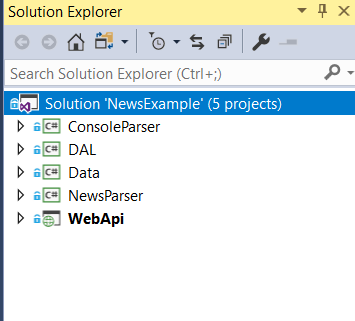
1. **Описание структуры проекта**



В солюшине 5 проектов:

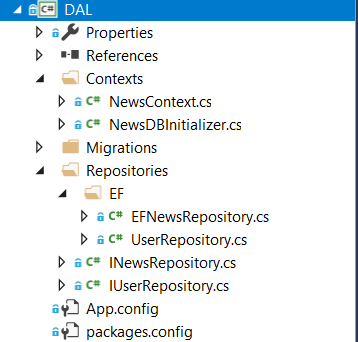
1. ConsoleParser – консольное приложение для запуска парсинга новостей и сохранения их в базу.
2. DAL – доступ к данным
3. Data – модель данных
4. NewsParser – бизнес логика парсера
5. WebApi – rest api
6. **DAL**

В данном проекте содержаться интерфейсы и классы для доступа к данным.

База данных в localdb data source=(localdb)\MSSQLLocalDB;Initial Catalog=newsdatabase

Для разработки базы выбрал подход Code First, т.к. модель данных простая, и бизнес-логика в базе не предусматривается, а также т.к. я единственный разработчик данного проекта. Также плюс легкое развертывание с помощью миграций.

Для доступа к данным был выбран EntityFramework в первую очередь из-за удобства работы с подходом Сode First.

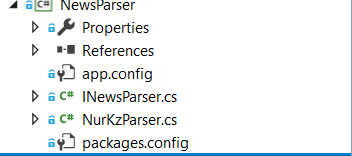


INewsRepository – доступ к базе новостей

IUserRepository – доступ к базе пользователей

Каталог Repositories/EF содержит классы реализации данных интерфейсов на основе EntityFramework.

1. **NewsParser**



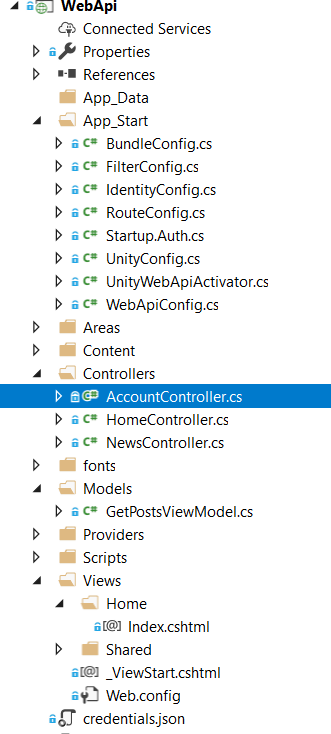
Содержит интерфейс INewsParser содержащий объявление метода GetNewsAsync для извлечения новостей.

NurKzParser – реализация интерфейса INewsParser для извлечения новостей с сайта nur.kz

В качестве HTTP клиента использован RestSharp, т.к. реализация предполагает извлечение данных в многопоточном режиме, а в стандартном HttpClient’е имеется «утечка соединений», что в многопоточном режиме приведет к переполнению открытых соединений.

Для парсинга HTML использована библиотека AngleSharp, в ней имеются методы для извлечения данных с помощью селекторов как в JavaScript.

1. **WebApi**



Стандартный проект WebAPI 2

Методы извлечения новостей находятся в контроллере NewsController

Для логирования использована библиотека log4net из-за очень широких возможностей библиотек и встроенной поддержки DI

В качестве DI используется Unity из-за простоты конфигурации для такого небольшого проекта.

Аутентификация на основе Bearer токена, доступ по Api ключу и IP адресу клиента.