LAMODA MARKETPLACE

Рекомендации по IT-интеграции

Версия 1.4

24 августа 2016

Оглавление

[IT интеграция 3](#_Toc459819767)

[Общая информация 3](#_Toc459819768)

[API интеграция 4](#_Toc459819769)

[Авторизация и доступ 5](#_Toc459819770)

[Управление товарами 6](#_Toc459819771)

[Загрузка товаров через csv-файлы 6](#_Toc459819772)

[Создание товаров через API 7](#_Toc459819773)

[Обновление данных о товарных запасах 16](#_Toc459819774)

[Полный список методов по созданию/обновлению продуктов 16](#_Toc459819775)

[Управление заказами 17](#_Toc459819776)

[Получение заказов 17](#_Toc459819777)

[Обработка заказов 17](#_Toc459819778)

[Обновление статусов заказов 17](#_Toc459819779)

[Полный объем интеграции 18](#_Toc459819780)

[Полный список методов по управлению заказами 19](#_Toc459819781)

[Упрощение интеграции 20](#_Toc459819782)

[История изменений 21](#_Toc459819783)

## IT интеграция

## Общая информация

Общий процесс работы по модели Marketplace состоит из следующих частей:

1. Компания-партнер загружает информацию о товарах (фотографии, описания, цены и категоризация)
2. Отдел контроля качества Lamoda проверяет загруженные товары и после прохождения проверки публикует на Lamoda.ua.
3. Клиенты создают заказы на Lamoda.ua. В Call Center Lamoda подтверждают все созданные клиентами заказы.
4. Компания-партнер получает уже подтвержденные заказы, собирает их и доставляет покупателю.

Управление продажами на Lamoda по модели marketplace происходит через Seller Center. Существует 3 основных способа взаимодействия с Seller Center:

1. API-интеграция. Рекомендованный вариант, позволяющий эффективно обмениваться большими объемами информации и минимизировать ручной труд.
2. Загрузка данных о товарах через csv-файл заданного формата.
3. Ручной ввод информации в web-интерфейсе.

##

## API интеграция

Предусмотрено несколько вариантов интеграции с Seller Center по API, в зависимости от объемов стока и возможностей ваших разработчиков.

#### Минимальный

Наиболее быстрый вариант интеграции, чтобы как можно скорее разместить товары на сайте и получать новые заказы.

1. Все операции с товарами и заказами проводить вручную в интерфейсе SC или с помощью csv-файлов. Подготовьте файлы с товарами и их изображениями в [соответствии с инструкцией](#_Работа_с_csv-файлами).
2. Обновляйте сток при его изменении или же не реже, чем раз в 10 минут, используя обмен [ProductUpdate](https://sellercenter.readme.io/docs/productupdate).

Работа с заказами полностью проходит в интерфейсе: SC автоматически загружает новые подтвержденные заказы. Их необходимо обработать (собрать и отправить клиенту) и отметить статусы доставки.

#### Оптимальный

Управление товарами проводить аналогично тому, как описано выше — с помощью csv-файлов, а управление заказами автоматизировать с помощью обменов:

1. [GetOrders](https://sellercenter.readme.io/docs/getorders) — чтобы загрузить все новые заказы
2. когда заказ собран, отправьте обмен [SetStatusToReadyToShip](https://sellercenter.readme.io/docs/setstatustoreadytoship)
3. когда заказ отправлен клиенту, — [SetStatusToShipped](https://sellercenter.readme.io/docs/setstatustoshipped)

Lamoda требует обработку заказов в течение заданного интервала времени. Свидетельством обработки заказа является изменение статуса заказа в Seller Center. Изменение статуса заказа на данный момент невозможно через csv-файл. Используйте интерфейс или API:

1. обмен SetStatusToDelivered для доставленных заказов
2. SetStatusToFailedDelivery, в случае, если что-то пошло не так.
3. Отменить заказ [setStatusToCanceled](https://sellercenter.readme.io/docs/setstatustocanceled) для заказов, которые еще не были собраны (до отправки обмена setStatusToReadyToShip).

#### Расширенный

Рекомендуется для крупных партнеров с большими объемами товаров — полностью автоматизировать работу с Seller Center. Подробное описание в разделах ниже.

#### Доставка LM Express

Если вы используете услугу доставки заказов курьерской службой LM Express, потребуются дополнительные действия. Все операционные требования описаны в отдельном документе **Operational guide**. В Seller Center потребуется создать исходящую поставку и информировать о том, что она отправлена от поставщика в ДЦ Lamoda. Все действия вы можете выполнить в интерфейсе SC, либо отправить обмены:

1. [CreateForwardManifest](https://sellercenter.readme.io/docs/createforwardmanifest): создать новую исходящую поставку, в которой отправлены посылки
2. [SetManifestStatusToShipped](https://sellercenter.readme.io/docs/setmanifeststatustoshipped): изменить статус созданной поставки на *Отправлена со склада*

Если интеграция с SC полностью автоматизирована, следует указать уникальные идентификаторы товаров (серийные номера в терминологии SC) перед созданием упаковки. Для этого отправьте обмен [SetStatusToPackedByMarketplaceV4](https://sellercenter.readme.io/docs/setstatustopackedbymarketplace). Подробности в документе ниже, раздел [Управление заказами](#_Управление_заказами).

## Авторизация и доступ

Для начала работы ваш менеджер по запуску предоставит вам доступ на Seller Center Staging, чтобы вы могли в тестовом режиме изучить возможности Seller Center. Для этого перейдите по ссылке из полученного письма, введите в окно логин **rocket** и пароль **rock4me**.



После авторизации вам предложат поменять ваш личный пароль. Для этого перейдите по предложенной ссылке и введите адрес электронной почты, который вы предоставили для регистрации в Seller Center, и придумайте пароль.

В последующем, для доступа в [Seller Center](https://sellercenter.lamoda.ua/user/user) вам будет необходимо вводить e-mail, на который зарегистрирован ваш аккаунт, и пароль, который вы создали.



В боевой системе аутентификация с вводом пароля rocket-rock4me отсутствует, вам будет необходимо сразу авторизоваться в приложении по вашему собственному адресу почты и паролю.

Полная документация по API с примерами запросов и ответов, описаниями полей и требованиям к ним доступна по ссылке: <https://sellercenter.readme.io/docs> (на английском).

Чтобы попробовать выполнить API-методы, авторизуйтесь на Seller Center Staging и откройте **Settings → Integration Management**. Чтобы получить доступ к API в Live-системе, сгенерируйте [API-ключ](https://sellercenter.readme.io/docs/signing-requests) по ссылке: <https://sellercenter.lamoda.ua/user/user>

## Управление товарами

### Загрузка товаров через csv-файлы

Подготовьте csv-файл в соответствии с форматом Lamoda. Чтобы загрузить шаблон, в интерфейсе Seller Center перейдите в раздел **Products** → **Import products**.



В правой части окна **Download Templates** размещены ссылки на все возможные файлы по заполнению продуктов. Вы можете скачать шаблоны и массово добавлять новые продукты. Скачайте шаблон для создания продуктов из категорий *Обувь, Одежда, Аксессуары и Красота* в формате csv-файлов в блоке **Product creation by category**.

Чтобы скачать файл со всеми категориями Seller Center, нажмите на ссылку **Download Category Tree.**

Чтобы отредактировать уже созданные продукты, скачайте соответствующий шаблон из Product Updates:

* Stock Update — обновить данные о товарных запасах
* Price Update — обновить цены
* Images Update — загрузить новые изображения
1. Заполните файл и загрузите его: для этого в блоке **Import products** раскройте список **Import Mode** и выберите **Create new products**.
2. Последующие редактирования рекомендуется выполнять с этим же файлом и загружать его с типом **Update products**. Если файл будет потерян, то скачать загруженные данные одним файлом будет невозможно — шаблон всегда формируется пустым.
3. Данные о товарных запасах обновляйте при каждом изменении стока, используя файл или обмен [productUpdate](https://sellercenter.readme.io/docs/productupdate).

### Создание товаров через API

Создание товаров в Seller Center возможно через API группы [Product Endpoints](https://sellercenter.readme.io/docs.). Однако запуск работы по Marketplace мы советуем начинать с импорта товаров с помощью csv-файла.

Seller Center предоставляет возможности по созданию продуктов по имеющейся структуре: дерево каталога и набор атрибутов для каждой категории определяются настройками систем Lamoda и синхронизированы с ними.

Порядок обменов для создания продукта в Seller Center:



1. [getBrands](https://sellercenter.readme.io/docs/getbrands) (необязательно): получить список всех доступных брендов для создания продукта. Бренд создает ваш аккаунт-менеджер. Обычно, к моменту вашей регистрации новый бренд уже заведен и вы можете использовать его в поле **Brand**.
2. [getCategoryTree](https://sellercenter.readme.io/docs/getcategorytree): получение дерева доступных категорий товаров.
3. [getCategoryAttributes](https://sellercenter.readme.io/docs/getcategoryattributes): получение набора всех атрибутов товаров для категории. Необходимо их сохранить и сделать маппинг с собственными категориями. Вы можете загрузить все дерево категорий и атрибуты к ним один раз. Для селлеров Lamoda регистрирует новые категории и дополняет наборы атрибутов для них. Поэтому для создания последующих новых товаров рекомендуем выполнять эти запросы.
4. [productCreate](https://sellercenter.readme.io/docs/productcreate): создание одного или нескольких продуктов. У каждого продукта должна быть указана категория и атрибуты (характеристики), описание, цена.

Категория и атрибуты этой категории вы получаете при выполнении запросов getCategoryTree и getCategoryAttibutes. При создании продукта необходимо перечислить все атрибуты, которые являются **isMandatory**. Набор атрибутов различается для каждой категории и делится на две группы:

* + Глобальные — обязательные для каждого товара. К ним относятся: brand, name, size\_scale, product\_type, description, category\_gender, season\_wear, color, style.
	+ Уникальные, в зависимости от категории товара (например, upper\_material, side\_length).

Важно! При создании продукта обязательными являются не только глобальные атрибуты, но и некоторые уникальные. Обязательно проверяйте атрибуты по каждой категории.

Вы можете выполнять тестовые запросы на странице [API Explorer](https://staging.sellercenter.lamoda.ru/api-explorer?powerswitch=1) (ссылка может различаться в зависимости от домена). Для этого:

1. Откройте главную страницу Seller Center.
2. На панели инструментов нажмите **Products → API Reference**.



1. Нажмите на ссылку **Product Call Examples**.
2. Выберите категорию, в которой хотите создать новый продукт. Например, *Женская обувь / Мужская обувь / Детская обувь*.



В результате вы получите пример XML-структуры запроса на создание продукта в выбранной категории:



В примере **Minimum API Call** представлен запрос с минимальным, но достаточным набором полей для создания продукта. В **Full API Call** перечислены все необязательные атрибуты продукта из заданной категории.

Обратите внимание на цветовое выделение: красным отмечены mandatory-поля, черным — опциональные, зеленым — уникальные в рамках категории (например, *Высота каблука* для продуктов из категории *Обувь*).

Для дополнительной проверки, является ли определенный атрибут обязательным, например, <SizeScaleMp> для категории *Обувь*, выполните API-запрос **getCategory**, чтобы определить номер категории, а затем — запрос **getCategoryAttributes** с номером категории (PrimaryCategory=11).



Нажмите **Regenerate Signature**, а затем — **Execute**.

В XML ниже вы можете найти атрибут: поле <isMandatory> для него имеет значение 1, а значит — обязательное.



Важно! Атрибуты продуктов, обязательные для передачи на уровне <Product>:

* Brand
* BrowseNodes
* Categories
* Condition
* Description
* Name
* ParentSku
* Price
* PrimaryCategory
* ProductData
* ProductGroup
* ProductId
* Quantity
* SaleStartDate
* SaleEndDate
* SalePrice
* SellerSku
* ShipmentType
* Status
* TaxClass
* Variation
* VolumetricWeight

Если вы неправильно укажете уровень вложенности атрибута (например, будете передавать их на уровне <ProductData>), Seller Center ответит ошибкой.

Поясним назначение некоторых полей продукта:

* **PrimaryCategory**: каждый продукт должен иметь хотя бы одну категорию. Поле обязательно для заполнения. Помимо нее, вы можете указать дополнительные категории, где должен отображаться продукт.
* **Categories**: перечислить через запятую id категории из того же дерева категорий, к которым относится товар. Например, продукту из категории *Джинсовые куртки* можно присвоить дополнительную категорию *Легкие куртки и ветровки*.
* **BrowseNodes**: перечислить через запятую id категорий из другого дерева категорий, к которым относится товар. Например, продукт из категории *Рубашки* показывать также в категории *Боди-блузы*.

Каждая модель в отдельном цвете является отдельным продуктом. Для создания продукта в разных размерах сначала создайте один размер через запрос [ProductCreate](https://sellercenter.readme.io/docs/productcreate), а затем SKU созданного продукта указывайте как **ParentSku** в последующих запросах на создание других размеров этого продукта.

Seller Center оперирует не только продуктами и заказами, но также учитывает способ, которым компания доставляет заказы покупателям. Возможные варианты:

* + **Dropshipping**: селлер доставляет заказы покупателям напрямую
	+ **Crossdocking**: селлер использует 3PL (Third Party logistics) услуги логистических компаний для доставки заказов.

На каждом продукте должен быть маркер используемой модели доставки — он указывается в поле **ShipmentType**. Изменить его нельзя, только создать продукт заново с другим типом. Seller Center предоставляет возможность работать одновременно по двум моделям доставки.

Если нет скидки на продукт, то в поле **SalePrice** указывать равный полю Price.

В ответе на **productCreate** в поле **RequestId** вернется уникальный идентификатор отправленного запроса. Используйте его, чтобы получить статус выполненного запроса через API-вызов **FeedStatus**.

1. [image](https://sellercenter.readme.io/docs/image): передача изображений для созданных продуктов. В запросе необходимо передать SKU (артикул) товара и ссылку, куда загружено изображение. Первое переданное изображение становится основным для продукта. В запросе можно передавать одно или несколько изображений как для одного продукта, так и сразу для нескольких. Для каждого товара можно загрузить не более 8 изображений. Для публикации товаров на Lamoda.ru изображение обязательно.
2. [FeedStatus](https://sellercenter.readme.io/docs/feedstatus): получить статус очереди запроса, в котором находятся записи о созданных продуктах в **productCreate**. Очередь может состоять из одного или нескольких запросов на создание продукта. Для вашего удобства рекомендуется отправлять запрос на создание / обновление атрибутов, состоящий из нескольких продуктов, чтобы получить один **requestID** на все продукты.

Возможные статусы запросов:

* + Queued: успешно добавлен к очереди и в ожидании обработки
	+ Processing: в процессе обработки сервером
	+ Canceled: отменен селлером
	+ Finished: обработан
	+ Error: ошибки в результате отправки запроса

Отменить очередь можно только в статусе Queued. Seller Center ограничивает возможное число очередей на создание/обновление продуктов: не более 50-ти в день.

По достижению этого дневного лимита вы можете отправлять не более одной очереди каждые 2 минуты. Таким образом, наибольшее количество созданных очередей в день — 770, но может варьироваться в зависимости от селлера.

1. [QCStatus](https://sellercenter.readme.io/docs/getqcstatus): получить результат проверки отделом Контроля качестве созданных или отредактированных товаров. По умолчанию приходит ответ на 100 последних созданных/измененных товаров.

### Обновление данных о товарных запасах

Возможно обновление информации о количестве товара в наличии, цен и других атрибутов товаров через API. Для иллюстрации простоты разработки приводим пример запроса:

<?xml version="1.0" encoding="UTF-8" ?>
 <Product>
 <SellerSku>4928a374c28ff1</SellerSku>
 <Quantity>4</Quantity>
 </Product>
</Request>

**Важно!** Обратите внимание, что при использовании этого метода посылайте только те значения, которые надо обновить. Отправка атрибутов товара (цвет, описание и др.) повлекут проверку отдела Контроля Качества, даже если они не были изменены. Если вы отправите только обновленные значения количества (quantity), проверки QC не потребуется.

### Полный список методов по созданию/обновлению продуктов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API-методы | Описание | Альтернатива | Обязательно | Частота обновления |
| [getBrands](https://sellercenter.readme.io/docs/getbrands) | Получение списка доступных брендов. | Наименование всех брендов согласовано с аккаунт-менеджером. | Нет  | при создании новых продуктов |
| [getCategoryTree](https://sellercenter.readme.io/docs/getcategorytree) | Получение дерева категорий. | Категории из дерева категорий полностью совпадают с наименованиями на Lamoda.ru. В csv-загрузке есть возможность выбора доступных категорий. | Нет | при создании новых продуктов |
| [getCategoryAttributes](https://sellercenter.readme.io/docs/getcategoryattributes) | Получение атрибутов, доступных для категории. | В csv-загрузке есть возможность выбора доступных атрибутов. | Нет | при создании новых продуктов |
| [productCreate](https://sellercenter.readme.io/docs/productcreate) | Создание продукта  | Загрузка csv-файла оптимально подходит для изначального создания продуктов, обновления коллекций. | Нет | при создании новых продуктов |
| [productUpdate](https://sellercenter.readme.io/docs/productupdate) | Обновление продукта (изменение характеристик)  | Загрузка csv-файла при глубоком товарном запасе | Обязательно для товарных запасов | допустимая частота обновления — раз в 10 минут.  |
| [image](http://postimage) | Добавление изображений | Массовая загрузка фотографий через интерфейс. | Нет  | при создании/обновлении продуктов |

## Управление заказами

Обработка заказов проходит следующие этапы:

1. загрузка новых заказов
2. сборка и упаковка посылок
3. передача в доставку
4. проставление статусов доставки

Seller Center не предоставляет возможности работать с заказами через csv-файлы. Вы можете обрабатывать заказы вручную через интерфейс или использовать обмены. Необходимый и достаточный набор обменов следующий:

* getOrders — получить новые заказы
* getMulitpleOrderItems — получить товары из заказа
* setStatusToReadyToShip — отметить собранные и упакованные заказы
* setStatusToShipped — отметить отправленные заказы
* setStatusToDelivered — статус заказа *Доставлен*
* setStatusToFailedDelivery — статус заказа *Не удалось доставить*

Подробное описание обменов представлено в разделах ниже.

### Получение заказов

Заказы, созданные на сайте Lamoda.ua и подтвержденные в Lamoda Call Center, передаются в Seller Center для дальнейшей обработки поставщиком. Если вы работаете через интерфейс, новые заказы загрузятся автоматически. Для работы по API используйте обмен getOrders и getMultipleOrderItems.

### Обработка заказов

Когда заказ собран и упакован, необходимо проинформировать об этом Seller Center обменом setStatusToReadyToShip. После передачи заказа в службу доставки отправьте обмен setStatusToShipped.

### Обновление статусов заказов

Lamoda требует, чтобы обработка заказов происходила строго в течение заданного интервала времени. Свидетельством обработки заказа является изменение статуса доставки заказа в Seller Center. Изменить статус возможно вручную через интерфейс или обменом через API.

Если вы работаете по модели Crossdocking — т.е. используете услуги LM Express по доставке заказов, то перевод заказа в статус Shipped (отправлена клиенту) будет происходить на стороне Lamoda.

### Полный объем интеграции



1. Получать заказы с помощью метода [getOrders](https://sellercenter.readme.io/docs/getorders) и товары из заказа с помощью [getMultipleOrderItems](https://sellercenter.readme.io/docs/getmultipleorderitems).
2. Сообщать при изменении статуса заказа с помощью API:
	* [setStatusToCanceled](https://sellercenter.readme.io/docs/setstatustocanceled): заказ отменен до начала сборки
	* [setStatusToReadyToShip](https://sellercenter.readme.io/docs/setstatustoreadytoship): заказ собран и упакован в посылку
	* [setStatusToShipped](https://sellercenter.readme.io/docs/setstatustoshipped): заказ передан в доставку (в случае собственной доставки — dropshipping, иначе метод недоступен)
	* [productUpdate](https://sellercenter.readme.io/docs/productupdate): обновить данные о товарных запасах (необязательно)
	* [setStatusToDelivered](https://sellercenter.readme.io/docs/setstatustodelivered): заказ доставлен
	* [setStatusToFailedDelivery](https://sellercenter.readme.io/docs/setstatustofaileddelivery): клиент отказался от заказа или не удалось доставить. Перед отправкой обмена необходимо получить список возможных причин недоставки методом [getFailureReasons](https://sellercenter.readme.io/docs/getfailurereasons).

Дополнительно, если подключена услуга доставки службой LM Express:

* [setStatusToPackedByMarketplace](https://sellercenter.readme.io/docs/setstatustopackedbymarketplace)V4: присвоить уникальный идентификатор Serial number для каждого SKU из заказа. Если у вас есть собственные идентификаторы товаров, укажите их в поле serialNumber. В противном случае оставьте пустыми.
* [createForwardManifest](https://sellercenter.readme.io/docs/createforwardmanifest): создать исходящую поставку с заказами, которые будут отправлены на склад в Быково.
* [setManifestStatusToShipped](https://sellercenter.readme.io/docs/setmanifeststatustoshipped): изменить статус созданной поставки на Shipped (отправлена со склада).

### Полный список методов по управлению заказами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API-методы | Описание | Альтернатива | Обязательно | Частота обновления |
| [getOrders](https://sellercenter.readme.io/docs/getorders) | Получить список заказов. | Интерфейс SC, csv-файл | Да | Раз в час. |
| [getOrder](https://sellercenter.readme.io/docs/getorder) | Получить заказ по id. | Интерфейс SC, csv-файл | Нет | По необходимости |
| [getMultipleOrderItems](https://sellercenter.readme.io/docs/getmultipleorderitems) | Получить товары из заказа. | Интерфейс SC, csv-файл | Да | После getOrders. |
| [setStatusToCanceled](https://sellercenter.readme.io/docs/setstatustocanceled) | Статус “отменен” может быть использован только до «Shipped» (передан в доставку), иначе — FailedDelivery (не удалось доставить).  | Интерфейс SC, csv-файл | Да | Не позже, чем через 2 часа после отмены заказа. |
| [setStatusToPackedByMarketplace](https://sellercenter.readme.io/docs/setstatustopackedbymarketplace) | Присвоить Serial numbers для каждого Sku в заказе  | Интерфейс SC, csv-файл | Для LM Express | Не позже, чем через 2 часа после getMultipleOrderItems |
| [setStatusToReadyToShip](https://sellercenter.readme.io/docs/setstatustoreadytoship) | Статус “собран” используется, когда товары собраны на складе. Для этого метода требуется указание [shipping provider.](https://sellercenter.lamoda.ru/user/shipping/)  | Интерфейс SC, csv-файл | Да | Не позже, чем через 2 часа после setStatusToPackedByMarketplace |
| [setStatusToShipped](https://sellercenter.readme.io/docs/setstatustoshipped) | Статус “передан в доставку” используется, когда заказ передан в службу доставки. | Интерфейс SC | Да  | Не позже, чем через 2 часа setStatusToReadyToShip |
| [setStatusToDelivered](https://sellercenter.readme.io/docs/setstatustodelivered) | Статус “доставлен” для товаров, действительно выкупленных клиентом.  | Интерфейс SC | Да  | Не позже, чем через сутки после setStatusToShipped |
| [setStatusToFailedDelivery](https://sellercenter.readme.io/docs/setstatustofaileddelivery) | Статус “не доставлен” для товаров, не выкупленных клиентом. | Интерфейс SC | Да | Не позже, чем через сутки после setStatusToShipped |
| [getFailureReasons](https://sellercenter.readme.io/docs/getfailurereasons) | Получить список возможных причин отказа/отмены. | Интерфейс SC | Нет | При изначальной интеграции и каждый раз перед setStatusToFailedDelivery. |
| [createForwardManifest](https://sellercenter.readme.io/docs/createforwardmanifest) | Создать исходящую поставку до склада приемки  | Интерфейс SC | Для LM Express | Не позже, чем через 2 часа после setStatusToReadyToShip |
| [setManifestStatusToShipped](https://sellercenter.readme.io/docs/setmanifeststatustoshipped) | Перевести статус отправки в Shipped (отправлена) | Интерфейс SC | Для LM Express | Не позже, чем через 2 часа после createForwardManifest |

## Упрощение интеграции

1. [Модуль для 1С](https://www.dropbox.com/s/cgvte902swb7brf/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%20%D0%B4%D0%BB%D1%8F%201%D0%A1.epf?dl=0)
2. [SDK для PHP](https://github.com/rocket-internet-berlin/SellerCenterSDK-PHP)
3. [SDK для Java](https://github.com/rocket-internet-berlin/SellerCenterSDK-Java)

## История изменений

| **Версия** | **Дата** | **Раздел** | **Комментарий** | **Автор** |
| --- | --- | --- | --- | --- |
| 0.2 | 16.06.2016 | Весь документ | Создание документа | Е. Леонова |
| 0.3 | 18.07.2016 | Управление товарами | Раздел дополнен подробными пояснениями | А. Булгакова |
| Управление заказами |
| 0.4 | 24.08.2016 | API интеграция | Добавлен раздел | А. Булгакова |