



РУКОВОДСТВО ИНТЕГРАТОРА

РЕАЛИЗАЦИЯ ФУНКЦИЙ ИНТЕГРАЦИИ С СЕРВИСОМ HEADO

ОБЩАЯ ИНФОРМАЦИЯ

Для реализации функций интеграции с сервисом HEADO лучшим решением будет разработка модуля интеграции с составе комплекса программного обеспечения торговой системы.

Настоящая инструкция адресована разработчику программного обеспечения модуля интеграции и включает описание взаимодействия торговой системы с сервисом HEADO, доступные функции, требования к инфраструктуре торговой системы и реализации отдельных функций.

Решение по составу реализуемых модулем интеграции функций (объем интеграции) принимается разработчиком модуля интеграции исходя из возможностей торговой системы, наличия в торговой системе источников данных, удовлетворяющих предъявляемым требованиям в данном руководстве. Решение по объему интеграции должно быть согласовано с HEADO.

HEADO предоставляет разработчику тестовый аккаунт для отладки и тестирования программного обеспечения модуля интеграции.

По окончании разработки обязательно проведение совместного комплексного тестирования разработанного модуля интеграции.

Порядок разработки:

1. Определение объема и плана интеграции, согласование решений с HEADO;
2. Разработка и подписание технического задания с учетом специфики торговой системы и согласованного объема интеграции.
3. Заключение договора, включающего условия использования модуля интеграции и сроки.
4. Разработка модуля интеграции, консультации технических специалистов.
5. Комплексное совместное тестирование, решение о начале эксплуатации.

Наши контакты для поддержки разработчиков

сайт: support.heado.ru

электронная почта: support@heado.ru

телефон: 8 800 700 1556

СОДЕРЖАНИЕ

Терминология и сокращения HEADO	4
Схема целевого решения интеграции с сервисом HEADO	5
Объем интеграции	5
Требования к источникам данных	7
Перечень источников данных:	7
Требования к справочникам торговых точек и устройств (кассы, сервера и т.п.)	7
Требования к справочнику товаров и категорий товаров.	7
Требования к данным о продажах (чекам)	8
Требования к надежности потоков данных	9
Автоматизация добавления торговых точек и устройств в сервис HEADO	9
Описание процесса и требования к организации	9
Добавление торговой точки/устройства	11
Параметры метода setCashBox	11
Передача данных о продажах	12
Требования к организации процесса извлечения и передачи данных оперативных продаж:	12
Регистрация продажи	12
Параметры метода purchaseRegisterBatch	13
Регистрация возврата	14
Регистрация чеков коррекции	14
Загрузка исторических данных	14
Сверка данных сервиса HEADO с торговой системой	15
Отчеты для поиска ошибок загрузки данных	15
Загрузка и обновление дополнительного справочника товаров	15
Параметры методов для загрузки доп. справочника	15
Отображение показателей на рабочем месте кассира (продавца)	16
Способ получения данных для отображения	16
Параметры метода reports	16
Установка и публикация плановых значений	17
Параметры метода storePlanSet	17
Параметры метода storePlanPublish	18
Сигнал открытия/закрытия торговой точки	18
Параметры метода registerEvent	19
Установка/обновление расписания работы торговых точек	20
Параметры метода storeSchedule	20
Обновление складских остатков	20

Параметры метода inventoryUpdateBatch	21
Установка/обновление расписания работы сотрудников	21
Параметры метода staffManning	22

1. Терминология и сокращения HEADO

Обозначение	Описание
ЛК HEADO	Личный кабинет HEADO доступный по адресу https://partner.heado.ru
Торговая система	Общее название программного обеспечения используемого на торговом предприятии. Как правило включает в себя несколько классов программ: Front office - кассовые ПО; Back office - ПО для анализа и учета произведенных продаж; ERP система - ПО для планирования ресурсами предприятия
Устройство	Закрепленный за торговой точкой виртуальный терминал, созданный в ЛК HEADO для идентификации устройств (кассы, сервера и т.п.) торгового предприятия и их взаимодействия с API HEADO (Device API , Management API). Терминал состоит из: Auth_Key , cashbox_id , extid (устройства) Существует 2 типа терминала:: <ul style="list-style-type: none"> - ККМ-сервис 2.0 (по умолчанию)- используется для взаимодействия с Device API и для установки ПО HEADO на рабочее место торгового предприятия для отображения показателей; - Административный терминал - используется для взаимодействия с Management API.
Справочник торговых точек	Справочник адресов торговых точек предприятия по классификаторам КЛАДР или ФИАС и содержащих их уникальный в пределах торговой сети идентификатор (store_ext_id)
Справочник устройств	Список устройств (кассы, сервера и т.п.) торгового предприятия с уникальным в пределах торговой точки идентификатором (extid устройства) для регистрации в ЛК HEADO
Auth_Key	Уникальный ключ авторизации устройства для взаимодействия с API HEADO (Device API , Management API) и установки ПО HEADO .
cashbox_id	Внутренний идентификатор устройства в ЛК HEADO
extid (устройства)	Уникальный идентификатор устройства (кассы, сервера и т.п.) в торговой системе предприятия, состоящий из латинских букв или цифр (до 100 символов).
store_ext_id	Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр (до 40 символов)
dAPI	Список методов для взаимодействия с Device API по URL <a href="https://api.heado.ru/device/<Auth_Key>">https://api.heado.ru/device/<Auth_Key> Тип терминала “ ККМ-сервис 2.0 ”

	Запросы применимы к торговой точке с доступом на уровне Auth Key устройства и торговой точки, к которой оно привязано
mAPI	Список методов для взаимодействия с Management API по URL <a href="https://api.heado.ru/management/<Auth_Key>">https://api.heado.ru/management/<Auth_Key> Тип терминала “ Административный терминал ” Запросы применимы для торговой сети в ЛК HEADO
OLTP	(англ. Online Transaction Processing), транзакционная система — обработка транзакций в реальном времени.
DWH	(англ. Data Warehouse) Хранилище данных торговой системы с данными о продажах (чеках)

2. Схема целевого решения интеграции с сервисом HEADO

Общая схема потоков данных целевого решения интеграции торговой системы с сервисом HEADO представлена на Рисунке 1.

Потоки данных реализуются путем JSON запроса к API HEADO ([Device API](#), [Management API](#))

Подробное описание методов API и их параметров можно найти по ссылкам, приведенным рядом с их упоминанием в документе. Приведенные в настоящем документе комментарии и указания имеют приоритет над таковыми в документации, размещенной по ссылкам.

Важно! Для взаимодействия с API HEADO на передающей стороне необходимо обеспечить доступность <https://api.heado.ru> по 80 и 443 портам

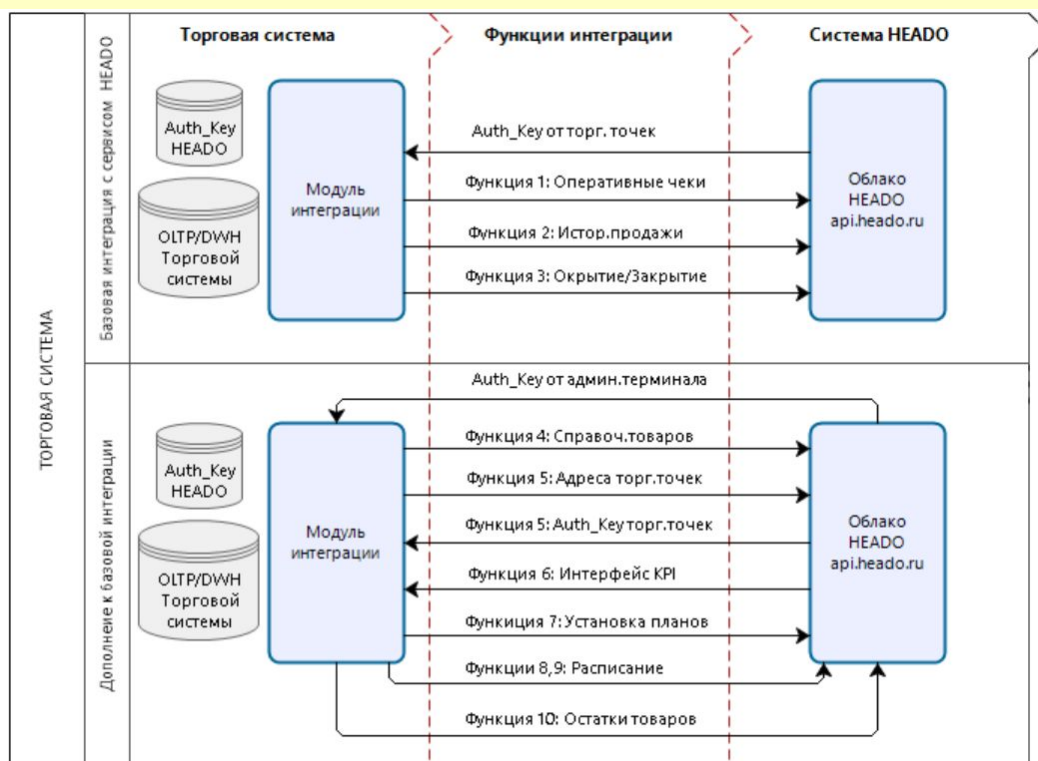


Рисунок 1

3. Объем интеграции

Для интеграции с сервисом HEADO необходимо реализовать базовый функционал в полном объеме. Дополнительный функционал реализуется при составлении и согласовании технического задания.

№ п.п.	Функция	Описание	Методы для реализации
Базовый функционал			
1.	Передача данных о продажах	Передаёт данные о продажах (чеки) в реальном времени.	dAPI purchaseRegisterBatch
2.	Загрузка исторических данных	Возможность выгружать данные о продажах (чеки) за указанный период	dAPI purchaseRegisterBatch
3.	Сигнал открытия/закрытия торговой точки	Позволяет отслеживать своевременное открытие и закрытие торговых точек.	dAPI registerEvent
Дополнительный функционал			
4.	Загрузка и обновление дополнительного справочника товаров	Реализация обязательна, если компонент торговой системы для которого разрабатывается модуль интеграции не содержит данные о наименовании и категориях (номенклатурных группах) товара. Функция должна быть реализована дополнительным модулем интеграции для компонента торговой системы, который содержит недостающие данные.	mAPI priceCategoryReset mAPI priceUpdateBatch
5.	Добавление торговых точек и устройств в сервисе HEADO	Позволяет добавлять новые торговые точки и устройства в системе HEADO (в автоматическом режиме при настройке подключения)	mAPI setCashbox
6.	Отображение показателей на рабочем месте кассира (продавца)	Графический интерфейс показателей торговой точки для торговых систем в которых рабочие места кассиров/продавцов реализованы не на Windows. Для Windows существует готовое ПО HEADO	dAPI reports
7.	Установка и публикация плановых значений	Функция является рекомендуемой, если в торговой системе есть функция планирования. Функция позволяет автоматизировать установку плановых значений показателей торговых точек и опубликовать их.	mAPI storePlanSet mAPI storePlanPublish
8.	Установка/обновление расписания работы торговых точек.	Функция является рекомендуемой при наличии в торговой системе данных о расписании работы торговых точек.	dAPI storeSchedule

9.	Установка/обновление расписания работы сотрудников	Функция является рекомендуемой при наличии в торговой системе данных о расписании работы сотрудников.	mAPI staffManning
10.	Обновление складских остатков	Позволяет автоматизированно отслеживать складские остатки не продающихся товаров на торговых точках.	dAPI inventoryUpdateBatch

4. Требования к источникам данных

4.1. Перечень источников данных:

№ п.п. Источник данных

1.	Справочник торговых точек и справочник устройств (кассы, сервера и т.п.)
2.	Справочник товаров и справочник категорий товаров
3.	Продажи (чеки), оперативные данные (OLTP)
4.	Продажи (чеки), исторические данные (DWH)

4.2. Требования к справочникам торговых точек и устройств (кассы, сервера и т.п.)

[Торговая система](#) должна использовать идентификаторы:

- [store_ext_id](#) - уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр (до 40 символов)
- [extid \(устройства\)](#) - уникальный идентификатор устройства (кассы, сервера и т.п.) в торговой системе предприятия, состоящий из латинских букв или цифр (до 100 символов).

Важно! Перечисленные идентификаторы используются сервисом HEADO в качестве ключей для привязки данных о продажах (чеках) к торговым точкам и устройствам.

Управление идентификаторами должно быть централизованным.

Чеки в [торговой системе](#) должны быть помечены уникальными идентификаторами торговой точки ([store_ext_id](#)) и [extid \(устройства\)](#).

Необходимо соответствие значений полей адресов торговых точек классификаторам КЛАДР или ФИАС.

4.3. Требования к справочнику товаров и категорий товаров.

Для корректного расчета показателей (KPI) справочник товаров [торговой системы](#) должен содержать иерархическую информацию о категориях товара (категории, подкатегории). Максимальная длина всех категорий товара не должна превышать 2000 символов и 200 символов каждой из них.

Имена категорий и подкатегорий, которые будут использоваться в фильтрах для настройки показателей (KPI) должны быть уникальными.

Важно! В случае, если при настройке фильтров по категориям товаров в показателях (KPI) будет выявлено дублирование имен категорий (подкатегорий) необходимо переименовать одну из конфликтующих категорий (подкатегорий) товаров.

Для реализации показателей (KPI) по спискам товаров справочник товаров, [торговая система](#) должна поддерживать дополнительные метки товаров и иметь средства управления ими.

В составе продаж (чеков) в HEADO должна передаваться информация о товарах:

- уникальный идентификатор товара;
- наименование товара;

- иерархия категорий товара и дополнительные текстовые метки (например название акции в которой участвует товар).

Наименования товаров, категории товара и метки следует передавать соответствующими полями и атрибутами чека при реализации функций

Важно! В случае, если [торговая система](#) не поддерживает в полном объеме необходимые для формирования показателей HEADO категории или метки товаров (или требуемая параметризация и актуальность не поддерживается по иным причинам), существует возможность загрузки в HEADO дополнительного справочника товаров, дополненного (относительно справочника [Торговой системы](#)) необходимыми параметрами товара (категории, метки). В таком случае требуется определить дополнительный источник информации для таких данных. Связность с чеками обеспечивается по уникальному идентификатору товара (должен совпадать с идентификатором товара получаемого из чека).

4.4. Требования к данным о продажах (чекам)

4.4.1. Для данных оперативных продаж следует использовать источники с минимальной задержкой относительно времени первичной регистрации продажи (чека, транзакции). Глубина данных не критична, но должна быть достаточной для гарантированной отправки данных в сервис HEADO с учетом возможных перерывов связи с сервисом. Лучшим решением будет использование источника в составе [OLTP](#) компонента [Торговой системы](#). В случае использования в качестве источника данных на промежуточных серверах (сервер торговой точки) необходимо убедиться, что задержка при доставке данных до такого источника не превышает 15 минут.

4.4.2. Для исторических данных продаж за указанный период, главным критерием является достаточная глубина архива (до 1 года). Важным показателем качества исторических данных являются однородность данных (отсутствие в загружаемом периоде влияния внешних факторов, влияющих на показатели торговой точки – смена формата, ремонт, временный простой). Источник данных должен обладать нагрузочной способностью, достаточной для обработки таких объемов данных без деградации для основных функций. Лучшим решением будет использование специализированных [DWH](#) компонент [торговой системы](#).

Важно! Источники данных должны быть одинаково представлены (тип данных, маски, форматы значений) в компонентах [Торговой системы](#) реализующих функционал [OLTP](#) и [DWH](#).

4.4.3. Необходимый набор параметров перечислен в разделе 7 «[Передача данных о продажах](#)».

4.4.4. Бизнес-логика Сервиса HEADO предполагает точные сведения о времени регистрации чеков. Системное время компонентов [торговой системы](#) должно быть синхронизировано по одному из доступных эталонов. Критичной для аналитических функций Сервиса HEADO будет несинхронность системных часов между компонентами [торговой системы](#) более чем на одну минуту.

Важно! HEADO определяет часовой пояс торговой точки по метке времени в формате [ISO-8601](#) (YYYY-MM-DDThh:mm:ss±hh:mm). Необходимо обязательно передавать метку времени регистрации чека с корректным смещением ±hh от [UTC](#), соответствующим временному поясу торговой точки.

4.4.5. Важным параметром, используемым в логике сервиса HEADO, является уникальный «Внешний идентификатор чека». Если в [торговой системе](#) уникальный внешний идентификатор чека отсутствует, то рекомендуем использовать в качестве его маски комбинацию из следующих идентификаторов:

- ИД торговой сети,
- ИД торговой точки ([store_ext_id](#)),
- ИД устройства ([extid \(устройства\)](#)),
- ИД кассовой смены,
- ИД номера чека в кассовой смене.

Важно! Способ формирования внешнего идентификатора (чека) должен гарантировать его уникальность хотя бы в пределах одной торговой сети и не превышать 40 символов.

5. Требования к надежности потоков данных

Надежность потоков данных должна обеспечиваться комплексом организационно-технических мероприятий отладки, поиска и исключения ошибок, а также технологическими процедурами резервирования.

5.1. При реализации потоков данных должны обеспечиваться:

- защита от потери данных в условиях возможных сбоев и отказов канала передачи данных, доступности API HEADO;
- диагностика сбоев и отказа канала передачи данных и доступности API HEADO;
- протоколирование событий при передаче данных с возможностью последующего анализа причин возникновения критических ситуаций.

5.2. В качестве мер по повышению надежности потоков данных должны быть предусмотрены:

- мониторинг доступности API HEADO;
- отложенная передача данных в случае недоступности API HEADO (прогрессивная задержка повторной передачи, например: 0, 3, 30, 60 сек, и т.д.);
- обработка ошибок, возвращаемых API HEADO (список ошибок описании [Device API](#));
- повторная отправка данных в случае наличия ошибок API HEADO;
- диагностика ошибок с помощью программных средств контроля.

6. Автоматизация добавления торговых точек и устройств в сервис HEADO

Автоматическое добавление торговых точек и [устройств](#) в сервис HEADO требуется для упрощения администрирования и для оперативного развертывания при большом количестве торговых точек и [устройств](#). Наибольший эффект автоматизации возможен при наличии в [торговой системе](#) функций централизованного управления торговой сетью.

6.1. Описание процесса и требования к организации

6.1.1. Перед добавлением торговых точек и [устройств торговой системы](#) в сервис HEADO необходимо выполнить следующие процедуры в [торговой системе](#):

- актуализировать [справочников торговых точек и устройств](#) в [торговой системе](#), выполнив проверку на предмет заполнения и соответствие содержимого полей: [store_ext_id](#), [extid \(устройства\)](#), адреса торговой точки по КЛАДР или ФИАС;
- синхронизировать распределенных справочников на всех узлах (компонентах) [торговой системы](#).

Важно! HEADO обеспечивает автоматическую регистрацию новых [устройств](#) торговой точки предприятия в [ЛК HEADO](#). Для этого при передаче данных продаж (чеков) должны передаваться оба параметра: [store_ext_id](#) и [extid \(устройства\)](#).

6.1.2. Для случая, когда [источником данных продаж \(чеков\)](#) является сервер торговой точки (передает продажи со всех устройств торговой точки) или сервер группы торговых точек (передает продажи со всех устройств всех торговых точек группы) достаточно зарегистрировать по одному [устройству](#) на каждую торговую точку. Полученные [Auth Key](#) при регистрации таких [устройств](#) в дальнейшем должны использоваться для взаимодействия с [API HEADO \(Device API, Management API\)](#). Рекомендуем присваивать [устройству](#) имя, ассоциируемое с его функцией, например “Сервер торговой точки”.

Важно! Для регистрации [устройства](#), предназначенного для передачи данных о продажах (сервер торговой точки или сервер группы торговых точек), достаточно при регистрации передать идентификатор торговой точки [store_ext_id](#). [extid \(устройства\)](#) можно не передавать, оставив "значение" пустым.

6.1.3. В случае, если [источником данных о продажах](#) является непосредственно [устройство](#), то регистрации подлежат все активные [устройства](#) торговых точек, выбранных оператором при настройке [торговой системы](#);

6.1.4. Зарегистрированные [устройства](#) сервисом HEADO группируются по идентификаторам торговых точек [store_ext_id](#);

Важно! При автоматизированной регистрации нескольких [устройств](#) одной торговой точки необходимо гарантированно обеспечить одинаковые значения всех параметров [устройств](#) (кроме [extid](#) устройства), относящихся к данной торговой точке.

6.1.5. Повторная регистрация [устройств](#) в сервисе HEADO должна быть исключена;

6.1.6. При успешной регистрации [устройства](#), сервис возвращает внутренний идентификатор устройства ([cashbox_id](#)) и ключ устройства ([Auth_Key](#));

6.1.7. Полученные в результате добавления [устройств](#) ключи ([Auth_Key](#)), необходимо передать средствами [торговой системы](#) на узлы (компоненты), с которых будет выполняться взаимодействие с сервисом HEADO (передача оперативных и исторических данных, загрузка справочника товаров, получение отчетов).

Важно! Удаление устройств и торговых точек не доступно через API, возможно только в личном кабинете HEADO с правами администратора.

На рисунке 2 отображена общая схема процесса добавления торговых точек/устройств в системе HEADO и получение [Auth_Key](#) от зарегистрированных [устройств](#) для дальнейшего взаимодействия с [dAPI](#) (продажи, сигналы и тд). Полученные [Auth_Key](#) также возможно использовать при установке [ПО HEADO](#) для отображения показателе по торговой точке.

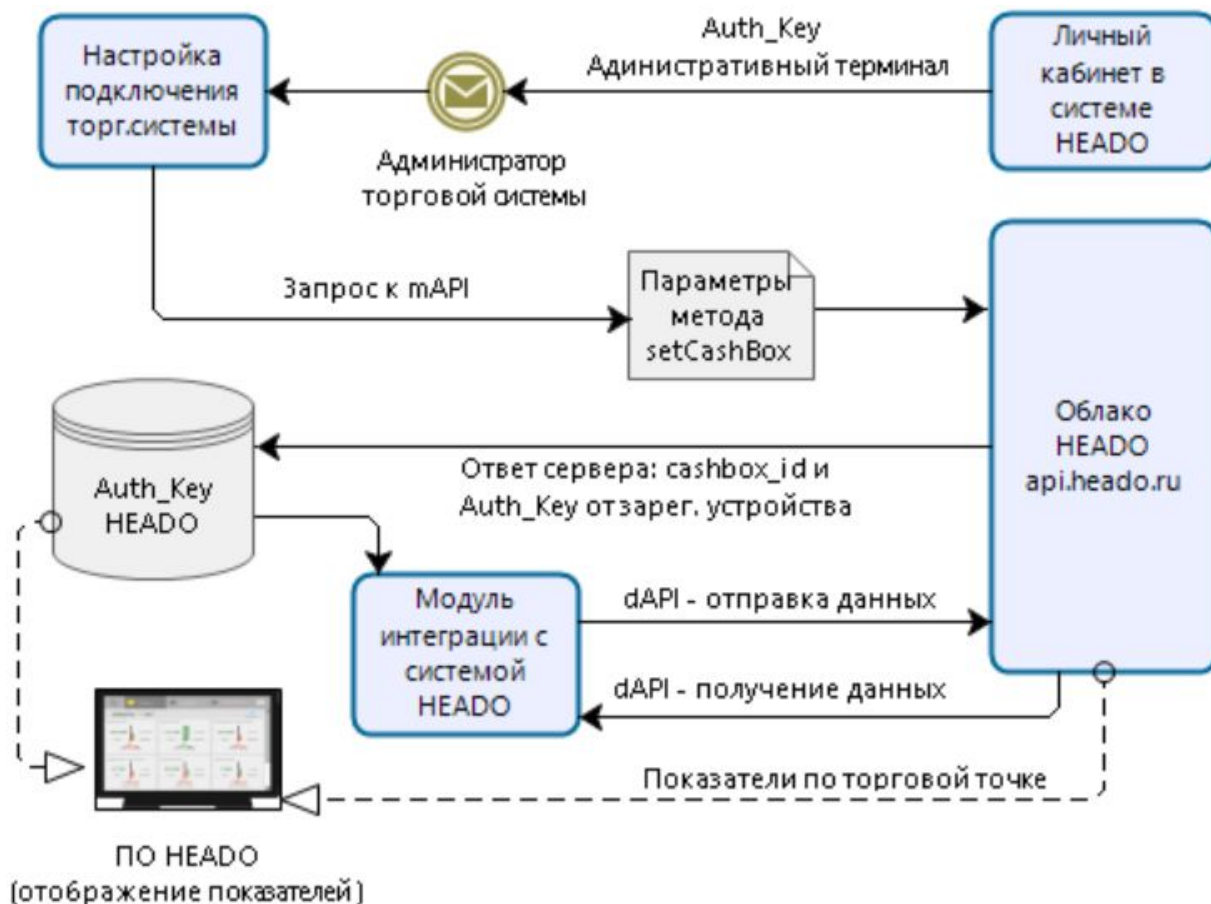


Рисунок 2

6.2. Добавление торговой точки/устройства

Для добавления торговой точки/устройства выполняется запрос к [mAPI](#) используя метод [setCashbox](#) и ключ доступа ([Auth_Key](#)) от [устройства](#) с типом терминала: "Административный терминал".

Важно! Ключ доступа ([Auth_Key](#)) от [устройства](#) с типом терминала: "Административный терминал" можно получить в [ЛК HEADO](#) или через администратора сервиса HEADO после регистрации личного кабинета торговой сети и должен быть сохранен на компоненте [торговой системы](#), выполняющей функции централизованного управления конфигурацией торговой сети.

Важно! На других компонентах [торговой системы](#) (сервер магазина, устройство) сохранять Ключ доступа ([Auth_Key](#)) от [устройства](#) с специального типа терминала: "Административный терминал" запрещается.

6.3. Параметры метода setCashBox

Комментарии содержат рекомендации по заполнению значений параметров запроса, правила применения и пример приведены в описании метода [setCashbox\(location. extid\)](#):

POST Web-адрес запроса ([mAPI](#)): <https://api.heado.ru/management/<authkey административный терминал>>

Имя	Тип	Комментарии
<i>location</i>	object	Объект локации (Описание Торговой точки): - country (string(60)): Название страны по КЛАДР или ФИАС - city (string(60)): Название города по КЛАДР или ФИАС - street (string(100)): Название улицы по КЛАДР или ФИАС

		<ul style="list-style-type: none"> - house (string(20)): Дом, включая корпуса, литеры и т.п. (например 10/1а) - store_ext_id (string(40)): Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр - name (string(100) опционально): Название торговой точки (можно изменить, отправив новое значение)
<i>extid</i>	string(100) опционально	<p>Уникальный идентификатор <u>устройства</u> (кассы, сервера и т.п.) в торговой системе предприятия, состоящий из латинских букв или цифр.</p> <p>Не передавать при регистрации устройства для передачи групповых данных о продажах (сервер торговой точки, группы торговых точек).</p>

Важно! При успешном выполнении запроса к [mAPI](#) используя метод [setCashBox](#) с параметрами [store_ext_id](#) и [extid \(ycmpойcma\)](#), сервис HEADO должен вернуть в ответ [cashbox_id](#) и ключ ([Auth_Key](#)) от зарегистрированного [ycmpойcma](#). Для получения ключа ([Auth_Key](#)) от основного [ycmpойcma](#) торговой точки, параметр [extid \(ycmpойcma\)](#) не передавать.

7. Передача данных о продажах

7.1. Требования к организации процесса извлечения и передачи данных оперативных продаж:

7.1.1. Отправка данных должна производиться непосредственно после завершения транзакции (печати чека);

7.1.2. Повторная отправка данных по одному чеку должна быть исключена;

7.1.3. Отмененные (удаленные из чека) до регистрации чека позиции не должны попадать в данные о чеке;

7.1.4. Сумма по позициям чека и общая сумма чека должны совпадать;

7.1.5. Сумма чека и суммы по позициям должны учитывать примененные скидки (Суммы передаются с учетом всех скидок);

7.1.6. Общая скидка на чек и скидка на позицию передаются отдельными параметрами чека и корзины.

Важно! В случае, если источник данных о чеках в торговой системе не содержит сведений о наименованиях товара и полную иерархию категории необходимо реализовать дополнительный поток данных для загрузки и обновления дополнительного справочника товаров - см. раздел 8 задания.

7.1.7. Необходимо реализовать обработку следующих событий:

- Регистрация продажи – чек продажи;
- Регистрация возврата чека – чек возврата (связанный с чеком продажи);
- Регистрация возврата позиции - частичный возврат по чеку продажи;
- Регистрация чеков произвольного возврата – чек возврата, не связанный с чеком продажи;
- Регистрация чеков коррекции – чеки, формируемые ККТ в соответствии с [п.4 ст.4.3. Федерального закона от 22.05.2003 N 54-ФЗ «о применении контрольно-кассовой техники при осуществлении наличных денежных расчетов и \(или\) расчетов с использованием электронных средств платежа»](#).

7.2. Регистрация продажи

Для регистрации продажи/возврата выполняется запрос к [dAPI](#) с использованием метода [purchaseRegisterBatch](#). Данный метод позволяет загрузить до 100 чеков в одном запросе.

Важно! Для торговых систем с предзаказом количества товара (продукта) необходимо передавать фактически отпущенное количество, зарегистрированное в чеке продажи.

7.3. Параметры метода purchaseRegisterBatch

Комментарии содержат рекомендации по заполнению значений параметров запроса к [dAPI](#), правила применения и пример приведены в описании метода [purchaseRegisterBatch\(receipts \)](#);

POST Web-адрес запроса: https://api.giftoman.ru/device/<Auth_Key_устройства>

Имя	Тип	Комментарии
receipts	Array of Objects	<p>Объект массива:</p> <ul style="list-style-type: none"> - amount (decimal(10,4)) : Итоговая сумма по чеку (скидки вычтены) - discount (decimal(10,4)) : Итоговая скидка по чеку (положительное число) - ts (ISO 8601: YYYY-MM-DDThh:mm:ss±hh:mm) : Точное время закрытия чека, см. п. 4.4.4. Необходимо обязательно передавать метку времени с корректным смещением ±hh от UTC, соответствующим временному поясу торговой точки. - cashier (string(200)) : ФИО Кассира - extId (string(40)): уникальный «Внешний идентификатор чека» в торговой системе, см. п. 4.4.5 - positions (array) : [Товарные позиции в чеке { <ul style="list-style-type: none"> -- name (string(200)) : Название товара -- count (decimal(10,4)) : Количество товара -- total (decimal(10,4)) : Общая сумма по позиции с учетом скидок -- seller (string(200)) : ФИО продавца для конкретной позиции. При наличии такого параметра существует возможность формировать показатели, мотивирующее данное звено продаж. -- category (string(200)) : Иерархическая структура категорий/подкатегорий товара и дополнительные метки. Структура передается последовательно, начиная с первого уровня с использованием разделителя '&&'. Общая длина строки 2000 символов . см. п.4.3 -- extId (string(40)) : Уникальный идентификатор товара, состоящий из цифр или латинских букв. см. п.4.3 - attributes (array) : [Массив свободных атрибутов чека (Атрибуты чека, предоставляющие Сервису HEADO возможности для дополнительной аналитики, см. п.п.7.3.1. Длина названия одного атрибута не более 20 символов. <ul style="list-style-type: none"> .) { <ul style="list-style-type: none"> -- name (string(20)) : Название атрибута (английские мнемоники), -- value (decimal(10,4)) : Значение атрибута - poskey (string(130)) Ключ маппинга к устройству или торговой точке, передаваемый в методе setCashbox() см. п.п.6.4 -- Общий формат: @store_ext_id/extid(устройства) --- store_ext_id - Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр (до 40 символов) --- extid - Уникальный идентификатор устройства (кассы, сервера) в торговой системе предприятия, состоящий из латинских букв или цифр (до 100 символов).

		-- Пример: @111/22
--	--	--------------------

7.3.1. Атрибуты чека

Ниже приведен типичный перечень атрибутов чека, конкретный перечень атрибутов зависит от наличия соответствующих источников данных в [торговой системе](#). Максимально развитый перечень атрибутов позволит в дальнейшем настроить в сервисе HEADO показатели, значимых для бизнеса торгового предприятия.

Параметр	Комментарии
<i>withcard</i> (value: 0,1)	Использование карты лояльности
<i>discount</i> (value: decimal(10,4))	Скидка на покупку
<i>bonus_sp</i> (value: decimal(10,4))	Использование бонусных баллов
<i>procpd</i> (value: int)	Время обработки чека в секундах
<i>paytype</i> /[cash,card,visa,master, e.t.c.] (value: decimal(10,4))	Вид или тип оплаты
<i>guestcount</i>	Количество гостей в чеке
<i>Свой атрибут на латинице, максимум 20 символов</i> (value: decimal(10,4) или int)	Все значения передаются в виде строки, в том числе и числовые типы скидок и флагов. Максимальная длина значения 10 символов.

7.4. Регистрация возврата

Для регистрации возврата чека и возврата позиции применяется тот же метод, который используется для регистрации чеков продажи с учетом следующих важных особенностей:

- При полном возврате, уникальный «Внешний идентификатор» чека, передаваемый в запросе должен соответствовать чеку продажи по которому производится возврат

Важно! При частичном возврате или в случае, если регламенты оператора торговой системы допускают проведение чеков возврата, без привязки к чеку продажи в качестве уникального «внешнего идентификатора» чека и параметров такого чека произвольного возврата необходимо применять данные самого чека произвольного возврата..

- Все параметры и атрибуты чека возврата, включая позиции чека (корзины) должны формироваться по данным чека продажи.
- Указанные ниже параметры в чеке возврата должны быть отрицательными (<0):

Параметр	Комментарии
<i>amount</i>	Итоговая сумма по чеку (скидки вычтены)
<i>attributes</i>	Массив свободных атрибутов чека (Атрибуты чека, предоставляющие Сервису HEADO возможности для дополнительной аналитики, см. п.п.7.3.1.

7.5. Регистрация чеков коррекции

Чеки коррекции регистрируются в сервисе HEADO способами, аналогичными для чеков продажи и чеков возврата. Параметры и атрибуты чеков, отсутствующие в случае регистрации чеков коррекции не передаются (остаются пустыми).

8. Загрузка исторических данных

Для загрузки исторических данных в сервис HEADO за указанный промежуток времени, необходимо реализовать утилиту для окружения [DWH](#), реализующую транспорт чеков в API HEADO, способом, аналогичным транспорту оперативных данных описанному в [разделе 7](#)

Важно! Загрузка больших объемов исторических данных в сервис HEADO возможна в часы наименьшей нагрузки с 23:00 до 07:00 по московскому времени (UTC+03:00).

Для автоматизации выгрузки целесообразно в утилите выгрузки исторических данных предусмотреть настройку заданий для выгрузки и выполнение заданий по расписанию – в часы наименьшей загрузки.

Важно! В случае, если источник исторических данных может содержать наименования товаров и категорий отличающиеся от источника данных оперативных продаж (компоненты разных разработчиков) рекомендуем реализовать дополнительный поток данных для загрузки и обновления дополнительного справочника товаров - см. [раздел 10](#).

При формировании задания необходимо предусмотреть выбор торговой точки, кассы и периода выгрузки.

Важно! Период выгрузки должен быть кратен одному календарному месяцу. В случае, если выгружаются данные за несколько периодов, не должно быть разрывов между ними, последний по хронологии период должен заканчиваться датой, после которой в сервисе HEADO существуют оперативные данные.

9. Сверка данных сервиса HEADO с торговой системой

При отладки интеграции и при развертывании сервиса HEADO на каждой торговой точке требуется проведение сверки данных между торговой системой и сервисом HEADO. Рекомендованный порядок сверки описан по ссылке <https://support.heado.ru/manuals/compare>

9.1. Отчеты для поиска ошибок загрузки данных

Для отладки реализованных интеграционных функций и локализации ошибок загрузки данных в [торговой системе](#) необходимо обеспечить возможность формирования инженерными средствами выгрузки данных в общепринятые форматы (xls, csv) представления, содержащего сведения о продажах, содержащие все использованные при отправке данных в API HEADO параметры [торговой системы](#) в неизменном виде (как в торговой системе).

Пример по ссылке: <https://partner.heado.ru/files/media/Mag58PurchasePositions61080064732179.xlsx>

10. Загрузка и обновление дополнительного справочника товаров

Важно! Загрузка и обновление дополнительного справочника товаров рекомендуется только для случаев, когда штатный справочник товаров и категорий/меток или средства его администрирования не позволяют реализовать необходимые для формирования показателей данные. В таком случае загружаемый дополнительный справочник товаров дополняет штатный, информация из которого передается в чеках. Должен быть определен достоверный источник данных для загрузки, актуальность которого поддерживается регулярно.

Перед загрузкой дополнительного справочника товаров необходимо сбросить категории/метки из ранее загруженного справочника - выполнив запрос к [mAPI](#) используя метод [priceCategoryReset\(\)](#), затем использовать метод [priceUpdateBatch\(items\)](#) для загрузки обновленного справочника товаров.

10.1. Параметры методов для загрузки доп.справочника

10.1.1. Метод [priceCategoryReset](#) не содержит параметров

10.1.2. Параметры метода [priceUpdateBatch](#) описаны в таблице ниже. Комментарии содержат рекомендации по заполнению значений параметров, правила применения и пример приведены в описании метода [priceUpdateBatch\(items\)](#)

POST Web-адрес запроса: <https://api.heado.ru/management/<authkey административный терминал>>

Имя	Тип	Комментарии
-----	-----	-------------

items	Array of Objects - Максимум 1000 элементов	<p>Объект массива:</p> <ul style="list-style-type: none"> - name (string(200)) : Название товара - params (array) : объект параметров товарной позиции { <ul style="list-style-type: none"> -- extid string(40) - Уникальный идентификатор товара, состоящий из цифр или латинских букв. см. п.4.3 Обязательно должен соответствовать внешнему уникальному идентификатору получаемого из чека. -- category string(200) (опционально) - Иерархическая структура категорий/подкатегорий товара и дополнительные метки. Структура передается последовательно, начиная с первого уровня с использованием разделителя '&&'. Общая длина строки 2000 символов. см. п.4.3. Опционально: перед доп.меткой в квадратных скобках можно через запятую указать идентификаторы торговых точек (store_ext_id) на которую применяется данная метка на товара (например метка акции для одного или нескольких магазинов) Пример: BSTR100&&[1,2,5]BSTR10 -- price decimal(10,4) - Цена позиции (допускается передавать 0)
-------	---	---

Важно! Связность с чеками выполняется по уникальному идентификатору товара - ExtId. Для корректного дополнения штатного справочника товаров [торговой системы](#) категориями и метками дополнительного справочника товаров необходимо обеспечить соответствие уникальных идентификаторов товара штатного и дополнительного справочников.

11. Отображение показателей на рабочем месте кассира (продавца)

Оперативное доведение достигнутых показателей до персонала на всех уровнях является важной задачей сервиса HEADO. Штатным и проверенным временем способом является установка на рабочие места персонала [приложения HEADO](#) для операционной системы Windows, которое отображает показатели HEADO в графическом интерфейсе.

Инструкция, в которой описан графический интерфейс, доступна по ссылке: https://partner.heado.ru/files/media/Инструкция_DirectorComputer.pdf

В случае, если отдельные рабочие места [торговой системы](#), работают на отличной от Windows операционной системе или имеют специфический, ориентированный на дисплеи небольшого физического размера, на которых графический интерфейс приложения HEADO будет конфликтовать с основным интерфейсом оператора (кассира), рекомендуем реализовать для таких рабочих мест адаптированное к пользовательскому интерфейсу оператора (кассира) отображение показателей HEADO.

11.1. Способ получения данных для отображения

Для получения показателей (отчетов) по торговой точке выполняется запрос [dAPI](#) с использованием метода [reports\(id \[, params \]\)](#):

Важно! Ответ системы HEADO на запрос с использованием метода [reports\(\)](#) предоставляет набор показателей торговой точки к которой относится используемый ключ ([Auth_Key](#)) [устройства](#). Для всех [устройств](#) торговой точки набор показателей одинаковый и [определяется настройками в личном кабинете](#).

В большинстве случаев для получения показателей рекомендуем использовать один ключ [Auth_Key](#) привязанный к торговой точке, получаемый при регистрации торговой точки. Исключением будет являться случай, когда данные о продажах передаются с использованием ключа [Auth_Key](#) сервера группы торговых точек предприятия.

11.2. Параметры метода reports

Важно! Границы временных периодов необходимо задавать точно - начало и конец месяца, начало и конец дня. Формирование показателей за неполный период не имеет практического смысла

Ответ на запросе через метод reports содержит набор таблиц показателей. Число таблиц определяется числом заданных временных периодов в параметрах команды.

Комментарии содержат рекомендации по заполнению значений параметров метода, правила применения и пример приведены в описании метода [reports\(id \[, params \]\)](#):

POST Web-адрес запроса: https://api.giftoman.ru/device/<Auth_Key устройства>

Имя	Тип	Комментарии
id	string(20)	- ' metrics ' - Получение общего отчёта "Показатели" (страница 4 инструкции) - ' bysellers ' - Получение отчёта "Продавцы" (страница 5 инструкции) - ' bytradepoints ' - Получение отчёта "Рейтинг торговых точек" (страница 6 инструкции)
params	object	{ "filter": { }, "range" : [{ "type": <id периода>, "from": "Y-m-d H:i", "to": "Y-m-d H:i"},...]

12. Установка и публикация плановых значений

При наличии в [торговой системе](#) функций планирования целесообразно реализовать автоматизированную установку и публикацию плановых значений. Для этого в [mAPI](#) HEADO предусмотрено 2 метода: [storePlanSet](#) и [storePlanPublish](#)

Схема потоков данных решения интеграции [торговой системы](#) с сервисом HEADO для передачи плановых значений представлена на Рисунке 3.



Рисунок 3

12.1. Параметры метода storePlanSet

Для установки или удаления планового значения на нужный период, необходимо выполнить запрос к [dAPI](#) с использованием метода [storePlanSet](#)

Важно! После установки всех плановых значений на период необходимо выполнить запрос к [mAPI](#) используя метод [storePlanPublish](#) для публикации плана и актуализации значений.

Комментарии содержат рекомендации по заполнению значений параметров метода, правила применения и пример в описании метода [storePlanSet](#)

POST Web-адрес запроса: <https://api.heado.ru/management/<authkey административный терминал>>

Имя	Тип	Комментарии
store_ext_id	string(20)	Уникальный идентификатор торговой точки в торговой

		системе предприятия, состоящий из латинских букв или цифр. Должен совпадать с store_ext_id передаваемого в методе setCashbox()
metric	string(40)	Мнемоника, либо прямой идентификатор правила: - income : Выручка - average : Средний чек - count : Кол-во продаж - category : Выручка по СТМ - complexity : Комплексность чека по кол-ву товаров - p/<идентификатор показателя> - прямой идентификатор показателя в системе HEADO
plan	DECIMAL(13,4)	Если >=0 , то установка планового значения = plan Если <0 , то удаление
year	int	Год
month (необязательный)	int	Месяц, если не указан или = -1 , то выставляется план на год
day (необязательный)	int	День, если не указан или = -1 , то выставляется план на месяц, иначе - на день

12.2. Параметры метода storePlanPublish

Данный метод следует выполнять только после установки плановых значений.

Комментарии содержат рекомендации по заполнению значений параметров метода, правила применения и пример приведены в описании метода [storePlanPublish\(store_ext_id, auto\)](#):

POST Web-адрес запроса: <https://api.heado.ru/management/<authkey административный терминал>>

Имя	Тип	Комментарии
store_ext_id	string(20)	Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр. Должен совпадать с store_ext_id передаваемого в методе setCashbox()
auto	boolean	- true - Использовать сервисную аналитику распределения по периодам. - false - Не использовать аналитику (равномерное распределение по нижним периодам)

13. Сигнал открытия/закрытия торговой точки

Для отправки сигнала и события открытия и закрытия торговой точки в [dAPI](#) HEADO предусмотрен метод [registerEvent](#).

Для реализации данного функционала необходимо определить источник и событие на которое будет отправляться сигнал об открытии закрытии торговой точки.

Рекомендуемые варианты реализации функции отправки сигналов открытия/закрытия:

- **Торговая система** самостоятельно определяет время открытия и закрытия торговой точки. Сигнал открытия или закрытия должен отправляться сразу при наступлении соответствующего события с использованием одного и того же **Auth Key** торговой точки. Данный вариант реализации рекомендуется, при наличии централизованного источника данных (например кассовый сервер торговой точки) для определения времени передаваемых в сигналах открытия/закрытия по имеющимся в **торговой системе** данных. Время открытия торговой точки определяется **торговой системой** по времени открытия самой первой смены кассира из всех касс торговой точки за текущий день. Время закрытия торговой точки определяется торговой системой по времени снятия последнего Z отчета среди всех касс торговой точки по которым были открыты смены.
- **Торговая система** при наступлении соответствующего события на каждой кассе, сразу отправляет сигнал открытия или закрытия с использованием **auth_key** кассы на котором произошло событие. В сигнале открытия, необходимо указывать время открытия смены кассира. В сигнале закрытия, необходимо указывать время снятия Z отчета на кассе за текущий день. При данном варианте реализации, система HEADO зафиксирует время (не корректируется при получении сигналов с другим временем) первого полученного сигнала открытия и самостоятельно определит время закрытия торговой точки (как только кол-во сигналов открытия и закрытия будут совпадать, последний сигнал закрытия будет считаться что торговую точку закрыли).

Схема потоков данных решения интеграции торговой системы с сервисом HEADO для передачи сигналов открытия/закрытия торговой точки представлена на Рисунке 4.



Рисунок 4

Важно! При наступлении соответствующих событий, сигнал открытия/закрытия должен передаваться сразу. При невозможности своевременной отправки сигнала из-за отсутствия тех.возможности, сигналы необходимо отправлять в порядке очередности: открытие>закрытие от самого раннего до самого позднего

13.1. Параметры метода registerEvent

Комментарии содержат рекомендации по заполнению значений параметров метода, правила применения и пример приведены в описании метода [registerEvent](#)

POST Web-адрес запроса: https://api.giftoman.ru/device/<Auth_Key_устройства>

Имя	Тип	Комментарии
name	string	- shift/start - сигнал открытия торговой точки/смены или - shift/stop - сигнал закрытия торговой точки/смены
ts	string	- shift/start , 'YYYY-MM-DD hh:mm:ss /*Локальное время начала смены*/ или - shift/stop , 'YYYY-MM-DD hh:mm:ss' /*Локальное время конца*/

params	object	в params можно передавать массив метаданных: - seller : string ФИО - localtimezone : offset +/- hours
--------	--------	---

14. Установка/обновление расписания работы торговых точек

Для обновления расписания работы торговых точек в [mAPI](#) HEADO предусмотрен метод [storeSchedule](#)

Схема потоков данных решения интеграции [торговой системы](#) с сервисом HEADO для установки/обновления расписания работы торговых точек представлена на Рисунке 5.



Рисунок 5

14.1. Параметры метода storeSchedule

Комментарии содержат рекомендации по заполнению значений параметров метода, правила применения и пример в описании метода [storeSchedule](#)

POST Web-адрес запроса: <https://api.heado.ru/management/><authkey административный терминал>

Имя	Тип	Комментарии
store_ext_id	string(20)	Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр. Должен совпадать с store_ext_id передаваемого в методе setCashbox()
schedule	Array of Objects	Объект массива: - wday (tinyint) : 1-Понедельник, ... 7 - Воскресенье. Если у торговой точки выходной, то расписание не указывается - from (string(5) ЧЧ:ММ) : Начало работы (00:00 для круглосуточных) - to (string(5) ЧЧ:ММ): Конец работы (24:00 для круглосуточных) - breaks (Array of objects опционально): [-- from (string(5) ЧЧ:ММ): Начало перерыва ТТ -- to (string(5) ЧЧ:ММ): Конец перерыва ТТ

15. Обновление складских остатков

Для получения остатков товаров по торговой точке в [mAPI](#) HEADO предусмотрен метод [inventoryUpdateBatch](#)

Для автоматизации и настройки выгрузки целесообразно предусмотреть и реализовать настройку заданий для выгрузки данных по расписанию – 1 раз в сутки в 00:00 часов по Мск или до 3-х раз при необходимости (увеличение частоты выгрузки увеличивает нагрузку).

Важно! При наличии у торговой точки нескольких складов, необходимо передавать в одном запросе все остатки товаров с использованием одного идентификатора ([store_ext_id](#)) торговой точки к которым они относятся.

Схема потоков данных решения интеграции [торговой системы](#) с сервисом HEADO для обновления складских остатков представлена на Рисунке 6.



Рисунок 6

Важно! Первая выгрузка остатков товаров по торговой точке, должна содержать все позиции действующих карточек товаров (включая товары с 0 остатком). Каждая последующая выгрузка остатков, должна содержать только позиции, по которым произошли изменения: кол-во, признак наличия в матрице товаров, цена (необходимо исключить повторную отправку данных без изменений).

15.1. Параметры метода `inventoryUpdateBatch`

Комментарии содержат рекомендации по заполнению значений параметров запроса к [mAPI HEADO](#), правила применения и пример приведены в описании метода [inventoryUpdateBatch](#)

POST Web-адрес запроса: <https://api.heado.ru/management/<authkey административный терминал>>

Имя	Тип	Комментарии
items	Array of Objects - Максимум 1000 элементов	<ul style="list-style-type: none"> -- store_ext_id (string(40)) - Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр. Должен совпадать с <code>store_ext_id</code> передаваемого в методе setCashbox() -- price_ext_id (string(40)) - Уникальный идентификатор товара, состоящий из цифр или латинских букв. см. п.4.3 Обязательно должен соответствовать внешнему уникальному идентификатору получаемого из чека. -- snapshot_datetime (string(ISO 8601 datetime)) - дата и время выгрузки актуального состояния инвентаря -- qty (decimal(13,4)) - количество товара на момент выгрузки состояния -- in_matrix (boolean) - признак наличия в товарной матрице торговой точки. 1-входит, 0-не входит, если нет данных - не передавать. -- purchase_price (decimal(12,4)) - цена продажи за единицу -- prime_cost (decimal(12,4)) - себестоимость товара "без НДС" за единицу (Стоимость всего остатка / количество = средневзвешенная цена)

16. Установка/обновление расписания работы сотрудников

Для обновления расписания работы сотрудников на торговых точках в Management API HEADO предусмотрена команда [staffManning](#)

Схема потоков данных решения интеграции торговой системы с сервисом HEADO для установки/обновления расписания работы сотрудников представлена на Рисунке 7.

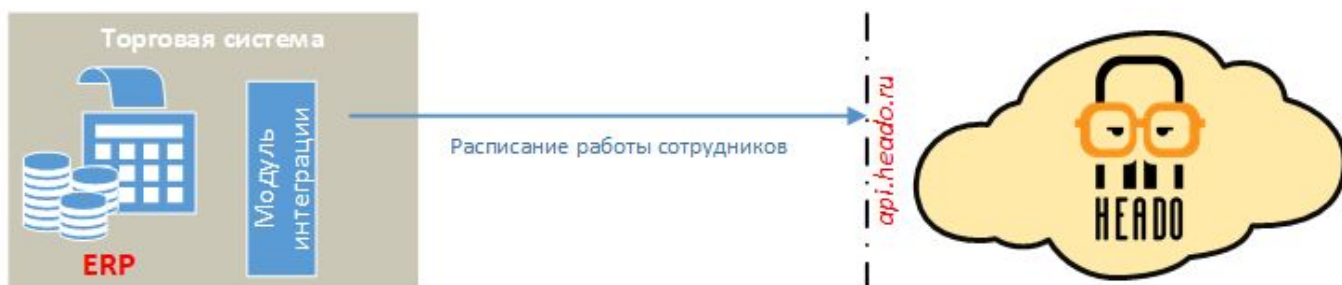


Рисунок 7

16.1. Параметры метода staffManning

Комментарии содержат рекомендации по заполнению значений параметров метода, правила применения и пример приведены в описании метода [staffManning\(items\)](#)

POST Web-адрес запроса: <https://api.heado.ru/management/<authkey административный терминал>>

Имя	Тип	Комментарии
items	Array of Objects	<ul style="list-style-type: none"> - store_ext_id (string(40)) - Уникальный идентификатор торговой точки в торговой системе предприятия, состоящий из латинских букв или цифр. Должен совпадать с store_ext_id передаваемого в методе setCashbox() - staff: <ul style="list-style-type: none"> -- fio (string(60)), - ФИО Сотрудника -- ext_id (string(40)) (опционально), - уникальный идентификатор в торговой системе предприятия (в пределах торговой точки) - date: Дата записи по ISO 8601 (YYYY-MM-DD) - workhours: Список рабочих периодов в указанный рабочий день <ul style="list-style-type: none"> -- from: <ЧЧ:ММ>, -- to: <ЧЧ:ММ> - breaks: (опционально) Периоды перерывов в указанный рабочий день. В случае отсутствия, система будет восстанавливать перерывы между рабочими периодами. <ul style="list-style-type: none"> -- from: <ЧЧ:ММ>, -- to: <ЧЧ:ММ>