**Backend**

**Уровень Aplication**

Набор объектов типа запрос – ответ. Есть возможность присоединять дополнительные поведения перед и после каждого запроса – это могут быть логирование, валидация, кеширование, авторизация и т.д.

**Commands** – Каждое действие, предназначенное для изменения состояния системы путем создания, обновления или удаления информации, должно бть представлено классом, реализующим либо **ICommand**, либо **ICommand<TResult>**

**Queries –** Классы**,** реализующие **IQuery<TResult>**, представлют собой операции чтения данных, которые не должны изменять состояние системы, иначе они должны рассматриваться как команды.

Mediator – Отделяет того, кто вызывает какое-то действие от того, кто исполняет его. Предоставляет общие методы для отправки команд, запросов, отвечает за поиск правильного обработчика и представлен интерфейсом **IMediator**

**Unit –** Тип означающий пустой ответ.

**Папка Common**

Содержит инфраструктуру – Все что относится к поведениям, исключениям, маппингу и т.д.

**Проект Presentation**

appsettings.json – Предназначен для хранения настроек (сроку подключения к БД)

CORS расшифровывается как **Cross-origin resourse sharing**, то есть совместное использование между разными источниками – технология современных браузеров, которая позволяет предоставить веб-страницам доступ к ресурсам другого домена.

Источник **origin** при этом состоит из

* **Протокол (например http)**
* **Хост (например example.com)**
* **Порт (например 8000)**

Валидация данных (Requiredб Range)

**Midleware –** фрагмент кода в конвейере приложения, используемый для обработки запросов и ответов (нужно добавлять в правильном порядке)

