**Inexis - Clean Architecture API Template – Getting Started**

Introduction

The template here provides a minimal startup solution for any API based project.

It is based on Clean Architecture pattern and to support DDD pattern.

The version of dotnet may be newer at the time one may use it, so please upgrade to the latest version of dotnet and all NuGet packages in the solution. Once NuGet packages have been upgraded, the build of the application will probably fail in case the APIs in the NuGet packages have changed, so fix it by reading the documentation of each failing NuGet package.

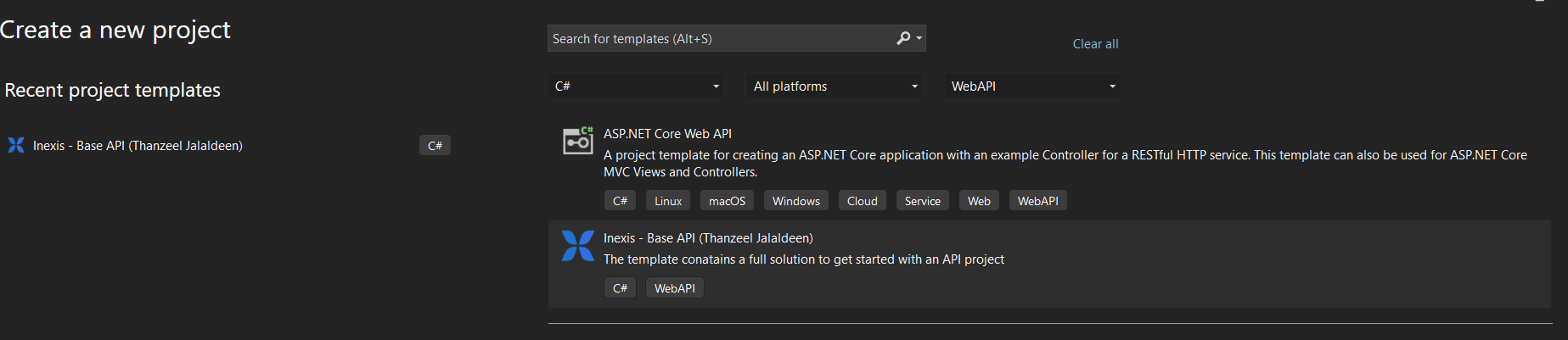
We will also try to keep this template up to date\*.

Installing the API template

1. Clone the repository from <https://github.com/tnlthanzeel/Inx.Project.Templates.git>
2. Navigate to the following folder ‘templates\Inexis.Clean.Architecture.Api.Template’
3. Open any command line tool at the above folder and type “dotnet new install . ”



1. once installed, open visual studio and search for “Inexis – Base API” to create a new project using the template installed.



* you can also create a new project using the dotnet cli by typing

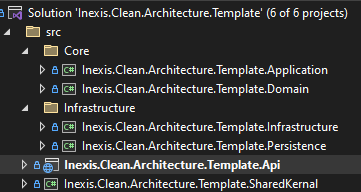
“dotnet new inx-api -o <application name>”. Just provide the name of the solution.



It consists of projects such as:

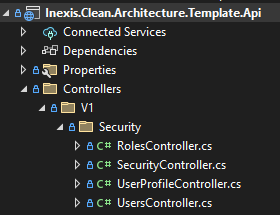
1. Application
2. Domain
3. Infrastructure
4. Persistence
5. API
6. Shared kernel

The projects are categorized by folder structure as below:



The API project contains controllers for

1. Role management,
2. User profile management
3. User management
4. Authentication management



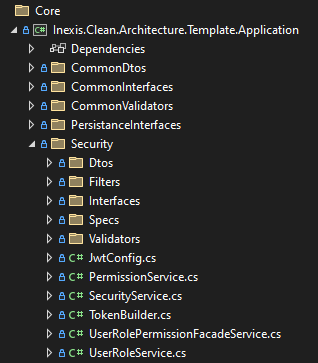
Shared kernel Project

This library project includes commonly used extensions, helpers, constants, interfaces, and models that are intended for application-wide use.

Feel free to add your common use cases to this project if required.

Core – Application Project

Each feature or module of the application is categorized as a folder to maintain vertical slice architecture.



The Core Application project contains the business, validation, services, specifications, filters and DTOs required by the feature.

Core – Domain Project

The core Domain project is responsible for encapsulating the domain knowledge of the domain entities.

Infrastructure Project

This project is responsible for handling cross-cutting concerns and non-domain related activities such as sending emails.

Persistence Project

This project is, as its name suggests, to do with the persistence of data to the data store. It can be MSSQL, Cosmos, or any other.

By default, we have added MSSQL using the Entity Framework Core ORM. You can change it as you please.