





We are going to create a solution which has 5 class library projects, Api & Contracts projects are comes under Presentation rest have project names as per the layer

1. **dotnet new sln -o CleanDinner**
2. Switch to CleanDinner path
3. dotnet new **webapi** -o CleanDinner.Api -controllers [by default it gives minimal api, if you want controllers then we need add -controllers
4. dotnet new **classlib** -o CleanDinner.Contracts
5. dotnet new **classlib** -o CleanDinner.Infrastructure
6. dotnet new **classlib** -o CleanDinner.Application
7. dotnet new **classlib** -o CleanDinner.Domain

If you try to build it throws exception/error because we need to add all these project to the solution [Step 1]

Run following command to add projects to the solution

**dotnet sln add (ls -r \*\*\\*.csproj) ,n**ow we can build the solution

Start adding the dependencies into the projects

1. dotnet add .\CleanDinner.Api\ reference .\CleanDinner.Contracts\ .\CleanDinner.Application .\CleanDinner.Infrastructure
2. dotnet add .\CleanDinner.Infrastructure\ reference .\CleanDinner.Application
3. dotnet add .\CleanDinner.Application\ reference .\CleanDinner.Domain\

To run the project in vs code **dotnet run --project .\CleanDinner.Api\** .

dotnet add .\CleanDinner.Application\ package Microsoft.Extensions.DependencyInjection.Abstractions

**If you see any error while building regarding copy dll’s issue run the following commands**

Get-Process | Where-Object {$\_.Id -eq 13580} | Stop-Process

dotnet clean

dotnet build

1.[API/Presentation]Controller Has two post methods one is login and another is register

2. IAuthenticationService will be used in controller, this interface will be part of Application

\* As Application is not dependent on Presentation, we are not going to use the models

which are defined in presentation, instead we use normal types not model

***Add Services to their respective projects, ex create following method in Application***

public static class DependencyInjection

{

    public static IServiceCollection AddApplication(this IServiceCollection services)

    {

        services.AddScoped<IAuthenticationService, AuthenticationService>();

        return services;

    }

}

And add this in Program.cs file

builder.Services.AddApplication();