**Описание**

* В hosts для arch.homework прописан IP, полученный через minikube ip.
* Установка приложения:

minikube addons enable ingress

kubectl create namespace otusapp

kubectl config set-context --current --namespace=otusapp

helm install app app-chart

* Идемпотентным реализован метод “CreateOrder” сервиса “Order”.
* Используемый паттерн – “ключ идемпотентности”. Предполагается, при создании заказа клиент генерирует ключ идемпотентности (GUID) и передаёт его в заголовке “IdempotencyKey”. При обработке запроса в методе “CreateOrder” (помимо проверки наличия и валидности значения заголовка) сначала проверяется, содержится ли уже в БД переданный в запросе ключ идемпотентности. Если имеется – метод возвращает HTTP ответ “Conflict”, иначе переданный ключ сохраняется в БД, затем создаётся заказ и возвращается HTTP ответ Created.
* Это предотвращает, двойное создание заказа, например, в следующих случаях:
  + если в клиентском приложении кнопка «Создать заказ» была нажата повторно;
  + если клиент переотправил запрос (например, при ожидании ответа на первоначальный запрос возник таймаут либо ошибка, но заказ, на самом деле, был создан).
* Для хранения ключей идемпотентности используется отдельная таблица (то есть она может быть использована для операций и сущностей любого типа). Для каждого ключа идемпотентности также сохраняется дата его создания (дата выполнения операции, в которой он был использован) – её потенциально можно использовать для удаления старых ключей (с целью предотвращения разрастания таблицы).
* Сохранение ключа идемпотентности и создание новой сущности (в данном случае - заказ) лучше производить в одной транзакции, чтобы ошибка при создании заказа (не связанная с ключом идемпотентности) не приводила к последующим отказам при повторных попытках создать заказ (уже по причине наличия используемого ключа в БД).