

---

# Python Paypal Api

*Release 0.1.1*

**Daniel Alvaro**

**Mar 19, 2023**



# CONTENTS

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Credentials</b>	<b>5</b>
<b>3</b>	<b>Current resources</b>	<b>11</b>
<b>4</b>	<b>Disclaimer</b>	<b>41</b>
	<b>Index</b>	<b>43</b>



This project helps you using python 3.8 to use the *Python Paypal API*.



## INSTALLATION

You can install using pip

```
pip install python-paypal-api
```





## CREDENTIALS

### 2.1 How to Authenticate

Please check as this example provides all ways to authenticate trough this API.

Paypal Rest API, support Basic and Bearer authentication so lets go from simple to complex. For any test you need have at least a `client_id` and a `client_secret` to run it.

**client\_id** string The ID obtained from Paypal

**client\_secret** string The password obtained from Paypal

**client\_mode** enum PRODUCTION or SANDBOX

#### 2.1.1 HTTPBasicAuth

The most simple way is to use the auth as a combination of `client_id` and `client_secret`, `requests` will create an HTTP-BasicAuth that will allow you to get access to Paypal Rest API:

#### Pass Credentials As Tuple

**credentials** tuple

```
credentials = ("your-client-id", "your-client-secret", "SANDBOX")
```

### Example python

```
from python_paypal_api.api import Identity
from python_paypal_api.base import PaypalApiException

my_credentials = ("your-client-id", "your-client-secret", "SANDBOX")

try:
    result = Identity(credentials=my_credentials).get_userinfo()
    payload = result.payload
    print(payload)
except PaypalApiException as error:
```

(continues on next page)

(continued from previous page)

```
print(error)
```

## 2.1.2 OAuth 2.0 Bearer

Bearer Tokens are the predominant type of access token used with OAuth 2.0. A Bearer Token is an opaque string, not intended to have any meaning to clients using it. This way will call the paypal oauth2 api and request for a token that could be used for all the api calls until get expired. There is 3 ways to achieve that: from environ, from code and from config.

### Pass Credentials From Environ

```
client_id os._Environ >> str required
client_secret os._Environ >> str required
client_mode os._Environ >> str
```

```
import os
os.environ["client_id"] = "your-client-id"
os.environ["client_secret"] = "your-client-secret"
os.environ["client_mode"] = "your-client-mode"
```

### Example python

```
import os
from python_paypal_api.api import Identity
from python_paypal_api.base import PaypalApiException

os.environ["client_id"] = "your-client-id"
os.environ["client_secret"] = "your-client-secret"
os.environ["client_mode"] = "your-client-mode"

try:
    result = Identity().get_userinfo()
    payload = result.payload
    print(payload)
except PaypalApiException as error:
    print(error)
```

### Pass Credentials From Code

credentials dict

```
credentials = dict(
    client_id="your-client-id",
    client_secret="your-client-secret",
    client_mode="SANDBOX"
)
```

### Example python

```

from python_paypal_api.api import Products
from python_paypal_api.base import PaypalApiException

my_credentials = dict(
    client_id="your-client-id",
    client_secret="your-client-secret",
    client_mode="SANDBOX"
)

try:
    result = Products(credentials=my_credentials, debug=True).list_products()
    payload = result.payload
    print(payload)
except PaypalApiException as error:
    print(error)

```

## Pass Credentials From Config

An example config file is provided in this repository, it supports multiple accounts. The confuse will search the system for a config file named config.yaml in the following search paths.

- macOS: ~/.config/python-paypal-api and ~/Library/Application Support/python-paypal-api
- Other Unix: ~/.config/python-paypal-api and /etc/python-paypal-api
- Windows: %APPDATA%\python-paypal-api where the APPDATA environment variable falls back to %HOME%\AppData\Roaming if undefined

Content example of a config.yaml file.

```

version: '1.0'

configuration:

  production:
    client_id: 'your-client-id'
    client_secret: 'your-client-secret'
    client_mode: 'PRODUCTION'

  default:
    client_id: 'your-client-id-sandbox'
    client_secret: 'your-client-secret-sandbox'

```

credentials str

```
credentials = "production"
```

### Example python

```

from python_paypal_api.api import Identity
from python_paypal_api.base import PaypalApiException

try:

```

(continues on next page)

(continued from previous page)

```
result = Identity(credentials="production", debug=True).get_userinfo(
    payload = result.payload
    print(payload)
except PaypalApiException as error:
    print(error)
```

View full possibilities From Config File

**Warning:** If you are using HTTPBasicAuth this do not applies as every call will be based on client\_id and client\_secret combination. No token is involved.

## 2.2 Storing Credentials

If you use OAuth 2.0 Bearer to avoid calls to the oauth paypal authentication api to obtain the token for Bearer authentication, there is 2 ways to deal with that:

### 2.2.1 Cache

The python paypal api will use the `cachetools` package to store the token in a LRU cache. This provides you a way that the token is stored in cache however it will be alive since the python script is executing and includes all possible calls to the api. When the script is ended the cache is flushed and other script will not be able to obtain a cached token one and will call the api to obtain a current or new one.

### 2.2.2 File

If you provide a store\_credentials bool in the client, True or False (default) The python paypal api will store the token in a file linked as an md5 of the client\_id. All next calls will find the file first, check if the token is not expired and use it. If the token is expired will recreate a new file with the new token and the actual expiring time. For security reasons the token file could be encrypted or not, using safe bool in the client, True or False (default). If safe=True, will use `cryptography` package to generate a Fernet key and encrypt the content. The .key file and the .token file will be stored in the confuse config folder `config = confuse.Configuration('python-paypal-api')` View more details about `confuse` in the From Config File help.

### Simple

Pass the keywords parameters to the client as bool:

```
Identity(store_credentials=True).get_userinfo() # Mode Store Safe No
```

```
Identity(store_credentials=True, safe=True).get_userinfo() # Mode Store Safe_
↪ Yes
```

## Complex

You could also pass the configuration as a dict. In that way you could overwrite the default storing folders for both token and key. Be aware that the folders need to exist and changing the configuration without moving the files will create a new token with a new key. Please check full examples to see the way you can use and choose it wisely.

**safe** bool Whether if the token will be encrypted or not. True or False

**token\_path** string The absolut path to the folder wish to store the token

**key\_path\_name** string The absolut path to the file where you wish to store the key.

**key\_value** string Fernet key must be 32 url-safe base64-encoded bytes. Use key\_value will invalidate a key\_path\_name.

This examples store an encrypted token in the folder `store_encrypted` and a key named `sandbox_secret.key` in the folder `store_key`.

```
config = \
{
    "safe": True,
    "token_path": "/Users/your-user/Desktop/store_encrypted",
    "key_path_name": "/Users/your-user/Desktop/store_key/sandbox_secret.key",
}

Identity(store_credentials=config, safe=True).get_userinfo() # Mode Store Safe Yes
```

You could also use your own key and not store it, so in that way you will not use a `key_path_name` and you should use a `key_value`.

```
config = \
{
    "safe": True,
    "token_path": "/Users/your-user/Desktop/store_encrypted",
    "key_value": "4BekoTZNO5aK4H0tIwkGYbq0IegqE5Y6w0bUoqVJqzk=",
}

Identity(store_credentials=config, safe=True).get_userinfo() # Mode Store Safe Yes
```

**Warning:** Take care about that the token filename is immutable and if you use diferent configuration could result in overwriting the token with a new key if you change it.



## CURRENT RESOURCES

Create and manage your site's payment functions using PayPal API collections.

### 3.1 Add Tracking API

#### Add Tracking API

Merchants can use the PayPal Add Tracking API to manage tracking information. Merchants can add tracking numbers and associated information to PayPal. After adding these details to PayPal, merchants can:

- Update tracking details.
- Show tracking details.
- Cancel tracking numbers.

For more information, see the [Add Tracking API Overview](#) and [Add Tracking API Integration Guide](#).

**class** `python_paypal_api.api.Tracking(*args, **kwargs)`

Use the `/trackers` resource to create and manage tracking information for PayPal transactions.

**put\_tracking**(*self*, *id*: str, \*\*kwargs) → ApiResponse

PUT `/v1/shipping/trackers/{id}`

Updates or cancels the tracking information for a PayPal transaction, by ID. To cancel tracking information, call this method and set the status to CANCELLED. For more information, see [Update or cancel tracking information](#).

**\*\*args:**

**id** string required

The ID of the tracker in the `transaction_id-tracking_number` format.

**\*\*kwargs:** Request body

**status** enum required

The status of the item shipment. For allowed values, see Shipping Statuses.

The possible values are:

- CANCELLED. The shipment was cancelled and the tracking number no longer applies.

- **DELIVERED.** The item was already delivered when the tracking number was uploaded.
- **LOCAL\_PICKUP.** Either the buyer physically picked up the item or the seller delivered the item in person without involving any couriers or postal companies.
- **ON\_HOLD.** The item is on hold. Its shipment was temporarily stopped due to bad weather, a strike, customs, or another reason.
- **SHIPPED.** The item was shipped and is on the way.
- **SHIPMENT\_CREATED.** The shipment was created.
- **DROPPED\_OFF.** The shipment was dropped off.
- **IN\_TRANSIT.** The shipment is in transit on its way to the buyer.
- **RETURNED.** The shipment was returned.
- **LABEL\_PRINTED.** The label was printed for the shipment.
- **ERROR.** An error occurred with the shipment.
- **UNCONFIRMED.** The shipment is unconfirmed.
- **PICKUP\_FAILED.** Pick-up failed for the shipment.
- **DELIVERY\_DELAYED.** The delivery was delayed for the shipment.
- **DELIVERY\_SCHEDULED.** The delivery was scheduled for the shipment.
- **DELIVERY\_FAILED.** The delivery failed for the shipment.
- **INRETURN.** The shipment is being returned.
- **IN\_PROCESS.** The shipment is in process.
- **NEW.** The shipment is new.
- **VOID.** If the shipment is cancelled for any reason, its state is void.
- **PROCESSED.** The shipment was processed.
- **NOT\_SHIPPED.** The shipment was not shipped.

**transaction\_id** string required

The PayPal transaction ID.

**carrier** enum required

The carrier for the shipment. Some carriers have a global version as well as local subsidiaries. The subsidiaries are repeated over many countries and might also have an entry in the global list. Choose the carrier for your country. If the carrier is not available for your country, choose the global version of the carrier. If your carrier name is not in the list, set `carrier_other_name` to `OTHER`. For allowed values, see Carriers.

The possible values are:

- **ACOMMERCE.** [aCommerce](#)



- PHL\_2GO. 2GO Philippines
- AU\_DHL\_EXPRESS. DHL Express Australia.
- BEL\_DHL. DHL Belgium.
- DE\_DHL\_DEUTSCHE\_POST\_INTL\_SHIPMENTS. Deutsche Post DHL Post International Germany
- IE\_AN\_POST\_REGISTERED. Post Registered Ireland
- AU\_AU\_POST. [Australian Postal Corporation](#)
- SPEEDEXCOURIER. Speedex Courier.
- UK\_ROYALMAIL\_SPECIAL. Royal Mail Special Delivery UK
- FR\_COLIS. Colis France.
- VNPOST\_EMS. Post EMS Vietnam.
- NL\_FEDEX. Federal Express Netherlands
- CN\_EMS. EMS China.
- IT\_POSTE\_ITALIANE. Poste Italiane.
- HK\_DHL\_ECOMMERCE. DHL eCommerce Hong Kong.
- ARAMEX. Aramex.
- AU\_OTHER. Other - Australia.
- TW\_CHUNGHWA\_POST. Chunghwa POST Taiwan
- DPEX. [DPEX Worldwide](#)
- POST\_SERBIA. [Pošta Srbije](#)
- PL\_POCZTEX. Pocztex
- CNEXPS. CN Express China.
- DIRECTLOG. Directlog.
- ES\_CORREOS\_DE\_ESPANA. Correos de Espana
- BE\_KIALA. Kiala Point Belgium
- ALPHAFAST. Alphafast.
- UKR\_POSHTA. [Ukrposhta - Ukraine's National Post](#)
- CN\_FEDEX. Federal Express China
- BUYLOGIC. [Buylogic](#)
- IT\_DHL\_ECOMMERCE. DHL eCommerce Italy.
- NINJAVAN\_MY. Ninjavan Malaysia.
- JPN\_YAMATO. Yamato Japan.
- POSTNORD\_LOGISTICS. Post Nord Logistics.
- US\_DHL\_GLOBALFORWARDING. DHL Global Forwarding US.
- IT\_SGT. SGT Corriere Espresso Italy.
- NINJAVAN\_PHILIPPINES. Ninja Van Philippines.

- EKART. Ekart.
- IDN\_WAHANA. Wahana Indonesia.
- FR\_GLS. [General Logistics Systems \(GLS\) France](#)
- IDN\_POS\_INT. Pos Indonesia International.
- DE\_HERMES. Hermes Germany.
- PRT\_CHRONOPOST. Chronopost Portugal.
- MYS\_MYS\_POST. [Pos Malaysia](#)
- WEBINTERPRET. WebInterpret.
- BG\_BULGARIAN\_POST. [Bulgarian Post](#)
- NL\_TPG. TPG Post Netherlands
- CA\_CANPAR. Canpar Courier Canada
- MYS\_AIRPAK. Airpak Malaysia.
- MEX\_SENDA. Senda Mexico.
- LANDMARK\_GLOBAL. Landmark Global.
- UK\_NIGHTLINE. Nightline UK.
- JP\_UPS. United Parcel Service Japan
- UK\_DHL. DHL UK.
- SG\_SG\_POST. Singapore Post.
- PHL\_AIRSPEED. Airspeed Philippines.
- DHL. DHL Express.
- KR\_KOREA\_POST. Korea Post
- JP\_KURO\_NEKO\_YAMATO\_UNYUU. Kuro Neko Yamato Unyuu Japan
- IE\_AN\_POST\_SWIFTPOST. Swift Post Ireland
- CUCKOOEXPRESS. Cuckoo Express.
- FR\_OTHER. Other - France.
- FASTRAK\_TH. Fastrak Thailand.
- AU\_DHL\_ECOMMERCE. DHL eCommerce Australia.
- DE\_UPS. United Parcel Service Germany
- ESHOPWORLD. EShopWorld.
- INTERNATIONAL\_BRIDGE. International Bridge.
- FR\_COLIPOSTE. Coliposte France
- AU\_AUSTRIAN\_POST. Austrian Post.
- IND\_DELHIVERY. [Delhivery India](#)
- DE\_TNT. TNT Germany.
- GLOBAL\_DHL. DHL Global.
- US\_DHL\_PARCEL. DHL Parcel US.

- NL\_UPS. United Parcel Service Netherlands
- GB\_APC. [APC Overnight UK](#)
- IDN\_TIKI. Tiki Indonesia.
- HERMES. Hermes.
- ESP\_NACEX. [Nacex Spain](#)
- NL\_TNT. TNT Netherlands.
- DE\_FEDEX. Federal Express Germany
- OTHER. Other.
- BONDSCOURIERS. Bonds Couriers.
- IT\_DHL\_GLOBALFORWARDING. DHL Global Forwarding Italy.
- IDN\_LION\_PARCEL. Lion Parcel Indonesia.
- UK\_YODEL. [Yodel UK](#)
- IT\_DHL\_EXPRESS. DHL Express Italy.
- PL\_DHL\_GLOBALFORWARDING. [DHL Global Forwarding Poland](#)
- DPD\_POLAND. DPD Poland.
- AU\_AUSTRALIA\_POST\_EXPRESS\_POST\_PLATINUM. Australia Post Express Post Platinum
- ES\_TNT. TNT Spain.
- CN\_DHL\_EXPRESS. DHL Express Canada.
- DE\_DPD. DPD Germany.
- DE\_DPD\_DELISTRACK. DPD Delistrack Germany
- CN\_DHL\_ECOMMERCE. DHL eCommerce China.
- JP\_TNT. TNT Japan.
- PRT\_CTT. [CTT Expresso Portugal](#)
- UK\_INTERLINK\_EXPRESS. Interlink Express UK.
- NLD\_POSTNL. [PostNL Netherlands](#)
- CA\_DHL\_ECOMMERCE. DHL eCommerce Canada.
- SWIFTAIR. Swift Air.
- NOR\_POSTEN. [Posten Norge](#)
- MEX\_REDPACK. Redpack Mexico.
- PL\_MASTERLINK. Masterlink Poland
- PL\_TNT. TNT Express Poland
- NIM\_EXPRESS. [Nim Express](#)
- PL\_UPS. United Parcel Service Poland
- UKR\_NOVA. [Nova Poshta](#)
- QUANTIUM. Quantum.

- SENDLE. Sendle.
- SG\_PARCELPOST. Parcel Post Singapore.
- SG\_NINJAVAN. Ninjavan Singapore.
- BQC\_EXPRESS. BQC Express.
- RPD2MAN. RPD2man Deliveries
- THA\_KERRY. Kerry Thailand.
- MEX\_AEROFLASH. Aeroflash Mexico.
- SPRADEL. Spreadel.
- ESP\_REDUR. [Redur Spain](#)
- JP\_JAPANPOST. Japan Post
- ARE\_EMIRATES\_POST. [Emirates Post Group](#)
- CN\_CHINA\_POST\_EMS. China Post EMS Express Mail Service
- UK\_DHL\_GLOBALFORWARDING. DHL Global Forwarding UK.
- CN\_SF\_EXPRESS. SF Express China.
- UK\_FEDEX. Federal Express UK
- POL\_POCZTA. Poczta Poland.
- YANWEN. [Yanwen Express](#)
- KOR\_CJ. CJ Korea.
- DE\_DEUTSCHE\_POST\_DHL\_WITHIN\_EUROPE\_TRACKNET. Deutsche Post DHL Tracknet Germany
- IND\_XPRESSBEES. XpressBees India.
- UK\_TNT. TNT UK.
- CJ\_KOREA\_THAI. [CJ Logistics in Thailand](#)
- CN\_OTHER. Other - China.
- IDN\_POS. Indonesia Post.
- ABC\_MAIL. ABC Mail.
- UK\_UPS. United Parcel Service UK
- CHINA\_POST. China Post.
- PL\_DHL\_EXPRESS. DHL Express Poland.
- ESP\_SPANISH\_SEUR. Spanish Seur Spain
- SG\_ZALORA. Zalora Singapore.
- MATKAHUOLTO. Matkahuoloto.
- FR\_LAPOSTE. Laposte France.
- KANGAROO\_MY. Kangaroo Express Malaysia.
- ESP\_CORREOS. [Sociedad Estatal Correos y Telégrafos](#)
- NL\_KIALA. KIALA Netherlands

- IND\_BLUEDART. [Blue Dart Express DHL](#)
- TUR\_PTT. PTT Turkey.
- CA\_CANNOT\_PROVIDE\_TRACKING. Cannot provide tracking - Canada.
- JPN\_SAGAWA. Sagawa Japan.
- MYS\_SKYNET. Skynet Malaysia.
- IT\_FERCAM. Fercam Italy.
- UK\_AIRBORNE\_EXPRESS. Airborne Express UK.
- CA\_OTHER. Other - Canada.
- DE\_DEUTSHCE\_POST\_DHL\_TRACK\_TRACE\_EXPRESS. Deutsche Post DHL Track Trace Express Germany
- CORREOS\_DE\_MEXICO. [Mex Post Correos de Mexico](#)
- FR\_DHL\_GLOBALFORWARDING. DHL Global Forwarding France.
- GLOBAL\_SKYNET. Skynet Global.
- AU\_DHL\_GLOBALFORWARDING. DHL Global Forwarding Australia.
- DE\_DHL\_GLOBALFORWARDING. DHL Global Forwarding Germany.
- SFC\_LOGISTICS. [SFC Logistics](#)
- US\_GLOBEGISTICS. Globeistics US.
- CA\_DHL\_GLOBALFORWARDING. DHL Global Forwarding Canada.
- OMNIPARCEL. Omni Parcel.
- PHL\_AIR21. Air21 Philippines
- CBL\_LOGISTICA. [CBL Logística](#)
- FR\_MONDIAL. Mondial France.
- DE\_DHL\_ECOMMERCE. DHL eCommerce Germany.
- ADICIONAL. Adicional.
- CH\_SWISS\_POST\_PRIORITY. Swiss Post Priority
- NL\_INTANGIBLE\_DIGITAL\_SERVICES. Intangible Digital Services
- DE\_ASENDIA. Asendia Germany.
- NL\_ABC\_MAIL. ABC Mail Netherlands
- UK\_DELTEC. Deltec UK.
- ONE\_WORLD. One World.
- AIRBORNE\_EXPRESS. Airborne Express.
- ES\_OTHER. Other - Spain.
- US\_DHL\_ECOMMERCE. [DHL eCommerce US](#)
- US\_ENSENDIA. Ensenda US.
- CPACKET. Cpacket.
- AXL. [AXL Express & Logistics](#)

- IND\_REDEXPRESS. Red Express India.
- NL\_LOCAL\_PICKUP. Local PickUp Netherlands
- UK\_ROYALMAIL\_AIRSURE. Royal Mail AirSure UK
- FR\_TNT. TNT France.
- USPS. [United States Postal Service \(USPS\)](#)
- RINCOS. Rincos.
- B2CEUROPE. B2C Europe
- PHL\_LBC. LBC Philippines.
- SG\_TAQBIN. TA-Q-BIN Parcel Singapore.
- GR\_ELTA. Elta Greece.
- WINIT. WinIt.
- NLD\_DHL. DHL Netherlands.
- FR\_GEODIS. Geodis France.
- DE\_DHL\_PACKET. DHL Packet Germany.
- ARG\_OCA. [OCA Argentina](#)
- JP\_DHL. DHL Japan.
- RUSSIAN\_POST. Russian Post.
- TW\_TAIWAN\_POST. [Chunghwa Post](#)
- UPS. [United Parcel Service of America, Inc.](#)
- BE\_BPOST. Bpost Belgium
- JP\_SAGAWA\_KYUU\_BIN. Sagawa Kyuu Bin Japan
- NATIONWIDE\_MY. Nationwide Malaysia.
- TNT. TNT Portugal.
- COURIERS\_PLEASE. Couriers Please.
- DMM\_NETWORK. DMM Network.
- TOLL. [Toll](#)
- NONE. None
- IDN\_FIRST\_LOGISTICS. First Logistics Indonesia.
- BH\_POSTA. [BH POŠTA](#)
- SENDIT. SendIt.
- US\_DHL\_EXPRESS. DHL Express US.
- FEDEX. FedEx.
- SWE\_POSTNORD. [PostNord Sverige](#)
- PHL\_XEND\_EXPRESS. Xend Express Philippines.
- POSTI. Posti.
- CA\_CANADA\_POST. [Canada Post](#)

- PL\_FEXEX. Fexex Poland
- CN\_EC. EC China.
- HK\_TAQBIN. TA-Q-BIN Parcel Hong Kong.
- UK\_AN\_POST. [AddressPay UK](#)
- WISELOADS. Wiseloads.
- PRT\_SEUR. [Seur Portugal](#)
- US\_ONTRAC. Ontrac US.
- THA\_THAILAND\_POST. Thailand Post.
- DPE\_EXPRESS. DPE Express.
- UK\_DHL\_EXPRESS. DHL Express UK.
- NL\_DHL. DHL Netherlands
- HK\_FLYT\_EXPRESS. Flyt Express Hong Kong
- UK\_HERMESWORLD. Hermesworld UK.
- IT\_REGISTER\_MAIL. Registered Mail Italy.
- ARG\_CORREO. [Correo Argentino](#)
- CA\_LOOMIS. Loomis Express Canada
- DTDC\_AU. DTDC Australia.
- DPD. DPD Global.
- ASENDIA\_HK. Asendia Hong Kong.
- UK\_ROYALMAIL\_RECORDED. Royal Mail Recorded UK
- PL\_POCZTA\_POLSKA. Poczta Polska
- EU\_IMX. IMX EU
- IDN\_PANDU. Pandu Indonesia.
- MEX\_ESTAFETA. [Estafeta Mexico](#)
- SREKOREA. [SRE Korea](#)
- CYP\_CYPRUS\_POST. [Cyprus Post](#)
- NZ\_COURIER\_POST. [CourierPost New Zealand](#)
- CN\_EMPS. EMPS China.
- AU\_TNT. TNT Australia.
- UK\_CANNOT\_PROVIDE\_TRACKING. Cannot provide tracking - UK.
- ES\_DHL. DHL Spain.
- CONTINENTAL. Continental.
- IND\_DTDC. DTDC India.
- DE\_GLS. [General Logistics Systems \(GLS\) Germany](#)
- NLD\_GLS. [General Logistics Systems \(GLS\) Netherlands](#)
- UK\_DPD. DPD UK.

- IT\_TNT. [TNT Italy](#)
- PL\_DHL. DHL Portugal.
- JP\_NITTSU\_PELICAN\_BIN. Nittsu Pelican Bin Japan
- THA\_DYNAMIC\_LOGISTICS. Dynamic Logistics Thailand.
- IT\_POSTE\_ITALIA. Poste Italia
- UK\_ROYALMAIL\_INTER\_SIGNED. Royal Mail International Signed UK
- HERMES\_IT. Hermes Italy.
- FR\_BERT. [Bert France](#)
- IND\_PROFESSIONAL\_COURIERS. Professional Couriers India.
- POL\_SIODEMKA. Siodemka Poland.
- IE\_AN\_POST\_SDS\_PRIORITY. Post SDS Priority Ireland
- ADSONE. [ADSone Cumulus](#)
- BRA\_CORREIOS. Correios Brazil.
- UBI\_LOGISTICS. UBI Logistics.
- ES\_CORREOS. [Sociedad Estatal Correos y Telégrafos](#)
- NGA\_NIPOST. [Nigerian Postal Service](#)
- AUT\_AUSTRIAN\_POST. Austrian Post.
- AU\_FASTWAY. Fastway Australia.
- AUS\_TOLL. Toll Australia.
- CA\_CANPAR\_COURIER. Canpar Courier Canada.
- SWE\_DIRECTLINK. [Direct Link Sweden](#)
- CZE\_CESKA. [Česká pošta](#)
- ROYAL\_MAIL. Royal Mail.
- SG\_SINGPOST. SingPost Singapore
- IT\_OTHER. Other - Italy.
- ZA\_FASTWAY. [Fastway Couriers \(South Africa\)](#)
- SEKOLOGISTICS. Seko Logistics.
- CN\_UPS. CN\_UPS
- HUNTER\_EXPRESS. Hunter Express.
- DE\_DHL\_PARCEL. DHL Parcel Germany.
- NLD\_TRANSMISSION. Transmission Netherlands.
- CN\_TNT. TNT China.
- DE\_DEUTSCHE. Deutsche Post Germany.
- AIRSURE. Airtsure.
- UK\_PARCELFORCE. Parcelforce UK.
- SWE\_DB. [DB Schenker Sweden](#)



- CN\_CHINA\_POST. China Post
- PL\_GLS. General Logistics Systems Poland
- EU\_BPOST. [bpost](#)
- RELAIS\_COLIS. [Relais Colis](#)
- UK\_DHL\_PARCEL. DHL Parcel UK.
- AUS\_STARTRACK. StarTrack Australia
- AU\_TOLL\_IPEC. Toll IPEC Australia
- CORREOS\_CHILE. [CorreosChile](#)
- CH\_SWISS\_POST\_EXPRES. Swiss Post Express
- MYS\_TAQBIN. TA-Q-BIN Parcel Malaysia.
- JET\_SHIP. Jetship.
- HK\_DHL\_EXPRESS. DHL Express Hong Kong.
- IT\_SDA. [SDA Express Courier](#)
- DE\_DHL\_DEUTSCHEPOST. DHL Deutsche Post Germany.
- HK\_DHL\_GLOBALFORWARDING. DHL Global Forwarding Hong Kong.
- PHL\_RAF. RAF Philippines.
- IT\_GLS. [General Logistics Systems \(GLS\) Italy](#)
- PANTOS. Pantos.
- KOR\_ECARGO. Ecargo Korea.
- AT\_AUSTRIAN\_POST\_EMS. EMS Express Mail Service Austria
- IT\_BRT. [BRT Corriere Espresso Italy](#)
- CHE\_SWISS\_POST. Swiss Post.
- FASTWAY\_NZ. Fastway New Zealand.
- IT\_EBOOST\_SDA. IT\_EBOOST\_SDA
- ASENDIA\_UK. Asendia UK.
- RRDONNELLEY. RR Donnelley.
- US\_RL. RL US.
- GR\_GENIKI. Geniki Greece.
- DE\_DHL\_EXPRESS. DHL Express Germany.
- CA\_GREYHOUND. Greyhound Canada.
- UK\_COLLECTPLUS. [CollectPlus UK](#)
- NINJAVAN\_THAI. Ninjavan Thailand.
- RABEN\_GROUP. Raben Group.
- CA\_DHL\_EXPRESS. DHL Express Canada.
- GLOBAL\_TNT. TNT Global.
- IN\_INDIAPOST. India Post

- ITIS. ITIS International
- PHL\_JAMEXPRESS. JamExpress Philippines.
- PRT\_INT\_SEUR. [International Seur Portugal](#)
- ESP\_ASM. [Parcel Monitor Spain](#)
- NINJAVAN\_ID. Ninjavan Indonesia.
- JP\_FEDEX. Federal Express Japan
- FR\_CHRONOPOST. Chronopost France.
- FR\_SUIVI. Suivi FedEx France
- FR\_TELIWAY. Teliway France.
- JPN\_JAPAN\_POST. [Japan Post](#)
- HRV\_HRVATSKA. [Hrvatska Pošta](#)
- AT\_AUSTRIAN\_POST\_PAKET\_PRIME. Austrian Post Paket Prime
- DE\_OTHER. Other - Germany.
- HK\_HONGKONG\_POST. Hong Kong Post.
- GRC\_ACS. ACS Greece.
- HUN\_MAGYAR. [Magyar Posta](#)
- FR\_DHL\_PARCEL. DHL Parcel France.
- UK\_OTHER. Other - UK.
- LWK\_HK. LWK Hong Kong.
- EFS. [Enterprise Freight Systems \(EFS\)](#)
- PL\_DHL\_PARCEL. DHL Parcel Poland.
- PARCELFORCE. Parcel Force.
- AU\_AUSTRALIA\_POST\_EMS. Australia Post EMS
- US\_ASCENDIA. Ascendia US.
- ROU\_POSTA. [Poșta Română](#)
- NZ\_NZ\_POST. [New Zealand Post Limited \(NZ\)](#)
- RPX. RPX International.
- POSTUR\_IS. Postur.
- IE\_AN\_POST\_SDS\_EMS. Post SDS EMS Express Mail Service Ireland
- UK\_UK\_MAIL. [UK Mail](#)
- UK\_FASTWAY. Fastway UK.
- CORREOS\_DE\_COSTA\_RICA. [Correos de Costa Rica](#)
- MYS\_CITYLINK. Citylink Malaysia.
- PUROLATOR. Purolator.
- IND\_DOTZOT. [DotZot India](#)
- NG\_COURIERPLUS. [Courier Plus Nigeria](#)

- HK\_FOUR\_PX\_EXPRESS. 4PX Express Hong Kong
- ROCKETPARCEL. [Rocket Parcel International](#)
- CN\_DHL\_GLOBALFORWARDING. DHL Global Forwarding China.
- EPARCEL\_KR. EParcel Korea.
- INPOST\_PACZKOMATY. InPost Paczkomaty.
- KOR\_KOREA\_POST. Korea Post.
- CA\_PUROLATOR. Purolator Canada
- APR\_72. APR 72.
- FR\_DHL\_EXPRESS. DHL Express France.
- IDN\_JNE. JNE Indonesia.
- AU\_AUSTRALIA\_POST\_EPARCEL. Australia Post Eparcel
- GLOBAL\_ESTES. Estes Global.
- LTU\_LIETUVOS. Lietuvos paštas Lithuania.
- THECOURIERGUY. [The Courier Guy](#)
- BE\_CHRONOPOST. Chronopost Belgium.
- VNM\_VIETNAM\_POST. Vietnam Post.
- AU\_STAR\_TRACK\_EXPRESS. StarTrack Express Australia
- RAM. [JP RAM Shipping](#)

**carrier\_name\_other** string

The name of the carrier for the shipment. Provide this value only if the carrier parameter is OTHER.

**last\_updated\_time** string

The date and time when the tracking information was last updated, in [Internet date and time format](#).

Minimum length: 20.

Maximum length: 64.

Pattern: `^[0-9]{4}-(0[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])[T,t]([0-1][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)([.][0-9]+)?([Zz]|[-+][0-9]{2}:[0-9]{2})$`.

**links** array (contains the [link\\_description](#) object)

An array of request-related [HATEOAS links](#).

**notify\_buyer** boolean

If `true`, sends an email notification to the buyer of the PayPal transaction. The email contains the tracking information that was uploaded through the API.

**postage\_payment\_id** string

The postage payment ID.

**quantity** integer

The quantity of items shipped.

**shipment\_date** string

The date when the shipment occurred, in [Internet date and time format](#).

Minimum length: 20.

Maximum length: 64.

Pattern: `^[0-9]{4}-(0[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])[T,t]([0-1][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)([.][0-9]+)?([Zz]|[-+][0-9]{2}:[0-9]{2})$`.

**tracking\_number** string

The tracking number for the shipment.

**tracking\_number\_type** enum

The type of tracking number.

The possible values are:

- `CARRIER_PROVIDED`. A merchant-provided tracking number.
- `E2E_PARTNER_PROVIDED`. A marketplace-provided tracking number.

**tracking\_number\_validated** boolean

Indicates whether the carrier validated the tracking number.

### Example python

```
from python_paypal_api.api import Tracking
from python_paypal_api.base import (
    ShipmentStatus,
    Carrier,
    TrackingNumberType,
    PaypalApiException
)
import logging

def py_put_tracking(composed_id: str, dictionary: dict):

    logger.info("-----")
    logger.info("Tracking > put_tracking(%s)" % str(dictionary))
    logger.info("-----")

    try:

        result = Tracking(debug=True).put_tracking(
            id=composed_id,
            body=dictionary
        )
        logger.info(result)

    except PaypalApiException as error:
        logger.error(error)

if __name__ == '__main__':
    logger = logging.getLogger("test")

    id_transaction = "9ST00334VA8626***"
    number_tracking = "443844607820"

    path = "{}-{}".format(id_transaction, number_tracking)

    update = \
    {
        "transaction_id": id_transaction,
        "tracking_number": number_tracking,
        "tracking_number_type": TrackingNumberType.CARRIER_PROVIDED.value,
        "status": ShipmentStatus.ON_HOLD.value,
        "carrier": Carrier.FEDEX.value,
        "notify_buyer": True
    }

    py_put_tracking(path, update)
```

### Response

A successful request returns the HTTP 204 OK status code with no JSON response body.

204 No Content
----------------

**get\_tracking**(self, id: str, \*\*kwargs) → ApiResponse

GET /v1/shipping/trackers/{id}

Shows tracking information, by tracker ID, for a PayPal transaction.

**\*\*args:**

**id** string required

The ID of the tracker in the transaction\_id-tracking\_number format.

### Example python

```
from python_paypal_api.api import Tracking
from python_paypal_api.base import (
    ShipmentStatus,
    Carrier,
    TrackingNumberType,
    PaypalApiException
)
import logging

def py_get_tracking(composed_id: str):

    logger.info("-----")
    logger.info("Tracking > get_tracking(%s)" % str(composed_id))
    logger.info("-----")

    try:

        result = Tracking(debug=True).get_tracking(
            id=composed_id
        )
        logger.info(result)

    except PaypalApiException as error:
        logger.error(error)

if __name__ == '__main__':
    logger = logging.getLogger("test")

    id_transaction = "9ST00334VA8626***"
    number_tracking = "443844607820"

    path = "{}-{}".format(id_transaction, number_tracking)

    py_get_tracking(path)
```

### Response JSON

A successful request returns the HTTP 200 OK status code and a JSON response body that shows tracking information.

```
{
  "transaction_id": "8MC585209K746392H",
  "tracking_number": "443844607820",
  "status": "SHIPPED",
  "carrier": "FEDEX",
  "links": [
    {
      "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers/
↪8MC585209K746392H-443844607820",
      "rel": "self"
    },
    {
      "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers/
↪8MC585209K746392H-443844607820",
      "rel": "replace",
      "method": "PUT"
    },
    {
      "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers-batch",
      "rel": "create",
      "method": "POST"
    }
  ]
}
```

**post\_tracking**(self, \*\*kwargs) → ApiResponse

POST /v1/shipping/trackers-batch

Adds tracking information, with or without tracking numbers, for multiple PayPal transactions. Accepts up to 20 tracking IDs. For more information, see [Add tracking information with tracking numbers](#) and [Add tracking information without tracking numbers](#).

**\*\*kwargs:**

**links** array (contains the [link\\_description](#) object)

An array of request-related [HATEOAS](#) links.

**trackers** array (contains the [tracker](#) object)

### Example python

```
from python_paypal_api.api import Tracking
from python_paypal_api.base import (
    ShipmentStatus,
    Carrier,
    TrackingNumberType,
    PaypalApiException
)
import logging

def py_post_tracking(dictionary: dict):
```

(continues on next page)

(continued from previous page)

```

logging.info("-----")
logging.info("Tracking > post_tracking(%s)" % str(dictionary))
logging.info("-----")

try:

    result = Tracking(debug=True).post_tracking(
        body=dictionary
    )
    logging.info(result)

except PaypalApiException as error:
    logging.error(error)

if __name__ == '__main__':
    logger = logging.getLogger("test")

    id_transaction = "9ST00334VA8626***"
    number_tracking = "443844607820"

    trackers = {
        "trackers": [
            {
                "transaction_id": id_transaction,
                "tracking_number": number_tracking,
                "status": "SHIPPED",
                "carrier": "FEDEX"
            }
        ]
    }

    py_post_tracking(trackers)

```

### ### Response JSON

A successful request returns the HTTP 200 OK status code and a JSON response body that shows tracking information.

```

{
  "tracker_identifiers":
  [
    {
      "transaction_id": "8MC585209K746392H",
      "tracking_number": "443844607820",
      "links": [
        {
          "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers/
↪8MC585209K746392H-443844607820",
          "rel": "self",
          "method": "GET"
        }
      ],
    }
  ],

```

(continues on next page)



(continued from previous page)

```

        {
            "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers/
↪8MC585209K746392H-443844607820",
            "rel": "replace",
            "method": "PUT"
        }
    ]
},
{
    "transaction_id": "53Y56775AE587553X",
    "tracking_number": "443844607821",
    "links":
    [
        {
            "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers/
↪53Y56775AE587553X-443844607821",
            "rel": "self",
            "method": "GET"
        },
        {
            "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers/
↪53Y56775AE587553X-443844607821",
            "rel": "replace",
            "method": "PUT"
        }
    ]
}
],
"errors":
[
    {
        "name": "RESOURCE_NOT_FOUND",
        "debug_id": "46735c7461f3d",
        "message": "The specified resource does not exist.",
        "details":
        [
            {
                "field": "/trackers/0/transaction_id",
                "value": "8MC585309K746392H",
                "location": "body",
                "issue": "INVALID_TRANSACTION_ID"
            }
        ]
    }
],
"links":
[
    {
        "href": "https://api-m.sandbox.paypal.com/v1/shipping/trackers-batch
↪",
        "rel": "self",
        "method": "POST"
    }
]

```

(continues on next page)

(continued from previous page)

```
}  
]  
}
```

## 3.2 Transaction Search API

### Transaction Search API

Use the Transaction Search API to get the history of transactions for a PayPal account. To use the API on behalf of third parties, you must be part of the PayPal partner network. Reach out to your partner manager for the next steps. To enroll in the partner program, see [Partner with PayPal](#). For more information about the API, see the [Transaction Search API Integration Guide](#).

---

**Note:** Note: To use the API on behalf of third parties, you must be part of the PayPal partner network. Reach out to your partner manager for the next steps. To enroll in the partner program, see [Partner with PayPal](#).

---

**class** `python_paypal_api.api.Transactions(*args, **kwargs)`

Use the /transactions resource to list transactions.

**get\_list\_transactions**(*self*, \*\*kwargs) → ApiResponse

GET /v1/reporting/transactions

Lists transactions. Specify one or more query parameters to filter the transaction that appear in the response.

---

**Note:**

- If you specify one or more optional query parameters, the ending\_balance response field is empty.
  - It takes a maximum of three hours for executed transactions to appear in the list transactions call.
  - This call lists transaction for the previous three years.
- 

**\*\*kwargs:**

**payment\_instrument\_type** string

Filters the transactions in the response by a payment instrument type. Value is either:

- CREDITCARD. Returns a direct credit card transaction with a corresponding value.
- DEBITCARD. Returns a debit card transaction with a corresponding value.

If you omit this parameter, the API does not apply this filter.

**end\_date** string required

Filters the transactions in the response by an end date and time, in [Internet date and time format](#). Seconds are required. Fractional seconds are optional. The maximum supported range is 31 days.

Minimum length: 20.

Maximum length: 64.

Pattern: `^[0-9]{4}-(0[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])[T,t]([0-1][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)([.][0-9]+)?([Zz]|[-+][0-9]{2}:[0-9]{2})$.`

**start\_date** string required

Filters the transactions in the response by a start date and time, in [Internet date and time format](#). Seconds are required. Fractional seconds are optional.

Minimum length: 20.

Maximum length: 64.

Pattern: `^[0-9]{4}-(0[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])[T,t]([0-1][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)([.][0-9]+)?([Zz]|[-+][0-9]{2}:[0-9]{2})$.`

**balance\_affecting\_records\_only** string

Indicates whether the response includes only balance-impacting transactions or all transactions.

Value is either:

- Y. The default. The response includes only balance transactions.
- N. The response includes all transactions.

**fields** string

Indicates which fields appear in the response. Value is a single field or a comma-separated list of fields. The `transaction_info` value returns only the transaction details in the response. To include all fields in the response, specify `fields=all`. Valid fields are:

- `transaction_info`. The transaction information. Includes the ID of the PayPal account of the payee, the PayPal-generated transaction ID, the PayPal-generated base ID, the PayPal reference ID type, the transaction event code, the date and time when the transaction was initiated and was last updated, the transaction amounts including

the PayPal fee, any discounts, insurance, the transaction status, and other information about the transaction.

- **payer\_info**. The payer information. Includes the PayPal customer account ID and the payer's email address, primary phone number, name, country code, address, and whether the payer is verified or unverified.
- **shipping\_info**. The shipping information. Includes the recipient's name, the shipping method for this order, the shipping address for this order, and the secondary address associated with this order.
- **auction\_info**. The auction information. Includes the name of the auction site, the auction site URL, the ID of the customer who makes the purchase in the auction, and the date and time when the auction closes.
- **cart\_info**. The cart information. Includes an array of item details, whether the item amount or the shipping amount already includes tax, and the ID of the invoice for PayPal-generated invoices.
- **incentive\_info**. An array of incentive detail objects. Each object includes the incentive, such as a special offer or coupon, the incentive amount, and the incentive program code that identifies a merchant loyalty or incentive program.
- **store\_info**. The store information. Includes the ID of the merchant store and the terminal ID for the checkout stand in the merchant store.

**page** integer

The zero-relative start index of the entire list of items that are returned in the response. So, the combination of `page=1` and `page_size=20` returns the first 20 items.

Minimum value: 1.

Maximum value: 2147483647.

**page\_size** integer

The number of items to return in the response. So, the combination of `page=1` and `page_size=20` returns the first 20 items. The combination of `page=2` and `page_size=20` returns the next 20 items.

Minimum value: 1.

Maximum value: 500.

**store\_id** string

Filters the transactions in the response by a store ID.

**terminal\_id** string

Filters the transactions in the response by a terminal ID.

**transaction\_amount** string

Filters the transactions in the response by a gross transaction amount range. Specify the range as <start-range> TO <end-range>, where <start-range> is the lower limit of the gross PayPal transaction amount and <end-range> is the upper limit of the gross transaction amount. Specify the amounts in lower denominations. For example, to search for transactions from \$5.00 to \$10.05, specify [500 TO 1005].

---

**Note:** The values must be URL encoded.

---

**transaction\_currency** string

Filters the transactions in the response by a [three-character ISO-4217 currency code](#) for the PayPal transaction currency.

**transaction\_id** string

Filters the transactions in the response by a PayPal transaction ID. A valid transaction ID is 17 characters long, except for an order ID, which is 19 characters long.

---

**Note:** A transaction ID is not unique in the reporting system. The response can list two transactions with the same ID. One transaction can be balance affecting while the other is non-balance affecting.

---

Minimum length: 17.

Maximum length: 19.

**transaction\_status** string

Filters the transactions in the response by a PayPal transaction status code. Value is:

Sta-tus code	Description
D	PayPal or merchant rules denied the transaction.
P	The transaction is pending. The transaction was created but waits for another payment process to complete, such as an ACH transaction, before the status changes to S.
S	The transaction successfully completed without a denial and after any pending statuses.
V	A successful transaction was reversed and funds were refunded to the original sender.

**transaction\_type** string

Filters the transactions in the response by a PayPal transaction event code. See [Transaction event codes](#).

#### Returns

ApiResponse

### Example python

```
from python_paypal_api.api import Transactions
from python_paypal_api.base import PaypalApiException
import logging
from datetime import datetime, timezone

def py_get_list_transactions(**kwargs):

    logger.info("-----")
    logger.info("Transactions > py_get_list_transactions({})".format(kwargs))
    logger.info("-----")

    try:

        result = Transactions(debug=True).get_list_transactions(
            **kwargs
        )
        logger.info(result)

    except PaypalApiException as error:
        logger.error(error)

if __name__ == '__main__':

    logger = logging.getLogger("test")
    date_start = datetime(2023, 2, 1, 0, 0, 0, 0, timezone.utc).isoformat()
    date_end = datetime(2023, 3, 1, 0, 0, 0, 0, timezone.utc).isoformat()

    py_get_list_transactions(
```

(continues on next page)

(continued from previous page)

```

        page_size=1,
        start_date=date_start,
        end_date=date_end
    )

```

### ### Response JSON

A successful request returns the HTTP 200 OK status code and a JSON response body that lists transactions.

```

{
    'account_number': '2LBUCGLCSB***',
    'end_date': '2023-03-01T00:00:00+0000',
    'last_refreshed_datetime': '2023-03-12T01:59:59+0000',
    'links': [
        {
            'href': 'https://api.sandbox.paypal.com/v1/reporting/transactions?start_
↪date=2023-02-01T00%3A00%3A00%2B00%3A00&end_date=2023-03-01T00%3A00%3A00%2B00
↪%3A00&page_size=1&page=19',
            'method': 'GET',
            'rel': 'last'
        },
        {
            'href': 'https://api.sandbox.paypal.com/v1/reporting/transactions?start_
↪date=2023-02-01T00%3A00%3A00%2B00%3A00&end_date=2023-03-01T00%3A00%3A00%2B00
↪%3A00&page_size=1&page=2',
            'method': 'GET',
            'rel': 'next'
        },
        {
            'href': 'https://api.sandbox.paypal.com/v1/reporting/transactions?start_
↪date=2023-02-01T00%3A00%3A00%2B00%3A00&end_date=2023-03-01T00%3A00%3A00%2B00
↪%3A00&page_size=1&page=1',
            'method': 'GET',
            'rel': 'self'
        }
    ],
    'page': 1,
    'start_date': '2023-02-01T00:00:00+0000',
    'total_items': 19,
    'total_pages': 19,
    'transaction_details': [
        {
            'transaction_info':
                {
                    'available_balance':
                        {
                            'currency_code': 'EUR',
                            'value': '165488.69'
                        },
                    'custom_field': 'S1121674440646-1563665460',
                    'ending_balance':
                        {
                            'currency_code': 'EUR',

```

(continues on next page)

(continued from previous page)

```
        'value': '165488.69'
    },
    'paypal_account_id': 'UYZPMCTQDV***',
    'paypal_reference_id': '4G8384171L5786***',
    'paypal_reference_id_type': 'TXN',
    'protection_eligibility': '02',
    'transaction_amount':
    {
        'currency_code': 'EUR',
        'value': '-50.00'
    },
    'transaction_event_code': 'T1110',
    'transaction_id': '18670957S08679***',
    'transaction_initiation_date': '2023-02-12T11:52:33+0000',
    'transaction_status': 'P',
    'transaction_updated_date': '2023-02-12T11:52:33+0000'
    }
}
]
```

**get\_balances**(*self*, *\*\*kwargs*) → ApiResponse

GET /v1/reporting/balances

List all balances. Specify date time to list balances for that time that appear in the response.

---

**Note:**

- It takes a maximum of three hours for balances to appear in the list balances call.
  - This call lists balances upto the previous three years.
- 

**\*\*kwargs:**

**as\_of\_time** string

List balances in the response at the date time provided, will return the last refreshed balance in the system when not provided.

Minimum length: 20.

Maximum length: 64.

Pattern: `^[0-9]{4}-(0[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])[T,t]([0-1][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)([.][0-9]+)?([Zz]|[-+][0-9]{2}:[0-9]{2})$`.

**currency\_code** string



Filters the transactions in the response by a [three-character ISO-4217 currency code](#) for the PayPal transaction currency.

Minimum length: 3.

Maximum length: 3.

### Example python

```
from python_paypal_api.api import Transactions
from python_paypal_api.base import PaypalApiException
import logging
from datetime import datetime, timezone

def py_get_balances(**kwargs):

    logger.info("-----")
    logger.info("Transactions > py_get_balances({})".format(kwargs))
    logger.info("-----")

    try:

        result = Transactions(debug=True).get_balances(
            **kwargs
        )
        logger.info(result)

    except PaypalApiException as error:
        logger.error(error)

if __name__ == '__main__':

    logger = logging.getLogger("test")
    date_start = datetime(2023, 2, 1, 0, 0, 0, 0, timezone.utc).isoformat()
    py_get_balances(
        as_of_time=date_start,
        currency_code="USD"
    )
```

### Response JSON

A successful request returns the HTTP 200 OK status code and a JSON response body that lists balances.

```
{
  'account_id': '2LBUCGLCSB***',
  'as_of_time': '2023-03-12T01:59:59Z',
  'balances': [
    {
      'available_balance':
        {
```

(continues on next page)

(continued from previous page)

```
        'currency_code': 'USD',
        'value': '1304.05'
    },
    'currency': 'USD',
    'total_balance':
    {
        'currency_code': 'USD',
        'value': '1304.05'
    },
    'withheld_balance':
    {
        'currency_code': 'USD',
        'value': '0.00'
    }
  },
  'last_refresh_time': '2023-03-12T01:59:59Z'
}
```

## 3.3 Catalog Products API

### Catalog Products API

Merchants can use the Catalog Products API to create products, which are goods and services.

**class** python\_paypal\_api.api.**Products**(\*args, \*\*kwargs)

Use /products resource to create and manage products.

**list\_products**(self, \*\*kwargs) → ApiResponse

GET /v1/catalogs/products

Lists products.

**\*\*kwargs:**

**page** integer

A non-zero integer which is the start index of the entire list of items that are returned in the response. So, the combination of **page**=1 and **page\_size**=20 returns the first 20 items. The combination of **page**=2 and **page\_size**=20 returns the next 20 items.

Minimum value: 1.

Maximum value: 100000.

**page\_size** integer

The number of items to return in the response.

Minimum value: 1.

Maximum value: 20.

**total\_required** boolean

Indicates whether to show the total items and total pages in the response.

**get\_product**(*self*, *productId*: str, *\*\*kwargs*) → ApiResponse

GET /v1/catalogs/products/{product\_id}

**update\_product**(*self*, *productId*: str, *\*\*kwargs*) → ApiResponse

PATCH /v1/catalogs/products/{product\_id}

**update\_product\_helper**(*self*, *productId*: str, *\*\*kwargs*) → ApiResponse

PATCH /v1/catalogs/products/{product\_id}

**post\_product**(*self*, *\*\*kwargs*) → ApiResponse

POST /v1/catalogs/products

**post\_product\_helper**(*self*, *body*: str, *\*\*kwargs*) → ApiResponse

POST /v1/catalogs/products



## DISCLAIMER

### MIT License

Copyright (c) 2021 denisneuf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

New in version 0.1.1: The *Products* is added replacing *Catalog* for best naming The *Config.yaml* is now the default for config credentials

```
from python_paypal_api.api import Identity, Products
from python_paypal_api.base import PayPalApiException
import logging

def py_list_products(**kwargs):

    logging.info("-----")
    logging.info("Catalog > list_products()")
    logging.info("-----")

    credentials = dict(
        client_id="your-client-id",
        client_secret="your-client-secret",
        client_mode="your-mode" # PRODUCTION OR SANDBOX(default)
    )

    try:

        result = Products(credentials=credentials, store_credentials=False, debug=True).
```

(continues on next page)

(continued from previous page)

```
↪ list_products(
    **kwargs
)
document_dict = result.payload
logging.info(result)

except Exception as error:
    logging.info(error)

def py_get_userinfo():

    logging.info("-----")
    logging.info("Identity > py_get_userinfo")
    logging.info("-----")

    try:

        # result = Identity(account="production", store_credentials=True, debug=True).
↪ get_userinfo(
        result = Identity(debug=True).get_userinfo(
        )
        logging.info(result)

    except PaypalApiException as error:
        logging.error(error)

    except Exception as error:
        logging.info(error)

if __name__ == '__main__':

    logger = logging.getLogger("test")

    py_get_userinfo()

    py_list_products(
        total_required=True,
        page_size=1,
        page=2
    )
```

## INDEX

### P

`Products` (class in `python_paypal_api.api`), 38  
`Products.get_product()` (in module `python_paypal_api.api.Products`), 39  
`Products.list_products()` (in module `python_paypal_api.api.Products`), 38  
`Products.post_product()` (in module `python_paypal_api.api.Products`), 39  
`Products.post_product_helper()` (in module `python_paypal_api.api.Products`), 39  
`Products.update_product()` (in module `python_paypal_api.api.Products`), 39  
`Products.update_product_helper()` (in module `python_paypal_api.api.Products`), 39

### T

`Tracking` (class in `python_paypal_api.api`), 11  
`Tracking.get_tracking()` (in module `python_paypal_api.api.Tracking`), 26  
`Tracking.post_tracking()` (in module `python_paypal_api.api.Tracking`), 27  
`Tracking.put_tracking()` (in module `python_paypal_api.api.Tracking`), 11  
`Transactions` (class in `python_paypal_api.api`), 30  
`Transactions.get_balances()` (in module `python_paypal_api.api.Transactions`), 36  
`Transactions.get_list_transactions()` (in module `python_paypal_api.api.Transactions`), 30