Contents

[Version 2](#_Toc520106301)

[NopCommerce 3](#_Toc520106302)

[Nop.Plugin.Api 4](#_Toc520106303)

[Installation 4](#_Toc520106304)

[Install API Sample Application 8](#_Toc520106305)

[Create Client API Access 9](#_Toc520106306)

[Call API with Postman 12](#_Toc520106307)

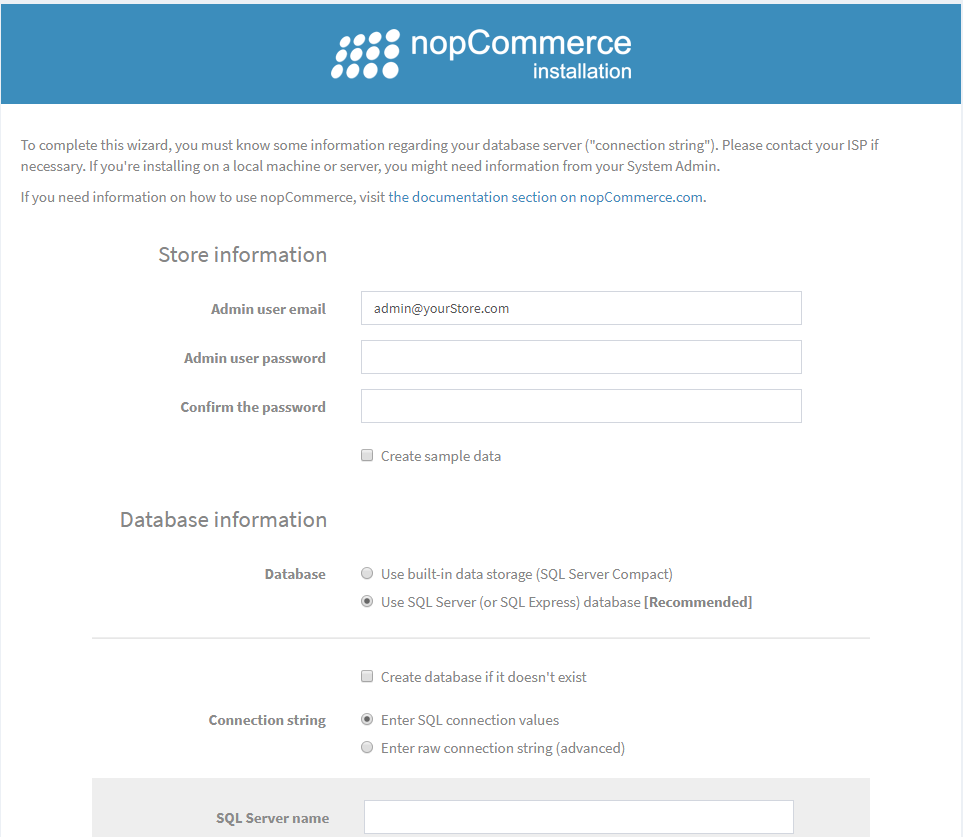
[Create New Entity (Table) 16](#_Toc520106308)

# Version

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Remark |
| 23/07/2018 | Eric Tan | V1.0 | Initial |
|  |  |  |  |

# NopCommerce

1. Download NopCommerce 4.0 from <https://www.nopcommerce.com>
2. Extract Nopcommerce file and open the solution.
3. Open Visual Studio 2017 and run as “Administrator” and open “NopCommerce”solution.
4. Set “Nop.Web” as start-up project and run.

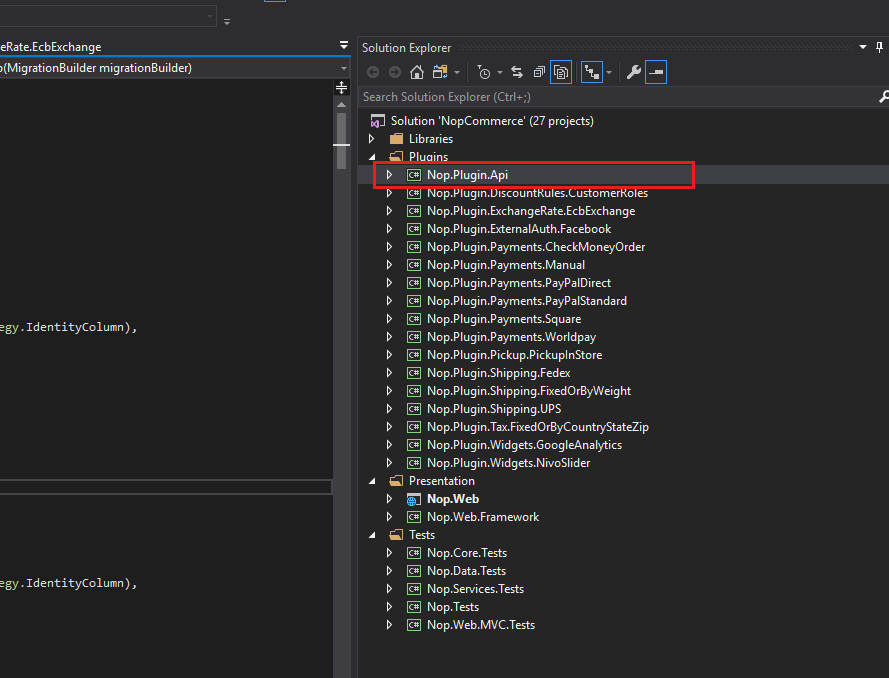


Fill up nopcommerce admin user password and check “Use SQL Server”, “Create database if doesn’t exist”, enter connection string info and click “Install”

# Nop.Plugin.Api

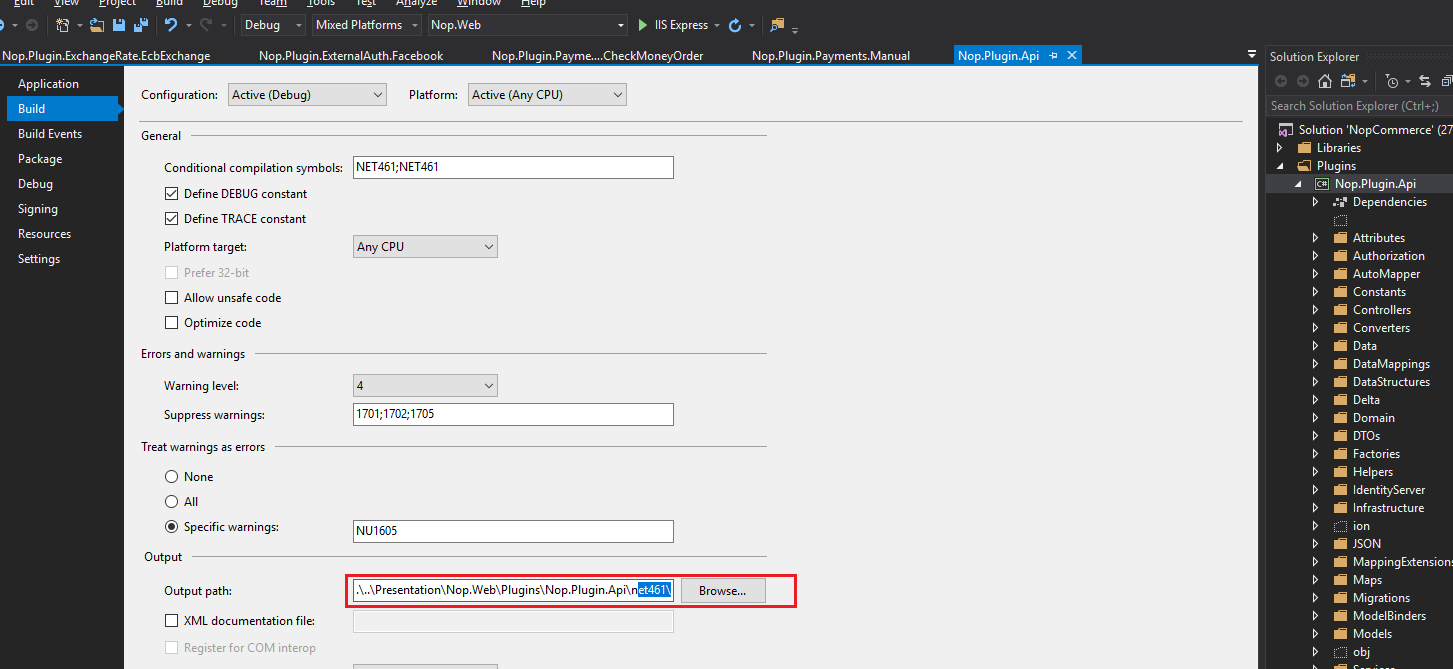
## Installation

1. Download “**Nop.Plugin.Api**” from https://github.com/SevenSpikes/api-plugin-for-nopcommerce
2. Extract zip file **Nop.Plugin.Api\_4.00.zip** and copy “**Nop.Plugin.Api**” into the same folder as other “**Plugin**” projects in NopCommerce solution.
3. Add the “**Nop.Plugin.Api**” project in “**NopCommerce**” solution.

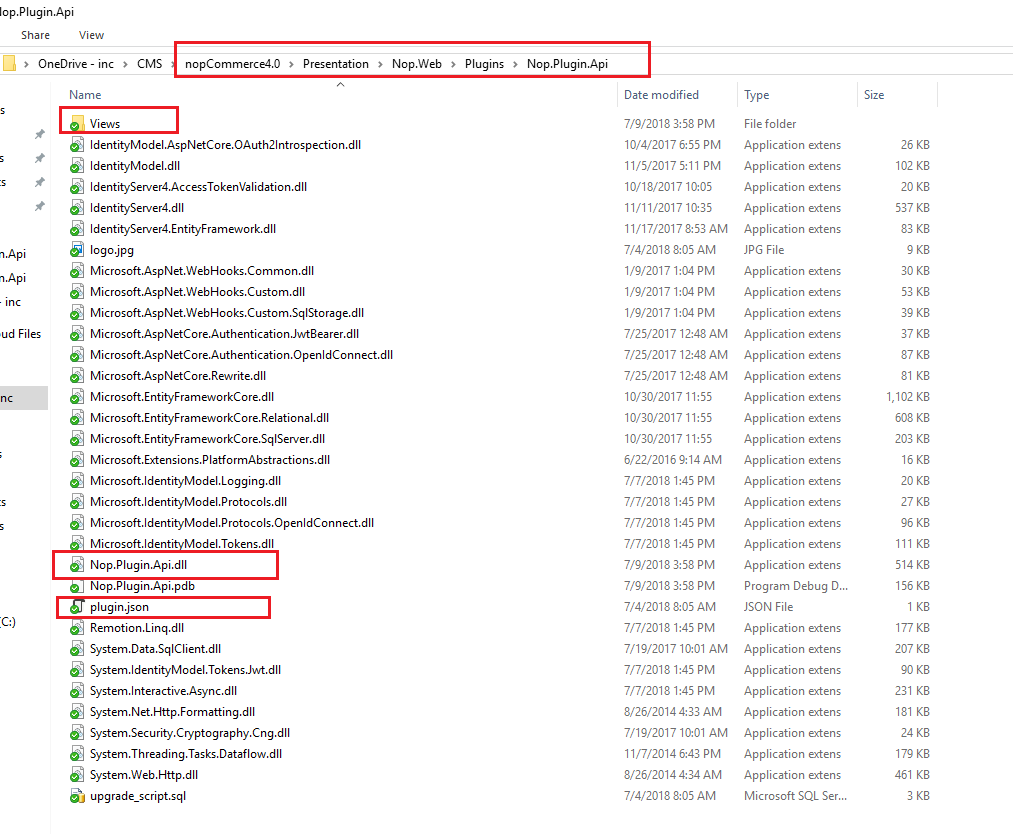


1. Add reference project:
2. Nop.Core
3. Nop.Service
4. Nop.Web.Framework
5. Ensure “**Nop.Plugin.Api**” build output to

“**nopCommerce4.0\Presentation\Nop.Web\Plugins\Nop.Plugin.Api**” folder

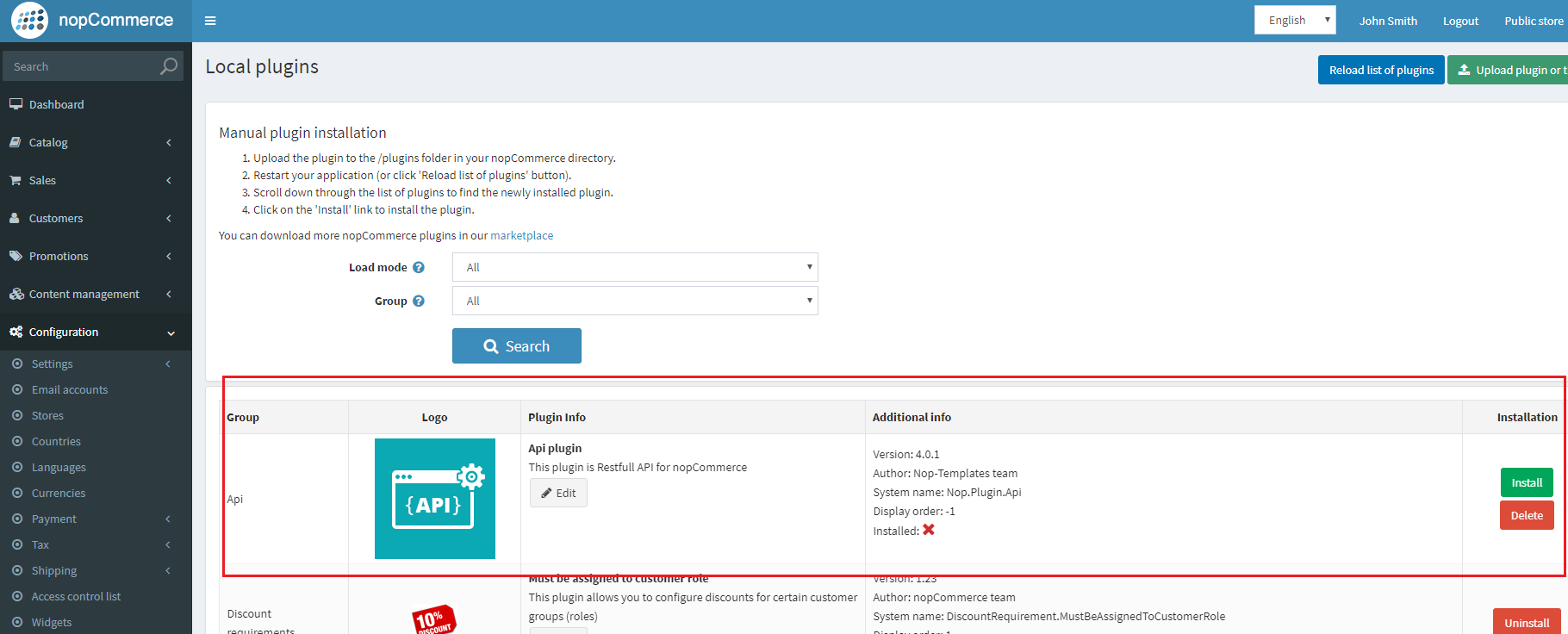


1. Rebuild “**Nop.Plugin.Api**” project.



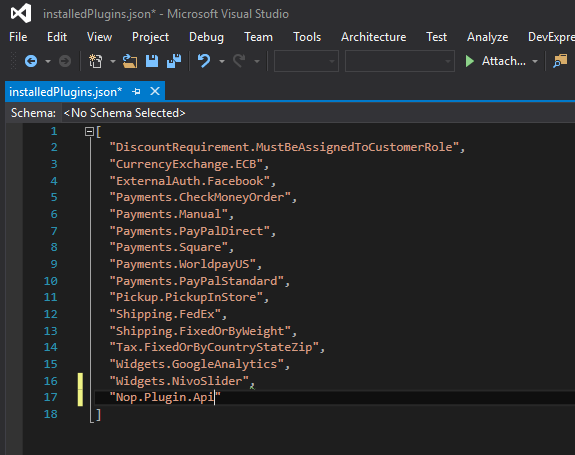
Ensure the files shown above are there after build

1. Run “**Nop.Web**”, login as administrator and go to admin site. Refer to this url <https://www.techadvisor.co.uk/how-to/windows/how-run-programs-as-administrator-in-windows-10-3632744/>. On how set to run Visual Studio 2017 as administrator by default.
2. Go to “**Configuration**” and “**Plugin**”

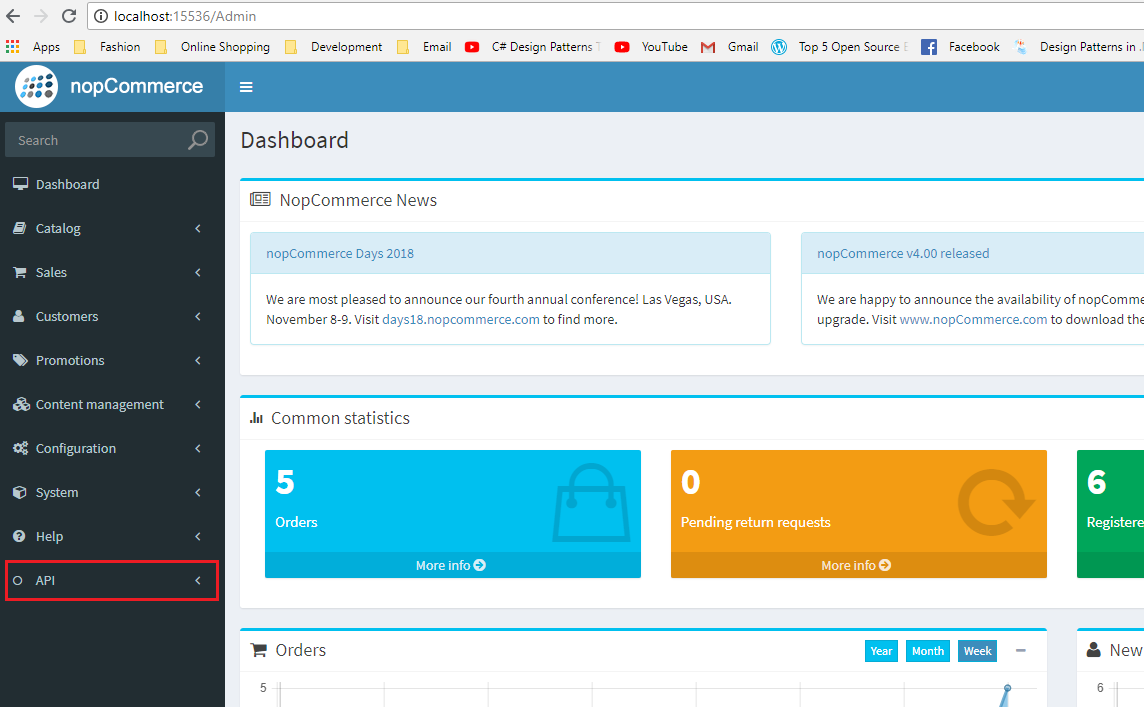


Click “**Install**”

1. After install “**Nop.Plugin.Api**” will be added in “**App\_Data\installedPlugins.json**” in “**Nop.Web**” project.

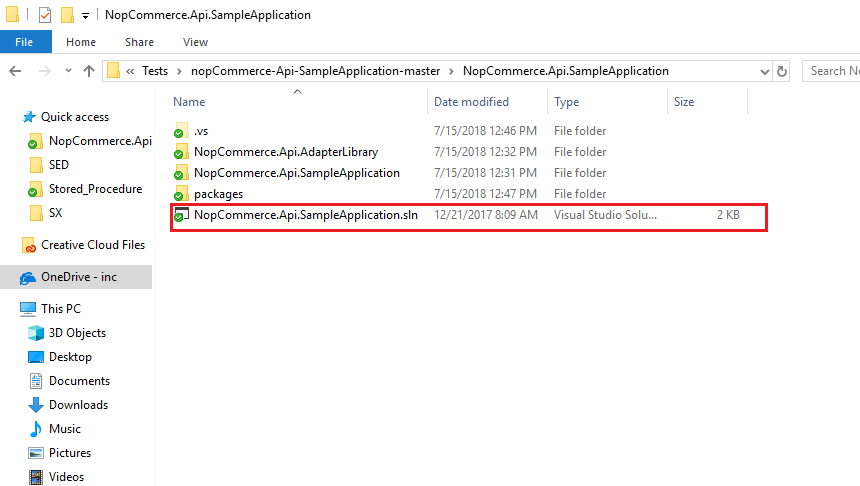


1. New “**API**” menu will appear in admin site

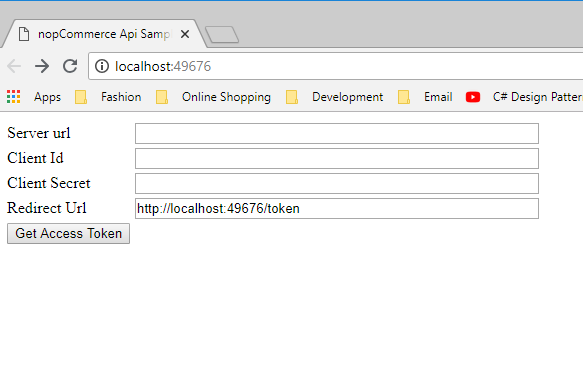


## Install API Sample Application

1. Extract zip file **nopCommerce-Api-SampleApplication-master.zip**
2. Open **Nop.Commerce.Api.SampleApplication.sln**

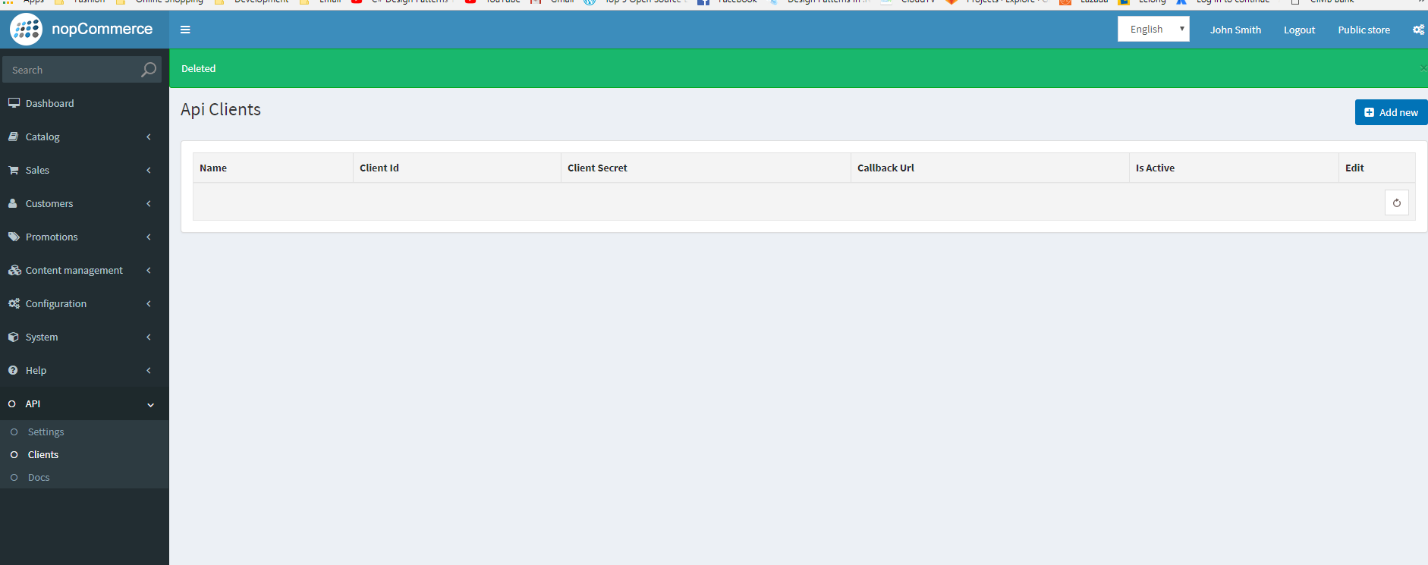


1. Run the project.

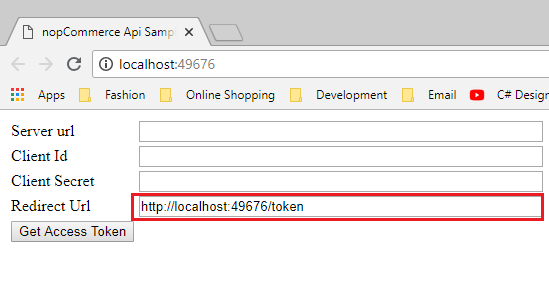


1. Run “**Nop.Web**” project and go to “**Administration”**, **“API”** and **“Clients”**

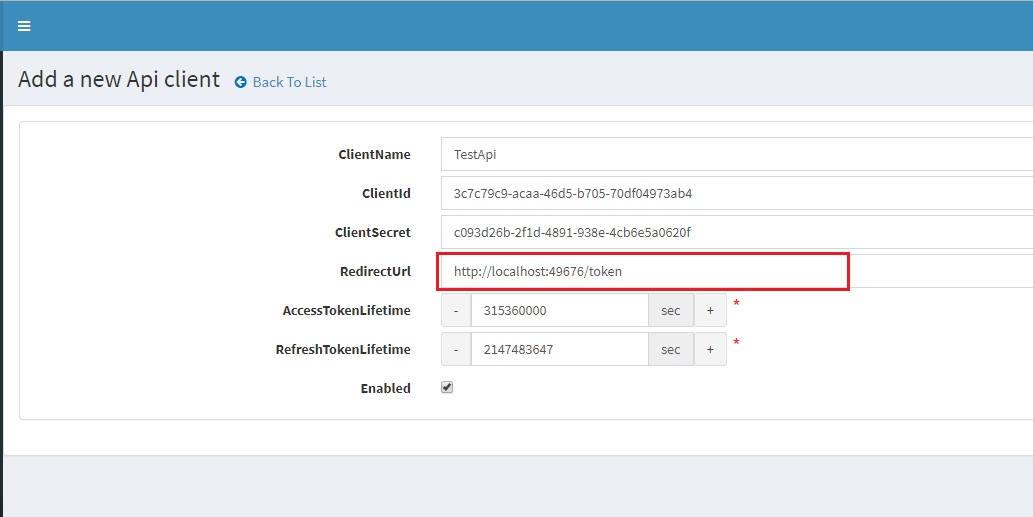
## Create Client API Access



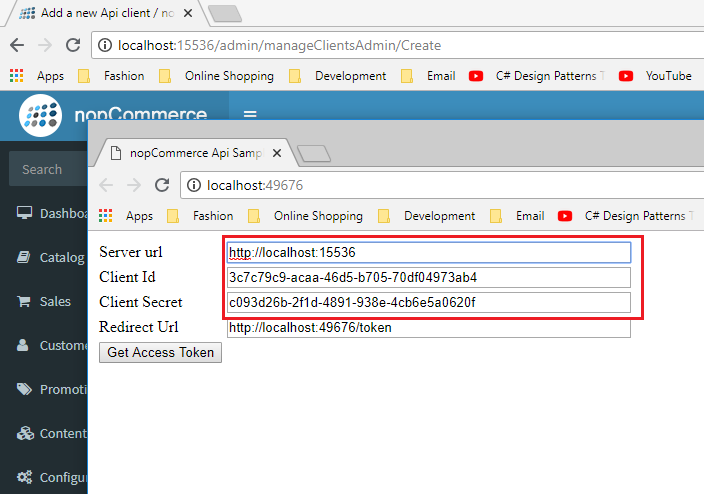
1. Click “**Add New**”
2. Go to previous running “**Nop.Commerce.Api.SampleApplication”** project web browser
3. Copy the redirect URL <http://localhost:49676/token>



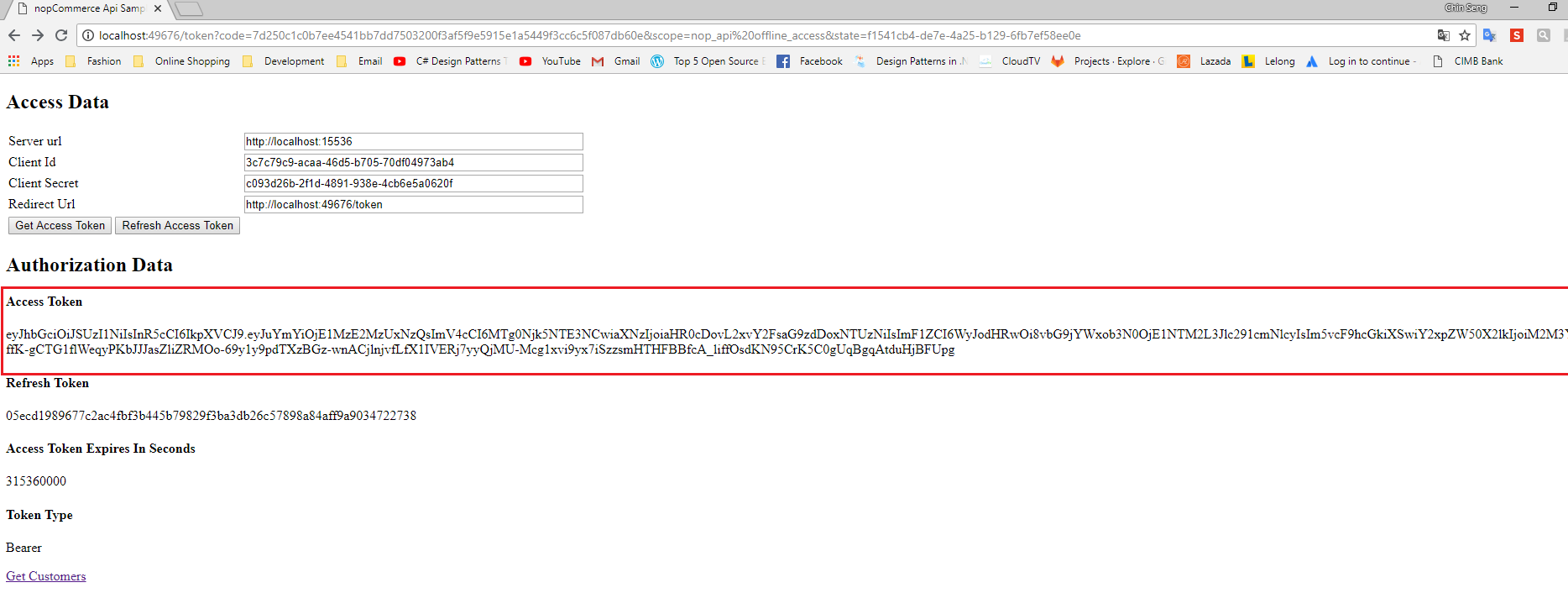
1. Paste it in “RedirectUrl”



1. Copy the **“ClientID”** and **“ClientSecret”** and paste it below and then click “**Save”**

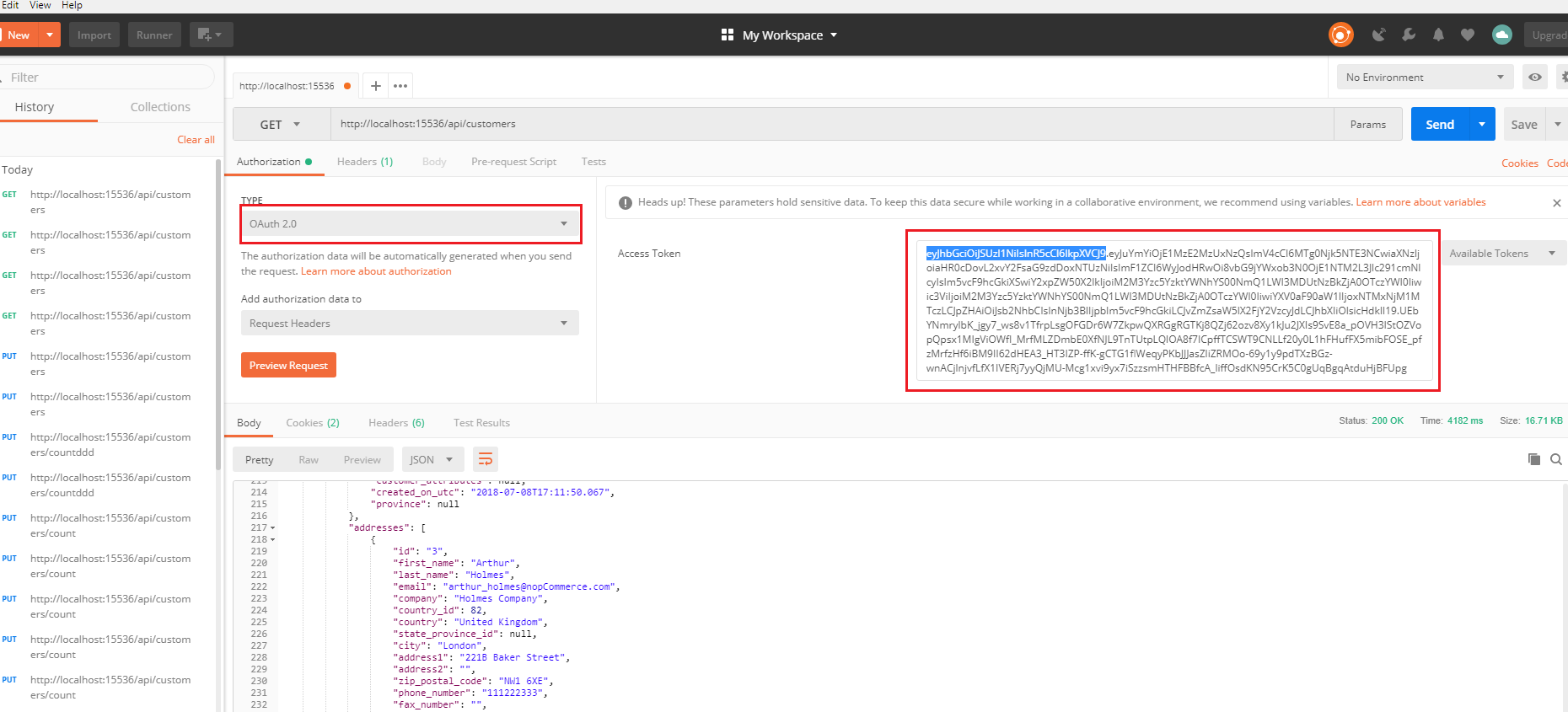


1. The “Server Url” – <http://localhost:15536> is the running “Nop.Web” url
2. Click “**Get Access Token**”

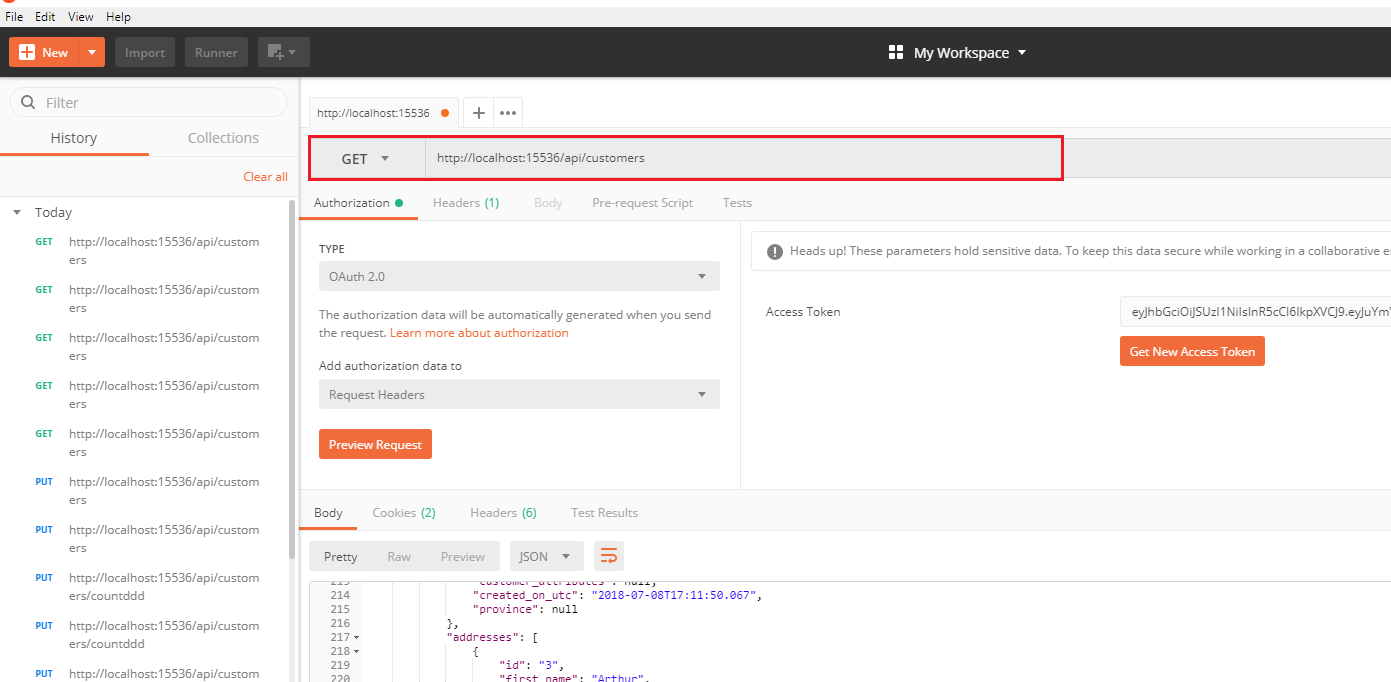


1. Copy the new generated Access token

## Call API with Postman



1. Select “**OAuth 2.0**” in authentication type above and paste the token.
2. You are not able to call API in “**Nop.Plugin.Api**”



|  |  |
| --- | --- |
| client\_id | 3c7c79c9-acaa-46d5-b705-70df04973ab4 |
| client\_secret | c093d26b-2f1d-4891-938e-4cb6e5a0620f |
| Code | 03b0e2801f4a34f3428957afb3803d332c8059e9a666a690b46ca4d3b77a1893 |
| grant\_type | authorization\_code |
|  |  |

<http://localhost:15536//oauth/authorize?client_id=3c7c79c9-acaa-46d5-b705-70df04973ab4&redirect_uri=http%3a%2f%2flocalhost%3a49676%2ftoken&response_type=code&state=2d17a2e9-4a11-4467-9f81-79da60618d07>

use POSTMan get method

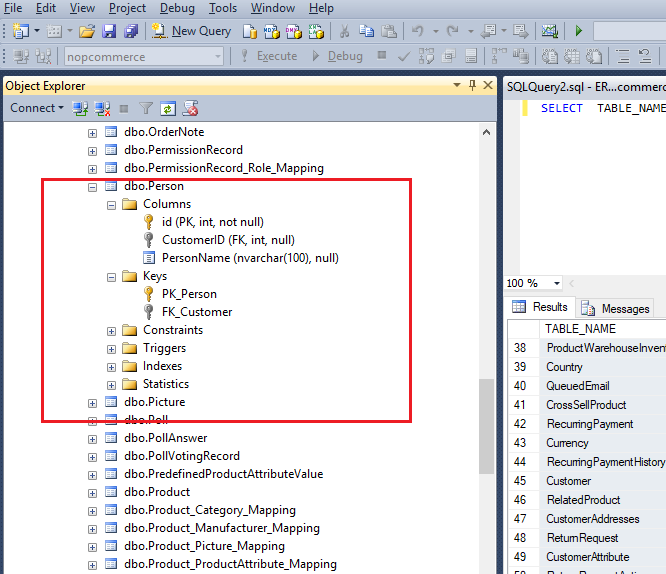
**NopCommerce and Plugin Tables**



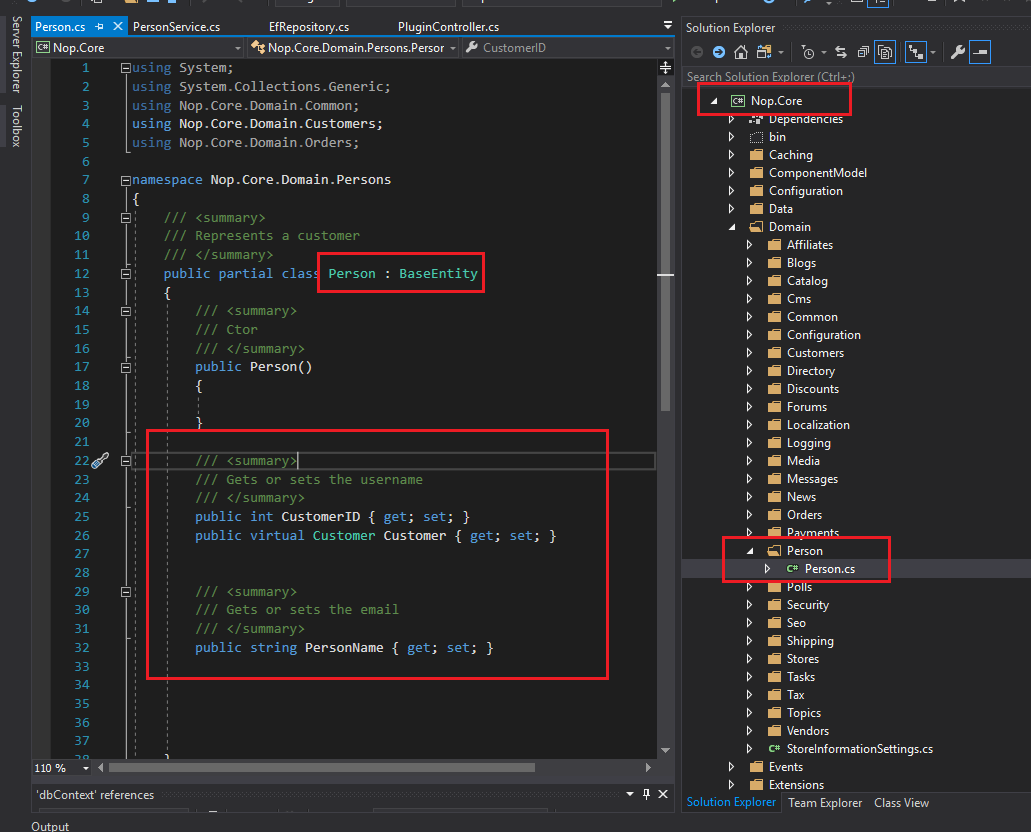
|  |  |  |
| --- | --- | --- |
| **Folder** |  |  |
| Attributes |  |  |
| Authorization |  |  |
| AutoMapper | AutoMapperApiConfiguration.cs | * Auto mapper class for Data Model mapping * Used in **ApiMapperConfiguration** |
| Constants | Configuration.cs | * Contains constant variables for API Settings |
| ViewNames.cs | * Contains API Admin view url |
| WebHookNames.cs |  |
| Controller |  |  |
| Converters | ApiTypeConverter.cs | * Implement **IApiTypeConverter** * Data Type Converter * Used in **ObjectConverter** |
| IApiTypeConverter.cs | * Interface Implemented by **ApiTypeConverter** |
| ObjectConverter.cs | * Implement **IObjectConverter** * Object Data Type that use ApiTypeConverter |
| IObjectConverter.cs |  |
| Data |  |  |
| DataMappings |  |  |
| DataStructures |  |  |
| Delta |  |  |
| Domain |  |  |
| DTOs |  |  |
| Factories |  |  |
| Infrastructure | DependencyRegister.cs |  |
|  |  |  |

# Create New Entity (Table)

1. Create a **Person** table with the following fields:
2. **Id** (int) – Primary key and identity
3. **CustomerID** (int) – Foreign key to Customer table
4. **PersonName** (nvarchar(100))

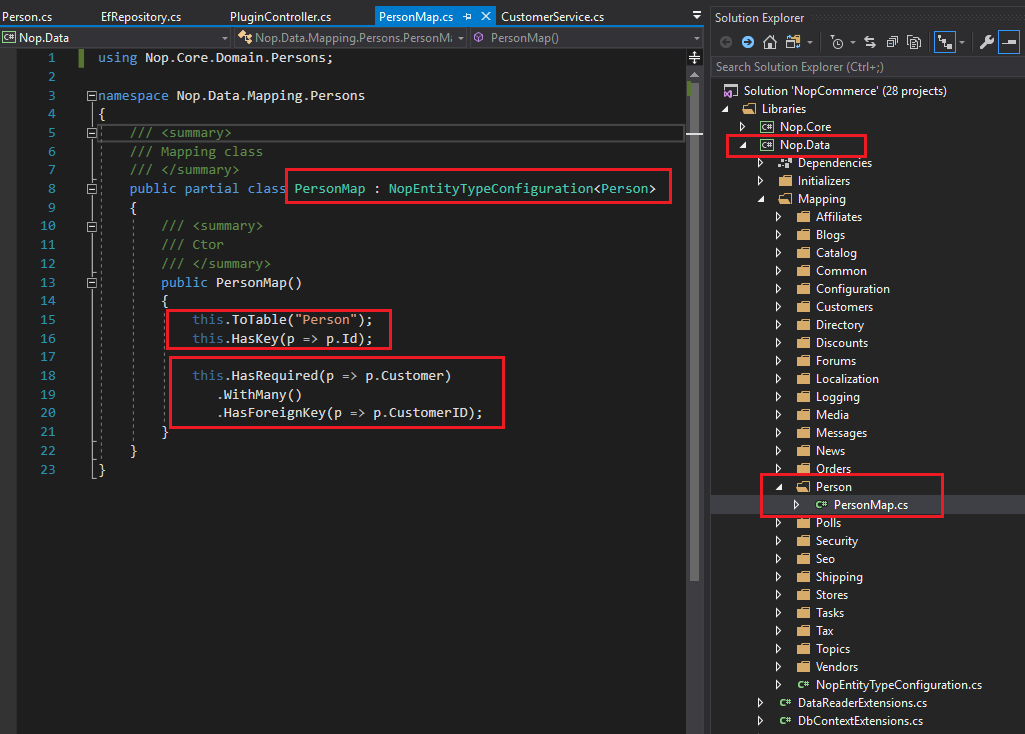


1. Nop.Core



* Create a class name “**Person.cs**” and inherit “**BaseEntity**” base class
* Create property “**CustomerID**”, “**Customer**” and “**PersonName**” accordingly in the table fields.

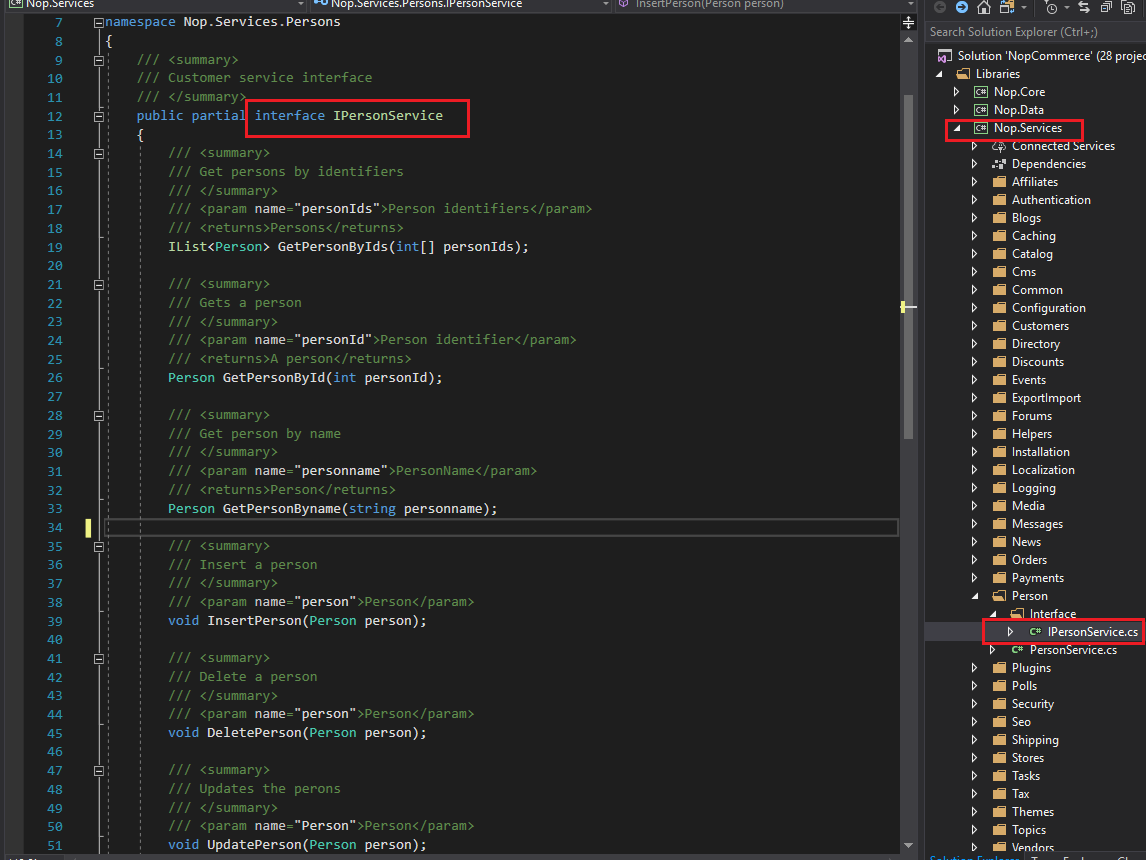
1. Nop.Data



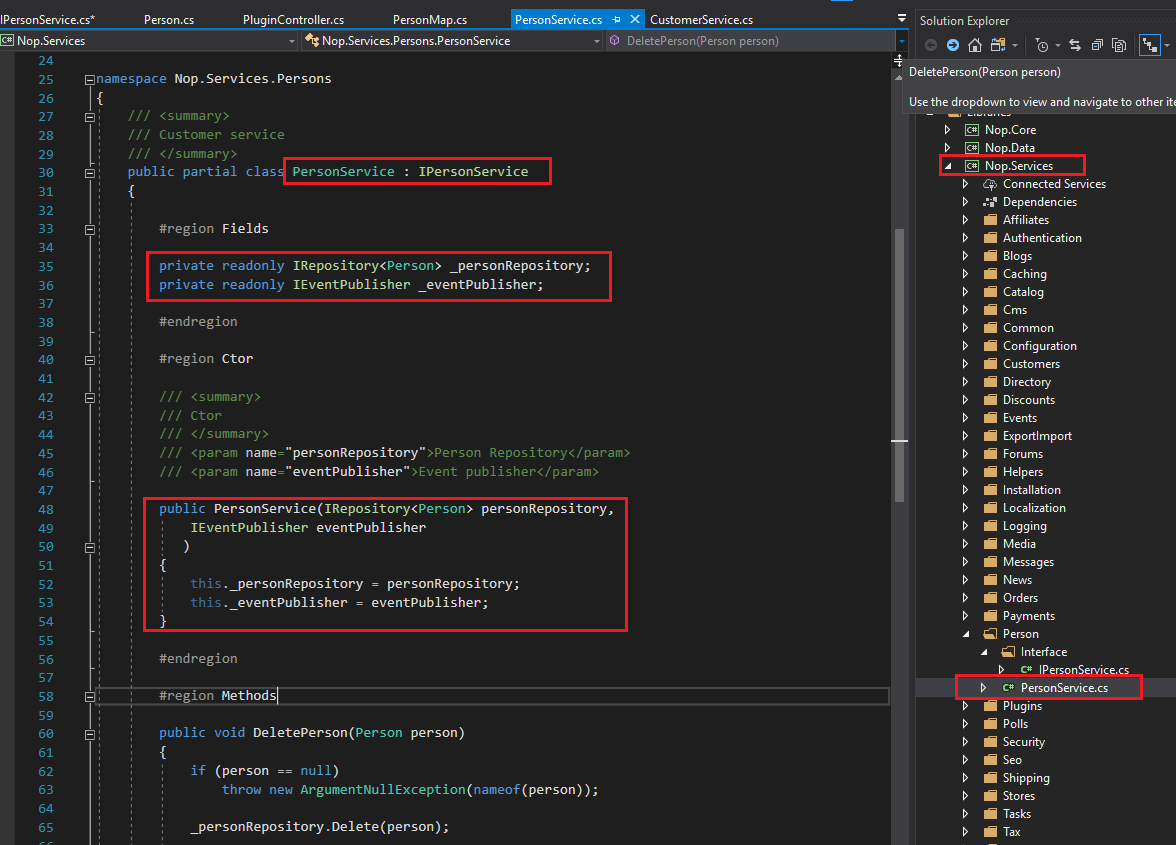
* Create a class name “**PersonMap.cs**” and inherit **“NopEntityTypeConfiguration<Person>”** of **Person** type

|  |  |
| --- | --- |
| this.ToTable("Person"); | Specify the Person table name |
| this.HasKey(p => p.Id); | Specify the ID as primary key |
| this.HasRequired(p => p.Customer)  .WithMany()  .HasForeignKey(p => p.CustomerID); | Specify CustomerID is foreign key to Customer table |

1. Nop.Service



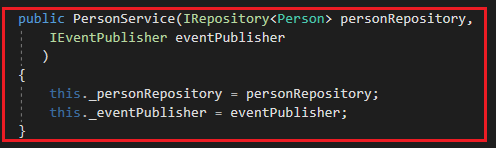
* Create “**IPersonService.cs**” interface
* Create “**GetPersonByIds**”, “**GetPersonById**”, “**GetPesonByname**”, “**InsertPerson**”, “**DeletePerson**” and “**UpdatePerson**” methods.



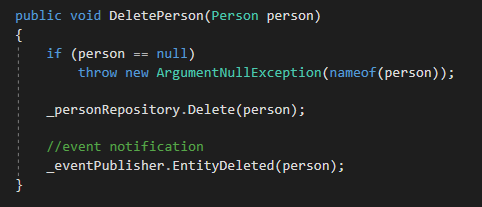
* Create “**PersonService.cs**” that implements all methods of “**IPersonService.cs**” interface
* Create two readonly properties:

1. “**\_personRepository**” of “**IRepository<Person>**”
2. “**\_eventPublisher”** of “**IEventPublisher**”

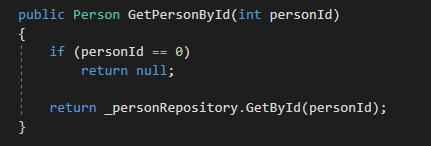
* And initialize in the constructor



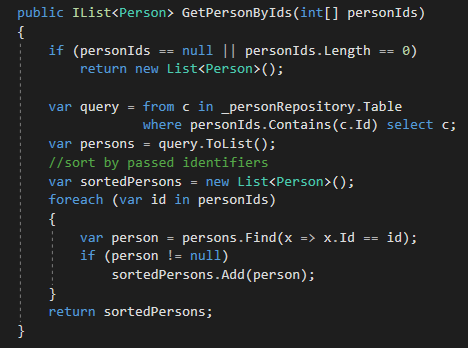
* **“DeletePerson”** method



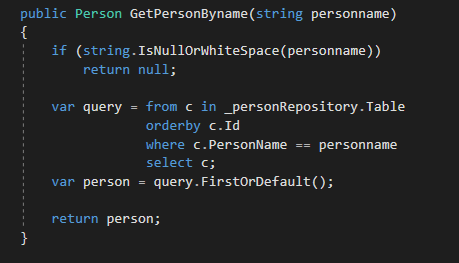
* **“GetPersonById”** method



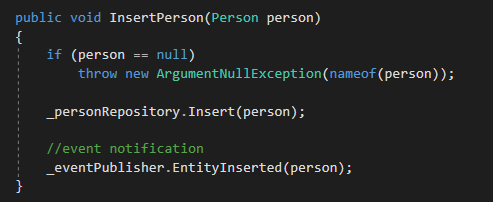
* **“GetPersonByIds”** method



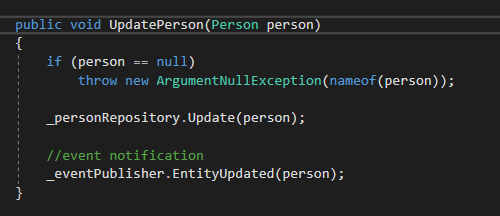
* **“GetPersonByname”** method



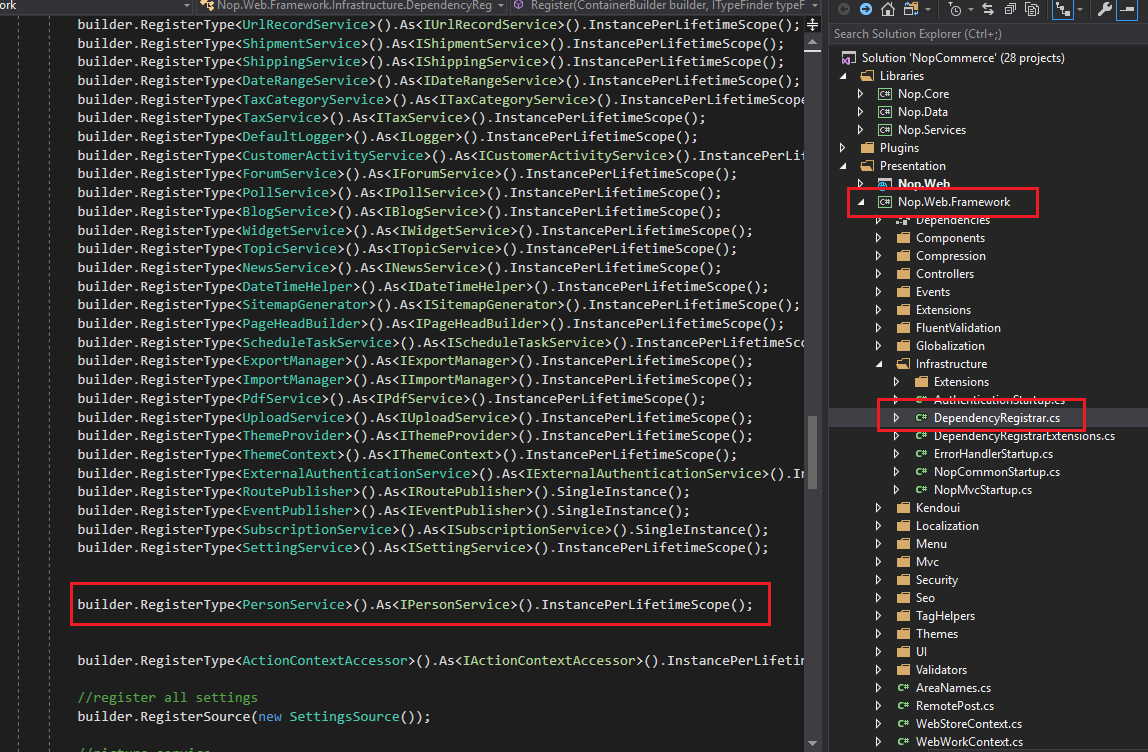
* **“InsertPerson”** method



* **“UpdatePerson”** method



1. Nop.Web.Framework (DependencyRegister.cs)



* Register **“PersonService”** in **“DependencyRegistrar.cs”**

