Table of Contents

[High Level API Diagram 2](#_Toc107177555)

[Basic Setup 2](#_Toc107177556)

[Repositories 2](#_Toc107177557)

[Interfaces 2](#_Toc107177558)

[Repositories 2](#_Toc107177559)

[Business Logic 2](#_Toc107177560)

[Interfaces 2](#_Toc107177561)

[BusinessLogic 3](#_Toc107177562)

[Dependency Injection 3](#_Toc107177563)

[Controllers 3](#_Toc107177564)

# High Level API Diagram

Diagram

Description automatically generated

# Basic Setup

First follow “[0004 Basic Setup - EntityFrameWork Setup Code First - DBContext - Sqlite.docx](0004%20Basic%20Setup%20-%20EntityFrameWork%20Setup%20Code%20First%20-%20DBContext%20-%20Sqlite.docx)” to setup basics.

# Repositories

## Interfaces

1. IUsersRepository

## Repositories

1. UsersRepository and implement IUsersRepository. Inject DataContext

# Business Logic

## Interfaces

1. IUsersBusinessLogic

## BusinessLogic

1. UsersBusinessLogic and implement IUsersBusinessLogic. Inject UsersRepository
   1. It has two methods and these currently return entity to the controller. This code is commented and now using DTO.

# Dependency Injection

Check document “0004 Basic Setup - EntityFrameWork Setup Code First - DBContext - Sqlite.docx”, section RegisterRepos Extension Method – Dependency Injection

# Controllers

Following controllers will be created. Check the source code for more details.

1. UsersController inject UsersBusinessLogic
   1. There are two methods that were created to pull the Users and User By Id initially. Both of these have commented since these will return a DTO and not an entity.