

---

# **Paddle Client**

**Matt Pye**

**Mar 26, 2021**



**CONTENTS:**

<b>1</b>	<b>Getting started</b>	<b>1</b>
1.1	Paddle authentication . . . . .	1
1.2	Installation . . . . .	1
1.3	Usage . . . . .	1
1.4	Paddle sandbox environment . . . . .	2
<b>2</b>	<b>API Reference</b>	<b>3</b>
2.1	Supported Paddle endpoints . . . . .	3
2.2	Full reference . . . . .	4
<b>3</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



## GETTING STARTED

Paddle Client is a python (3.5+) wrapper around the [Paddle.com API](#)

### 1.1 Paddle authentication

Before you get started you will need an account on [Paddle.com](#) to generate authentication details.

Once you have an account go to the [Paddle authentication page](#) and find out both your `Vendor ID` and `API key`

### 1.2 Installation

Install of the paddle client can be done via pip:

```
pip install paddle-client
```

### 1.3 Usage

```
from paddle import PaddleClient

paddle = PaddleClient(vendor_id=12345, api_key='myapikey')
paddle.list_products()
```

If `vendor_id` and `api_key` are not passed through when initialising Paddle will fall back and try and use environmental variables called `PADDLE_VENDOR_ID` and `PADDLE_API_KEY`

```
export PADDLE_VENDOR_ID=12345
export PADDLE_API_KEY="myapikey"
```

```
from paddle import PaddleClient

paddle = PaddleClient()
paddle.list_products()
```

### 1.4 Paddle sandbox environment

The **Paddle sandbox environment** is a separate Paddle environment which can be used for development and testing. You are required to create a new account in this environment, different to your production account.

Once you have this account setup and configured you can use the sandbox account by passing `sandbox=True` when initialising the Paddle Client. This will send all API calls to the Paddle sandbox URLs instead of the production URLs

```
from paddle import PaddleClient

paddle = PaddleClient (vendor_id=12345, api_key='myapikey', sandbox=True)
```

It is also possible to turn the sandbox environment on using an environmental variable called `PADDLE_SANDBOX`:

```
export PADDLE_SANDBOX="true"
```

```
from paddle import PaddleClient

paddle = PaddleClient (vendor_id=12345, api_key='myapikey')
```

## API REFERENCE

Below is a *full list* of the Paddle Client methods. *Supported Paddle endpoints* lists the Paddle API endpoints in the same order as they are in the [Paddle API Reference](#)

There are also several *helper* methods implemented on top of the Paddle endpoints to make usage of the API easier.

For example the `pause_subscription` and `resume_subscription` methods have been added on top of `update_subscription` as to pause or resume a subscription the `update_subscription` endpoint is used but only the `pause` parameter can be set (and will fail if any others are).

### 2.1 Supported Paddle endpoints

As listed in the [Paddle API Reference](#)

#### Checkout API

- *Get Order Details*
- *Get User History*
- *Get Prices*

#### Product API

- *List Coupons*
- *Create Coupon*
- *Delete Coupon*
- *Update Coupon*
- *List Products*
- *List Transactions*
- *Refund Payment*

#### Subscription API

- *List Plans - (Including Get Plan)*
- *Create Plan*
- *List Subscription Users*
- *Cancel Subscription*
- *Update Subscription - (Including Pause Subscription and Resume Subscription)*
- *Add Modifier*

- *Delete Modifier*
- *List Modifiers*
- *List Payments*
- *Reschedule Payment*
- *Create One-off Charge*

### Alert API

- *Get Webhook History*

## 2.2 Full reference

For reference on how to use. An example on how to use *List Products*:

```
from paddle import PaddleClient

paddle = PaddleClient(vendor_id=12345, api_key='myapikey')
paddle.list_products()
```

**class** `paddle.PaddleClient` (*vendor\_id=None, api\_key=None, sandbox=None*)

If `vendor_id` and `api_key` are not passed through when initialising Paddle will fall back and try and use environmental variables called `PADDLE_VENDOR_ID` and `PADDLE_API_KEY`

#### Parameters

- **vendor\_id** (`typing.Optional[int]`) –
- **api\_key** (`typing.Optional[str]`) –
- **sandbox** (`typing.Optional[bool]`) –

**add\_modifier** (*subscription\_id, modifier\_amount, modifier\_recurring=True, modifier\_description=""*)

[Add Modifier Paddle docs](#)

#### Parameters

- **subscription\_id** (`int`) –
- **modifier\_amount** (`float`) –
- **modifier\_recurring** (`bool`) –
- **modifier\_description** (`str`) –

**Return type** `dict`

**cancel\_subscription** (*subscription\_id*)

[Cancel Subscription Paddle docs](#)

**Parameters** **subscription\_id** (`int`) –

**Return type** `bool`

**create\_coupon** (*coupon\_type, discount\_type, discount\_amount, allowed\_uses, recurring, currency, product\_ids=None, coupon\_code=None, coupon\_prefix=None, num\_coupons=None, description=None, expires=None, minimum\_threshold=None, group=None*)

[Create Coupon Paddle docs](#)



currency appears to be required:

Paddle error 134 - The given coupon currency is invalid. The currency must match your balance currency.

Even though the docs states:

“Required if discount\_amount is flat.”

#### Parameters

- **coupon\_type** (str) –
- **discount\_type** (str) –
- **discount\_amount** (float) –
- **allowed\_uses** (int) –
- **recurring** (bool) –
- **currency** (str) –
- **product\_ids** (typing.Optional[list]) –
- **coupon\_code** (typing.Optional[str]) –
- **coupon\_prefix** (typing.Optional[str]) –
- **num\_coupons** (typing.Optional[int]) –
- **description** (typing.Optional[str]) –
- **expires** (typing.Union[str, datetime.datetime, None]) –
- **minimum\_threshold** (typing.Optional[int]) –
- **group** (typing.Optional[str]) –

**Return type** dict

**create\_one\_off\_charge** (*subscription\_id, amount, charge\_name*)

Create One-off Charge [Paddle docs](#)

This endpoint currently only supports x-www-form-urlencoded not json unlike most of the other Paddle endpoints

Note from Paddles API docs:

Please put a proper product / charge name here to make it clear what buyers are buying. Here are some examples:

- [add-on name] x [quantity] e.g. Supercool design plugin x 1
- [number of credits] credits for [subscription plan] e.g. 200 credits for Monthly talk time
- [quantity] [units] [product description] e.g. 124 MB data usage

#### Parameters

- **subscription\_id** (int) –
- **amount** (float) –
- **charge\_name** (str) –

**Return type** dict

```
create_pay_link (product_id=None, title=None, webhook_url=None, prices=None,
                  recurring_prices=None, trial_days=None, custom_message=None,
                  coupon_code=None, discountable=None, image_url=None, return_url=None,
                  quantity_variable=None, quantity=None, expires=None, affiliates=None, recur-
                  ring_affiliate_limit=None, marketing_consent=None, customer_email=None,
                  customer_country=None, customer_postcode=None, passthrough=None,
                  vat_number=None, vat_company_name=None, vat_street=None, vat_city=None,
                  vat_state=None, vat_country=None, vat_postcode=None)
```

[Create Pay Link Paddle docs](#)

product\_id appears to be required:

Paddle error 108 - Unable to find requested product

Even though the docs states:

“If no product\_id is set, custom non-subscription product checkouts can be generated instead by specifying title, webhook\_url and prices.”

Sending an invalid coupon code will result in the request failing with “Paddle error 101 - Bad method call”

#### Parameters

- **product\_id** (typing.Optional[int]) –
- **title** (typing.Optional[str]) –
- **webhook\_url** (typing.Optional[str]) –
- **prices** (typing.Optional[typing.List[str]]) –
- **recurring\_prices** (typing.Optional[typing.List[str]]) –
- **trial\_days** (typing.Optional[int]) –
- **custom\_message** (typing.Optional[str]) –
- **coupon\_code** (typing.Optional[str]) –
- **discountable** (typing.Optional[bool]) –
- **image\_url** (typing.Optional[str]) –
- **return\_url** (typing.Optional[str]) –
- **quantity\_variable** (typing.Optional[bool]) –
- **quantity** (typing.Optional[int]) –
- **expires** (typing.Union[str, datetime.datetime, None]) –
- **affiliates** (typing.Optional[typing.List[str]]) –
- **recurring\_affiliate\_limit** (typing.Optional[int]) –
- **marketing\_consent** (typing.Optional[bool]) –
- **customer\_email** (typing.Optional[str]) –
- **customer\_country** (typing.Optional[str]) –
- **customer\_postcode** (typing.Optional[str]) –
- **passthrough** (typing.Optional[str]) –
- **vat\_number** (typing.Optional[str]) –
- **vat\_company\_name** (typing.Optional[str]) –

- **vat\_street** (typing.Optional[str]) –
- **vat\_city** (typing.Optional[str]) –
- **vat\_state** (typing.Optional[str]) –
- **vat\_country** (typing.Optional[str]) –
- **vat\_postcode** (typing.Optional[str]) –

**Return type** dict

**create\_plan** (*plan\_name*, *plan\_trial\_days*, *plan\_length*, *plan\_type*, *main\_currency\_code*='USD', *initial\_price\_usd*=None, *initial\_price\_gbp*=None, *initial\_price\_eur*=None, *recurring\_price\_usd*=None, *recurring\_price\_gbp*=None, *recurring\_price\_eur*=None)

[Create Plan Paddle docs](#)

In the docs all of the price variables are noted as strings except for USD which is a number. This is probably an error in the docs.

**Parameters**

- **plan\_name** (str) –
- **plan\_trial\_days** (int) –
- **plan\_length** (int) –
- **plan\_type** (str) –
- **main\_currency\_code** (str) –
- **initial\_price\_usd** (typing.Optional[float]) –
- **initial\_price\_gbp** (typing.Optional[float]) –
- **initial\_price\_eur** (typing.Optional[float]) –
- **recurring\_price\_usd** (typing.Optional[float]) –
- **recurring\_price\_gbp** (typing.Optional[float]) –
- **recurring\_price\_eur** (typing.Optional[float]) –

**Return type** dict

**delete\_coupon** (*coupon\_code*, *product\_id*=None)

[Delete Coupon Paddle docs](#)

**Parameters**

- **coupon\_code** (str) –
- **product\_id** (typing.Optional[int]) –

**Return type** dict

**delete\_modifier** (*modifier\_id*)

[Delete Modifier Paddle docs](#)

**Parameters** **modifier\_id** (int) –

**Return type** dict

**generate\_license** (*product\_id*, *allowed\_uses*, *expires\_at*=None)

[Generate License Paddle docs](#)

**Parameters**

- **product\_id** (int) –

- **allowed\_uses** (int) –
- **expires\_at** (typing.Union[str, datetime.datetime, None]) –

**Return type** dict

**get\_order\_details** (*checkout\_id*)

[Get Order Details Paddle docs](#)

**Parameters** **checkout\_id** (str) –

**Return type** dict

**get\_plan** (*plan*)

[List Plans Paddle docs](#)

There is no Get Plan endpoint in Paddle's API but list plans does support sending through a plan ID. This is a helper method to get a single plan where the plan parameter is required.

**Parameters** **plan** (int) –

**Return type** dict

**get\_prices** (*product\_ids*, *customer\_country=None*, *customer\_ip=None*, *coupons=None*)

[Get Prices Paddle docs](#)

**Parameters**

- **product\_ids** (list) –
- **customer\_country** (typing.Optional[str]) –
- **customer\_ip** (typing.Optional[str]) –
- **coupons** (typing.Optional[list]) –

**Return type** dict

**get\_user\_history** (*email*, *vendor\_id=None*, *product\_id=None*)

[Get User History Paddle docs](#)

**Parameters**

- **email** (str) –
- **vendor\_id** (typing.Optional[int]) –
- **product\_id** (typing.Optional[int]) –

**Return type** str

**get\_webhook\_history** (*page=None*, *alerts\_per\_page=None*, *query\_head=None*,  
*query\_tail=None*)

[Get Webhook History Paddle docs](#)

**Parameters**

- **page** (typing.Optional[int]) –
- **alerts\_per\_page** (typing.Optional[int]) –
- **query\_head** (typing.Union[str, datetime.datetime, None]) –
- **query\_tail** (typing.Union[str, datetime.datetime, None]) –

**Return type** dict

**list\_coupons** (*product\_id*)

[List Coupon Paddle docs](#)

**Parameters** `product_id(int)` –

**Return type** `typing.List[dict]`

**list\_modifiers** (*subscription\_id=None, plan\_id=None*)

[List Modifiers Paddle docs](#)

**Parameters**

- **subscription\_id** (`typing.Optional[int]`) –
- **plan\_id** (`typing.Optional[int]`) –

**Return type** `typing.List[dict]`

**list\_plans** (*plan=None*)

[List Plans Paddle docs](#)

**Parameters** **plan** (`typing.Optional[int]`) –

**Return type** `typing.List[dict]`

**list\_products** ()

[List Products Paddle docs](#)

**Return type** `dict`

**list\_subscription\_payments** (*subscription\_id=None, plan=None, is\_paid=None, \_from=None, to=None, is\_one\_off\_charge=None*)

[List Payment \(subscription\) Paddle docs](#)

**Parameters**

- **subscription\_id** (`typing.Optional[int]`) –
- **plan** (`typing.Optional[int]`) –
- **is\_paid** (`typing.Union[bool, int, None]`) –
- **\_from** (`typing.Union[str, datetime.date, None]`) –
- **to** (`typing.Union[str, datetime.date, None]`) –
- **is\_one\_off\_charge** (`typing.Optional[bool]`) –

**Return type** `typing.List[dict]`

**list\_subscription\_users** (*subscription\_id=None, plan\_id=None, state=None, page=None, results\_per\_page=None*)

[List Users \(subscription\) Paddle docs](#)

Note: response does not include any information in pages and totals

**Parameters**

- **subscription\_id** (`typing.Optional[int]`) –
- **plan\_id** (`typing.Optional[int]`) –
- **state** (`typing.Optional[int]`) –
- **page** (`typing.Optional[int]`) –
- **results\_per\_page** (`typing.Optional[int]`) –

**Return type** `typing.List[dict]`

**list\_subscriptions** (*subscription\_id=None, plan\_id=None, state=None, page=None, results\_per\_page=None*)

List Users (subscription) [Paddle docs](#)

Note: response does not include any information in pages and totals

**Parameters**

- **subscription\_id** (`typing.Optional[int]`) –
- **plan\_id** (`typing.Optional[int]`) –
- **state** (`typing.Optional[int]`) –
- **page** (`typing.Optional[int]`) –
- **results\_per\_page** (`typing.Optional[int]`) –

**Return type** `typing.List[dict]`

**list\_transactions** (*entity, entity\_id, page=None*)

List Transactions [Paddle docs](#)

**Parameters**

- **entity** (`str`) –
- **entity\_id** (`typing.Union[str, int]`) –
- **page** (`typing.Optional[int]`) –

**Return type** `typing.List[dict]`

**pause\_subscription** (*subscription\_id*)

Update Subscription [Paddle docs](#)

There is no Pause Subscription endpoint in Paddle's API. This is a convenient helper method for update\_subscription as no extra data can be sent when pausing/resuming subscriptions

**Parameters** **subscription\_id** (`int`) –

**Return type** `dict`

**refund\_product\_payment** (*order\_id, amount=None, reason=None*)

Refund Product Payment [Paddle docs](#)

**Parameters**

- **order\_id** (`typing.Union[str, int]`) –
- **amount** (`typing.Optional[float]`) –
- **reason** (`typing.Optional[str]`) –

**Return type** `dict`

**reschedule\_subscription\_payment** (*payment\_id, date*)

Reschedule Payment (subscription) [Paddle docs](#)

**Parameters**

- **payment\_id** (`int`) –
- **date** (`typing.Union[str, datetime.date]`) –

**Return type** `bool`

**resume\_subscription** (*subscription\_id*)

[Update Subscription Paddle docs](#)

There is no Resume Subscription endpoint in Paddle's API. This is a convenient helper method for update\_subscription as no extra data can be sent when pausing/resuming subscriptions

**Parameters** **subscription\_id** (int) –

**Return type** dict

**update\_coupon** (*coupon\_code=None, new\_coupon\_code=None, group=None, new\_group=None, product\_ids=None, expires=None, allowed\_uses=None, currency=None, minimum\_threshold=None, discount\_amount=None, recurring=None*)

[Update Coupon Paddle docs](#)

**Parameters**

- **coupon\_code** (typing.Optional[str]) –
- **new\_coupon\_code** (typing.Optional[str]) –
- **group** (typing.Optional[str]) –
- **new\_group** (typing.Optional[str]) –
- **product\_ids** (typing.Optional[list]) –
- **expires** (typing.Union[str, datetime.datetime, None]) –
- **allowed\_uses** (typing.Optional[int]) –
- **currency** (typing.Optional[str]) –
- **minimum\_threshold** (typing.Optional[int]) –
- **discount\_amount** (typing.Optional[float]) –
- **recurring** (typing.Optional[bool]) –

**Return type** dict

**update\_subscription** (*subscription\_id, quantity=None, currency=None, recurring\_price=None, bill\_immediately=None, plan\_id=None, prorate=None, keep\_modifiers=None, passthrough=None, pause=None*)

[Update Subscription Paddle docs](#)

**Parameters**

- **subscription\_id** (int) –
- **quantity** (typing.Optional[int]) –
- **currency** (typing.Optional[str]) –
- **recurring\_price** (typing.Optional[float]) –
- **bill\_immediately** (typing.Optional[bool]) –
- **plan\_id** (typing.Optional[int]) –
- **prorate** (typing.Optional[bool]) –
- **keep\_modifiers** (typing.Optional[bool]) –
- **passthrough** (typing.Optional[str]) –
- **pause** (typing.Optional[bool]) –

**Return type** dict





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## INDEX

### A

`add_modifier()` (*paddle.PaddleClient* method), 4

### C

`cancel_subscription()` (*paddle.PaddleClient* method), 4

`create_coupon()` (*paddle.PaddleClient* method), 4

`create_one_off_charge()` (*paddle.PaddleClient* method), 5

`create_pay_link()` (*paddle.PaddleClient* method), 5

`create_plan()` (*paddle.PaddleClient* method), 7

### D

`delete_coupon()` (*paddle.PaddleClient* method), 7

`delete_modifier()` (*paddle.PaddleClient* method), 7

### G

`generate_license()` (*paddle.PaddleClient* method), 7

`get_order_details()` (*paddle.PaddleClient* method), 8

`get_plan()` (*paddle.PaddleClient* method), 8

`get_prices()` (*paddle.PaddleClient* method), 8

`get_user_history()` (*paddle.PaddleClient* method), 8

`get_webhook_history()` (*paddle.PaddleClient* method), 8

### L

`list_coupons()` (*paddle.PaddleClient* method), 8

`list_modifiers()` (*paddle.PaddleClient* method), 9

`list_plans()` (*paddle.PaddleClient* method), 9

`list_products()` (*paddle.PaddleClient* method), 9

`list_subscription_payments()` (*paddle.PaddleClient* method), 9

`list_subscription_users()` (*paddle.PaddleClient* method), 9

`list_subscriptions()` (*paddle.PaddleClient* method), 9

`list_transactions()` (*paddle.PaddleClient* method), 10

### P

*PaddleClient* (class in *paddle*), 4

`pause_subscription()` (*paddle.PaddleClient* method), 10

### R

`refund_product_payment()` (*paddle.PaddleClient* method), 10

`reschedule_subscription_payment()` (*paddle.PaddleClient* method), 10

`resume_subscription()` (*paddle.PaddleClient* method), 10

### U

`update_coupon()` (*paddle.PaddleClient* method), 11

`update_subscription()` (*paddle.PaddleClient* method), 11