 references
public CustomersController(
90 %
Find Results 1 Output Error List ...
Ready
Ln 43 Col 37 Ch 37 INS
Add to Source Control

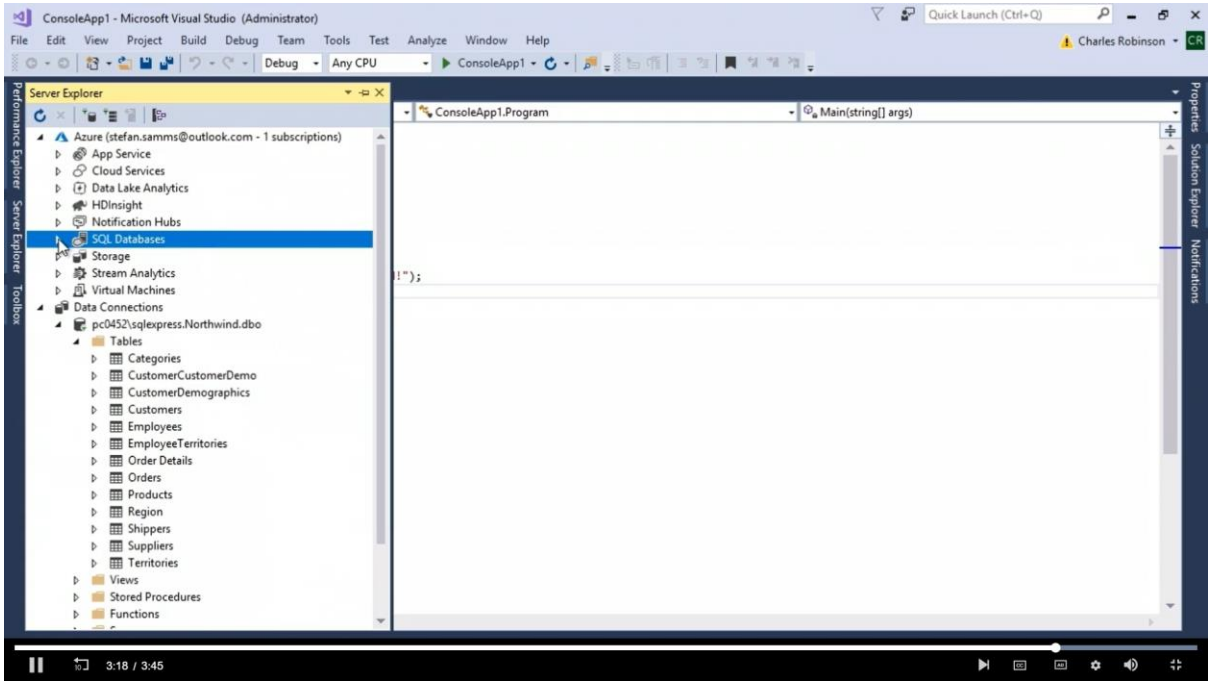

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
CustomersController.cs
CustomersControll...etCustomerById.cs
Nop.Plugin.Api.Tests
Nop.Plugin.Api.Tests.ControllersTests.Customers.CustomersControlle
WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPositive
Assert.AreEqual(HttpStatusCode.NotFound, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
0 references
public void WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPositiveCustomerId)
{
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
autoMocker.Get<IJsonFieldsSerializer>().Stub(x => x.Serialize(Arg<CustomersRootObject>.Is.Anything, Arg<string>.Is.Anything))
.IgnoreArguments()
.Return(string.Empty);
// Act
IActionResult result = autoMocker.ClassUnderTest.GetCustomerById(nonPositiveCustomerId);
// Assert
var statusCode = ActionResultExecutor.ExecuteResult(result);
Assert.AreEqual(HttpStatusCode.BadRequest, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
0 references
public void WhenIdEqualsToZeroOrLess_ShouldNotCallCustomerApiService(int negativeCustomerId)
{
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
90 %
Find Results 1 Output Error List ...
Ready
Ln 49 Col 19 Ch 19 INS
Add to Source Control

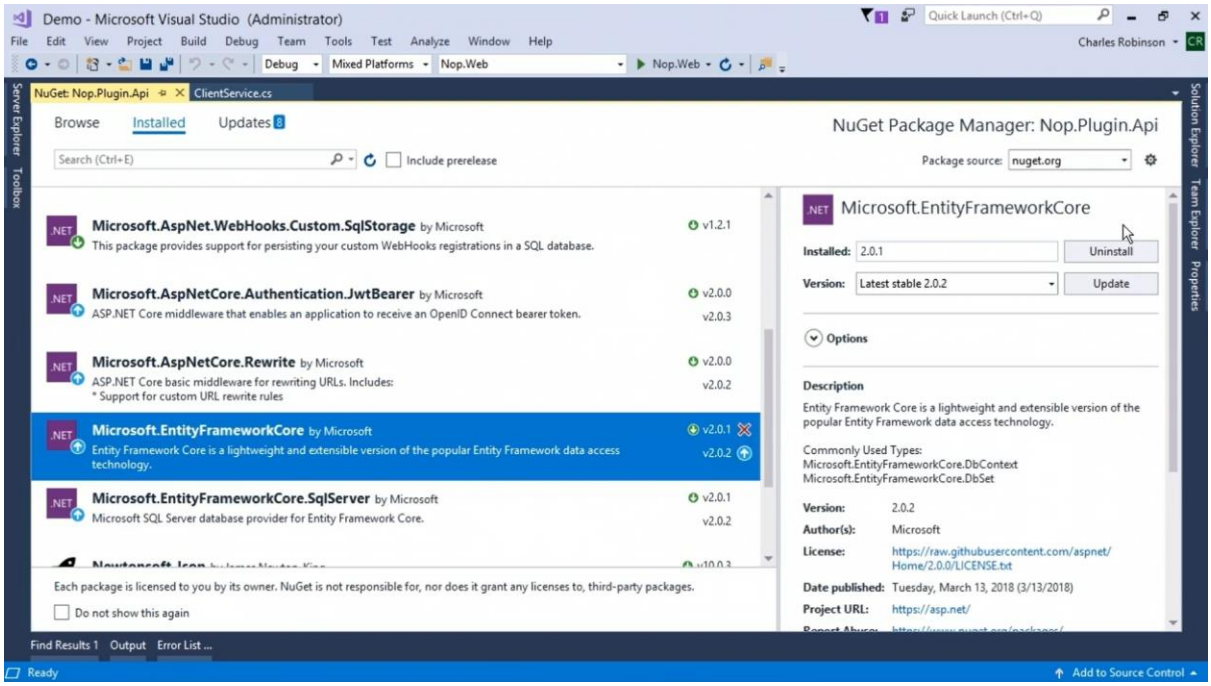
Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
CustomersController.cs
CustomersControll...etCustomerById.cs
Nop.Plugin.Api.Tests
Nop.Plugin.Api.Tests.ControllersTests.Customers.CustomersControlle
WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPositive
Assert.AreEqual(HttpStatusCode.NotFound, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
0 references
public void WhenIdEqualsToZeroOrLess_ShouldReturnBadRequest(int nonPositiveCustomerId)
{
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
autoMocker.Get<IJsonFieldsSerializer>().Stub(x => x.Serialize(Arg<CustomersRootObject>.Is.Anything, Arg<string>.Is.Anything))
.IgnoreArguments()
.Return(string.Empty);
// Act
IActionResult result = autoMocker.ClassUnderTest.GetCustomerById(nonPositiveCustomerId);
// Assert
var statusCode = ActionResultExecutor.ExecuteResult(result);
Assert.AreEqual(HttpStatusCode.BadRequest, statusCode);
}
[Test]
[TestCase(0)]
[TestCase(-20)]
0 references
public void WhenIdEqualsToZeroOrLess_ShouldNotCallCustomerApiService(int negativeCustomerId)
{
// Arange
var autoMocker = new RhinoAutoMocker<CustomersController>();
90 %
Find Results 1 Output Error List ...
Ready
Ln 46 Col 68 Ch 68 INS
Add to Source Control

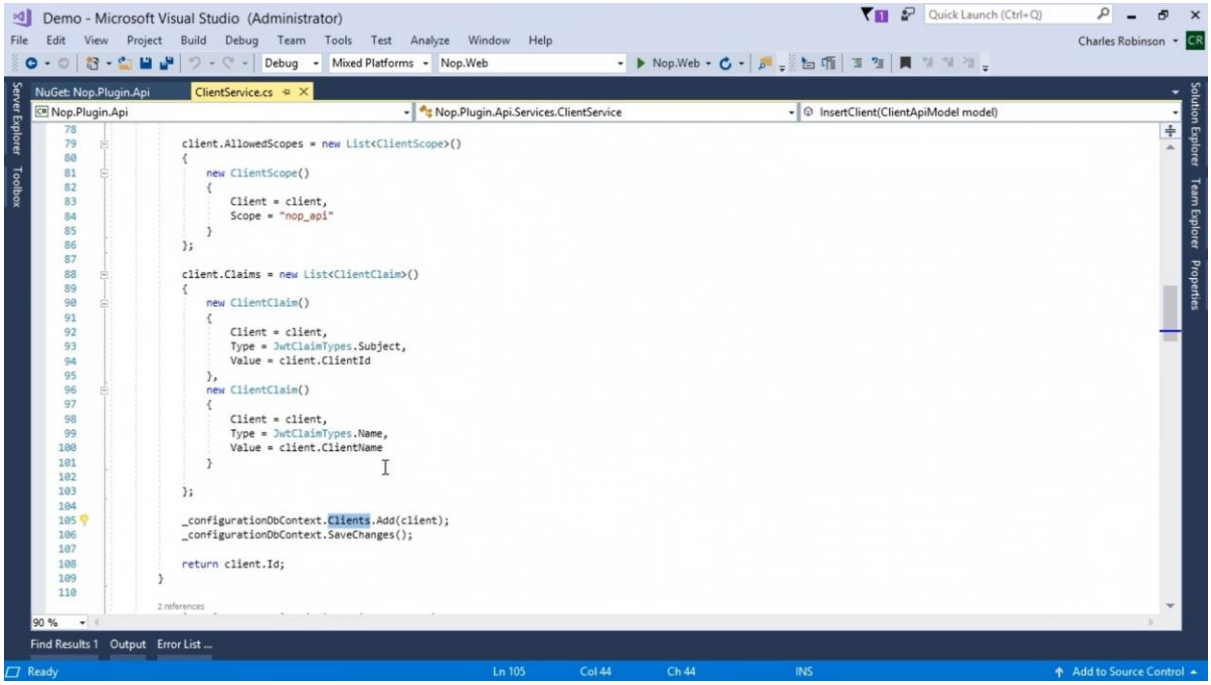

ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tool
Quick Launch (Ctrl+Q)
Charles Robinson
Server Explorer
Azure (stefan.samms@outlook.com - 1 subscriptions)
Data Connections
pc0452\sqlexpress.Northwind.dbo
Servers
PC0452
Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
pc0452\sqlexpress
Refresh
Log on to the server
Authentication:
Windows Authentication
User name:
Password:
Save my password
Connect to a database
Select or enter a database name:
Northwind
Attach a database file:
Browse...
Logical name:
Advanced...
Test Connection
OK
Cancel
Package Manager Console
Error List
Output
Restore completed in 64.76 ms for C:\Users\Charles\Documents\Visual Studio 2017\Projects\ConsoleApp1\ConsoleApp1.csproj.
Add to Source Control

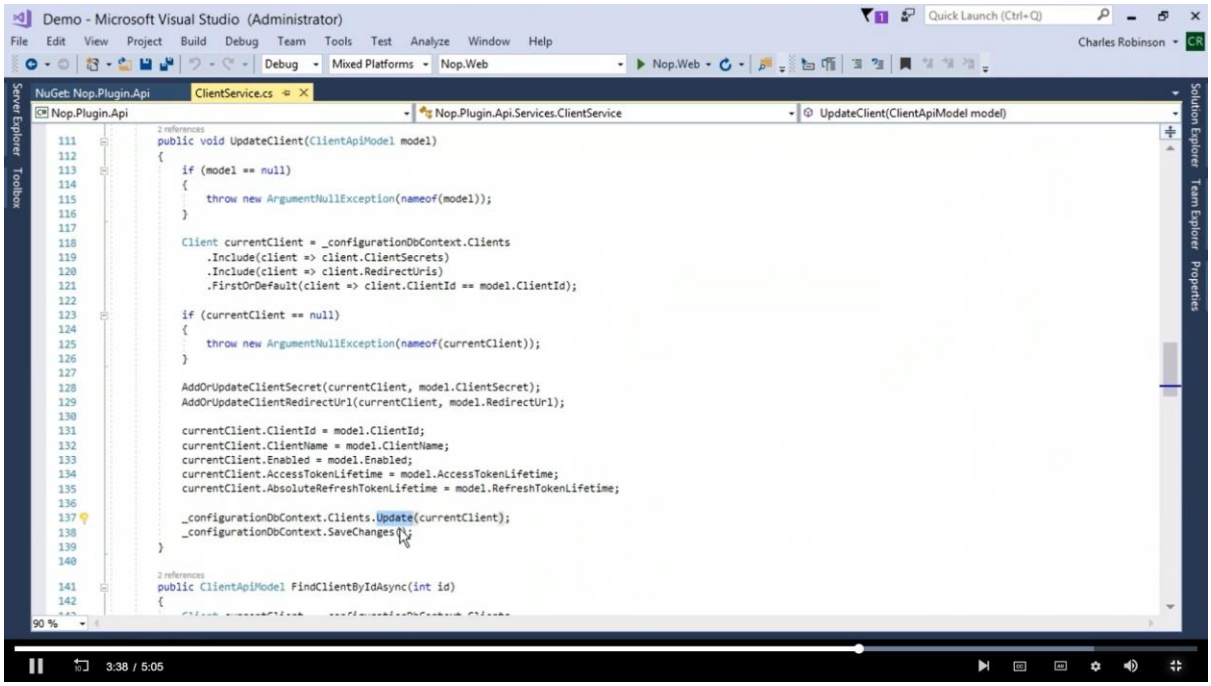

ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Debug
Any CPU
ConsoleApp1
Program.cs
ConsoleApp1
Add New Item - ConsoleApp1
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ASP.NET Core
SQL Server
Storm Items
Test
Online
Sort by: Default
Search (Ctrl+E)
DataSet
Visual C# Items
Service-based Database
Visual C# Items
XML File
Visual C# Items
XML Schema
Visual C# Items
XSLT File
Visual C# Items
Type: Visual C# Items
A DataSet for using data in your application
Name:
DataSet1.xsd
Add
Cancel
100 %
2:40 / 3:45

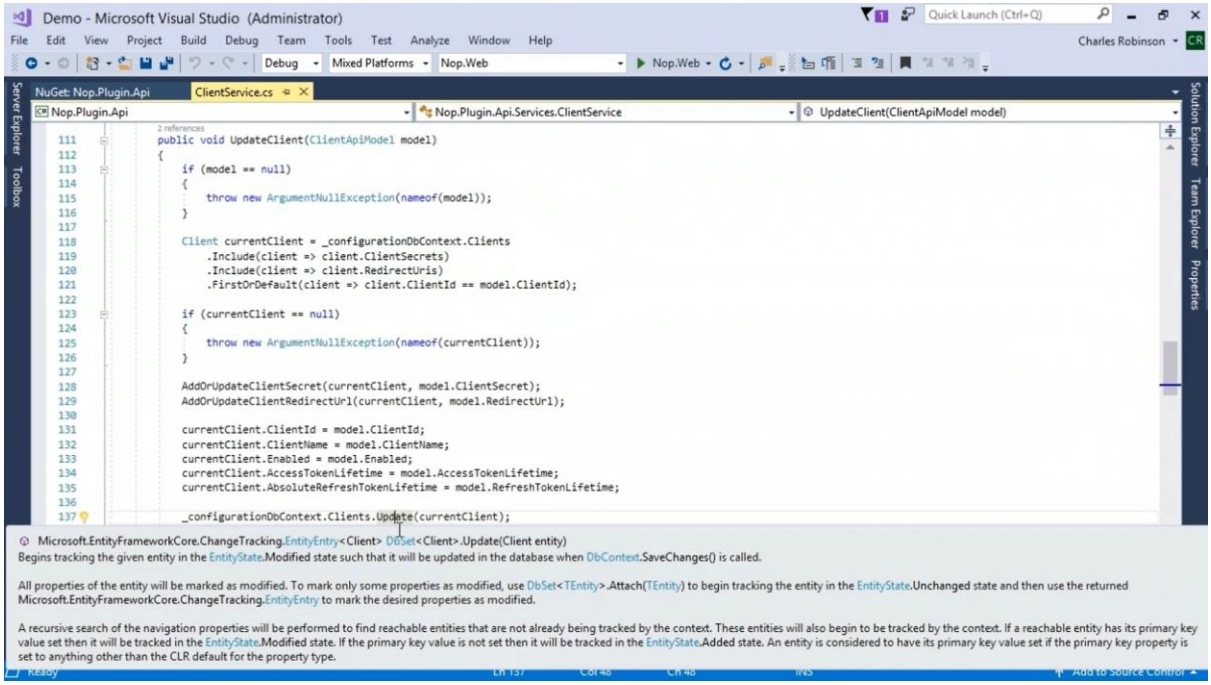

ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Debug
Any CPU
ConsoleApp1
Program.cs
ConsoleApp1
Add New Item - ConsoleApp1
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ASP.NET Core
SQL Server
Storm Items
Test
Online
Sort by: Default
Search (Ctrl+E)
DataSet
Service-based Database
XML File
XML Schema
XSLT File
Visual C# Items
Type: Visual C# Items
An empty SQL Server database for service-based data access
Name: Database1.mdf
Add
Cancel
100 %
2:44 / 3:45

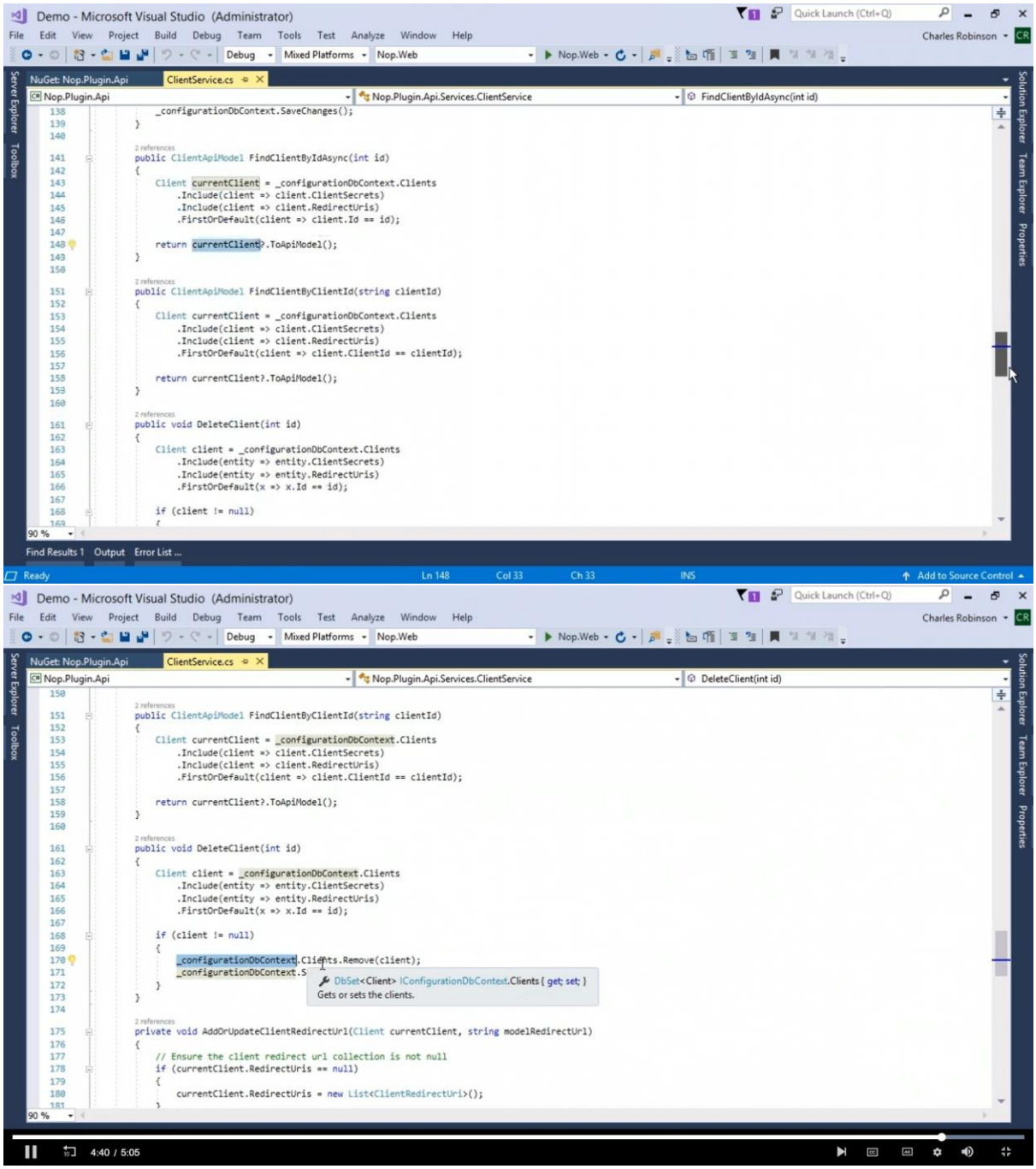

ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Debug
Any CPU
ConsoleApp1
Server Explorer
Azure (stefan.samms@outlook.com - 1 subscriptions)
App Service
Cloud Services
Data Lake Analytics
HDInsight
Notification Hubs
SQL Databases
Storage
Stream Analytics
Virtual Machines
Data Connections
pc0452\sqlexpress.Northwind.dbo
Tables
Categories
CustomerCustomerDemo
CustomerDemographics
Customers
Employees
EmployeeTerritories
Order Details
Orders
Products
Region
Shippers
Suppliers
Territories
Views
Stored Procedures
Functions
ConsoleApp1.Program
Main(string[] args)
3:18 / 3:45


Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api
ClientService.cs
Browse
Installed
Updates 8
NuGet Package Manager: Nop.Plugin.Api
Search (Ctrl+E)
Include prerelease
Package source: nuget.org
Microsoft.AspNet.WebHooks.Custom.SqlStorage by Microsoft
This package provides support for persisting your custom WebHooks registrations in a SQL database.
v1.2.1
Microsoft.AspNetCore.Authentication.JwtBearer by Microsoft
ASP.NET Core middleware that enables an application to receive an OpenID Connect bearer token.
v2.0.0
v2.0.3
Microsoft.AspNetCore.Rewrite by Microsoft
ASP.NET Core basic middleware for rewriting URLs. Includes:
* Support for custom URL rewrite rules
v2.0.0
v2.0.2
Microsoft.EntityFrameworkCore by Microsoft
Entity Framework Core is a lightweight and extensible version of the popular Entity Framework data access technology.
v2.0.1
v2.0.2
Microsoft.EntityFrameworkCore.SqlServer by Microsoft
Microsoft SQL Server database provider for Entity Framework Core.
v2.0.1
v2.0.2
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again
Microsoft.EntityFrameworkCore
Installed: 2.0.1
Uninstall
Version: Latest stable 2.0.2
Update
Options
Description
Entity Framework Core is a lightweight and extensible version of the popular Entity Framework data access technology.
Commonly Used Types:
Microsoft.EntityFrameworkCore.DbContext
Microsoft.EntityFrameworkCore.DbSet
Version: 2.0.2
Author(s): Microsoft
License: https://raw.githubusercontent.com/aspnet/Home/2.0.0/LICENSE.txt
Date published: Tuesday, March 13, 2018 (3/13/2018)
Project URL: https://asp.net/
Find Results 1 Output Error List ...
Ready
Add to Source Control

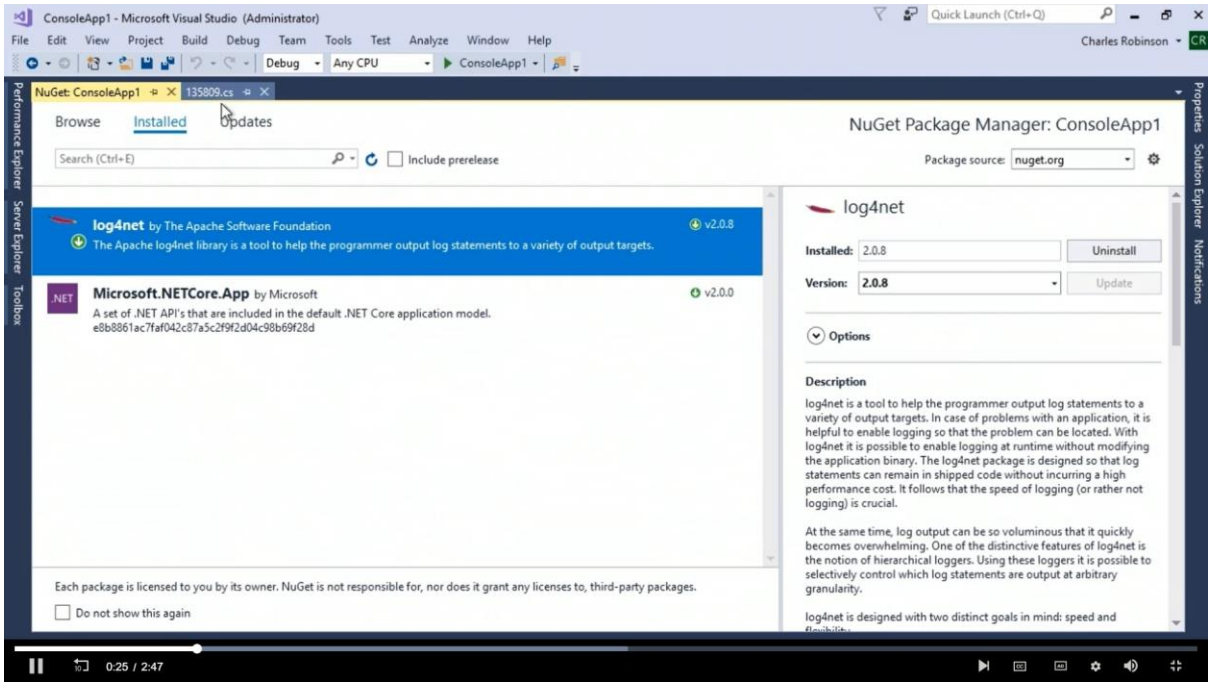

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api
ClientService.cs
Nop.Plugin.Api
Nop.Plugin.Api.Services.ClientService
InsertClient(ClientApiModel model)
client.AllowedScopes = new List<ClientScope>()
{
new ClientScope()
{
Client = client,
Scope = "nop_api"
}
};
client.Claims = new List<ClientClaim>()
{
new ClientClaim()
{
Client = client,
Type = JwtClaimTypes.Subject,
Value = client.ClientId
},
new ClientClaim()
{
Client = client,
Type = JwtClaimTypes.Name,
Value = client.ClientName
}
};
_configurationDbContext.Clients.Add(client);
_configurationDbContext.SaveChanges();
return client.Id;
}
2 references
90 %
Find Results 1 Output Error List ...
Ready
Ln 105 Col 44 Ch 44 INS
Add to Source Control

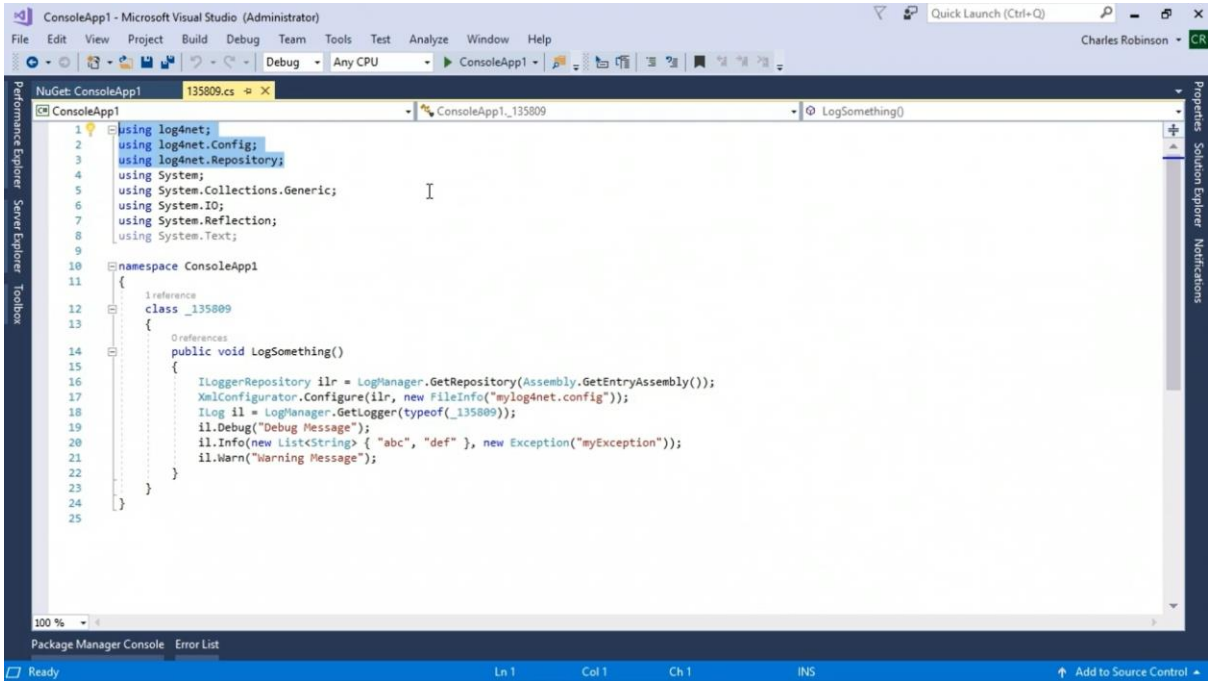

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api
ClientService.cs
Nop.Plugin.Api
Nop.Plugin.Api.Services.ClientService
UpdateClient(ClientApiModel model)
2 references
public void UpdateClient(ClientApiModel model)
{
if (model == null)
{
throw new ArgumentNullException(nameof(model));
}
Client currentClient = _configurationDbContext.Clients
.Include(client => client.ClientSecrets)
.Include(client => client.RedirectUris)
.FirstOrDefault(client => client.ClientId == model.ClientId);
if (currentClient == null)
{
throw new ArgumentNullException(nameof(currentClient));
}
AddOrUpdateClientSecret(currentClient, model.ClientSecret);
AddOrUpdateClientRedirectUrl(currentClient, model.RedirectUrl);
currentClient.ClientId = model.ClientId;
currentClient.ClientName = model.ClientName;
currentClient.Enabled = model.Enabled;
currentClient.AccessTokenLifetime = model.AccessTokenLifetime;
currentClient.AbsoluteRefreshTokenLifetime = model.RefreshTokenLifetime;
_configurationDbContext.Clients.Update(currentClient);
_configurationDbContext.SaveChanges();
}
2 references
public ClientApiModel FindClientByIdAsync(int id)
{
90 %
3:38 / 5:05

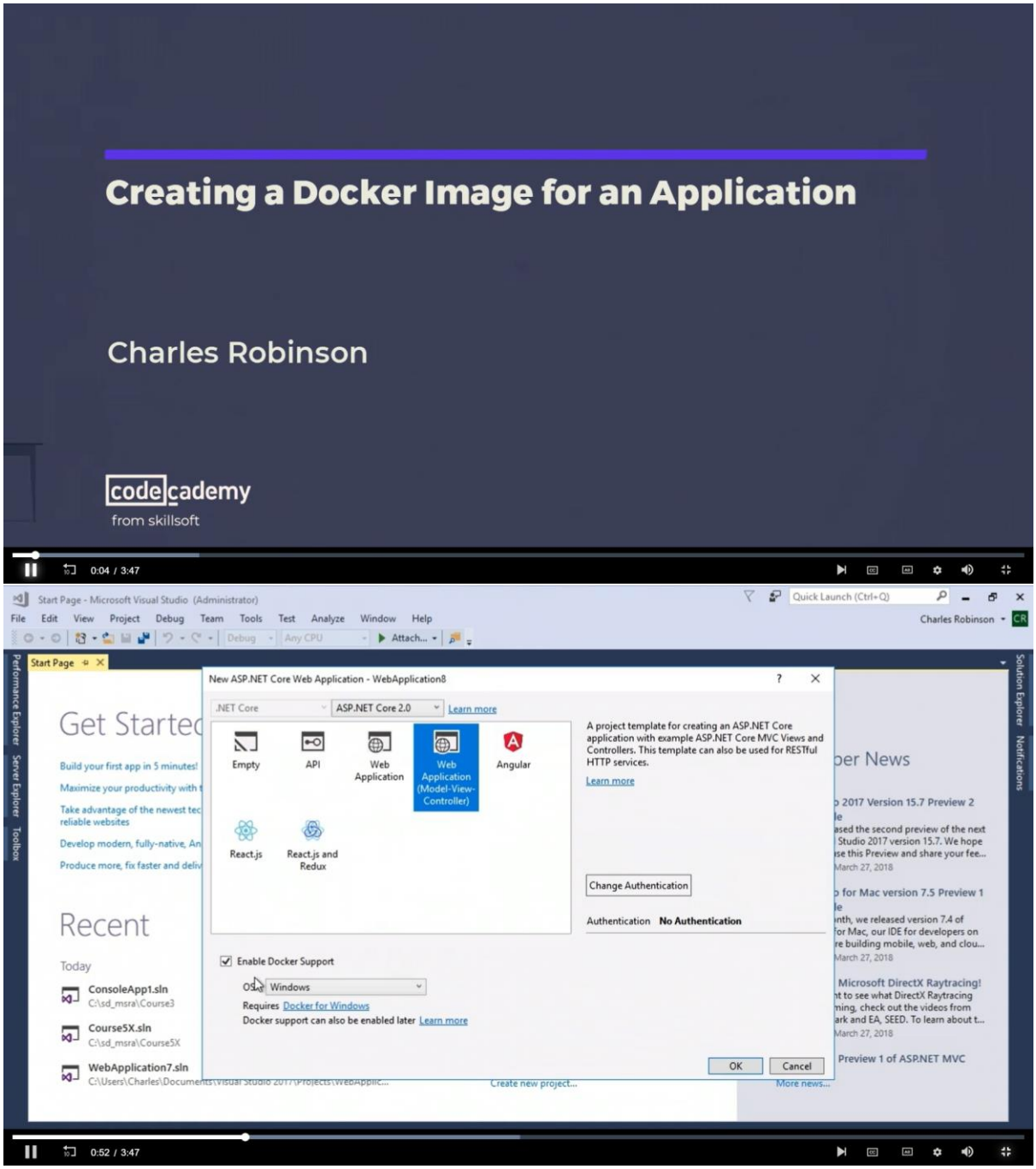

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api
ClientService.cs
Nop.Plugin.Api
Nop.Plugin.Api.Services.ClientService
UpdateClient(ClientApiModel model)
2 references
public void UpdateClient(ClientApiModel model)
{
if (model == null)
{
throw new ArgumentNullException(nameof(model));
}
Client currentClient = _configurationDbContext.Clients
.Include(client => client.ClientSecrets)
.Include(client => client.RedirectUris)
.FirstOrDefault(client => client.ClientId == model.ClientId);
if (currentClient == null)
{
throw new ArgumentNullException(nameof(currentClient));
}
AddOrUpdateClientSecret(currentClient, model.ClientSecret);
AddOrUpdateClientRedirectUrl(currentClient, model.RedirectUrl);
currentClient.ClientId = model.ClientId;
currentClient.ClientName = model.ClientName;
currentClient.Enabled = model.Enabled;
currentClient.AccessTokenLifetime = model.AccessTokenLifetime;
currentClient.AbsoluteRefreshTokenLifetime = model.RefreshTokenLifetime;
_configurationDbContext.Clients.Update(currentClient);
Microsoft.EntityFrameworkCore.ChangeTracking.EntityEntry<Client> DbSet<Client>.Update(Client entity)
Begins tracking the given entity in the EntityState.Modified state such that it will be updated in the database when DbContext.SaveChanges() is called.
All properties of the entity will be marked as modified. To mark only some properties as modified, use DbSet<TEntity>.Attach(TEntity) to begin tracking the entity in the EntityState.Unchanged state and then use the returned Microsoft.EntityFrameworkCore.ChangeTracking.EntityEntry to mark the desired properties as modified.
A recursive search of the navigation properties will be performed to find reachable entities that are not already being tracked by the context. These entities will also begin to be tracked by the context. If a reachable entity has its primary key value set then it will be tracked in the EntityState.Modified state. If the primary key value is not set then it will be tracked in the EntityState.Added state. An entity is considered to have its primary key value set if the primary key property is set to anything other than the CLR default for the property type.
Ready
Ln 137
Col 48
Ch 48
INS
Add to Source Control

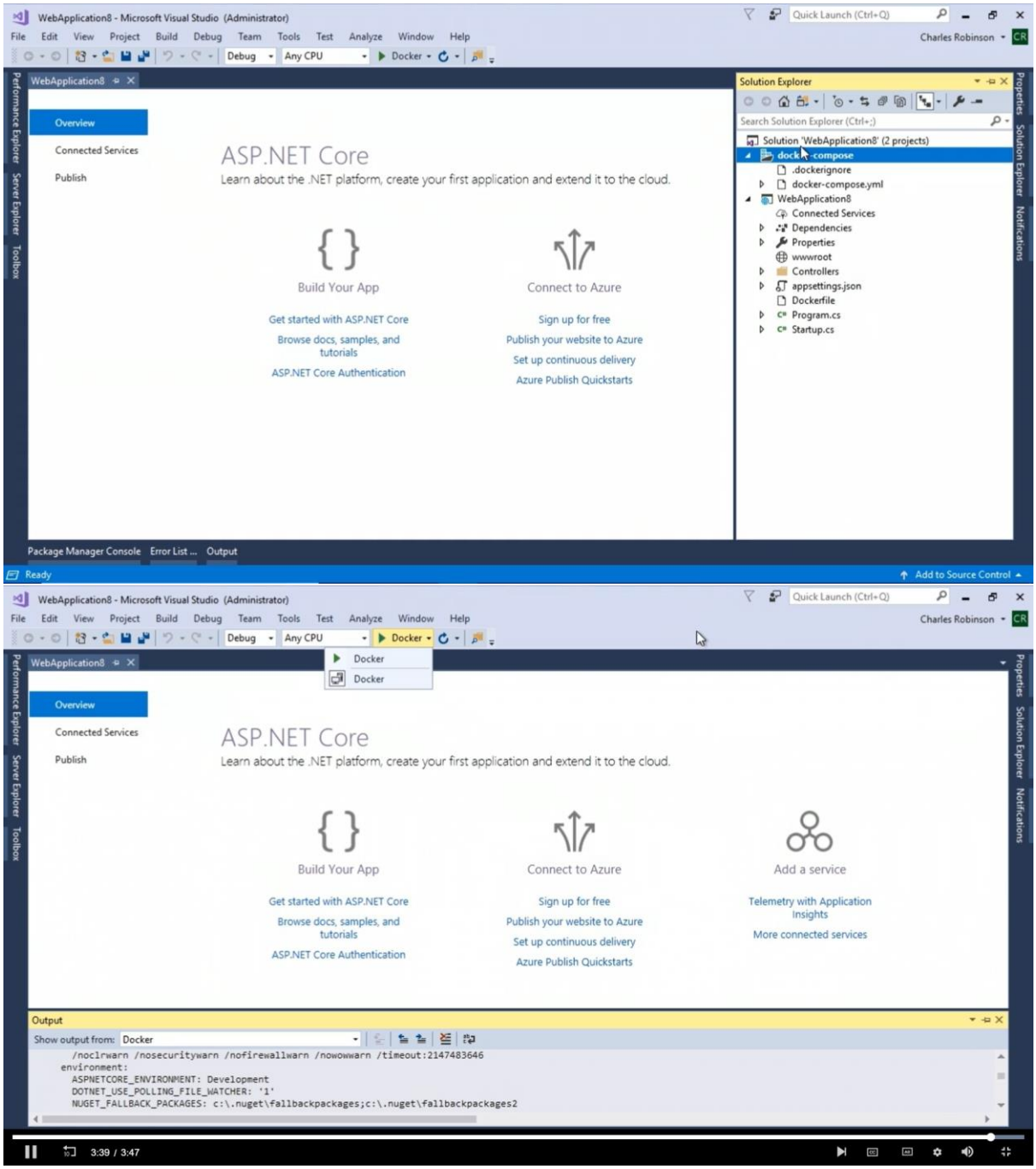

Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api
ClientService.cs
Nop.Plugin.Api
Nop.Plugin.Api.Services.ClientService
FindClientByIdAsync(int id)
_configurationDbContext.SaveChanges();
}
2 references
public ClientApiModel FindClientByIdAsync(int id)
{
Client currentClient = _configurationDbContext.Clients
.Include(client => client.ClientSecrets)
.Include(client => client.RedirectUris)
.FirstOrDefault(client => client.Id == id);
return currentClient?.ToApiModel();
}
2 references
public ClientApiModel FindClientByClientId(string clientId)
{
Client currentClient = _configurationDbContext.Clients
.Include(client => client.ClientSecrets)
.Include(client => client.RedirectUris)
.FirstOrDefault(client => client.ClientId == clientId);
return currentClient?.ToApiModel();
}
2 references
public void DeleteClient(int id)
{
Client client = _configurationDbContext.Clients
.Include(entity => entity.ClientSecrets)
.Include(entity => entity.RedirectUris)
.FirstOrDefault(x => x.Id == id);
if (client != null)
{
90 %
Find Results 1 Output Error List ...
Ready
Ln 148 Col 33 Ch 33 INS
Add to Source Control
Demo - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Mixed Platforms
Nop.Web
NuGet: Nop.Plugin.Api
ClientService.cs
Nop.Plugin.Api
Nop.Plugin.Api.Services.ClientService
DeleteClient(int id)
2 references
public ClientApiModel FindClientByClientId(string clientId)
{
Client currentClient = _configurationDbContext.Clients
.Include(client => client.ClientSecrets)
.Include(client => client.RedirectUris)
.FirstOrDefault(client => client.ClientId == clientId);
return currentClient?.ToApiModel();
}
2 references
public void DeleteClient(int id)
{
Client client = _configurationDbContext.Clients
.Include(entity => entity.ClientSecrets)
.Include(entity => entity.RedirectUris)
.FirstOrDefault(x => x.Id == id);
if (client != null)
{
_configurationDbContext.Clients.Remove(client);
_configurationDbContext.S
}
}
DbSet<Client> IConfigurationDbContext.Clients { get; set; }
Gets or sets the clients.
2 references
private void AddOrUpdateClientRedirectUrl(Client currentClient, string modelRedirectUrl)
{
// Ensure the client redirect url collection is not null
if (currentClient.RedirectUris == null)
{
currentClient.RedirectUris = new List<ClientRedirectUri>();
90 %
4:40 / 5:05

Logging

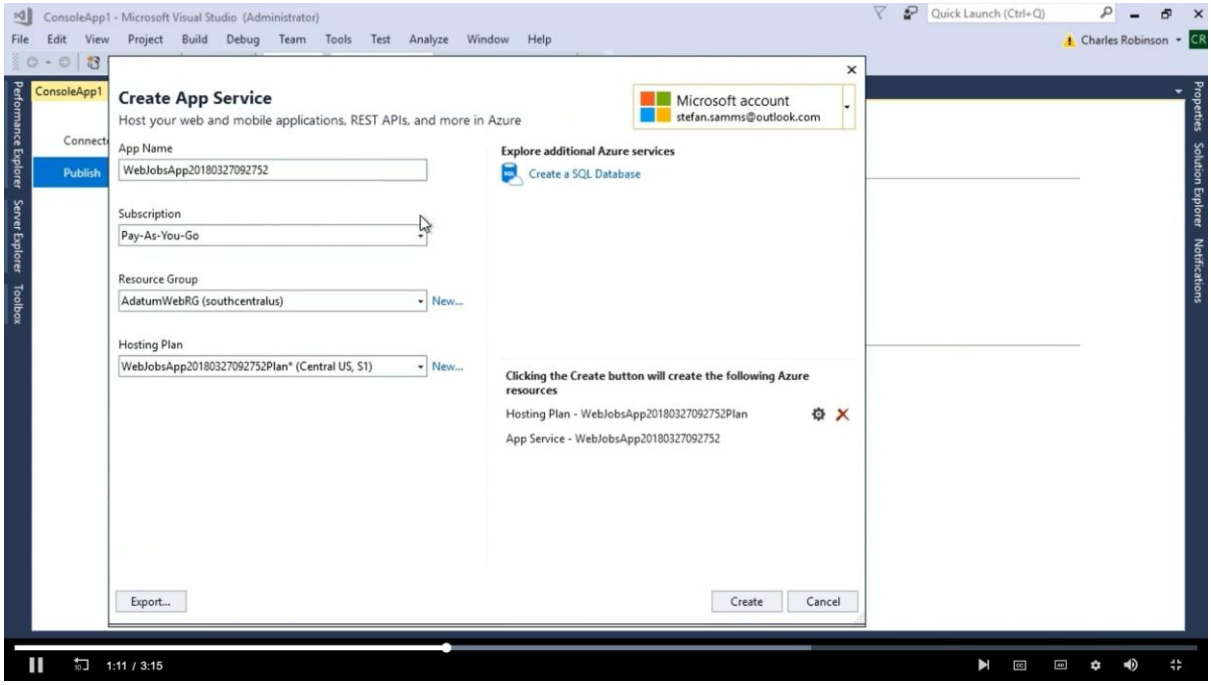

ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Any CPU
ConsoleApp1
NuGet: ConsoleApp1
135809.cs
Browse
Installed
Updates
NuGet Package Manager: ConsoleApp1
Search (Ctrl+E)
Include prerelease
Package source: nuget.org
log4net by The Apache Software Foundation
v2.0.8
The Apache log4net library is a tool to help the programmer output log statements to a variety of output targets.
Microsoft.NETCore.App by Microsoft
v2.0.0
A set of .NET API's that are included in the default .NET Core application model.
e8b8861ac7faf042c87a5c2f9f2d04c98b69f28d
log4net
Installed: 2.0.8
Uninstall
Version: 2.0.8
Update
Options
Description
log4net is a tool to help the programmer output log statements to a variety of output targets. In case of problems with an application, it is helpful to enable logging so that the problem can be located. With log4net it is possible to enable logging at runtime without modifying the application binary. The log4net package is designed so that log statements can remain in shipped code without incurring a high performance cost. It follows that the speed of logging (or rather not logging) is crucial.
At the same time, log output can be so voluminous that it quickly becomes overwhelming. One of the distinctive features of log4net is the notion of hierarchical loggers. Using these loggers it is possible to selectively control which log statements are output at arbitrary granularity.
log4net is designed with two distinct goals in mind: speed and
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again
0:25 / 2:47

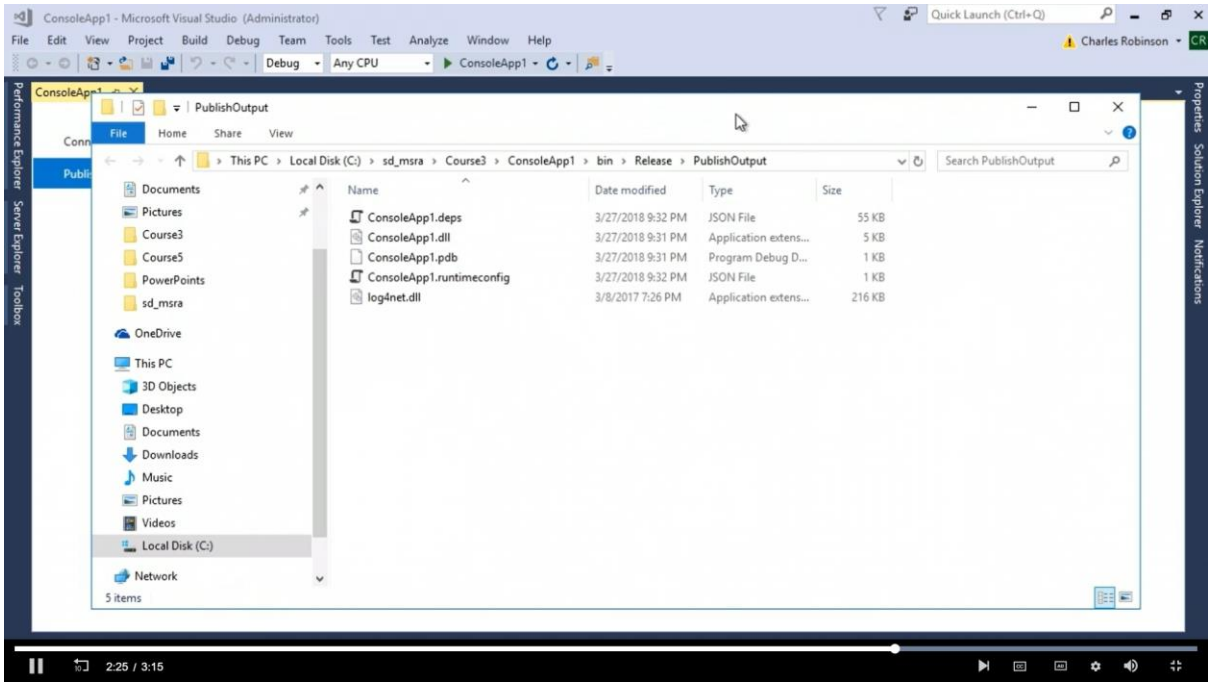

ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Charles Robinson
Debug
Any CPU
ConsoleApp1
NuGet: ConsoleApp1
135809.cs
ConsoleApp1
ConsoleApp1._135809
LogSomething()
using log4net;
using log4net.Config;
using log4net.Repository;
using System;
using System.Collections.Generic;
using System.IO;
using System.Reflection;
using System.Text;
namespace ConsoleApp1
{
1 reference
class _135809
{
0 references
public void LogSomething()
{
ILoggerRepository ilr = LogManager.GetRepository(Assembly.GetEntryAssembly());
XmlConfigurator.Configure(ilr, new FileInfo("mylog4net.config"));
ILog il = LogManager.GetLogger(typeof(_135809));
il.Debug("Debug Message");
il.Info(new List<String> { "abc", "def" }, new Exception("myException"));
il.Warn("Warning Message");
}
}
}
100 %
Package Manager Console
Error List
Ready
Ln 1
Col 1
Ch 1
INS
Add to Source Control


Creating a Docker Image for an Application
Charles Robinson
codecademy
from skillsoft
0:04 / 3:47
Start Page - Microsoft Visual Studio (Administrator)
File Edit View Project Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Start Page
New ASP.NET Core Web Application - WebApplication8
.NET Core
ASP.NET Core 2.0
Learn more
Empty
API
Web Application
Web Application (Model-View-Controller)
Angular
React.js
React.js and Redux
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
Learn more
Change Authentication
Authentication No Authentication
Enable Docker Support
OS Windows
Requires Docker for Windows
Docker support can also be enabled later Learn more
OK
Cancel
Get Started
Build your first app in 5 minutes!
Recent
Today
ConsoleApp1.sln
C:\sd_msra\Course3
Course5X.sln
C:\sd_msra\Course5X
WebApplication7.sln
Create new project...
More news...
Performance Explorer Server Explorer Toolbox
Solution Explorer Notifications
0:52 / 3:47


WebApplication8 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Debug
Any CPU
Docker
WebApplication8
Overview
Connected Services
Publish
ASP.NET Core
Learn about the .NET platform, create your first application and extend it to the cloud.
Build Your App
Get started with ASP.NET Core
Browse docs, samples, and tutorials
ASP.NET Core Authentication
Connect to Azure
Sign up for free
Publish your website to Azure
Set up continuous delivery
Azure Publish Quickstarts
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'WebApplication8' (2 projects)
docker-compose
.dockerignore
docker-compose.yml
WebApplication8
Connected Services
Dependencies
Properties
wwwroot
Controllers
appsettings.json
Dockerfile
Program.cs
Startup.cs
Package Manager Console
Error List ...
Output
Ready
Add to Source Control
Docker
Docker
Add a service
Telemetry with Application Insights
More connected services
Output
Show output from: Docker
/noclrwarn /nosecuritywarn /nofirewallwarn /nowowwarn /timeout:2147483646
environment:
ASPNETCORE_ENVIRONMENT: Development
DOTNET_USE_POLLING_FILE_WATCHER: '1'
NUGET_FALLBACK_PACKAGES: c:\.nuget\fallbackpackages;c:\.nuget\fallbackpackages2
3:39 / 3:47

Deploying a Microservice to a Server
Charles Robinson
codecademy
from skillsoft
0:03 / 3:15


ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Create App Service
Host your web and mobile applications, REST APIs, and more in Azure
Microsoft account
stefan.samms@outlook.com
App Name
WebJobsApp20180327092752
Subscription
Pay-As-You-Go
Resource Group
AdatumWebRG (southcentralus)
New...
Hosting Plan
WebJobsApp20180327092752Plan* (Central US, S1)
New...
Explore additional Azure services
Create a SQL Database
Clicking the Create button will create the following Azure resources
Hosting Plan - WebJobsApp20180327092752Plan
App Service - WebJobsApp20180327092752
Export...
Create
Cancel
1:11 / 3:15


ConsoleApp1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Charles Robinson
Debug
Any CPU
ConsoleApp1
PublishOutput
File Home Share View
This PC › Local Disk (C:) › sd_msra › Course3 › ConsoleApp1 › bin › Release › PublishOutput
Search PublishOutput
Documents
Pictures
Course3
Course5
PowerPoints
sd_msra
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Network
5 items
Name
Date modified
Type
Size
ConsoleApp1.deps 3/27/2018 9:32 PM JSON File 55 KB
ConsoleApp1.dll 3/27/2018 9:31 PM Application extens... 5 KB
ConsoleApp1.pdb 3/27/2018 9:31 PM Program Debug D... 1 KB
ConsoleApp1.runtimeconfig 3/27/2018 9:32 PM JSON File 1 KB
log4net.dll 3/8/2017 7:26 PM Application extens... 216 KB
Performance Explorer
Server Explorer
Toolbox
Properties
Solution Explorer
Notifications
2:25 / 3:15