Project Structure

Introduction

This document will introduce the structure of the project. The project itself use and follow the ABP Framework product and documentation. ABP Framework created by Volosoft

ABP Framework is a complete infrastructure based on the ASP.NET Core to create modern web applications and APIs by following the software development best practices and the latest technologies.

ABP Framework is free and open source.

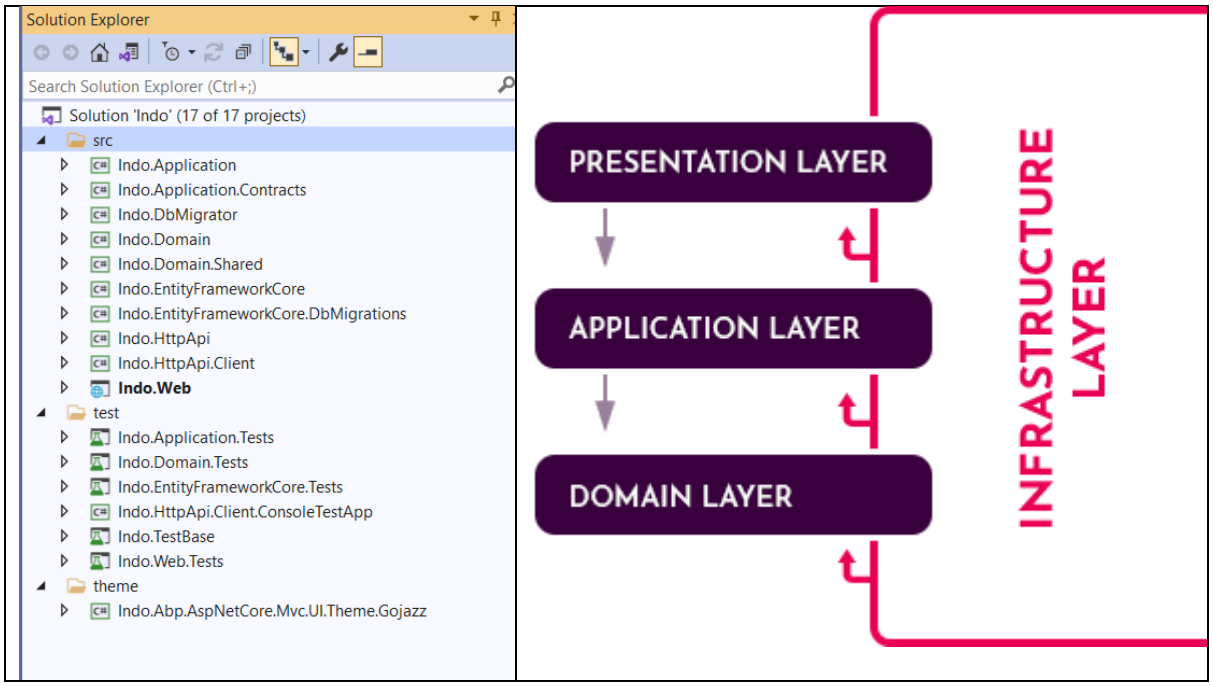
More about ABP Framework can be read on their official sites:

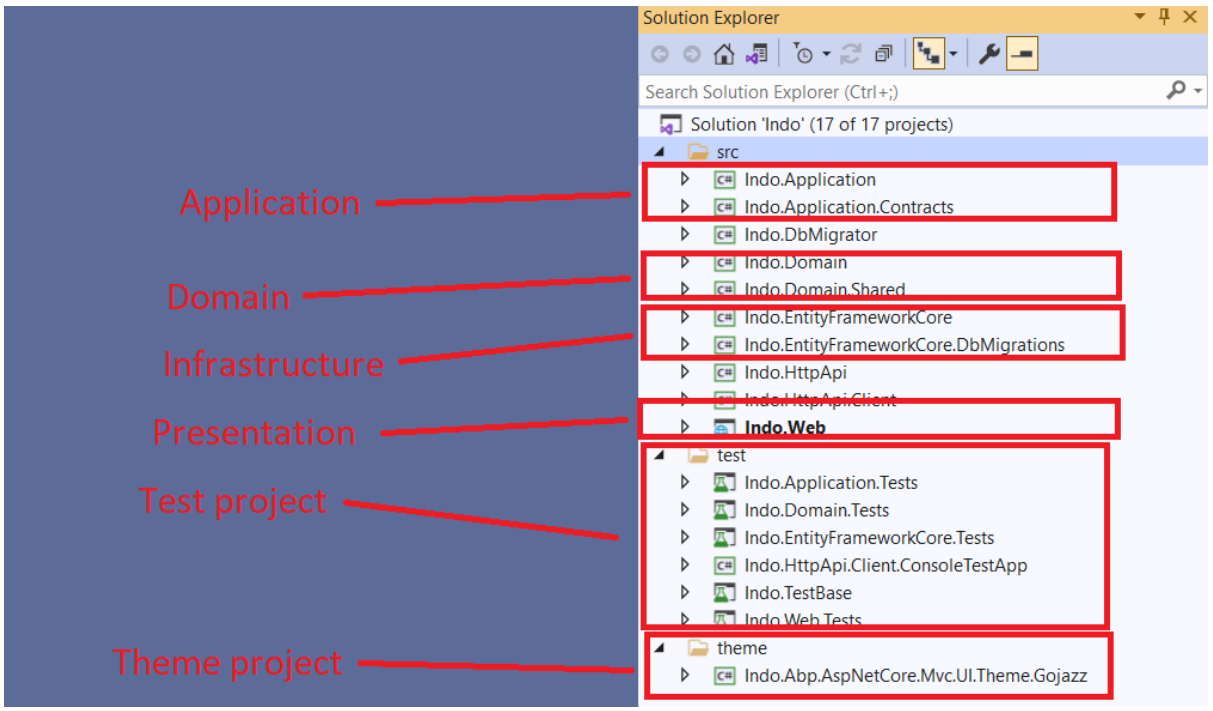
- Documentation: https://docs.abp.io/en/abp/latest
- Project Source Code: https://github.com/abpframework/abp
- Official Volosoft Company Website: https://volosoft.com/

ASP.NET Core 5 Razor Pages

ABP Framework can be implemented using various Web UI technologies such as: Angular, Blazor, MVC and Razor Pages.

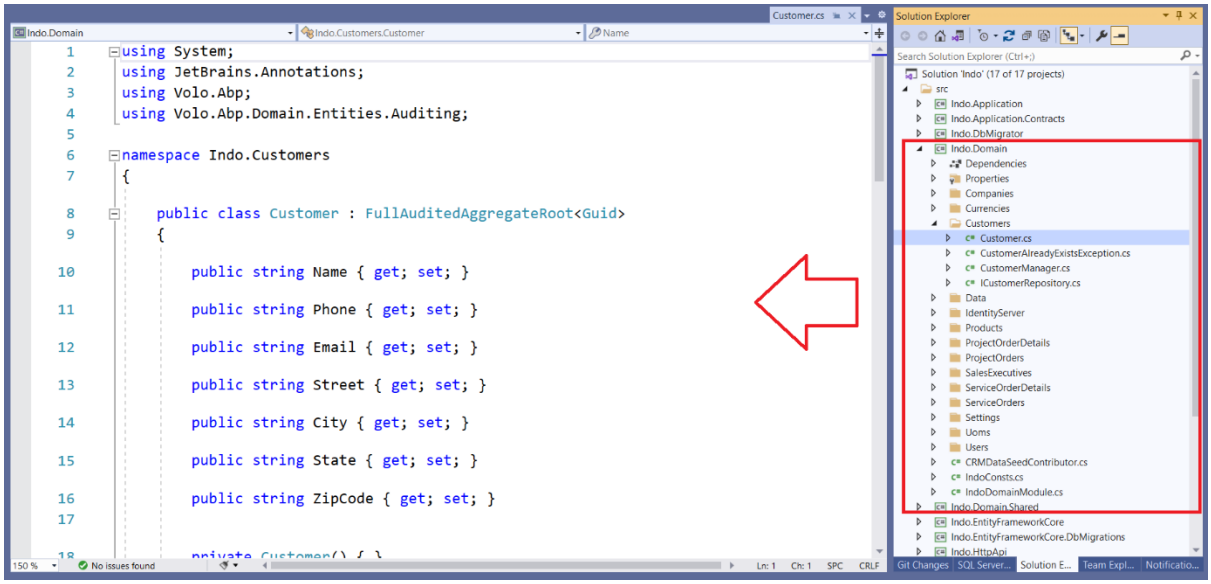
This product designed specifically to be used on ASP.NET Core 5 Razor Pages. Below is the complete project structure that follow the clean architecture:





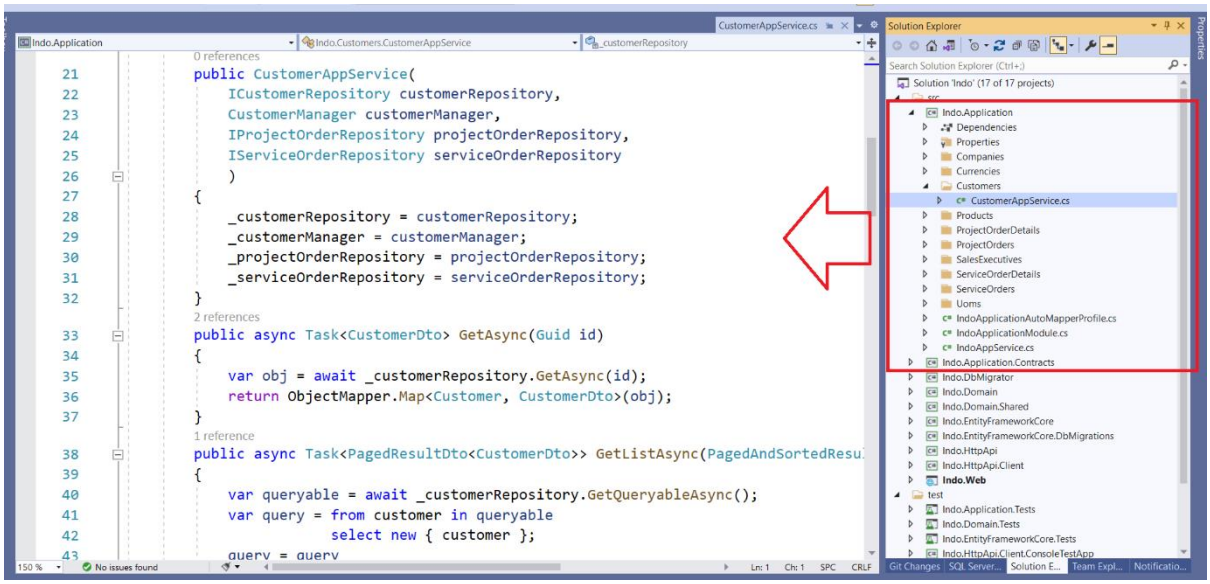
Domain Layer

Includes business objects and the core (domain) business rules. This is the heart of the application.



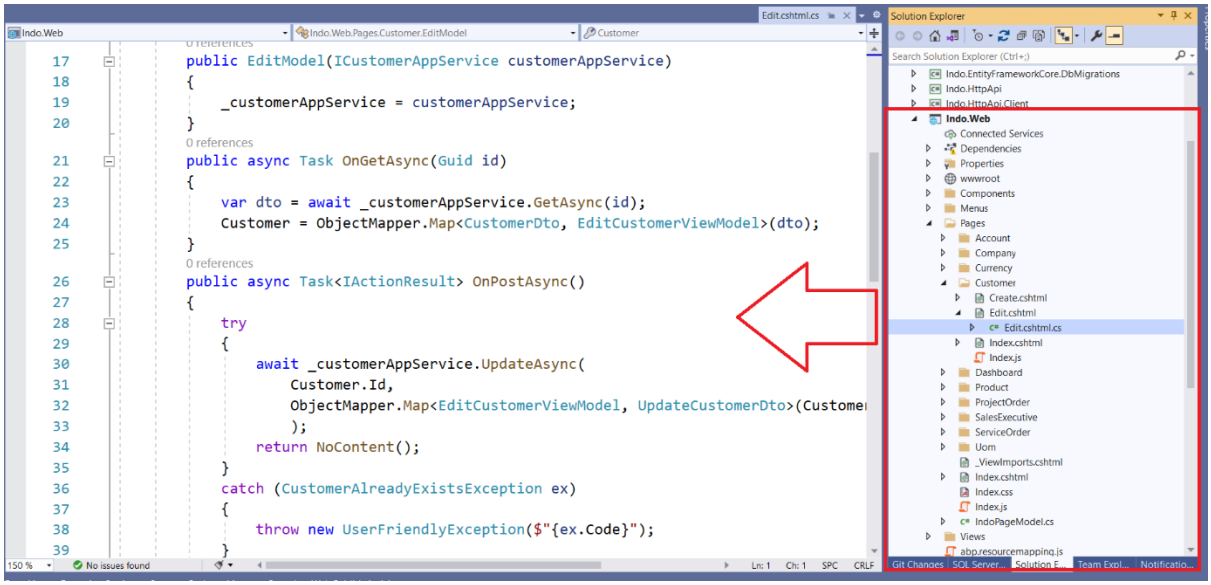
Application Layer

Mediates between the presentation and domain layers. Orchestrates business objects to perform specific application tasks. Implements use cases as the application logic.



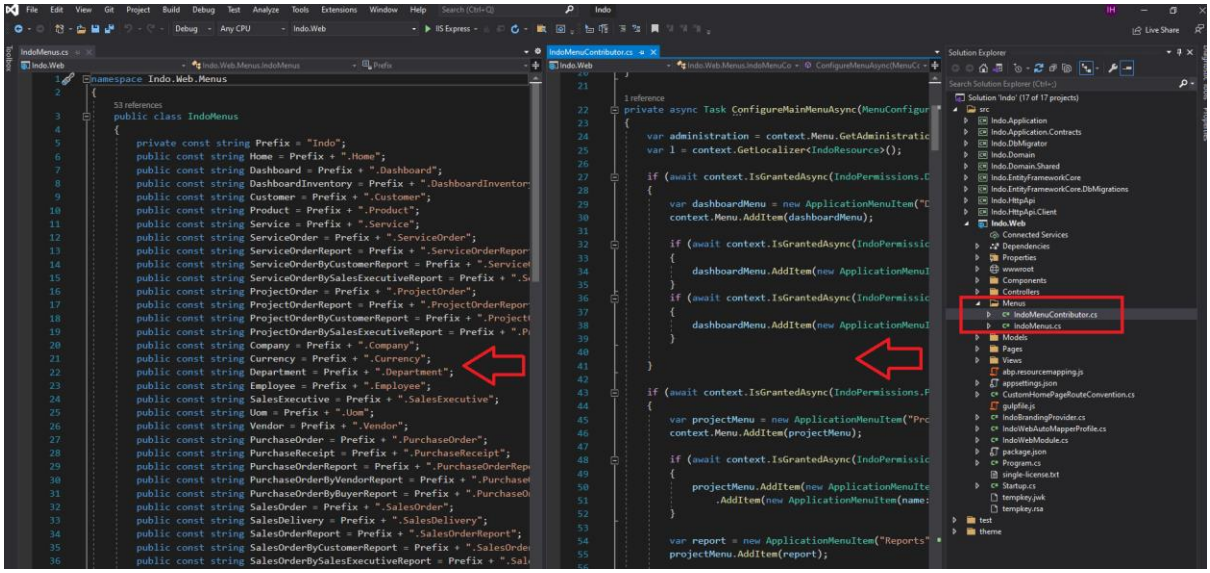
Presentation Layer

Provides an interface to the user. Uses the application layer to achieve user interactions.



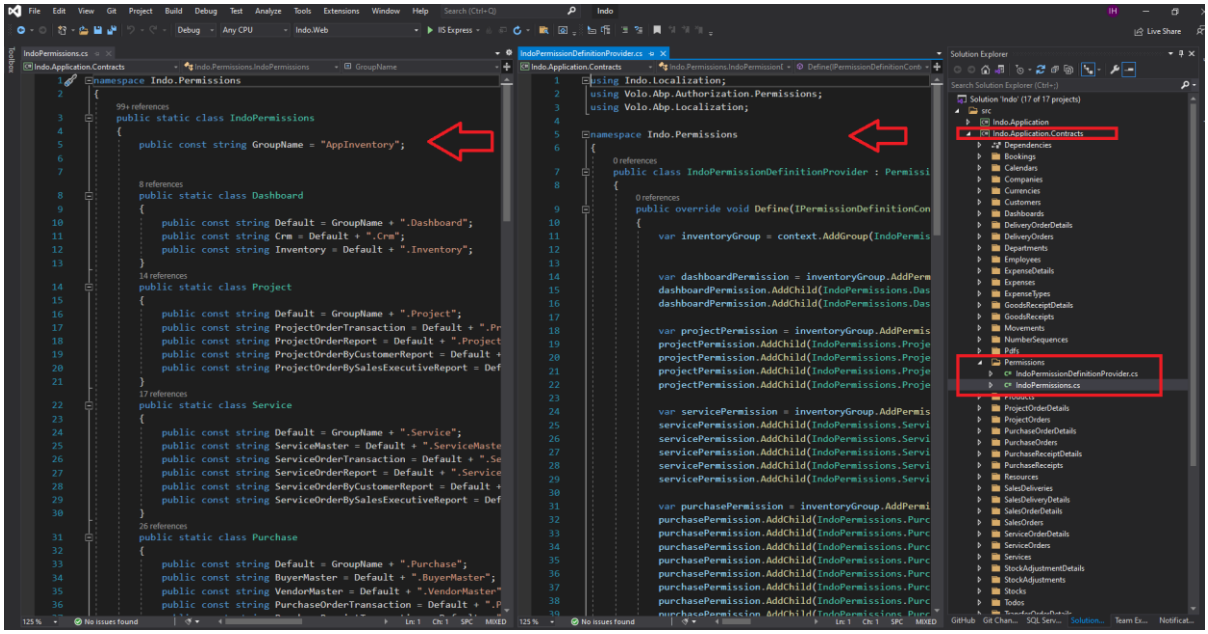
Menu Management

Provides mechanism to manage the menus.



RBAC Management

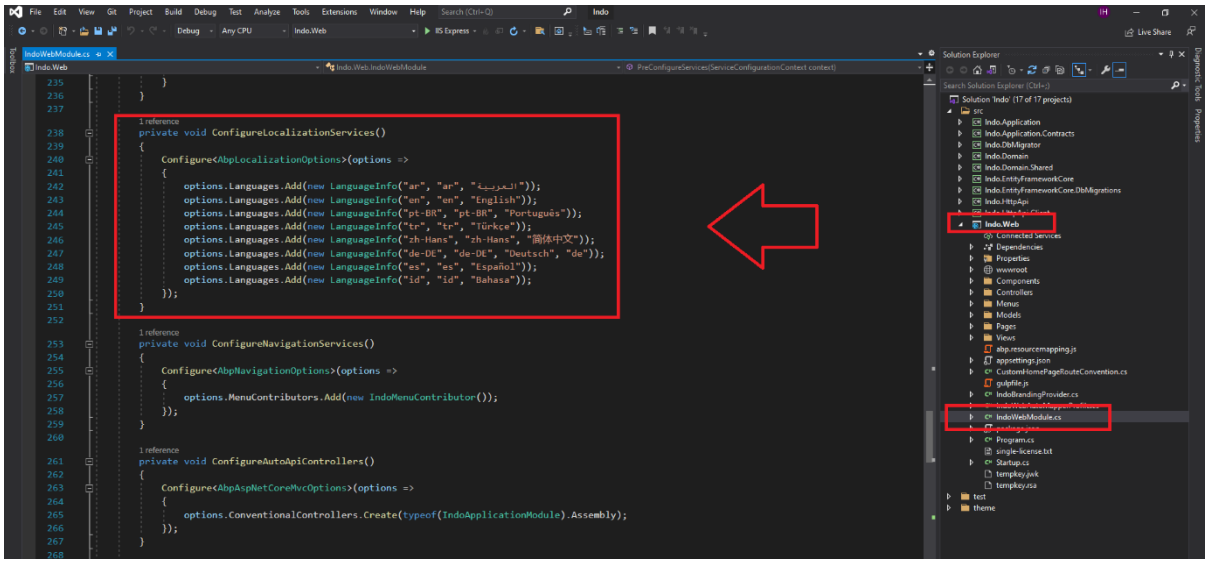
RBAC or Roles Based Access Control, before can be used at user interface level, it definition should be created first. The source code located at Application Contracts project.



Multi Language Management

Localization for multi language can be achieved easily by configuring each language with its translations in the format of JSON.

Language that will be supported, can be configure at Web project:

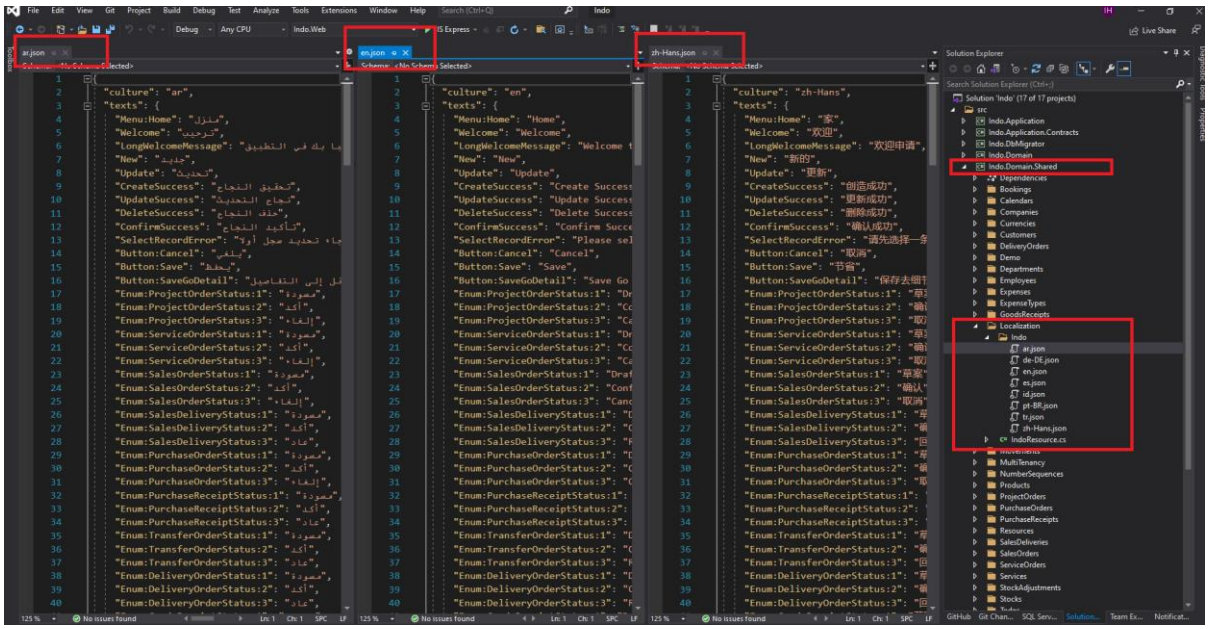


```
private void ConfigureLocalizationServices()
{
    Configure<AbpLocalizationOptions>(options =>
    {
        options.Languages.Add(new LanguageInfo("ar", "ar", "العربية"));
        options.Languages.Add(new LanguageInfo("en", "en", "English"));
        options.Languages.Add(new LanguageInfo("pt-BR", "pt-BR", "Português"));
        options.Languages.Add(new LanguageInfo("tr", "tr", "Türkçe"));
        options.Languages.Add(new LanguageInfo("zh-Hans", "zh-Hans", "简体中文"));
        options.Languages.Add(new LanguageInfo("de-DE", "de-DE", "Deutsch", "de"));
        options.Languages.Add(new LanguageInfo("es", "es", "Español"));
        options.Languages.Add(new LanguageInfo("id", "id", "Bahasa"));
    });
}

private void ConfigureNavigationServices()
{
    Configure<AbpNavigationOptions>(options =>
    {
        options.MenuContributors.Add(new IndoMenuContributor());
    });
}

private void ConfigureAutoApiControllers()
{
    Configure<AbpAspNetCoreMvcOptions>(options =>
    {
        options.ConventionalControllers.Create(typeof(IndoApplicationModule).Assembly);
    });
}
```

The translations, can be configured at Domain Shared project:



```
{
  "culture": "ar",
  "texts": {
    "Menu:Home": "منزل",
    "Welcome": "ترحيب",
    "LongWelcomeMessage": "مرحبًا بك في التطبيق",
    "New": "جديد",
    "Update": "تحديث",
    "CreateSuccess": "تم الإنشاء بنجاح",
    "UpdateSuccess": "نجاح التحديث",
    "DeleteSuccess": "حذف النجاح",
    "ConfirmSuccess": "تأكيد النجاح",
    "SelectRecordError": "الرجاء تحديد سجل أولاً",
    "Button:Cancel": "يلغي",
    "Button:Save": "يحفظ",
    "Button:SaveGoDetail": "احفظ وانتقل إلى التفاصيل",
    "Enum:ProjectOrderStatus:1": "مسودة",
    "Enum:ProjectOrderStatus:2": "أكد",
    "Enum:ProjectOrderStatus:3": "ألغاء",
    "Enum:ServiceOrderStatus:1": "مسودة",
    "Enum:ServiceOrderStatus:2": "أكد",
    "Enum:ServiceOrderStatus:3": "ألغاء",
    "Enum:SalesOrderStatus:1": "مسودة",
    "Enum:SalesOrderStatus:2": "أكد",
    "Enum:SalesOrderStatus:3": "ألغاء",
    "Enum:SalesDeliveryStatus:1": "مسودة",
    "Enum:SalesDeliveryStatus:2": "أكد",
    "Enum:SalesDeliveryStatus:3": "عاد",
    "Enum:PurchaseOrderStatus:1": "مسودة",
    "Enum:PurchaseOrderStatus:2": "أكد",
    "Enum:PurchaseOrderStatus:3": "ألغاء",
    "Enum:PurchaseReceiptStatus:1": "مسودة",
    "Enum:PurchaseReceiptStatus:2": "أكد",
    "Enum:PurchaseReceiptStatus:3": "عاد",
    "Enum:TransferOrderStatus:1": "مسودة",
    "Enum:TransferOrderStatus:2": "أكد",
    "Enum:TransferOrderStatus:3": "عاد",
    "Enum:DeliveryOrderStatus:1": "مسودة",
    "Enum:DeliveryOrderStatus:2": "أكد",
    "Enum:DeliveryOrderStatus:3": "عاد",
```

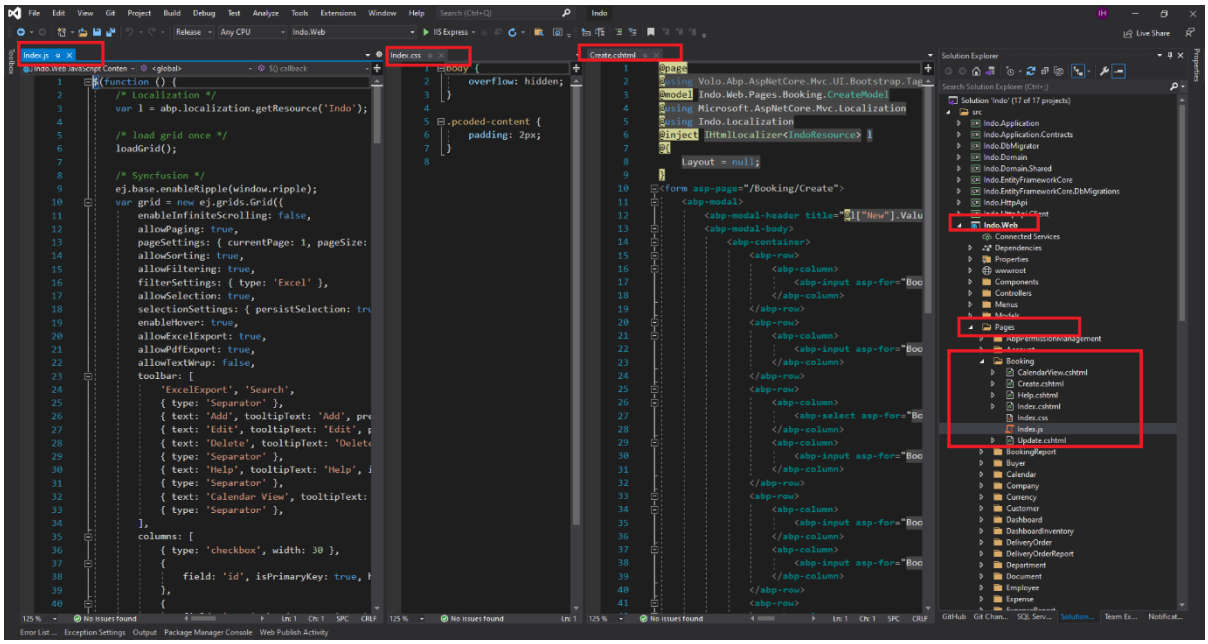
```
{
  "culture": "en",
  "texts": {
    "Menu:Home": "Home",
    "Welcome": "Welcome",
    "LongWelcomeMessage": "Welcome t
    "New": "New",
    "Update": "Update",
    "CreateSuccess": "Create Success
    "UpdateSuccess": "Update Success
    "DeleteSuccess": "Delete Success
    "ConfirmSuccess": "Confirm Succe
    "SelectRecordError": "Please sel
    "Button:Cancel": "Cancel",
    "Button:Save": "Save",
    "Button:SaveGoDetail": "Save Go
    "Enum:ProjectOrderStatus:1": "Dr
    "Enum:ProjectOrderStatus:2": "Co
    "Enum:ProjectOrderStatus:3": "Ca
    "Enum:ServiceOrderStatus:1": "Dr
    "Enum:ServiceOrderStatus:2": "Co
    "Enum:ServiceOrderStatus:3": "Ca
    "Enum:SalesOrderStatus:1": "Draf
    "Enum:SalesOrderStatus:2": "Conf
    "Enum:SalesOrderStatus:3": "Canc
    "Enum:SalesDeliveryStatus:1": "D
    "Enum:SalesDeliveryStatus:2": "C
    "Enum:SalesDeliveryStatus:3": "R
    "Enum:PurchaseOrderStatus:1": "D
    "Enum:PurchaseOrderStatus:2": "C
    "Enum:PurchaseOrderStatus:3": "C
    "Enum:PurchaseReceiptStatus:1":
    "Enum:PurchaseReceiptStatus:2":
    "Enum:PurchaseReceiptStatus:3":
    "Enum:TransferOrderStatus:1": "D
    "Enum:TransferOrderStatus:2": "C
    "Enum:TransferOrderStatus:3": "R
    "Enum:DeliveryOrderStatus:1": "D
    "Enum:DeliveryOrderStatus:2": "C
    "Enum:DeliveryOrderStatus:3": "R
```

```
{
  "culture": "zh-Hans",
  "texts": {
    "Menu:Home": "家",
    "Welcome": "欢迎",
    "LongWelcomeMessage": "欢迎申请",
    "New": "新的",
    "Update": "更新",
    "CreateSuccess": "创造成功",
    "UpdateSuccess": "更新成功",
    "DeleteSuccess": "删除成功",
    "ConfirmSuccess": "确认成功",
    "SelectRecordError": "请先选择一
    "Button:Cancel": "取消",
    "Button:Save": "节省",
    "Button:SaveGoDetail": "保存去细
    "Enum:ProjectOrderStatus:1": "草
    "Enum:ProjectOrderStatus:2": "确
    "Enum:ProjectOrderStatus:3": "取
    "Enum:ServiceOrderStatus:1": "草
    "Enum:ServiceOrderStatus:2": "确
    "Enum:ServiceOrderStatus:3": "取
    "Enum:SalesOrderStatus:1": "草案
    "Enum:SalesOrderStatus:2": "确认
    "Enum:SalesOrderStatus:3": "取消
    "Enum:SalesDeliveryStatus:1": "草
    "Enum:SalesDeliveryStatus:2": "确
    "Enum:SalesDeliveryStatus:3": "回
    "Enum:PurchaseOrderStatus:1": "草
    "Enum:PurchaseOrderStatus:2": "确
    "Enum:PurchaseOrderStatus:3": "取
    "Enum:PurchaseReceiptStatus:1":
    "Enum:PurchaseReceiptStatus:2":
    "Enum:PurchaseReceiptStatus:3":
    "Enum:TransferOrderStatus:1": "草
    "Enum:TransferOrderStatus:2": "确
    "Enum:TransferOrderStatus:3": "回
    "Enum:DeliveryOrderStatus:1": "草
    "Enum:DeliveryOrderStatus:2": "确
    "Enum:DeliveryOrderStatus:3": "回
```

User Interface (UI)

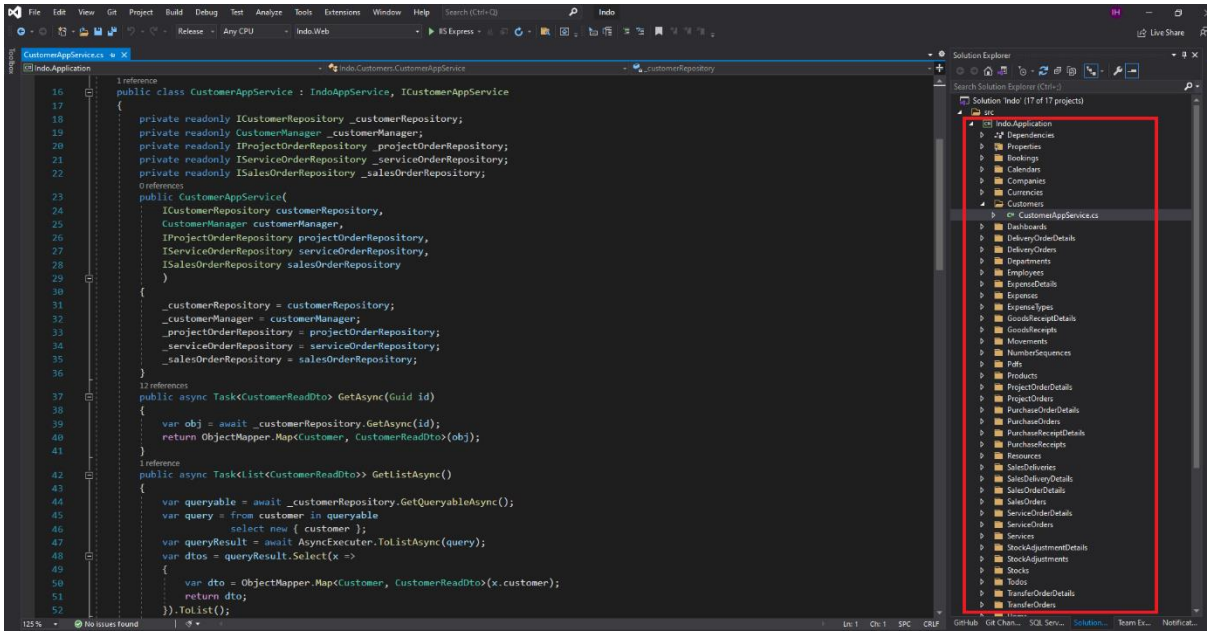
The UI following Microsoft official ASP.NET Core Razor Pages approach, which is the desired and recommended approach by Microsoft compare to classic MVC approach.

All the UI resides under the Pages folder of Web project and grouped by folders which contains CSHTML, CSS and JavaScript.



REST API

All the method at Application project, will be auto magically converted into REST API



```
public class CustomerAppService : IndoAppService, ICustomerAppService
{
    private readonly ICustomerRepository _customerRepository;
    private readonly CustomerManager _customerManager;
    private readonly IProjectOrderRepository _projectOrderRepository;
    private readonly IServiceOrderRepository _serviceOrderRepository;
    private readonly ISalesOrderRepository _salesOrderRepository;
    public CustomerAppService(
        ICustomerRepository customerRepository,
        CustomerManager customerManager,
        IProjectOrderRepository projectOrderRepository,
        IServiceOrderRepository serviceOrderRepository,
        ISalesOrderRepository salesOrderRepository
        )
    {
        _customerRepository = customerRepository;
        _customerManager = customerManager;
        _projectOrderRepository = projectOrderRepository;
        _serviceOrderRepository = serviceOrderRepository;
        _salesOrderRepository = salesOrderRepository;
    }
    public async Task<CustomerReadDto> GetAsync(Guid id)
    {
        var obj = await _customerRepository.GetAsync(id);
        return ObjectMapper.Map<Customer, CustomerReadDto>(obj);
    }
    public async Task<List<CustomerReadDto>> GetListAsync()
    {
        var queryable = await _customerRepository.GetQueryableAsync();
        var query = from customer in queryable
                    select new { customer };
        var queryResult = await AsyncExecuter.ToListAsync(query);
        var dtos = queryResult.Select(x =>
        {
            var dto = ObjectMapper.Map<Customer, CustomerReadDto>(x.customer);
            return dto;
        }).ToList();
```

Check and test using swagger (already pre configured, no additional steps required)

Using URL: https://localhost:xxx/swagger