Проектная работа по курсу «Архитектор программного обеспечения 2019-11» на тему: «Система автоматизации складского предприятия»

Автор: Киселёв М.В.

# Аннотация

Данное приложение является учебным и разрабатывается в рамках курса Software Architect 2019-11 образовательной компании [OTUS](https://otus.ru/). Предполагается использовать результаты разработки данного приложения для построения реального приложения автоматизации складского хозяйства группы компаний [Легион](https://www.legion.ru/).

# Краткое описание

Система управления складским предприятием. Предприятие оказывает услуги по хранению товаров клиентов. Приёмка и отгрузка товаров клиентам осуществляется по заявкам клиентов, пользующихся услугами складского предприятия. Клиенты складского предприятия подают заявку на приёмку/получения товаров в виде списка единиц хранения (SKU). Сотрудник складского предприятия подтверждает принятие заявки, с указанием даты и времени приёмки/отгрузки, либо отклоняет, с указанием причины. Заявка клиента представляет собой список мест хранения (коробка, поддон и пр.) для каждого из которых определён перечень SKU (с количеством). Идентификация SKU осуществляется по штрих-коду. В системе должна быть описана структура мест хранения (адресное хранение). Сотрудники предприятия используют программные средства поддержки адресного хранения для подбора и размещения товаров клиентов. Любые перемещения товаров в рамках складского предприятия осуществляются только на основе данных предоставляемых системой - что и где (SKU и в какой ячейке) взять или положить для выполнения заявок клиентов. Дополнительно, должны быть разработаны следующие отчёты: отчёт об инвентаризации как для клиентов, так и для сотрудников предприятия; отчёт о заполнении складских мощностей; отчёт о движении SKU за период.

# Используемые технологии

Предполагается использовать следующие технологии: Microsoft® SQL Server™, Node.js, SPA – Angular®, брокер сообщений RabbitMQ®. Выбор технологий обусловлен наличием знаний и опыта у группы разработчиков и требованием руководства (выбор СУРБД), участвующих в проекте. СУРБД Microsoft® SQL Server™ принята как стандартная в группе компаний Легион. Платформа разработки WEB-приложений Angular® выбрана из соображений возможности обучения новым технологиям и инструментам.

# Термины

## Единица хранения

Единица хранения (SKU – stock keeping unit) – это минимальная значимая единица товара, единичная упаковка товара. Единица хранения не может включать в себя другие единицы хранения, т.е. является атомарной. Каждая единица хранения идентифицируется по уникальному штрих-коду (соответствует артикулу товара по производителю, не серийный номер, а именно артикул) на своей упаковке. Для каждой единицы хранения указывается название (артикул) – для визуальной идентификации, вес, объём и площадь (стороны) основания.

## Поддон

Поддон (pallet) – совокупность SKU, физически размещённых на поддоне. Сущность поддон используется при приёмке и отгрузке товаров по заявке клиент. При приёмке товаров, в заявке, указывается номер поддона (сквозной уникальный номер для данного контрагента), состав поддона – список SKU и их (SKU) количество. Допускается приёмка отдельных SKU, без паллетирования.

## Структурная единица описания хранилища

Структурная единица описания хранилища (единица хранилища) – узел в иерархическом описании структуры хранилища. Структура хранилища представляет собой дерево. Например, в здании находится помещение, в помещении размещены стеллажи, у стеллажей имеются полки, на полках размещены ячейки. Каждая структурная единица описывается уникальным идентификатором, именем, например, стеллаж А, путём к узлу структурной единицы в рамках дерева хранилища, например, для единицы *«Помещение А\Стеллаж С1\Ячейка Я22»* - путь - *«А\С1\Я22»*. Путь структурной единицы хранилища уникален (на уровне базы данных). Каждая структурная единица имеет соответствующий тип: здание, помещение, комната, шкаф, стеллаж, полка, ячейка. Для каждого типа определён атрибут *Capacity Range*, используемый процедурой валидации при формировании описания структуры хранилища (в ячейку нельзя поместить стеллаж и т.д.). Данный атрибут – число, в элемент с большим числом нельзя вставить элемент с меньшим числом.

## Состояние хранилища

Состояние хранилища (единицы хранилища) – запись содержащая идентификатор единицы хранилища и SKU, находящейся в данной единице хранилища, с указанием количества. SKU не обязательно хранятся в структурных единицах хранилища, являющихся листьями дерева хранилища. Данное поведение определяется настройками приложения – для всех узлов или персональными настройками узлов. Выбор стратегии для реализации требует дополнительного исследования.

# Архитектура приложения

В соответствии с требованиями, предъявляемыми к курсовому проекту, архитектура курсового проекта должна быть реализована на основе микросервисной парадигмы. Диаграмма микросервисной архитектуры приложения приведена здесь. Описание API приложения приведено [здесь](https://app.swaggerhub.com/apis/Kiselb/APW/1.0.0-oas3#/info) (Swagger). Исходный код приложение размещён [здесь](https://github.com/Kiselb/Software-Architect-2019-11) (GitHub).

## Сервис операций по изменению состояния хранилищ

Основу приложения определяет сервис операций по изменению состояния хранилищ. База данных данного сервиса содержит описания всех операций, которые привели к изменению состояния хранилищ. Если последовательно применить все операции к начальному состоянию хранилищ в хронологическом порядке, то будет получено текущее состояние хранилищ. Т.е. сервис операций является источником истины в системе.

Сервис операций по изменению состояния хранилищ взаимодействует со следующими сервисами:

* *сервис состояния хранилищ предприятия*. Данный сервис соответствует контексту интегрального состояния хранилищ предприятия: обеспечивает актуальное состояние счётчиков в разрезе *Хранилище-Товар* (ответ на вопрос: что где находится, и в каком количестве). *Данный сервис является потребителем данных (событий) сервиса операций*. Сервис поддерживает актуальными не только данные о количестве того или иного товара в структурной единице хранилища, но и данные об весе, объёме в разрезе всех элементов в иерархии хранилища. Сервис предоставляет данные, необходимые при поиске и размещении SKU по заявкам потребителей и сотрудников предприятия. Дополнительно, сервис обеспечивает данные о физическом состоянии хранилищ (физическая нагрузка).
* *Сервис структуры хранилищ предприятия*. Данный сервис предназначен для обеспечения актуального описания структуры хранилищ предприятия. Структура хранилищ необходима при поиске SKU при отгрузке, и при поиске свободного места при приёмке SKU. Т.е. структура хранилищ – это своего рода адресное бюро, позволяющее определить физическое местоположение единицы хранилища. При изменении структуры хранилища формируется список операций по изменению состояния хранилища, передаваемый в сервис операций по изменению состояния хранилища. *Т.е. данный сервис является источником данных (событий) для сервиса операций*;
* *Сервис отчётов*. Сервис отчётов обеспечивает получение отчётов как контрагентами компании, так и сотрудниками компании. Данные сервиса представляют собой сверх (как только возможно) денормализованные данные об операциях по управлению хранилищем (приёмка, отгрузка, перемещение SKU). Данные формируются на ежедневной основе, т.е. квантом данных является день (счета, предъявляемые клиентам основаны на данных за день – точность учёта составляет один день). *Данный сервис является потребителем данных (событий) сервиса операций*.
* *Сервис заявок на обслуживание*. Данные сервис предназначен для регистрации и обеспечения заявок клиентов и сотрудников предприятия. При размещении заявки формируется список операций по изменению состояния хранилищ, необходимых для исполнения заявок. *Данный сервис является источником данных (событий) для сервиса операций*.

Особо следует остановиться на взаимодействии сервиса операций и сервиса состояния хранилищ. Поскольку счётчики и операции должны быть согласованы в произвольный момент времени, то сервис операций и сервис состояний определяют единый контекст транзакции, рамках которого, операции регистрируются и соответственно изменяются счётчики.

Для обеспечения транзакционности взаимодействия между сервисами операций и состояния выбран шаблон *«Публикация событий»* и шаблон *«Опрашивающий издатель»*. Ниже приведена диаграмма, поясняющая реализацию взаимодействия.

**Сервис операций**

**ис опера**Таблица операций

Таблица блоков операций

**ций**

Таблица операций

Состояние с учётом неопубликованных блоков

Флаг публикации

Брокер сообщений

Опрашивающий издатель

**База данных**

Транзакция (ACID)

Таблица блоков операций

Идемпотентное событие. Полезная нагрузка: состояние SKU в разрезе хранилище-SKU-клиент только по данным блока

**Сервис состояния**

**База данных**

Транзакция (ACID)

Таблица состояния

На вход сервиса операций поступает список операций, которые необходимо выполнить над хранилищем, для выполнения заявки. Различают следующие типы операций (связаны с состоянием заявки):

* *Блокирование*. Блокируют SKU в разрезе хранилище-SKU-клиент в соответствии с заявкой. Промежуточное состояние хранилища на момент выполнения заявки. Блокирование применяется только при отгрузке SKU при изменении состояния заявки с *Approved* на *In Progress*. Блокирование необходимо для обработки конкурирующих (в разрезе хранилище-SKU-клиент) заявок;
* *Выполнение*. Списание или начисление количеств SKU в разрезе хранилище-SKU-клиент в соответствии с заявкой. Операции данного типа порождаются при изменении заявки из состояния In Progress в состояние Completed.

Для сокращения числа операций, хранящихся в базе данных сервиса операций, планируется использовать шаблон *«Снимок»*. Для реализации данного шаблона вводится тип операции *«Состояние»*.

Брокеру сообщений предъявляются следующие требования: строгая последовательность сообщений и соответствие стратегии «доставка сообщения как минимум один раз».

## Сервис заявок на обслуживание

Сервис заявок на обслуживание предназначен для регистрации и управления выполнением заявок клиентов на приёмку, отгрузку принадлежащим им (клиентам) SKU. Диаграмма состояний заявки представлена ниже.

Accepted (Принята) 🡪 Approved (Подтверждена)🡪In Progress (Выполняется)🡪Completed (Завершена) 🡪 Archived (Перемещена в Архив)

Accepted (Принята) 🡪 Rejected (Отклонена) 🡪 Archived (Перемещена в Архив)

Approved (Подтверждена)🡪 Canceled (Отменена) 🡪 Archived (Перемещена в Архив)

In Progress (Выполняется)🡪 Canceled (Отменена) 🡪 Archived (Перемещена в Архив)

Начальное состояние – Accepted (Принята)

Конечное состояние – Archived (Перемещена в Архив)

Данный сервис взаимодействует со следующими сервисами:

* *Сервис SKU*. Данный сервис заполняет ключевые атрибуты SKU, включённых в заявку: заполняет уникальный идентификатор SKU по штрих-коду SKU и идентификатору клиента. Если SKU, указанный в заявке отсутствует в базе данных сервиса SKU, то сервис регистрирует новый SKU. Сервис возвращает заявку, обогащённую идентификаторами SKU. Для отслеживания конкретного запроса используется идентификатор запроса, назначаемый сервисом заявок при отправке запроса сервису SKU.
* *Сервис описаний клиентов (мини CRM)*. Данный сервис разрешает/запрещает приём заявки на выполнение, основываясь на текущем состоянии взаимоотношений с клиентом. Идентификация клиента осуществляется по ключу (внешнему ключу сервиса заявок).
* *Сервис состояния хранилищ*. На основе состояния хранилищ определяется возможность выполнения заявки в плане доступности того или иного SKU в количественном выражении. Заявка выполняется только целиком: если количества на складе достаточно для выполнения заявки по всем SKU заявки, то заявка принимается на выполнение. Так осуществляется первичный контроль возможности выполнения заявки. Основной контроль осуществляется непосредственно при размещении заявки – порождения операций по исполнению заявки. При переходе заявки в состояние *In Progress*, на основе данных, задаваемых пользователем, об местоположении принимаемых/отгружаемых SKU, сервис заявок порождает список операций по изменению состояния хранилищ. При размещении операций на отгрузку осуществляется контроль на возможность выполнения операций по количеству в разрезе *Хранилище-SKU-Клиент*.
* *Сервис извещений*. Позволяет информировать клиентов об изменении статуса заявки. Сервис извещений в свою очередь, взаимодействует с сервисами отправки сообщений по e-mail и отправки PUSH-уведомлений.
* *Сервис структуры хранилищ*. Данный сервис предоставляет данные о структуре хранилищ, необходимые для размещения заявки пользователем системы. Данные о структуре хранилища позволяют пользователю указать системе что откуда следует забрать или что куда положить.

## Сервис структуры хранилищ

Сервис структуры хранилищ предприятия предназначен для управления физической структурой мест хранения предприятия. Структура мест хранения представляет собой иерархию хранилищ. Например, *помещение🡪стеллаж🡪полка🡪ячейка*. Хранение (привязка) SKU допустима только для узлов иерархии, являющихся листьями.

Сервис структуры хранилищ взаимодействует с сервисом операций и сервисом состояния. Данный сервис порождает список операций при перемещении SKU между листьями без физического перемещения. Например, при слиянии двух соседних ячеек. Для физического перемещения используется заявка с типом *«Перемещение»*.

Дополнительно сервис структуры предоставляет функциональность по автоматическому формированию плана выполнения заявки (что куда следует положить или что откуда взять). Для этого используются данные, предоставляемые сервисом состояния хранилищ. Следует отметить, что формирования плана осуществляется строго последовательно: параллельное формирование планов различных заявок запрещено.

## Сервис описаний клиентов (мини-CRM)

Сервис описаний клиентов содержит полную информацию о клиентах и отвечает за идентификацию клиентов в системе:

* Идентификация. Глобальный идентификатор, название и прочие идентификаторы, используемые в коммерческой деятельности;
* Состояние. Состояние определяет, допустимо ли выполнение заявок от данного клиента;
* Контактная информация.
* Данные общего характера.

Агрегат описания клиента связан с агрегатом заявка, агрегатом состояния хранилища, агрегатом пользователь по внешнему ключу идентификатор клиента.

## Сервис пользователей

Сервис пользователей отвечает за аутентификацию и авторизацию пользователей. База данных сервиса содержит описания пользователей, перечень разрешений для конкретного пользователя и пароли.

Каждый пользователь, кроме внутренних, привязан к конкретному клиенту. У одного клиента может быть несколько пользователей.

Аутентификация пользователей осуществляется с использованием JWT.

# Диаграммы бизнес-транзакций

## Регистрация заявки

Сервис Клиентов

Сервис SKU

Сервис заявок

Загрузка заявки

XLS, XML, JSON

Заявка JSON

Заявка JSON + Параметры Клиента

Заявка JSON + Параметры Клиента

Заявка JSON + Параметры Клиента + SKU UID

Заявка

зарегистрирована

## Размещение заявки и выполнение (отгрузка)

Сервис состояния

Сервис состояния

Сервис структуры

Сервис операций

Сервис заявок

Сервис состояния

Заявка

OK

Заявка

Запрос состояния

Состояние по заявке

Список операций (блокирование)

План

OK

Список операций (списание)

OK