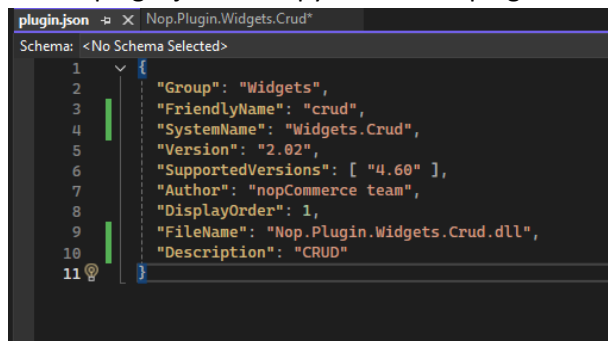


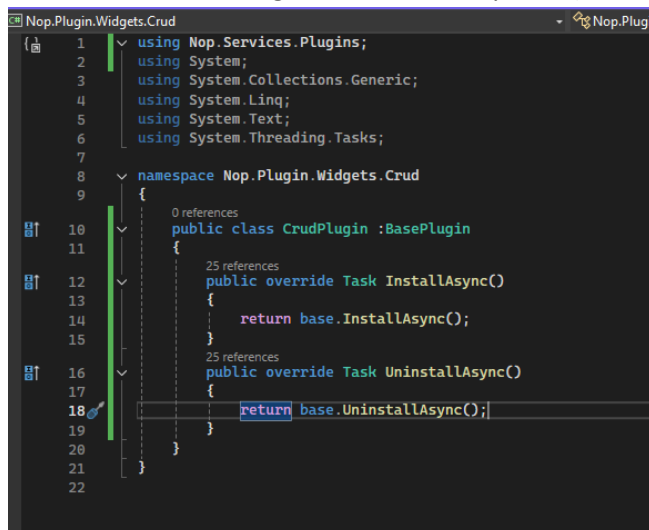
## How to create CRUD plugin in nopCommerce 4.60.0

1. Download source 4.60.0 code from official website
2. Set database and run project
3. Create new plugin Nop.Plugin.Widgets.Crud and select .NET 7
4. add Nop.Web project reference
5. Create plugin.json or copy from other plugin and set dll



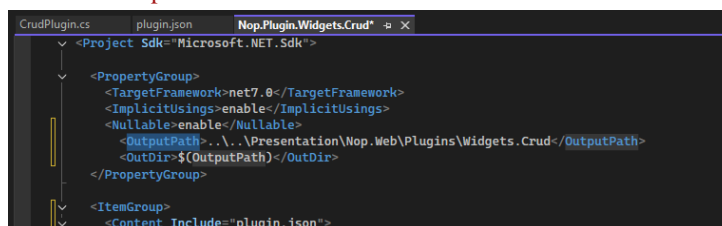
```
plugin.json  X Nop.Plugin.Widgets.Crud*
Schema: <No Schema Selected>
1 {
2   "Group": "Widgets",
3   "FriendlyName": "crud",
4   "SystemName": "Widgets.Crud",
5   "Version": "2.02",
6   "SupportedVersions": [ "4.60" ],
7   "Author": "nopCommerce team",
8   "DisplayOrder": 1,
9   "FileName": "Nop.Plugin.Widgets.Crud.dll",
10  "Description": "CRUD"
11 }
```

6. Create a class CrudPlugin.cs and make it public (note: most of the classes should be public)



```
Nop.Plugin.Widgets.Crud  Nop.Plugin
1 using Nop.Services.Plugins;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Nop.Plugin.Widgets.Crud
9 {
10     0 references
11     public class CrudPlugin : BasePlugin
12     {
13         25 references
14         public override Task InstallAsync()
15         {
16             return base.InstallAsync();
17         }
18         25 references
19         public override Task UninstallAsync()
20         {
21             return base.UninstallAsync();
22         }
23     }
24 }
```

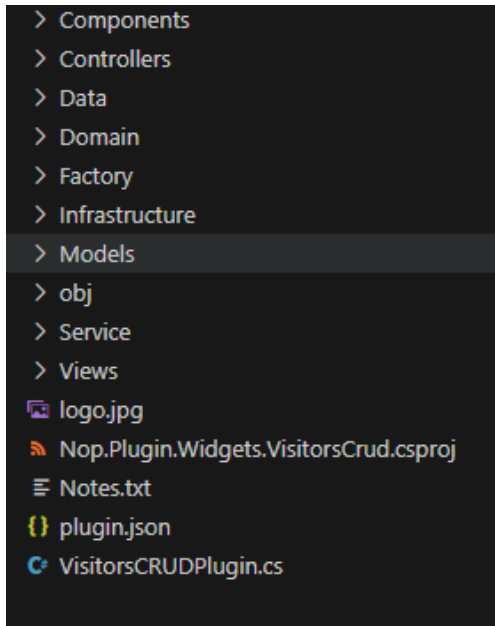
7. Now set **OutputPath** in



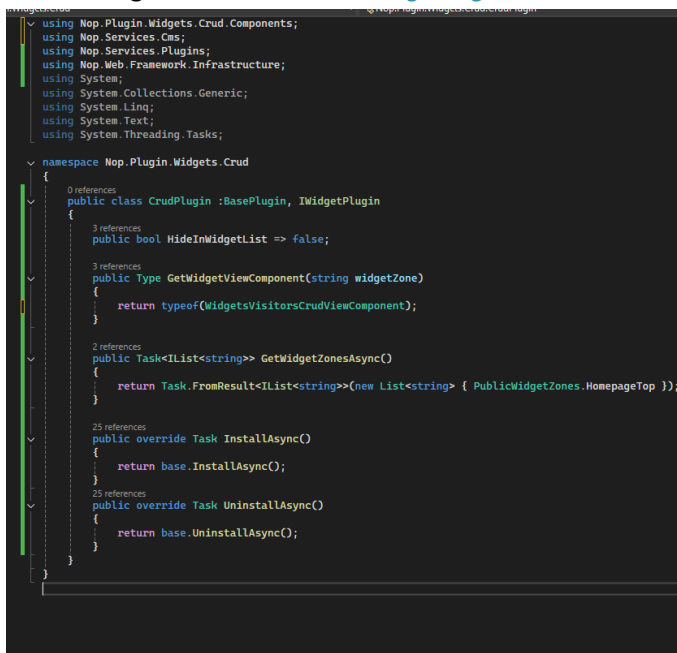
```
CrudPlugin.cs  plugin.json  Nop.Plugin.Widgets.Crud*  X
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net7.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <OutputPath>..\..\Presentation\Nop.Web\Plugins\Widgets.Crud</OutputPath>
    <OutDir>$(OutputPath)</OutDir>
  </PropertyGroup>
  <ItemGroup>
    <Content Include="plugin.json">
```

8. Now build it and check it from admin side

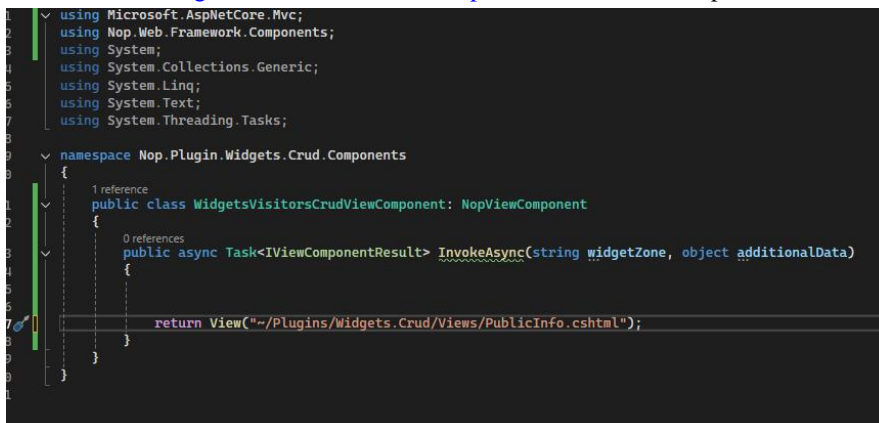
9. Now create all folder



10. In CrudPlugin.cs inherit from [IWidgetPlugin](#)



11. Now create [WidgetsVisitorsCrudViewComponent](#) class inside component folder



12. Now create **PublicInfo.cshtml** inside view folder
13. Add model **PublicInfoModel.cs** ConfigurationModel.cs
14. In ConfigurationModel.cs add DataAnnotations

```

using Microsoft.AspNetCore.Mvc.Rendering;
using Nop.Web.Framework.Models;
using Nop.Web.Framework.Mvc.ModelBinding;
using System.ComponentModel.DataAnnotations;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Nop.Plugin.Widgets.Crud.Models
{
    public record ConfigurationModel:BaseNopModel
    {
        public int Id { get; set; }

        [NopResourceDisplayName("Name")]
        public string Name { get; set; }

        [NopResourceDisplayName("Age")]
        public int Age { get; set; }

        [NopResourceDisplayName("Gender")]
        public string Gender { get; set; }

        public IEnumerable<SelectListItem> GenderSelection { get; set; }

        [DataType(DataType.PhoneNumber)]
        [NopResourceDisplayName("Phone")]
        public string Phone { get; set; }
    }
}

```

15. PublicInfoModel.cs

```

using Nop.Web.Framework.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Nop.Plugin.Widgets.Crud.Models
{
    public record PublicInfoModel:BaseNopModel
    {
        public int Id { get; set; }

        public string Name { get; set; }

        public int Age { get; set; }

        public string Gender { get; set; }

        public string Phone { get; set; }
    }
}

```

## 16. Add a class in Domain Visitor.cs

```
1 using Nop.Core;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Nop.Plugin.Widgets.Crud.Domain
9 {
10     public class Visitor : BaseEntity
11     {
12         public string Name { get; set; }
13         public int Age { get; set; }
14         public string Gender { get; set; }
15         public string Phone { get; set; }
16     }
17 }
18
```

## 17. difference between Visitor and PublicInfoModel class

### 1. Visitor (Domain Model)

- Located in: `Nop.Plugin.Widgets.Crud.Domain`
- Inherits from: `BaseEntity`
- Purpose: Represents the database entity.
- Usage:
  - This class is mapped to the database table.
  - It is used by the data access layer (repositories and services).
  - It interacts with Entity Framework Core to perform CRUD operations.

```
csharp Copy Edit

namespace Nop.Plugin.Widgets.Crud.Domain
{
    public class Visitor : BaseEntity
    {
        public string Name { get; set; }
        public int Age { get; set; }
        public string Gender { get; set; }
        public string Phone { get; set; }
    }
}
```

- Key Characteristic:
  - The `Visitor` class extends `BaseEntity`, which typically includes properties like `Id`, `CreatedOnUtc`, `UpdatedOnUtc`, etc.
  - It is responsible for persistence (saving data to and retrieving data from the database).

18.

## 2. `PublicInfoModel` (View Model)

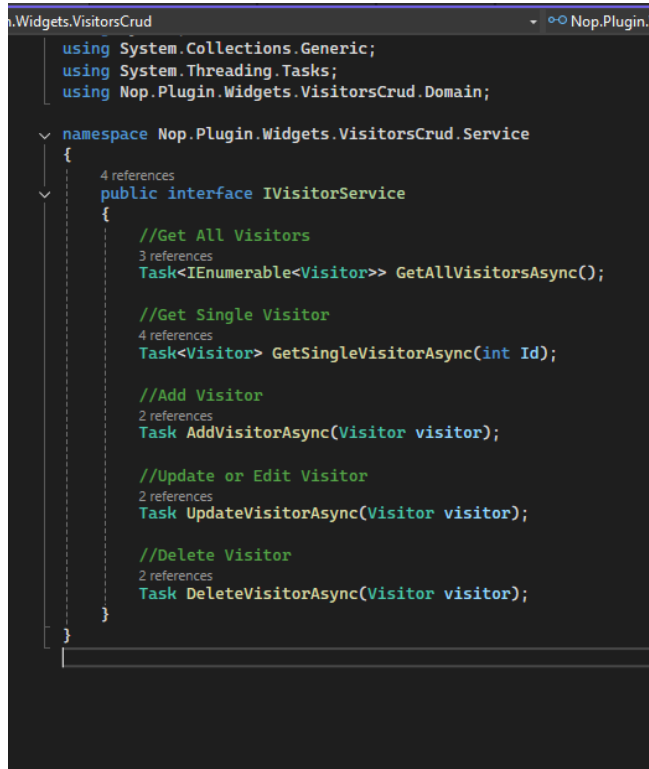
- **Located in:** `Nop.Plugin.Widgets.Crud.Models`
- **Inherits from:** `BaseNopModel`
- **Purpose:** Represents the data for the view (UI layer).
- **Usage:**
  - This class is used in the presentation layer (views or API responses).
  - It is returned to the UI or API, containing only necessary data for display.
  - It is separate from the database entity to enforce separation of concerns.

```
csharp Copy Edit  
  
namespace Nop.Plugin.Widgets.Crud.Models  
{  
    public record PublicInfoModel : BaseNopModel  
    {  
        public int Id { get; set; }  
        public string Name { get; set; }  
        public int Age { get; set; }  
        public string Gender { get; set; }  
        public string Phone { get; set; }  
    }  
}
```

- **Key Characteristic:**
  - `PublicInfoModel` extends `BaseNopModel`, which is used for models in the nopCommerce UI layer.
  - It is designed to be a DTO (Data Transfer Object) that prevents exposing unnecessary database details to the UI.



## 19. Create IVisitorService inside service folder



The screenshot shows a code editor window with the file path `Widgets.VisitorsCrud` and a solution explorer on the left showing the `Nop.Plugin` project. The code defines the `IVisitorService` interface within the `Nop.Plugin.Widgets.VisitorsCrud.Service` namespace. The interface includes five methods: `GetAllVisitorsAsync()`, `GetSingleVisitorAsync(int Id)`, `AddVisitorAsync(Visitor visitor)`, `UpdateVisitorAsync(Visitor visitor)`, and `DeleteVisitorAsync(Visitor visitor)`. Each method is preceded by a comment and a reference count.

```
using System.Collections.Generic;
using System.Threading.Tasks;
using Nop.Plugin.Widgets.VisitorsCrud.Domain;

namespace Nop.Plugin.Widgets.VisitorsCrud.Service
{
    4 references
    public interface IVisitorService
    {
        //Get All Visitors
        3 references
        Task<IEnumerable<Visitor>> GetAllVisitorsAsync();

        //Get Single Visitor
        4 references
        Task<Visitor> GetSingleVisitorAsync(int Id);

        //Add Visitor
        2 references
        Task AddVisitorAsync(Visitor visitor);

        //Update or Edit Visitor
        2 references
        Task UpdateVisitorAsync(Visitor visitor);

        //Delete Visitor
        2 references
        Task DeleteVisitorAsync(Visitor visitor);
    }
}
```

## 20. Create VisitorService inside service folder

```
using Nop.Data;
using Nop.Plugin.Widgets.Crud.Domain;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Nop.Plugin.Widgets.Crud.Services
{
    1 reference
    public class VisitorService:IVisitorService
    {
        private readonly IRepository<Visitor> _visitorRepository;

        0 references
        public VisitorService(IRepository<Visitor> visitorRepository)
        {
            _visitorRepository = visitorRepository;
        }

        1 reference
        public async Task<IEnumerable<Visitor>> GetAllVisitorsAsync()
        {
            return await _visitorRepository.GetAllAsync(query =>
            {
                return from v in query where v.Id > 0 select v;
            });
        }

        1 reference
        public async Task<Visitor> GetSingleVisitorAsync(int Id)
        {
            return await _visitorRepository.GetByIdAsync(Id);
        }

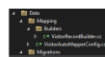
        1 reference
        public async Task AddVisitorAsync(Visitor visitor)
        {
            await _visitorRepository.InsertAsync(visitor);
        }

        1 reference
        public async Task DeleteVisitorAsync(Visitor visitor)
        {
            await _visitorRepository.DeleteAsync(visitor);
        }

        1 reference
        public async Task UpdateVisitorAsync(Visitor visitor)
        {
            await _visitorRepository.UpdateAsync(visitor);
        }
    }
}
```

## steps for Data folder

21. Add Mapping Migrations folder inside Data folder
22. Add Builders folder and VisitorAutoMapperConfig.cs inside Mapping folder
23. Add VisitorRecordBuilder.cs inside Builders folder
24. Add DbMigration inside Migrations folder



25.

```

VisitorRecordBuilder.cs
Nop.Plugin.Widgets.Crud

1 using FluentMigrator.Builders.Create.Table;
2 using Nop.Data.Mapping.Builders;
3 using Nop.Plugin.Widgets.Crud.Domain;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace Nop.Plugin.Widgets.Crud.Data.Mapping.Builders
11 {
12     public class VisitorRecordBuilder : NopEntityBuilder<Visitor>
13     {
14         public override void MapEntity(CreateTableExpressionBuilder table)
15         {
16             table.WithColumn(nameof(Visitor.Name)).AsString(100)
17                 .WithColumn(nameof(Visitor.Age)).AsInt64()
18                 .WithColumn(nameof(Visitor.Gender)).AsString(10)
19                 .WithColumn(nameof(Visitor.Phone)).AsString(20);
20         }
21     }
22 }

```

26.

```

VisitorAutoMapperConfig.cs
Nop.Plugin.Widgets.Crud

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Nop.Plugin.Widgets.Crud.Data.Mapping
8 {
9     public class VisitorAutoMapperConfig
10     {
11     }
12 }

```

27.

```

DbMigrations.cs
Nop.Plugin.Widgets.Crud

1 using FluentMigrator;
2 using Nop.Data.Extensions;
3 using Nop.Data.Migrations;
4 using Nop.Plugin.Widgets.Crud.Domain;
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10
11 namespace Nop.Plugin.Widgets.Crud.Data.Migrations
12 {
13     [DepMigration("2020-02-20 19:45:08:9843077", "Custom Visitor base schema", MigrationProcessType.Installation)]
14     public class DbMigration : AutoReversingMigration
15     {
16         public override void Up()
17         {
18             Create.TableFor<Visitor>();
19         }
20     }
21 }

```

## 28. Now time to Factory

29. Add VisitorModelFactory.cs IVisitorModelFactory.cs inside Factory folder

```

IVisitorModelFactory.cs
Nop.Plugin.Widgets.Crud

1 using Nop.Plugin.Widgets.Crud.Models;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Nop.Plugin.Widgets.Crud.Factory
9 {
10     public interface IVisitorModelFactory
11     {
12         Task<ConfigurationModel> PrepareVisitorModelAsync(ConfigurationModel configurationModel);
13
14         Task<IEnumerable<ConfigurationModel>> PrepareVisitorModelListAsync();
15
16         Task<IEnumerable<PublicInfoModel>> PreparePublicVisitorModelListAsync();
17
18         Task<ConfigurationModel> AddVisitorModelAsync(ConfigurationModel configurationModel);
19
20         Task<ConfigurationModel> GetVisitorModelAsync(int Id);
21
22         Task<ConfigurationModel> EditVisitorModelAsync(ConfigurationModel configurationModel);
23
24         Task<ConfigurationModel> DeleteVisitorModelAsync(int Id);
25     }
26 }

```



```
VisitorModelFactory.cs* ×
Nop.Plugin.Widgets.Crud Nop.Plugin.Widgets.Crud.Factory.VisitorModelFactory

1 using Microsoft.AspNetCore.Mvc.Rendering;
2 using Nop.Plugin.Widgets.Crud.Domain;
3 using Nop.Plugin.Widgets.Crud.Models;
4 using Nop.Plugin.Widgets.Crud.Services;
5 using System;
6 using System.Collections.Generic;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10
11 namespace Nop.Plugin.Widgets.Crud.Factory
12 {
13     1 reference
14     public class VisitorModelFactory : IVisitorModelFactory
15     {
16         private readonly IVisitorService _visitorService;
17
18         0 references
19         public VisitorModelFactory(IVisitorService visitorService)
20         {
21             _visitorService = visitorService;
22
23         1 reference
24         public async Task<ConfigurationModel> PrepareVisitorModelAsync(ConfigurationModel configurationModel)
25         {
26             configurationModel.GenderSelection = new List<SelectListItem>
27             {
28                 new SelectListItem { Text = "Select a gender", Value = "" },
29                 new SelectListItem { Text = "Male", Value = "Male" },
30                 new SelectListItem { Text = "Female", Value = "Female" },
31                 new SelectListItem { Text = "Other", Value = "Other" }
32             };
33             return configurationModel;
34         }
35
36         1 reference
37         public async Task<IEnumerable<ConfigurationModel>> PrepareVisitorModelListAsync()
38         {
39             var visitors = _visitorService.GetAllVisitorsAsync().Result;
40             var visitorList = new List<ConfigurationModel>();
41
42             foreach (var visitor in visitors)
43             {
44                 visitorList.Add(new ConfigurationModel
45                 {
46                     Id = visitor.Id,
47                     Name = visitor.Name,
48                     Age = visitor.Age,
49                     Gender = visitor.Gender,
50                     Phone = visitor.Phone
51                 });
52             }
53             return visitorList;
54         }
55     }
56 }
```

```
VisitorModelFactory.cs* X
Nop.Plugin.Widgets.Crud Nop.Plugin.Widgets.Crud.Factory.VisitorModelFactory

83         return null;
84     }
85
86     1 reference
87     public async Task<ConfigurationModel> GetVisitorModelAsync(int Id)
88     {
89         var getVisitor = _visitorService.GetSingleVisitorAsync(Id).Result;
90         var sendVisitor = new ConfigurationModel
91         {
92             Id = Id,
93             Name = getVisitor.Name,
94             Age = getVisitor.Age,
95             Gender = getVisitor.Gender,
96             Phone = getVisitor.Phone
97         };
98
99         sendVisitor.GenderSelection = new List<SelectListItem>
100         {
101             new SelectListItem { Text = "Select a gender", Value = "" },
102             new SelectListItem { Text = "Male", Value = "Male" },
103             new SelectListItem { Text = "Female", Value = "Female" },
104             new SelectListItem { Text = "Other", Value = "Other" }
105         };
106
107         return sendVisitor;
108     }
109
110     1 reference
111     public async Task<ConfigurationModel> EditVisitorModelAsync(ConfigurationModel configurationModel)
112     {
113         var newVisitor = _visitorService.GetSingleVisitorAsync(configurationModel.Id).Result;
114
115         newVisitor.Name = configurationModel.Name;
116         newVisitor.Phone = configurationModel.Phone;
117         newVisitor.Age = configurationModel.Age;
118         newVisitor.Gender = configurationModel.Gender;
119
120         await _visitorService.UpdateVisitorAsync(newVisitor);
121
122         return null;
123     }
124
125     1 reference
126     public async Task<ConfigurationModel> DeleteVisitorModelAsync(int Id)
127     {
128         var newVisitor = _visitorService.GetSingleVisitorAsync(Id).Result;
129
130         await _visitorService.DeleteVisitorAsync(newVisitor);
131
132         return null;
133     }
134 }
```

```

VisitorModelFactory.cs*  X
Nop.Plugin.Widgets.Crud  Nop.Plugin.Widgets.Crud.Factory.VisitorModelFactory

46         Phone = visitor.Phone
47     });
48 }
49
50     return visitorList;
51 }
52
53 1 reference
54 public async Task<IEnumerable<PublicInfoModel>> PreparePublicVisitorModelListAsync()
55 {
56     var visitors = _visitorService.GetAllVisitorsAsync().Result;
57     var visitorList = new List<PublicInfoModel>();
58
59     foreach (var visitor in visitors)
60     {
61         visitorList.Add(new PublicInfoModel
62         {
63             Id = visitor.Id,
64             Name = visitor.Name,
65             Age = visitor.Age,
66             Gender = visitor.Gender,
67             Phone = visitor.Phone
68         });
69     }
70
71     return visitorList;
72 }
73
74 1 reference
75 public async Task<ConfigurationModel> AddVisitorModelAsync(ConfigurationModel configurationModel)
76 {
77     var newVisitor = new Visitor();
78
79     newVisitor.Name = configurationModel.Name;
80     newVisitor.Phone = configurationModel.Phone;
81     newVisitor.Age = configurationModel.Age;
82     newVisitor.Gender = configurationModel.Gender;
83
84     await _visitorService.AddVisitorAsync(newVisitor);
85
86     return null;
87 }
88
89 1 reference
90 public async Task<ConfigurationModel> GetVisitorModelAsync(int Id)
91 {
92     var getVisitor = _visitorService.GetSingleVisitorAsync(Id).Result;
93     var sendVisitor = new ConfigurationModel
94     {
95         Id = Id,
96         Name = getVisitor.Name,
97         Age = getVisitor.Age,
98         Gender = getVisitor.Gender,
99         Phone = getVisitor.Phone
100     };
101 }

```

### 30. Add NopStartup.cs inside infrastructure folder

```

NopStartup.cs*  X
Nop.Plugin.Widgets.Crud  Nop.Plugin.Widgets.Crud.Infrastructure.NopStartup

1  using Microsoft.AspNetCore.Builder;
2  using Microsoft.Extensions.Configuration;
3  using Microsoft.Extensions.DependencyInjection;
4  using Nop.Core.Infrastructure;
5  using Nop.Plugin.Widgets.Crud.Factory;
6  using Nop.Plugin.Widgets.Crud.Services;
7  using System;
8  using System.Collections.Generic;
9  using System.Linq;
10 using System.Text;
11 using System.Threading.Tasks;
12
13 namespace Nop.Plugin.Widgets.Crud.Infrastructure
14 {
15     0 references
16     public class NopStartup : INopStartup
17     {
18         3 references
19         public int Order => 0;
20
21         2 references
22         public void Configure(IApplicationBuilder application)
23         {
24         }
25
26         2 references
27         public void ConfigureServices(IServiceCollection services, IConfiguration configuration)
28         {
29             services.AddScoped<IVisitorService, VisitorService>();
30             services.AddScoped<IVisitorModelFactory, VisitorModelFactory>();
31         }
32     }
33 }

```

### 31. WidgetsVisitorsCrudController.cs add in Controller folder

```
WidgetsVisitorsCrudController.cs* X
Nop.Plugin.Widgets.Crud Nop.Plugin.Widgets.Crud.Controllers.WidgetsVisitorsCrud

1 using Microsoft.AspNetCore.Mvc;
2 using Nop.Plugin.Widgets.Crud.Factory;
3 using Nop.Plugin.Widgets.Crud.Models;
4 using Nop.Services.Localization;
5 using Nop.Services.Messages;
6 using Nop.Services.Security;
7 using Nop.Web.Framework;
8 using Nop.Web.Framework.Controllers;
9 using Nop.Web.Framework.Mvc.Filters;
10 using System;
11 using System.Collections.Generic;
12 using System.Linq;
13 using System.Text;
14 using System.Threading.Tasks;
15
16 namespace Nop.Plugin.Widgets.Crud.Controllers
17 {
18     [AuthorizeAdmin]
19     [Area("AreaName.Admin")]
20     [AutoValidateAntiforgeryToken]
21     public class WidgetsVisitorsCrudController : BasePluginController
22     {
23         private readonly ILocalizationService _localizationService;
24         private readonly INotificationService _notificationService;
25         private readonly IPermissionService _permissionService;
26         private readonly IVisitorModelFactory _visitorModelFactory;
27
28         public WidgetsVisitorsCrudController(ILocalizationService localizationService,
29             INotificationService notificationService,
30             IVisitorModelFactory visitorModelFactory,
31             IPermissionService permissionService)
32         {
33             _localizationService = localizationService;
34             _notificationService = notificationService;
35             _visitorModelFactory = visitorModelFactory;
36             _permissionService = permissionService;
37         }
38
39         [HttpPost]
40         public async Task<ActionResult> Configure()
41         {
42             if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
43                 return AccessDeniedView();
44
45             var model = await _visitorModelFactory.PrepareVisitorModelListAsync();
46             return View("~/Plugins/Widgets.Crud/Views/Configure.cshtml", model);
47         }
48
49         [HttpGet]
50         public async Task<ActionResult> Create()
51         {
52             if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
53                 return AccessDeniedView();
54
55             var model = await _visitorModelFactory.PrepareVisitorModelAsync(new ConfigurationModel());
56             return View("~/Plugins/Widgets.Crud/Views/Create.cshtml", model);
57         }
58
59         [HttpPost]
60         public async Task<ActionResult> Create(ConfigurationModel model)
61         {
62             if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
63                 return AccessDeniedView();
64         }
65     }
66 }
```

```
WidgetsVisitorsCrudController.cs* X
Nop.Plugin.Widgets.Crud Nop.Plugin.Widgets.Crud.Controllers.WidgetsVisitorsCrud

59 {
60     if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
61         return AccessDeniedView();
62
63     await _visitorModelFactory.AddVisitorModelAsync(model);
64     _notificationService.SuccessNotification(await _localizationService.GetResourceAsync("Admin.Plugins.Saved"));
65     return await Configure();
66 }
67
68 [HttpGet]
69 public async Task<ActionResult> Edit(int Id)
70 {
71     if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
72         return AccessDeniedView();
73
74     var visitor = await _visitorModelFactory.GetVisitorModelAsync(Id);
75     return View("~/Plugins/Widgets.Crud/Views/Edit.cshtml", visitor);
76 }
77
78 [HttpPost]
79 public async Task<ActionResult> Edit(ConfigurationModel model)
80 {
81     if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
82         return AccessDeniedView();
83
84     await _visitorModelFactory.EditVisitorModelAsync(model);
85     _notificationService.SuccessNotification(await _localizationService.GetResourceAsync("Admin.Plugins.Saved"));
86     return await Configure();
87 }
88
89 [HttpGet]
90 public async Task<ActionResult> Delete(int Id)
91 {
92     if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
93         return AccessDeniedView();
94
95     var visitor = await _visitorModelFactory.GetVisitorModelAsync(Id);
96     return View("~/Plugins/Widgets.Crud/Views/Delete.cshtml", visitor);
97 }
98
99 [HttpPost, ActionName("Delete")]
100 public async Task<ActionResult> DeleteConfirmed(int Id)
101 {
102     if (!await _permissionService.AuthorizeAsync(StandardPermissionProvider.ManageWidgets))
103         return AccessDeniedView();
104
105     await _visitorModelFactory.DeleteVisitorModelAsync(Id);
106     _notificationService.SuccessNotification(await _localizationService.GetResourceAsync("Admin.Plugins.Saved"));
107     return await Configure();
108 }
109
110 }
```

32. Now create all view according to controller and also add \_ViewImports.cshtml

```
_ViewImports.cshtml
1 @inherits Nop.Web.Framework.Mvc.Razor.NopRazorPage<TModel>
2 @inject INopHtmlHelper NopHtml
3 @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
4 @addTagHelper *, Nop.Web.Framework
5
6 @using Microsoft.AspNetCore.Mvc.ViewFeatures
7 @using Nop.Web.Areas.Admin.Components
8 @using Nop.Web.Framework.UI
9 @using Nop.Web.Framework.Extensions
10 @using System.Text.Encodings.Web
```

```
Configure.cshtml
1 @using Nop.Plugin.Widgets.Crud.Models
2 @model List<ConfigurationModel>
3
4 @{
5     Layout = "_ConfigurePlugin";
6     //active menu item (system name)
7     NopHtml.SetActiveMenuItemSystemName("Widgets.Crud");
8 }
9
10 <div class="content-header clearfix">
11     <h1 class="float-left">
12         Visitors
13     </h1>
14     <div class="float-right">
15         <a asp-controller="WidgetsVisitorsCrud" asp-action="Create" method="Get"><i class="fas fa-plus-square"></i>Add new visitor</a>
16     </div>
17 </div>
18
19 <br />
20 <br />
21
22 <section class="content">
23     <div class="container-fluid">
24         <div class="cards-group">
25             <div class="card card-default">
26                 <div class="card-body">
27                     <table class="table">
28                         <thead>
29                             <tr>
30                                 <th>Name</th>
31                                 <th>Age</th>
32                                 <th>Gender</th>
33                                 <th>Phone</th>
34                             </tr>
35                         </thead>
36                     </thead>
37                     <tbody>
38                         @foreach (var item in Model)
39                         {
40                             <tr>
41                                 <td>@item.Name</td>
42                                 <td>@item.Age</td>
43                                 <td>@item.Gender</td>
44                                 <td>@item.Phone</td>
45                                 <a asp-controller="WidgetsVisitorsCrud" asp-action="Edit" asp-route-id="@item.Id" class="btn btn-primary">Edit</a> |
46                                 <a asp-controller="WidgetsVisitorsCrud" asp-action="Delete" asp-route-id="@item.Id" class="btn btn-info">Details</a>
47                             </tr>
48                         }
49                     </tbody>
50                 </table>
51             </div>
52         </div>
53     </div>
54 </section>
```

Create.cshtml

Nop.Plugin.Widgets.Crud

```
1  @model Nop.Plugin.Widgets.Crud.Models.ConfigurationModel
2
3  @{
4      Layout = "_ConfigurePlugin";
5      //active menu item (system name)
6      NopHtml.SetActiveMenuItemSystemName("Widgets.Crud");
7  }
8
9  <form asp-controller="WidgetsVisitorsCrud" asp-action="Create" method="post">
10     <div asp-validation-summary="ModelOnly"></div>
11     <input asp-for="Id" type="hidden" />
12     <section class="content">
13         <div class="container-fluid">
14             <div class="cards-group">
15                 <div class="card card-default">
16                     <div class="card-header">
17                         @T("Plugins.Widgets.Crud.AddVisitorButton")
18                     </div>
19                     <div class="card-body">
20                         <div class="form-group row">
21                             <div class="col-md-3">
22                                 <nop-label asp-for="Name" />
23                             </div>
24                             <div class="col-md-9">
25                                 <nop-editor asp-for="Name" asp-required="true" />
26                                 <span asp-validation-for="Name"></span>
27                             </div>
28                         </div>
29                         <div class="form-group row">
30                             <div class="col-md-3">
31                                 <nop-label asp-for="Age" />
32                             </div>
33                             <div class="col-md-9">
34                                 <nop-editor asp-for="Age" asp-required="true" />
35                                 <span asp-validation-for="Age"></span>
36                             </div>
37                         </div>
38                         <div class="form-group row">
39                             <div class="col-md-3">
40                                 <nop-label asp-for="Gender" />
41                             </div>
42                             <div class="col-md-9">
43                                 <nop-select asp-for="Gender" asp-items="Model.GenderSelection" />
44                             </div>
45                         </div>
46                         <div class="form-group row">
47                             <div class="col-md-3">
48                                 <nop-label asp-for="Phone" />
49                             </div>
50                             <div class="col-md-9">
51                                 <nop-editor asp-for="Phone" asp-required="true" />
52                                 <span asp-validation-for="Phone"></span>
53                             </div>
54                         </div>
55                     </div>
56                 </div>
57                 <div class="card card-default">
58                     <div class="card-body">
59                         <div class="form-group row">
60                             <div class="col-md-9 offset-md-3">
61                                 <button type="submit" name="save" class="btn btn-primary">@T("Admin.Common.Save")</button>
62                             </div>
63                         </div>
64                     </div>
65                 </div>
66             </div>
67         </div>
68     </section>
69 </form>
```

```
Delete.cshtml  X
Nop.Plugin.Widgets.Crud
1  @model Nop.Plugin.Widgets.Crud.Models.ConfigurationModel
2
3  @{
4      Layout = "_ConfigurePlugin";
5      //active menu item (system name)
6      NopHtml.SetActiveMenuItemSystemName("Widgets.Crud");
7  }
8
9  <h1>Details</h1>
10
11  <div>
12      <h4>Visitor</h4>
13      <hr />
14
15      <dl class="row">
16          <dt class="col-sm-2">Name</dt>
17          <dd class="col-sm-10">@Model.Name</dd>
18          <dt class="col-sm-2">Age</dt>
19          <dd class="col-sm-10">@Model.Age</dd>
20          <dt class="col-sm-2">Gender</dt>
21          <dd class="col-sm-10">@Model.Gender</dd>
22          <dt class="col-sm-2">Phone</dt>
23          <dd class="col-sm-10">@Model.Phone</dd>
24      </dl>
25
26      <form asp-controller="WidgetsVisitorsCrud" asp-action="Delete" asp-route-id="@Model.Id" method="post">
27          <input type="hidden" asp-for="Id" />
28          <a asp-action="Edit" method="Get" class="btn btn-primary">Edit</a> |
29          <input type="submit" value="Delete" class="btn btn-danger" onclick="return confirm('Are you sure you want to delete this item?');"/>
30          <a asp-action="Configure" class="btn btn-dark">Back to List</a>
31      </form>
32  </div>
```

```

Edit.cshtml  X
Nop.Plugin.Widgets.Crud

4   @{
5       Layout = "_ConfigurePlugin";
6       //active menu item (system name)
7       NopHtml.SetActiveMenuItemSystemName("Widgets.Crud");
8   }
9
10  <h1>Edit</h1>
11
12  <h4>Visitor</h4>
13  <hr />
14  <div class="editSection">
15      <form asp-controller="WidgetsVisitorsCrud" asp-action="Edit" method="post">
16          <div asp-validation-summary="ModelOnly" class="text-danger"></div>
17          <input type="hidden" asp-for="Id" />
18
19          <div class="editFields">
20              <div class="form-group">
21                  <nop-label asp-for="Name" />
22                  <input asp-for="Name" class="form-control" />
23                  <span asp-validation-for="Name" class="text-danger"></span>
24              </div>
25              <div class="form-group">
26                  <nop-label asp-for="Age" />
27                  <input asp-for="Age" class="form-control" />
28                  <span asp-validation-for="Age" class="text-danger"></span>
29              </div>
30              <div class="form-group">
31                  <nop-label asp-for="Gender" />
32                  <nop-select asp-for="Gender" asp-items="Model.GenderSelection" />
33                  <span asp-validation-for="Gender" class="text-danger"></span>
34              </div>
35              <div class="form-group">
36                  <nop-label asp-for="Phone" />
37                  <input asp-for="Phone" class="form-control" />
38                  <span asp-validation-for="Phone" class="text-danger"></span>
39              </div>
40          </div>
41          <br />
42          <div class="saveButton">
43              <div class="form-group">
44                  <input type="submit" value="Save" class="btn btn-primary" />
45              </div>
46          </div>
47      </form>
48  </div>
49  <br />
50  <div>
51      <a asp-controller="WidgetsVisitorsCrud" asp-action="Configure">Back to List</a>
52  </div>
53

```

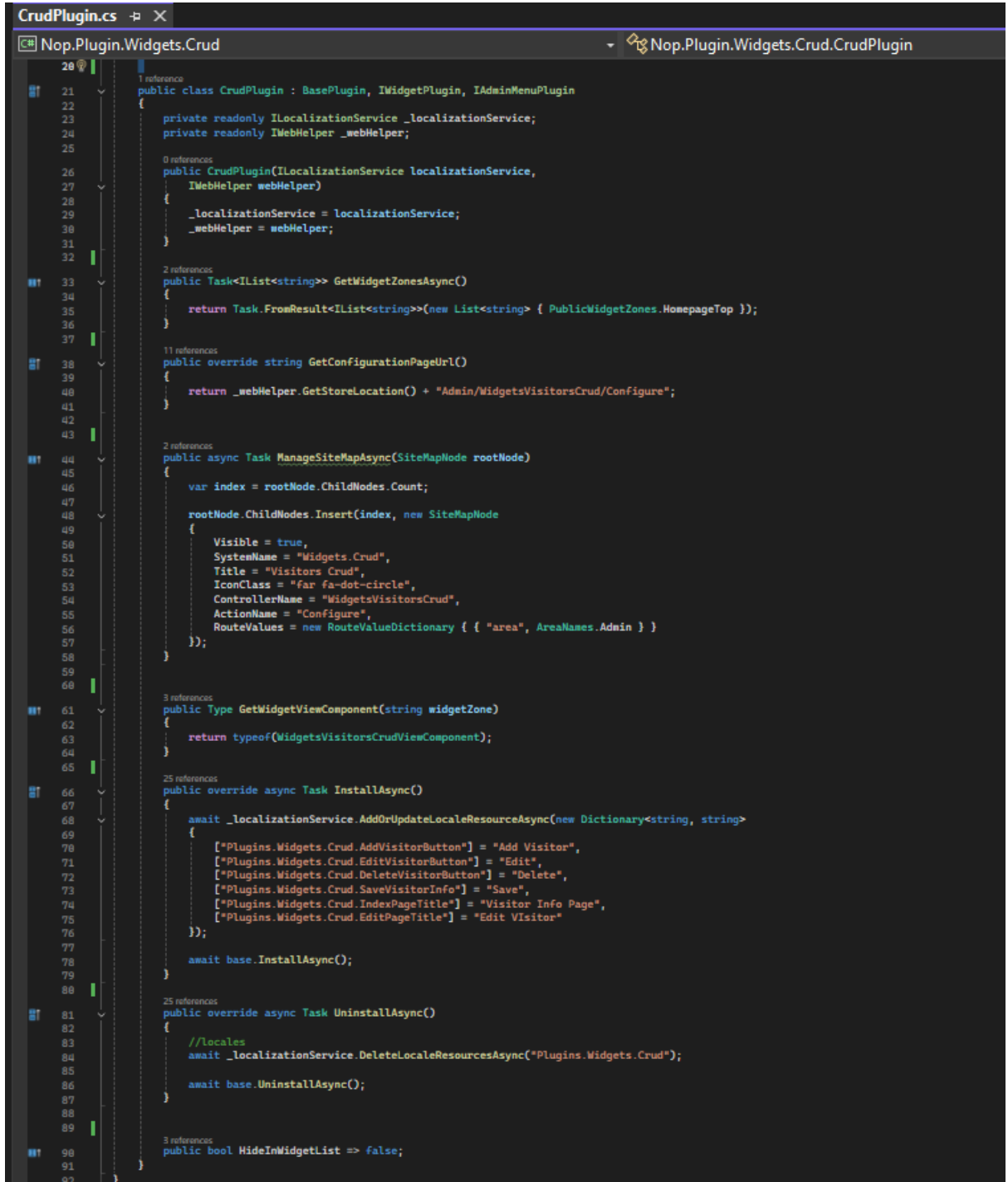


PublicInfo.cshtml

Nop.Plugin.Widgets.Crud

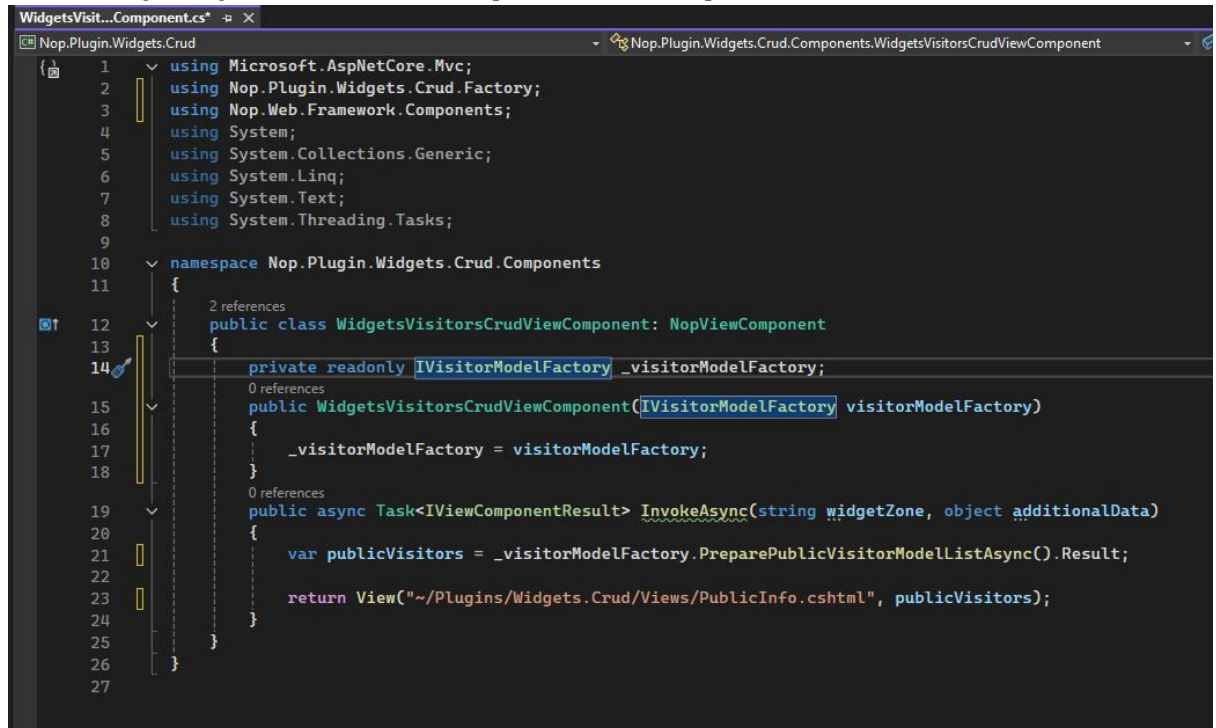
```
1  @using Nop.Plugin.Widgets.Crud.Models
2  @model IEnumerable<PublicInfoModel>
3  @{
4      Layout = "";
5  }
6
7  <div class="slider-wrapper theme-custom">
8      <table class="table">
9          <thead>
10             <tr>
11                 <th>Name</th>
12                 <th>Age</th>
13                 <th>Gender</th>
14                 <th>Phone</th>
15             </tr>
16         </thead>
17
18         <tbody>
19             @{
20                 foreach (var item in Model)
21                 {
22                     <tr>
23                         <td>@item.Name</td>
24                         <td>@item.Age</td>
25                         <td>@item.Gender</td>
26                         <td>@item.Phone</td>
27                     </tr>
28                 }
29             }
30         </tbody>
31     </table>
32 </div>
```

### 33. CrudPlugin inherit from IAdminMenuPlugin



```
20 1 reference
21 public class CrudPlugin : BasePlugin, IWidgetPlugin, IAdminMenuPlugin
22 {
23     private readonly ILocalizationService _localizationService;
24     private readonly IWebHelper _webHelper;
25
26     0 references
27     public CrudPlugin(ILocalizationService localizationService,
28         IWebHelper webHelper)
29     {
30         _localizationService = localizationService;
31         _webHelper = webHelper;
32     }
33
34     2 references
35     public Task<IList<string>> GetWidgetZonesAsync()
36     {
37         return Task.FromResult<IList<string>>(new List<string> { PublicWidgetZones.HomepageTop });
38     }
39
40     11 references
41     public override string GetConfigurationPageUrl()
42     {
43         return _webHelper.GetStoreLocation() + "Admin/WidgetsVisitorsCrud/Configure";
44     }
45
46     2 references
47     public async Task ManageSiteMapAsync(SiteMapNode rootNode)
48     {
49         var index = rootNode.ChildNodes.Count;
50
51         rootNode.ChildNodes.Insert(index, new SiteMapNode
52         {
53             Visible = true,
54             SystemName = "Widgets.Crud",
55             Title = "Visitors Crud",
56             IconClass = "far fa-dot-circle",
57             ControllerName = "WidgetsVisitorsCrud",
58             ActionName = "Configure",
59             RouteValues = new RouteValueDictionary { { "area", AreaNames.Admin } }
60         });
61     }
62
63     3 references
64     public Type GetWidgetViewComponent(string widgetZone)
65     {
66         return typeof(WidgetsVisitorsCrudViewComponent);
67     }
68
69     25 references
70     public override async Task InstallAsync()
71     {
72         await _localizationService.AddOrUpdateLocaleResourceAsync(new Dictionary<string, string>
73         {
74             [*Plugins.Widgets.Crud.AddVisitorButton] = "Add Visitor",
75             [*Plugins.Widgets.Crud.EditVisitorButton] = "Edit",
76             [*Plugins.Widgets.Crud.DeleteVisitorButton] = "Delete",
77             [*Plugins.Widgets.Crud.SaveVisitorInfo] = "Save",
78             [*Plugins.Widgets.Crud.IndexPageTitle] = "Visitor Info Page",
79             [*Plugins.Widgets.Crud.EditPageTitle] = "Edit Visitor"
80         });
81         await base.InstallAsync();
82     }
83
84     25 references
85     public override async Task UninstallAsync()
86     {
87         //locales
88         await _localizationService.DeleteLocaleResourcesAsync("Plugins.Widgets.Crud");
89         await base.UninstallAsync();
90     }
91
92     3 references
93     public bool HideInWidgetList => false;
94 }
```

34. Now time to go WidgetsVisitorsCrudViewComponent.cs inside components



```
1 using Microsoft.AspNetCore.Mvc;
2 using Nop.Plugin.Widgets.Crud.Factory;
3 using Nop.Web.Framework.Components;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace Nop.Plugin.Widgets.Crud.Components
11 {
12     2 references
13     public class WidgetsVisitorsCrudViewComponent: NopViewComponent
14     {
15         private readonly IVisitorModelFactory _visitorModelFactory;
16         0 references
17         public WidgetsVisitorsCrudViewComponent(IVisitorModelFactory visitorModelFactory)
18         {
19             _visitorModelFactory = visitorModelFactory;
20         }
21         0 references
22         public async Task<IViewComponentResult> InvokeAsync(string widgetZone, object additionalData)
23         {
24             var publicVisitors = _visitorModelFactory.PreparePublicVisitorModelListAsync().Result;
25             return View("~/Plugins/Widgets.Crud/Views/PublicInfo.cshtml", publicVisitors);
26         }
27     }
```