**Wildberries Markeplace**

Проект «Продажи со склада поставщика» позволяет поставщикам не только значительно расширить ассортимент на витрине **Wildberries** за счет, например, крупногабаритных товаров, но и предложить покупателям товары с невысоким спросом, держать по которым остатки на складах WB зачастую было экономически нецелесообразно.

Новый формат сотрудничества подразумевает:

* Размещение практически любого ассортимента на витрине **Wildberries**;
* Консолидацию клиентских заказов в течение дня;
* Ежедневную доставку консолидированного заказа на склад WB;
* Доставку до клиента силами **Wildberries**.

В рамках проекта **Marketplace** партнёр поставляет товар только под оформленные и уже оплаченные заказы. При этом контроль над ассортиментом на витрине, ценообразование, продвижение товара, работа с отзывами по-прежнему осуществляется через *Личный Кабинет Поставщика*.

Данные о заказах будут поступать в вашу учётную систему в реальном времени. Консолидированный к концу дня заказ необходимо подтвердить и доставить на выбранный склад в установленные сроки. Дальнейшую обработку заказа берет на себя **Wildberries**. Интерфейс для управления поставками и ассортиментом будет доступен в интерфейсе поставщика в ближайшее время.

В случае возникновения вопросов и пожеланий: **marketplace-support@wildberries.ru**.
Также мы приглашаем Вас вступить в официальный Telegram-канал нашего Проекта: [**https://t.me/marketplacewb**](https://t.me/marketplacewb)

**Страница разработчика | Проект «Продажа со склада поставщика»**

Данный документ описывает API, используемое для интеграции между системой поставщика и проектом «продажа со склада поставщика».

Авторизация

Авторизация реализуется с использованием заголовка **Authorization Bearer**

header: Authorization Bearer {token}

{token} — значение токена поставщика.

Потоки данных

Основные методы обмена данными:

1. Получение списка складов;
2. Передача данных и статусов сборочных заданий;
3. Передача данных об остатках на складах поставщика;

Доступно два способа обмена данными:

* Отправка данных о событии от одной системы к другой в момент совершения события (**POST**);
* Отложенный запрос данных о произошедних событиях от одной системы к другой (**GET**);

Формат сообщений

Запросы и данные ответов передаются в формате JSON.

При успешной обработке запроса каждая из систем должна возвращать следующее минимальное сообщение с использованием HTTP-кода 200:

{ success: true }

При неудачной обработке запроса ожидается следующий формат сообщения:

{ success: false, errors: [] }

Массив errors должен содержать список ошибок.

Список складов

Получение актуального списка складов поставщика:

**Production URL** https://marketplace-remotewh.wildberries.ru/api/public/v1/stores
**Test URL** https://marketplace-remotewh-test.wildberries.ru/api/public/v1/stores

Пример ответа

 [{

 "stores\_id": string, // Идентификатор склада поставщика

 "wb\_wh\_id": int, // Идентификатор склада WB

 "name": string, // Название склада

 "lat": float, // Широта

 "lon": float, // Долгота

 "border\_time": [string], // Время загрузки машины для доставки (например, ["9:00","18:30"])

 "nextday\_time": string, // Время, после которого заказы принимаются на следующий день

 "delivery\_hours": int, // Cрок доставки товаров на склад WB

 "priority\_wb\_status": int, // Приоритет относительно склада WB (1 - выше, 0 - ниже)

 "priority\_supplier": int, // Приоритет склада среди складов поставщика

 "status": int // Cтатус склада (1 - активен, 0 - не активен)

}]

Сборочные задания (заказы)

**Важно:**

* При отправке статусов заказов, ожидается, что поставщик передаст значение штрих-кода Wildberries, значение штрих-кода поставщика и значение chrt\_id.
* Ожидается, что поставщик предоставит информацию о распределении заказов по полибоксам с указанием идентификатор сортировочного центра, для которого предназначается каждый товар.
* В момент получения уведомления об окончании отгрузки производится автоматическая регистрация данных о поставке на портале поставщиков.

Передача данных о новом сборочном задании:

**URL** https://supplier-host/api/order/create

При повторном получении сборочного задания с тем же идентификатором новое задание формироваться не должно.

**Формат передаваемого объекта**

{

 "id": int, // Идентификатор задания на сборку

 "stores\_id": string, // Идентификатор склада поставщика, с которого отправляется заказ

 "wb\_wh\_id": int, // Идентификатор склада WB, на которой заказ должен быть доставлен

 "office\_id": int, // Идентификатор сортировочного центра WB

 "items": [

 {

 "chrt\_id": int, // Идентификатор артикула

 "request": int, // Количество вещей для сборки

 "price": int, // Стоимость товара

 "details": [

 {

 "wbsticker\_encoded": string, // Закодированное значение стикера (представляется на этикетке в формате Code-128)

 "wbsticker\_id": int, // Идентификатор этикетки

 "wbsticker\_svg\_base64": string, // Полное представление этикетки в векторном формате

 "wbsticker\_zpl": string, // Полное представление этикетки в формате ZPL

 "wbsticker\_id\_parts": {

 "a": string, // Первая часть идентификтаора этикетки (для печати подписи)

 "b": string // Вторая часть идентификатора этикетки

 },

 }

 ]

 }

 ]

}

Статусы сборочного задания

Сборочные задания могут находиться в следующих статусных состояниях:

|  |  |
| --- | --- |
| **Код** | **Статус** |
| 0 | Сборка закончена |
| 1 | Принято |
| 2 | Обработано частично |
| 3 | Отменено |
| 4 | Отгружено |

Метод проверки статуса сборочного задания на стороне поставщика:

**URL** https://supplier-host/api/order/{id\_задания}/status

Если задание отсутствует, ожидается ответ: { success: false, errors: ["Сборочное задание не найдено"] }.

**Формат объекта сборочного задания со статусом**

{

 "id": int, // Идентификатор задания на сборку

 "stores\_id": string, // Идентификатор склада поставщика, с которого отправляется заказ

 "wb\_wh\_id": int, // Идентификатор склада WB, на которой заказ должен быть доставлен

 "office\_id": int, // Идентификатор сортировочного центра WB

 "state": int // Статус задания

 "items":

 [

 {

 "chrt\_id": int, // Идентификатор артикула

 "request": int, // Количество вещей для сборки

 "price": int, // Стоимость товара

 "satisfied": int, // Количество позиций, принятых на сборку

 "rejected": int, // Количество отклонённых позиций

 "details": [

 {

 "wbsticker\_id": int, // Идентификатор этикетки

 ]

 ]

 }

 ]

}

Изменение статуса сборочного задания со стороны поставщика:

**Production URL** https://marketplace-remotewh.wildberries.ru/api/public/v1/supply\_tasks/status
**Test URL** https://marketplace-remotewh-test.wildberries.ru/api/public/v1/supply\_tasks/status

При передаче статусов сборочных заданий действуют следующие требования:

* В случае полной обработки задания (статус 0), в сообщении должны присутствовать данные для всего перечня товаров;
* При частичной сборке (статус 2), то пакет (со статусом 2) может содержать данные лишь о части вещей;
* В случае невозможности обработать задание полностью - ожидается пакет со статусом 3;
* Если вещи собраны и отгружены со склада, то ожидается статус 4;

Отмена сборочного задания:

**URL** https://supplier-host/api/order/{id\_задания}/cancel

Сервис поставщика должен включать в себя возможность отмены сборочного задания.

Остатки товаров на складах поставщика

Передача остатков на складах поставщика:

**Production URL** https://marketplace-remotewh.wildberries.ru/api/public/v1/stocks
**Test URL** https://marketplace-remotewh-test.wildberries.ru/api/public/v1/stocks

Рекомендуемая периодичность обмена остатками: полная выгрузка 1 раз в сутки, дельта-изменения в реальном времени, либо c периодичностью в несколько минут.

**Формат запроса от сервиса поставщика**

{

"data":

 [

 {

 "store\_id": string, // Идентификатор магазина

 "package\_id": int, // Идентификатор пакета

 "action\_code": string, // Режим передачи (FQNT полный набор значений остатков, DQNT набор изменившихся остатков)

 "stocks":

 [

 {

 "chrt": int, // Артикул

 "qnt": int // Величина остатка

 },

 {

 "chrt": int,

 "qnt": int

 }

 ]

 }

 ]

}

Получение списка соответствия значений артикулов

**URL** https://suppliers-stats.wildberries.ru/api/v1/supplier/nomenclatures?key={ключ поставщика}

Пример записи:

[

 {

 "brandName": "MERLION",

 "subjectName": "Куртки",

 "chrtId": 22943847,

 "sa": "ML17101/Синий",

 "nmId": 6502087,

 "tsName": "S",

 "saImt": "ML17101/",

 "saColor": "Синий",

 "barcode": "2003000220728",

 "priceRu": 5900,

 "contentsNamesList": ""

 }

]

Запуск интеграции

Для перевода интеграции в режим промышленной эксплуатации необходимо направить соответствующее уведомление в службу поддержки.

Перед началом получения информации о заказах необходимо:

* Убедиться в успешной загрузке данных складов;
* Убедиться в успешной загрузке данных об остатках;

**Документация по работе с сервисом карточек**

Данное API предназначено для работы по схеме “Со склада поставщика”.

Для работы с запросами требуется пройти аутентификацию пользователя. Процесс аутентификации описан в самом низу.

* Создание карточки/карточек товаров.
Для создания карточек существует 2 метода:
	+ Create. https://content-suppliers.wildberries.ru/card/create
	**Описание**: метод позволяет создать одну новую карточку товара. Возвращает созданную карточку.
	Тело запроса выглядит следующим образом:
* {
* "id": 1, // ID запроса для сопоставления ответа отправленному запросу, необходимы уникальные идентификаторы.
* // Для каждого запроса, рекомендуется избегать простых идентификаторов, таких как 1, 2, 3 и т.д.
* "jsonrpc": "2.0", // Версия протокола. Всегда должна быть "2.0".
* "params": { // Параметры.
* "card": {
* "countryProduction": "string", // Страна проиводитель.
* "object": "string" // Категория товара (Jeans, Books и другие).
* "addin": [ // Структура, содержащая характеристики карточки, общие для всех номенклатур и размеров.
* {
* "type": "string", // Название характеристики. Пример: "Composition".
* "params": [ // Массив значений характеристики. Пример для 1 эл-та: "{"type": "Brand", "params": [{"value": "brandName"}]}".
* {
* "count": 0, // Численное значение характеристики.
* "units": "string", // Единицы измерения характеристики ("см", "%" и другие).
* "value": "string" // Текстовое значение характеристики ("brandName").
* }
* ]
* }
* ],
* "nomenclatures": [ // Массив номенклатур товара.
* {
* "vendorCode": "string", // Артикул товара.
* "variations": [ // Массив вариаций товара. Одна цена - одна вариация.
* {
* "barcode": "string", // Штрихкод товара.
* "addin": [ // Структура, содержащая характеристики конкретной вариации товара.
* {
* "type": "string", // Название характеристики.
* // Массив значений хар-ки. Пример: "{"type": "Size", "params": [{"value": "S"}]}".
* "params": [ // У хар-ик, содержащих одно значение, массив будет содержать только 1 элемент.
* {
* "count": 0, // Численное значение характеристики.
* "units": "string", // Единицы измерения характеристики.
* "value": "string" // Текстовое значение характеристики.
* }
* ]
* }
* ]
* }
* ],
* "addin": [ // Структура, содержащая характеристики конкретной номенклатуры.
* {
* "type": "string", // Название характеристики.
* // Массив значений характеристики. Пример: "{"type": "Color", "params": [{"value": "Красный"}]}".
* "params": [ // У хар-ик, содержащих одно значение массив будет содержать только 1 элемент.
* {
* "count": 0, // Численное значение характеристики.
* "units": "string", // Единицы измерения характеристики.
* "value": "string" // Текстовое значение характеристики.
* }
* ]
* }
* ]
* }
* ]
* },
* "supplierID": "00000000-0000-0000-0000-000000000000" // Идентификатор поставщика. Выдаётся при регистрации поставщика.
* }
* }

* + BatchCreate https://content-suppliers.wildberries.ru/card/batchCreate
	**Описание**: метод позволяет создавать сразу много карточек. Возвращает, соответственно, много созданных карточек.
	Тело запроса выглядит следующим образом:

{

 "id": 1, // ID запроса для сопоставления ответа отправленному запросу, необходимы уникальные идентификаторы.

 // для каждого запроса, рекомендуется избегать простых идентификаторов, таких как 1, 2, 3 и т.д.

 "jsonrpc": "2.0", // Версия протокола. Всегда должна быть "2.0".

 "params": { // Параметры.

 "card": [ // Первый параметр запроса - массив карточек.

 {

 "countryProduction": "string", // Страна проиводитель.

 "object": "string" // Категория товара (Jeans, Books и другие).

 "addin": [ // Структура, содержащая характеристики карточки, общие для всех номенклатур и размеров.

 {

 "type": "string", // Название характеристики. Пример: "Composition".

 // Массив значений характеристики. У характеристик, содержащих одно значение, массив будет

 // содержать только 1 элемент. Пример: "{"type": "Brand", "params": [{"value": "brandName"}]}".

 "params": [

 {

 "count": 0, // Численное значение характеристики.

 "units": "string", // Единицы измерения характеристики ("см", "%" и другие).

 "value": "string" // Текстовое значение характеристики ("brandName").

 }

 ]

 }

 ],

 "nomenclatures": [ // Массив номенклатур товара.

 {

 "vendorCode": "string", // Артикул товара.

 "variations": [ // Массив вариаций товара. Одна цена - одна вариация.

 {

 "barcode": "string", // Штрихкод товара.

 "addin": [ // Структура, содержащая характеристики конкретной вариации товара.

 {

 "type": "string", // Название характеристики.

 // Массив значений характеристики. Пример: "{"type": "Size", "params": [{"value": "S"}]}".

 "params": [ // У хар-ик, содержащих одно значение, массив будет содержать только 1 элемент.

 {

 "count": 0, // Численное значение характеристики.

 "units": "string", // Единицы измерения характеристики.

 "value": "string" // Текстовое значение характеристики.

 }

 ]

 }

 ]

 }

 ],

 "addin": [ // Структура, содержащая характеристики конкретной номенклатуры.

 {

 "type": "string", // Название характеристики.

 // Массив значений характеристики. Пример эл-та массива: "{"type": "Color", "params": [{"value": "Красный"}]}".

 "params": [ // У хар-ик, содержащих одно значение, массив будет содержать только 1 элемент.

 {

 "count": 0, // Численное значение характеристики.

 "units": "string", // Единицы измерения характеристики.

 "value": "string" // Текстовое значение характеристики.

 }

 ]

 }

 ]

 }

 ]

 },

 // Следующая карточка и так далее.

 {

 }

 ],

 "supplierID": "00000000-0000-0000-0000-000000000000" // ID поставщика. Выдаётся при регистрации поставщика.

 }

}

**ADDIN**
На данный момент реализована поддержка только следующих полей для параметров Addin:
Для карточки: Brand, Title, Description.
Для номенклатуры: Collection, Season, Sex, AdditionalColors, MainColor, Composition.
Для вариации: Price, Size.
type должны совпадать с названиями из этого списка.

* Получение карточки/карточек.
Для получения карточек существует 2 метода:
	+ CardByID. https://content-suppliers.wildberries.ru/card/cardById
	**Описание**: метод позволяет получить карточку поставщика с указанным ID.
	Тело запроса выглядит следующим образом:
* {
* "id": 1,
* "jsonrpc": "2.0",
* "params": {
* "cardID": "00000000-0000-0000-0000-000000000000", // ID карточки. Можно получить в ответе сервиса после создания карточки или методом "List", который рассмотрен ниже.
* "supplierID": "00000000-0000-0000-0000-000000000000" // ID поставщика. Выдаётся при регистрации поставщика.
* }
* }

* + List. https://content-suppliers.wildberries.ru/card/list
	**Описание**: метод позволяет получить список карточек поставщика по указанным параметрам. Фильтрация позволяет исключить из итогового списка карточек те, в которых указанное поле column содержит значение excludedValues. Поиск позволяет оставить в итоговом списке карточек только те, у которых поле column имеет значение search, если это число, если это строка, то значение поля начинается с search. Сортировка позволяет отсортировать список по полю column в порядке order. query содержит значения limit и offset, который указывают соответственно на макисмальное количество карточек, которые надо вывести и на количество карточек, которые с самого начала списка нужно пропустить.
	Наименования полей соответствуют пути до этого поля через точку. Так, например, чтобы добраться до поля barcode, понадобится путь nomenclatures.variations.barcode. Исключения - параметры, лежащие внутри структуры addin. Чтобы добраться до них, требуется в пути после addin через точку указать type этого параметра. Например, чтобы получить brand, потребуется путь addin.Brand. Brand написан с большой буквы, так как type был написан тоже с большой.
	Тело запроса выглядит следующим образом:

{

 "id": 1,

 "jsonrpc": "2.0",

 "params": {

 "filter": {

 "filter": [ // Исключение карточек указанные поля которых содержат указанные значения.

 {

 "column": "nomenclatures.addin.Color",

 "excludedValues": "Красный"

 },

 {

 "column": "object",

 "excludedValues": "Ноутбуки"

 }

 ],

 "find": [ // Исключение карточек, указанные поля которых не содержат указанные значения.

 {

 "column": "nomenclatures.variations.addin.Size",

 "search": "M"

 }

 ],

 "order": { // Сортировка по указанному полю в указанном порядке.

 "column": "createdAt",

 "order": "asc"

 }

 },

 "query": { // Пагинация.

 "limit": 10,

 "offset": 20

 },

 "supplierID": "00000000-0000-0000-0000-000000000000" // Идентификатор поставщика.

 }

}

В результате вышеприведённого запроса нам вернутся все карточки поставщика 00000000-0000-0000-0000-000000000000, среди которых не будет карточек с красными цветами и ноутбуков, у всех карточек будут товары с размером “S”. И все эти карточки будут отсортированы по дате создания в порядке возрастания (сначала старые).

* Аутентификация.
Авторизация происходит через СМС-уведомление или уведомление в приложении Wildberries. Для этого необходимо выполнить 2 простых шага:
	+ Запросить код авторизации. https://content-suppliers.wildberries.ru/passport/api/v2/auth/login\_by\_phone
	В запросе требуется ввести свой номер телефона phone. В ответе придёт структура, состоящая из токена и времени в милисекундах, за которое нужно ввести код. Код придёт на телефон в виде СМС или или уведомления в приложении.
	Тело запроса:
	+ {
	+ "phone": ,,"79000000000",
	+ "is\_terms\_and\_conditions\_accepted": true
	+ }

* + Подтверждение токена и кода. https://content-suppliers.wildberries.ru/passport/api/v2/auth/login
	В запрос требуется вставить token, пришедший из предыдущего запроса и notify\_code из СМС.
	Тело запроса:
	+ {
	+ "token":"AcDT8wyihNLuCzBD3ClM\_ZA8pXzdDbSzGxrPQre6WfrScRrpTsZ4HMiAEu140KBtbhFJp5F7RujinaY",
	+ "options": {
	+ "notify\_code":"000000"
	+ }
	+ }