DPD API Reference Documentation

Release 2.0

Portal Labs, LLC

Oct 06, 2022

CONTENTS

Contents:

CHAPTER

ONE

DPD API

Note: This API is supported on the hosted site (*.dpdcart.com) *only. We highly recommend upgrading to the hosted site as soon as possible as the lightbox cart does not support all DPD features.*

1.1 About

Important: This document is intended for programmers only.

The DPD API is a read-only window into your DPD account. With the API you are able to view all of the data that is available in your account.

The API does not support modifying any data at this time.

CHAPTER

TWO

AUTHENTICATION

Authentication is performed using standard HTTP Basic Authentication. The username is your account username and the password is the API password available on your profile.

Your username and API password are securely encrypted by the SSL connection.

You can test your username and password on the CLI using cURL:

```
curl -u [username] https://api.getdpd.com/v2/
```

Enter your API password when prompted.

CHAPTER

THREE

LIMITATIONS

3.1 Pagination

All collection GET methods are limited to a result set of 100 records and support paging by means of the “page” querystring parameter. Omission of this parameter will attempt to return all results; this is not guaranteed to work.

> GET /purchases?page=2

To retreive the entire set, start with page=1 and increment the page parameter until you recieve the response:

{“status”: “NOTFOUND”}

3.2 Time Zone

All datetime parameters are assumed to be in your account’s time zone. You can specify a different time zone by using a datetime format that includes a time zone such as RFC 3389/ISO 8601 (2004-02-12T15:19:21+00:00) or RFC 2822 (Thu, 21 Dec 2000 16:01:07 +0200).

CHAPTER

FOUR

RESPONSE AND HTTP STATUS CODES

The payload is a JSON object on success. This object is either a hash for single resources (such as show) or a list of hashes for multiple resources (such as list).

Standard HTTP status codes are used:

- 200 OK - The request has succeeded
- 401 Unauthorized - Your credentials are invalid
- 403 Forbidden - You do not have access to the resource
- 404 Not Found - The resource could not be found
- 412 Precondition Failed - The request could not be completed because of a validation error
- 500 Internal Server Error - The server encountered an unexpected condition which prevented it from fulfilling the request.
- 503 Service Unavailable - The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

CHAPTER

FIVE

GENERAL METHODS

5.1 Ping

You can ping the API by issuing a GET to the root URL.

GET /

Response:

HTTP/1.0 200 OK

{“status”:”SUCCESS”}

5.2 Verify Notification

You can verify a POST from the URL integration. You must POST all of the parameters that were POSTed to your site.

POST /notification/verify

Response:

The response will either be *VERIFIED* or *INVALID*.

CHAPTER

SIX

STOREFRONTS

Storefronts are also called stores and websites in the DPD admin.

6.1 List Storefronts

Returns a list of storefronts in your account:

```
GET /storefronts
```

Response:

```
[{"id":1,"name":"...",...},{"id":2,"name":"...",...}]
```

6.2 Get Storefront

Returns information about a single storefront:

```
GET /storefronts/:id
```

The response is a JSON hash with the following attributes:

- `id` (integer) - Unique ID
- `created_at` (UNIX timestamp) - When the storefront was created.
- `updated_at` (UNIX timestamp) - When the storefront was last updated.
- `name` (string) - Storefront name.
- `url` (string) - Storefront URL.
- `contact_name` (string) - Contact name.
- `contact_email` (string) - Contact email.
- `currency` (string) - Currency.
- `type` (string) - Type of storefront. `cart`, `subscription`, or `clickbank`.
- `subdomain` (string) - Subdomain (v2 storefronts only). Append to dpdcart.com to get the full domain.

CHAPTER

SEVEN

PRODUCTS

7.1 List Products

Returns a list of products in your account.:

```
GET /products
```

Parameters

- storefront_id - Storefront to filter on. Default is all storefronts.

Response:

```
[{"id":1,"name":"...",...},{"id":2,"name":"...",...}]
```

7.2 Get Product

Returns information about a single product.:

```
GET /products/:id
```

The response is a JSON hash with the following attributes:

- id (integer) - Unique ID
- created_at (UNIX timestamp) - When the product was created.
- updated_at (UNIX timestamp) - When the product was last updated.
- storefront_id (integer) - Storefront ID.
- name (string) - Name of the product.
- description (string) - Short description.
- long_description (string) - Long description.
- price (string) - Default price.
- sku (string) - SKU
- image_file_name (string) -
- image_updated_at (integer; UNIX timestamp)
- mime_type (string)

- `file_size` (integer)
- `file_name` (string)
- `prices` (array)
- `weight` (integer)
- `visibility` (integer; 0 = hidden, 1 = displayed)

The *prices* array has the following attributes:

- `id` (integer)
- `name` (string)
- `price` (string)

7.3 Product Image URL

When an image is uploaded to a product, multiple sizes are created. You can use `50x50`, `125x125`, `100x100`, `150x150`, `250x250`, `300x300`, `350x350`, `400x400`, `450x450`, `500x500`, `550x550`, `600x600`, `800x800`, `1000x1000` for the `size` parameter. The image URL is:

```
https://d2beuh40lcdzfb.cloudfront.net/products/ID/SIZE/IMAGE_FILE_NAME
```

CHAPTER

EIGHT

PURCHASES

8.1 List Purchases

Returns a list of purchases in your account.:

```
GET /purchases
```

Parameters

- `status` - Purchase status. See below for a full list of status codes.
- `product_id` - Product ID
- `storefront_id` - Storefront ID
- `customer_id` - Customer ID
- `subscriber_id` - Subscriber ID
- `customer_email` - Customer email address. Does a "startswith" match. Not case sensitive.
- `customer_first_name` - Customer first name. Does a "startswith" match. Not case sensitive.
- `customer_last_name` - Customer last name. Does a "startswith" match. Not case sensitive.
- `date_min` - Purchase created after. Accepts any format that PHP's strtotime accepts.
- `date_max` - Purchase created before. Accepts any format that PHP's strtotime accepts.
- `total` - Purchase total. Behavior is controlled by the `total_op` param.
- `total_op` - How to compare the `total` parameter. eq (equals; default), ne (not equal), gt (greater than), lt (less than)
- `ship` - Purchases with tangible goods that have not been shipped.

Response:

```
[{"id":1,"status":"...",...},{"id":2,"status":"...",...}]
```

8.2 Get Purchase

Returns information about a single purchase.:

```
GET /purchases/:id
```

The response is a JSON hash with the following attributes:

- `id` (integer) - Unique ID
- `created_at` (UNIX timestamp) - When the purchase was created.
- `updated_at` (UNIX timestamp) - When the purchase was last updated.
- `storefront_id` (integer) - Storefront ID.
- `salt` (string) - Unique hash code.
- `status` (string) - Purchase status (see below).
- `currency` (string) - Currency.
- `subtotal` (string) - Subtotal.
- `discount` (string) - Discount amount.
- `tax` (string) - Tax amount.
- `shipping` (string) - Shipping amount.
- `total` (string) - Total (`subtotal` + `discount` + `tax` + `shipping`)
- `processor_fee` (string) - Total fee collected by the payment processor. Not recorded for all payment processors and is not reflected in the `total`.
- `buyer_email` (string) - Buyer's email address.
- `buyer_firstname` (string) - Buyer's first name.
- `buyer_lastname` (string) - Buyer's last name.
- `customer` (object) - Customer information object.
- `customer_notes` (string) - Notes from the customer.
- `custom_fields` (array) - A list of custom fields and their responses.
- `shipping_firstname` (string) - Shipping first name.
- `shipping_lastname` (string) - Shipping last name.
- `shipping_street` (string) - Shipping street address.
- `shipping_street2` (string) - Shipping street address line 2.
- `shipping_city` (string) - Shipping city.
- `shipping_state` (string) - Shipping state/province code.
- `shipping_zip_code` (string) - Shipping ZIP/postal code.
- `shipping_country_code` (string) - Shipping country code.
- `shipping_country` (string) - Full shipping country name.
- `shipping_phone` (string) - Shipping phone number.
- `shipping_vatin` (string) - Shipping VAT identifier.

- `shipping_business_name` (string) - Shipping business name.
- `shipping_carrier` (string) - Shipping carrier (UPS, FedEx, etc)
- `shipping_tracking_number` (string) - Tracking number for the shipment.
- `requires_shipping` (boolean) - `true` if the purchase requires shipping to fulfill.
- `tangibles_to_ship` (integer) - Number of goods that have not been shipped. `0` if all have been shipped. `null` if shipping wasn't required.
- `billing_firstname` (string) - Billing first name.
- `billing_lastname` (string) - Billing last name.
- `billing_street` (string) - Billing street address.
- `billing_street2` (string) - Billing street address line 2.
- `billing_city` (string) - Billing city.
- `billing_state` (string) - Billing state/province code.
- `billing_zip_code` (string) - Billing ZIP/postal code.
- `billing_country_code` (string) - Billing country code.
- `billing_card_type` (string) - Billing card type.
- `billing_card_number` (string) - Partial billing card number, padded with asterisks.
- `line_items` (array)
- `coupons` (array) - The coupons used with this purchase (if any)
- `ip_address` (string) - IP address of the checkout location
- `marketing_optin` (boolean) - `true` if the customer opted to receive marketing communications.

The `line_items` array has the following attributes:

- `id` (integer) Unique ID
- `purchase_id` (integer) Purchase ID
- `product_id` (integer) Product ID
- `price` (string) Line price.
- `quantity` (integer) Line quantity.
- `subtotal` (string) Line subtotal (`price` * `quantity`)
- `tax` (string) Line tax.
- `discount` (string) Line discount.
- `total` (string) Line total (`subtotal` + `tax` + `discount`)
- `product_name` (string) Product name.
- `download_limit` (integer) How many times the product may be downloaded. 0 is "unlimited".
- `download_count` (integer) How many times the product has been downloaded.
- `expires_at` (UNIX timestamp) When the download expires.
- `product_keys` (array) All of the product keys delivered with this line item.
- `has_all_keys` (boolean) `true` if all of the keys were successfully generated for the line item.

The custom_fields array has the following attributes:

- label (string) Custom field label.
- response (string) Customer's response.

The coupons array has the following attributes:

- name (string) The coupon name.
- code (string) The code used to redeem the coupon.
- discount_amount (string) The discount. Either as a percentage or as money.
- discount_type (string) The type of discount. Either money or percent.

The customer object has the following attributes:

- id (integer) - Unique ID
- firstname (string) - Customer's first name.
- lastname (string) - Customer's last name.
- email (string) - Customer's email address.
- receives_email (boolean) - Boolean on whether your customer recieves email from your product updates.
- created_at (UNIX timestamp) - When the purchase was created.
- updated_at (UNIX timestamp) - When the purchase was last updated.

8.2.1 Purchase Status Codes

A purchase's status will be one of the following:

- ACT - Active. The purchase has completed and is paid.
- PND - Pending. The purchase is pending payment.
- RFD - Refunded. The purchase has been refunded in partial or full.
- ERR - Error. There was an error verifying the purchase's payment.
- CAN - Canceled. The purchase was canceled.
- HLD - Held. The purchase was held due to fraud or blacklist checks.

8.3 Reactivate Purchase

Reactivates a purchase. This method is identical in behavior to reactivating a purchase through the web site:

```
POST /purchases/:id/reactivate
```

Parameters

- customer_email (string) - Customer email address. Omit or leave blank to send to the email address associated with the purchase.
- refulfill (boolean) - Set to refulfill the purchase.

The response is a JSON hash with the following attributes:

- status (string) - "OK" on success

CHAPTER

NINE

SUBSCRIBERS

9.1 List Subscribers

Returns a list of subscribers in your account:

```
GET /storefronts/:storefront_id/subscribers
```

Parameters

- `storefront_id` - **Required** Storefront
- `username` - Username (the subscriber's email address)

Response:

```
[{"id":1,"username":"...",...},{"id":2,"username":"...",...}]
```

9.2 Get Subscriber

Returns information about a single subscriber:

```
GET /storefronts/:storefront_id/subscribers/:id
```

The response is a JSON hash with the following attributes:

- `id` (integer)
- `created_at` (UNIX timestamp)
- `updated_at` (UNIX timestamp)
- `username` (string)
- `last_login_at` (UNIX timestamp)
- `subscription` (hash)

The *subscription* hash has the following attributes:

- `id` (integer)
- `status` (string) - See below for values.
- `price` (string)
- `period` (integer)

- unit (char)
- tax_amount (string)
- trial_price (string)
- trial_period (integer)
- trial_unit (char)
- trial_tax_amount (string)
- created_at (UNIX timestamp)
- updated_at (UNIX timestamp)
- started_at (UNIX timestamp)
- ended_at (UNIX timestamp)
- trial_started_at (UNIX timestamp)
- last_payment_at (UNIX timestamp)
- next_payment_at (UNIX timestamp)

9.3 Subscription Statuses

- NEW - Subscription has not been activated. Subscriber is in the process of checking out or has abandoned signup.
- TRIAL - Subscription is in the trial period.
- ACTIVE - Subscription is active without any payment due.
- CANCELED - Subscription has been canceled but the current term has not ended.
- CLOSED - Subscription has been canceled and closed.
- PAST_DUE - Subscription is active but payment is past due. Subscription will be CLOSED if payment is not received.

You can validate a subscriber's subscription by looking at the subscription.status field. Values of ACTIVE, TRIAL, and PAST_DUE receiving service.

9.4 Verify Subscriber

Simple method to get a subscriber's subscription status:

```
GET /storefronts/:storefront_id/subscribers/verify
```

Parameters

- storefront_id - **Required** Storefront
- username - Username (the subscriber's email address)
- id - Subscriber ID

Either the username or subscriber_id parameter must be provided. If the subscriber is found it will return the subscriber's subscription status.

Example

```
GET /storefronts/1/subscribers/verify?username=test%40example.com
```

CHAPTER

TEN

CUSTOMERS

10.1 List Customers

Returns a list of customers in your account:

```
GET /customers
```

Parameters

- `email` - Customer email address. Does a “startswith” match. Not case sensitive.
- `first_name` - Customer first name. Does a “startswith” match. Not case sensitive.
- `last_name` - Customer last name. Does a “startswith” match. Not case sensitive.
- `product_id` - Product ID that the customer has purchased.
- `receives_newsletters` - Boolean on whether your customer recieves email from your product updates.
- `date_min` - Customer created after. Accepts any format that PHP’s strtotime accepts.
- `date_max` - Customer created before. Accepts any format that PHP’s strtotime accepts.

Response:

```
[{"id":1,"status":"...",...},{"id":2,"status":"...",...}]
```

10.2 Get Customer

Returns information about a single purchase.:

```
GET /customers/:id
```

The response is a JSON hash with the following attributes:

- `id` (integer) - Unique ID
- `firstname` (string) - Customer’s first name.
- `lastname` (string) - Customer’s last name.
- `email` (string) - Customer’s email address.
- `receives_email` (boolean) - Boolean on whether your customer recieves email from your product updates.
- `created_at` (UNIX timestamp) - When the purchase was created.

- `updated_at` (UNIX timestamp) - When the purchase was last updated.