

Package ‘paddleR’

June 24, 2025

Type Package

Title Wrapper for 'Paddle' API (Application Programming Interface)

Version 0.1.2

Description

A wrapper for 'Paddle' - The Merchant of Record for digital products API (Application Programming Interface) <<https://developer.paddle.com/api-reference/overview>>. Provides functions to manage and analyze products, customers, invoices and many more.

License MIT + file LICENSE

Encoding UTF-8

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports httr2, rlang

Depends R (>= 4.1.0)

RoxygenNote 7.3.2

VignetteBuilder knitr

URL <https://arnold-kakas.github.io/paddleR/>,
<https://github.com/Arnold-Kakas/paddleR>

BugReports <https://github.com/Arnold-Kakas/paddleR/issues>

NeedsCompilation no

Author Arnold Kakas [aut, cre, cph]

Maintainer Arnold Kakas <kakasarnold@gmail.com>

Repository CRAN

Date/Publication 2025-06-24 08:20:02 UTC

Contents

drop_nulls	3
get	3
get_paddle_key	4

get_paddle_mode	4
get_paddle_url	5
handle_paddle_response	5
paddle_activate_trial_subscription	6
paddle_cancel_subscription	6
paddle_create_customer	7
paddle_create_customer_address	8
paddle_create_customer_business	9
paddle_create_discount	10
paddle_create_price	11
paddle_create_product	12
paddle_create_transaction	13
paddle_generate_auth_token	15
paddle_get_events	16
paddle_get_update_payment_transaction	16
paddle_list_credit_balances	17
paddle_list_customers	18
paddle_list_customer_addresses	19
paddle_list_customer_businesses	20
paddle_list_discounts	21
paddle_list_prices	22
paddle_list_products	23
paddle_list_subscriptions	24
paddle_list_transactions	25
paddle_pause_subscription	26
paddle_preview_one_time_charge	27
paddle_preview_subscription_update	28
paddle_preview_transaction	29
paddle_resume_subscription	31
paddle_update_customer	31
paddle_update_customer_address	32
paddle_update_customer_business	34
paddle_update_discount	35
paddle_update_price	36
paddle_update_product	37
paddle_update_subscription	38
paddle_update_transaction	40
post	41
post_excl_body	42
set_paddle_mode	42
update	43

drop_nulls	<i>Drop NULL values from a list</i>
------------	-------------------------------------

Description

Drop NULL values from a list

Usage

```
drop_nulls(x)
```

Arguments

x	A list from which to remove NULL values.
---	--

Value

A list with NULL values removed.

get	<i>Make a GET request to Paddle API</i>
-----	---

Description

Make a GET request to Paddle API

Usage

```
get(link)
```

Arguments

link	The API endpoint URL.
------	-----------------------

Value

Parsed JSON response from the API.

get_paddle_key	<i>Get Paddle API Key Based on Current Mode</i>
----------------	---

Description

Resolves the Paddle API key based on environment variables and the selected mode (live or sandbox).

Usage

```
get_paddle_key()
```

Value

A character string representing the Paddle API key.

Examples

```
get_paddle_key() # returns token for current mode
```

get_paddle_mode	<i>Get Paddle Mode This function retrieves the current mode set for Paddle API requests.</i>
-----------------	--

Description

Get Paddle Mode This function retrieves the current mode set for Paddle API requests.

Usage

```
get_paddle_mode()
```

Value

The current mode for Paddle API requests, either "live" or "sandbox".

Examples

```
get_paddle_mode() # should return "live" by default
```

get_paddle_url	<i>Get Paddle URL This function retrieves the base URL for Paddle API requests based on the current mode.</i>
----------------	---

Description

Get Paddle URL This function retrieves the base URL for Paddle API requests based on the current mode.

Usage

```
get_paddle_url()
```

Value

The base URL for Paddle API requests.

Examples

```
get_paddle_url() # returns token for current mode
```

handle_paddle_response	<i>Handle Paddle API Response</i>
------------------------	-----------------------------------

Description

Parses and returns the response if successful, otherwise stops with an error message.

Usage

```
handle_paddle_response(resp)
```

Arguments

resp	Response object from 'httr2::req_perform()'
------	---

Value

Parsed JSON content from response, or throws error if HTTP status is not 2xx.

```
paddle_activate_trial_subscription
```

Activate a trialing subscription

Description

Activates a trialing subscription using its ID. Only automatically-collected subscriptions with status = "trialing" can be activated.

Usage

```
paddle_activate_trial_subscription(id)
```

Arguments

id Character. Required. The Paddle subscription ID (e.g. "sub_abc123").

Details

This triggers an immediate charge and recalculates billing dates from activation time.

Value

A list with updated subscription entity and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_activate_trial_subscription(id = "sub_123")
```

```
paddle_cancel_subscription
```

Cancel a Paddle subscription

Description

Cancels a subscription using its ID. Defaults to cancel at next billing period unless 'effective_from' is set to "immediately".

Usage

```
paddle_cancel_subscription(id, effective_from = NULL)
```

Arguments

`id` Character. Required. Paddle subscription ID, e.g. "sub_abc123".

`effective_from` Character or NULL. Optional. One of "next_billing_period" or "immediately". Defaults to "next_billing_period".

Value

A list with the updated subscription entity and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_cancel_subscription(
  id = "sub_123",
  effective_from = "immediately"
)
```

`paddle_create_customer`*Create a Customer.*

Description

If successful, your response includes a copy of the new customer entity.

Usage

```
paddle_create_customer(email, name = NULL, custom_data = NULL, locale = NULL)
```

Arguments

`email` Character. Email address for the customer. Required.

`name` Character. Full name of the customer. Optional.

`custom_data` Named list of custom metadata. Optional.

`locale` Character. Locale string (IETF BCP 47). Optional, defaults to "en".

Value

A data frame with the new customer info.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_create_customer(
  email = "test@example.com",
  name = "Test User",
  custom_data = list(plan = "demo"),
  locale = "en-US"
)
```

paddle_create_customer_address

Create an Address for a Customer

Description

Creates a new address for a given customer in the Paddle API.

Usage

```
paddle_create_customer_address(
  id,
  country_code,
  description = NULL,
  first_line = NULL,
  second_line = NULL,
  city = NULL,
  postal_code = NULL,
  region = NULL,
  custom_data = NULL
)
```

Arguments

id	Character. Paddle customer ID (e.g., "ctm_123"). Required.
country_code	Character. Two-letter ISO 3166-1 alpha-2 country code. Required.
description	Character. Optional description for internal reference.
first_line	Character. Optional. First line of address.
second_line	Character. Optional. Second line of address.
city	Character. Optional. City name.
postal_code	Character. Optional. ZIP or postal code. Required for some countries.
region	Character. Optional. Region, state, or county.
custom_data	Named list. Optional custom metadata.

Value

A list containing the created address entity and metadata.

Examples

```
result <- paddle_create_customer_address(  
  id = "ctm_123",  
  country_code = "US",  
  city = "New York",  
  postal_code = "10001"  
)
```

```
paddle_create_customer_business  
  Create a Business for a Customer
```

Description

Creates a new business entity associated with a specific customer in the Paddle API.

Usage

```
paddle_create_customer_business(  
  id,  
  name,  
  company_number = NULL,  
  tax_identifier = NULL,  
  contacts = NULL,  
  custom_data = NULL  
)
```

Arguments

id	Character. Paddle customer ID (e.g., "ctm_123"). Required.
name	Character. Name of the business. Required.
company_number	Character. Optional company number.
tax_identifier	Character. Optional tax/VAT ID.
contacts	List of contact objects (each a named list with "name" and required "email"). Optional.
custom_data	Named list of additional custom fields. Optional.

Value

A list containing the created business entity and metadata.

Examples

```
result <- paddle_create_customer_business(  
  id = "ctm_123",  
  name = "Acme Inc.",  
  tax_identifier = "123456789",  
  contacts = list(list(email = "ceo@acme.com")) )  
)
```

paddle_create_discount

Create a Paddle Discount

Description

Sends a POST request to the Paddle API to create a new discount.

Usage

```
paddle_create_discount(  
  amount,  
  description,  
  type,  
  enabled_for_checkout = NULL,  
  code = NULL,  
  mode = NULL,  
  currency_code = NULL,  
  recur = NULL,  
  maximum_recurring_intervals = NULL,  
  usage_limit = NULL,  
  restrict_to = NULL,  
  expires_at = NULL,  
  custom_data = NULL  
)
```

Arguments

amount	Character. Amount to discount by. Required.
description	Character. Internal description of the discount. Required.
type	Character. Type of discount ("percentage", "flat", or "flat_per_seat"). Required.
enabled_for_checkout	Logical. Can be used at checkout? Optional.
code	Character. Optional promo code (letters/numbers, max 32 characters).
mode	Character. Filter discounts by mode (one of "standard", "custom"). Optional.
currency_code	Character. Required for "flat" or "flat_per_seat". Optional.

recur	Logical. Should it recur for subscriptions? Optional.
maximum_recurring_intervals	Integer. Number of times to recur. Optional.
usage_limit	Integer. Maximum redemptions. Optional.
restrict_to	Character vector of product or price IDs. Optional.
expires_at	Character. RFC 3339 datetime string. Optional.
custom_data	Named list. Custom metadata. Optional.

Value

A list containing the created discount and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_create_discount(
  amount = "10.00",
  description = "10% off",
  type = "percentage")
```

paddle_create_price *Create a Paddle Price*

Description

Creates a new price for a product. The price defines how much customers pay, how often, and under what terms.

Usage

```
paddle_create_price(
  description,
  product_id,
  unit_price,
  type = NULL,
  name = NULL,
  billing_cycle = NULL,
  trial_period = NULL,
  tax_mode = NULL,
  unit_price_overrides = NULL,
  quantity = NULL,
  custom_data = NULL
)
```

Arguments

description	Internal description for your team. Required.
product_id	ID of the product this price belongs to. Required.
unit_price	A list with 'amount' (string, lowest denomination, e.g. for 10 USD write 1000 (lowest denomination = cents)) and 'currency_code' (3-letter ISO). Required.
type	Type of item (one of "standard" and "custom"). Optional. Defaults to "standard".
name	Name of this price (shown at checkout/invoices). Optional.
billing_cycle	List with 'frequency' and 'interval' (one of "day", "week", "month", "year"). Optional. Omit for one-time pricing.
trial_period	List with 'frequency' and 'interval' (one of "day", "week", "month", "year"). Optional. Requires 'billing_cycle'.
tax_mode	One of "account_setting", "external", "internal". Optional.
unit_price_overrides	A list of overrides with 'country_codes' (Supported two-letter ISO 3166-1 alpha-2 country code) and 'unit_price' (same list as in unit_price parameter). Optional.
quantity	List with 'minimum' and 'maximum' quantity limits. Optional.
custom_data	Named list of custom metadata. Optional.

Value

A list representing the created price entity and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_create_price(
  description = "Standard monthly subscription",
  product_id = "pro_123",
  unit_price = list(
    amount = "1000", # 10.00 USD
    currency_code = "USD"
  )
)
```

paddle_create_product *Create a Paddle Product*

Description

Creates a new product in Paddle. You must specify a name and tax category (chosen from a predefined set supported by Paddle).

Usage

```
paddle_create_product(
  name,
  tax_category,
  description = NULL,
  type = NULL,
  image_url = NULL,
  custom_data = NULL
)
```

Arguments

name	Name of the product. Required.
tax_category	Character vector of tax categories to filter. One of "digital-goods", "ebooks", "implementation-services", "professional-services", "saas", "software-programming-services", "standard", "training-services", and "website-hosting". Required.
description	Short description of the product. Optional.
type	Character. Type of product (one of "standard" and "custom"). Optional, defaults to "standard".
image_url	HTTPS URL for the product image (1:1 recommended). Optional.
custom_data	Named list of your own structured key-value metadata. Optional.

Value

A list representing the newly created product.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_create_product(
  name = "My Product",
  tax_category = "digital-goods",
  description = "A great product"
)
```

paddle_create_transaction

Create a Transaction

Description

Creates a new transaction in the Paddle API.

Usage

```

paddle_create_transaction(
  items,
  status = NULL,
  customer_id = NULL,
  address_id = NULL,
  business_id = NULL,
  custom_data = NULL,
  currency_code = NULL,
  collection_mode = NULL,
  discount_id = NULL,
  billing_details = NULL,
  billing_period = NULL,
  checkout = NULL,
  include = NULL
)

```

Arguments

<code>items</code>	List of item objects. Each must contain <code>'price_id'</code> (Character, required) and <code>'quantity'</code> (integer, required).
<code>status</code>	Character. Optional. Must be <code>"billed"</code> or skipped to let Paddle set the status.
<code>customer_id</code>	Character. Optional. Paddle customer ID (e.g., <code>"ctm_123"</code>).
<code>address_id</code>	Character. Optional. Paddle address ID (e.g., <code>"add_456"</code>). Requires <code>'customer_id'</code> .
<code>business_id</code>	Character. Optional. Paddle business ID. Requires <code>'customer_id'</code> .
<code>custom_data</code>	Named list. Optional structured metadata.
<code>currency_code</code>	Character. Optional. Must be <code>"USD"</code> , <code>"EUR"</code> , or <code>"GBP"</code> if <code>'collection_mode' = "manual"</code> .
<code>collection_mode</code>	Character. Optional. Must be one of <code>"automatic"</code> or <code>"manual"</code> . Defaults to <code>"automatic"</code> .
<code>discount_id</code>	Character. Optional. Paddle discount ID.
<code>billing_details</code>	List. Required if <code>'collection_mode' = "manual"</code> . May include: - <code>'payment_terms'</code> → list with <code>'interval'</code> (Character, must be one of <code>"day"</code> , <code>"week"</code> , <code>"month"</code> , <code>"year"</code>) and <code>'frequency'</code> (integer) - <code>'enable_checkout'</code> (logical) - <code>'purchase_order_number'</code> (Character) - <code>'additional_information'</code> (Character or null)
<code>billing_period</code>	List with <code>'starts_at'</code> and <code>'ends_at'</code> in RFC 3339 format. Optional.
<code>checkout</code>	List with <code>'url'</code> (Character) to override payment link. Optional.
<code>include</code>	Character vector. Must be one of <code>"address"</code> , <code>"adjustments"</code> , <code>"adjustments_totals"</code> , <code>"available_payment_methods"</code> , <code>"business"</code> , <code>"customer"</code> , <code>"discount"</code> . Optional.

Value

A list containing the created transaction and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_create_transaction(
  items = list(
    list(price_id = "pri_123", quantity = 1)
  )
)
```

paddle_generate_auth_token

Generate an Authentication Token for a Paddle Customer

Description

Creates a temporary authentication token for a customer. The token allows Paddle.js to present saved payment methods at checkout.

Usage

```
paddle_generate_auth_token(id)
```

Arguments

id	Paddle customer ID (required).
----	--------------------------------

Value

A list containing the auth token and expiry details.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_generate_auth_token("ctm_123")
```

paddle_get_events *Get Paddle Events*

Description

Retrieves a list of event entities from Paddle. You can optionally filter by event type.

Usage

```
paddle_get_events(after = NULL, order_by = NULL, per_page = NULL)
```

Arguments

after	Character. Return entities after the specified Paddle ID when working with paginated endpoints. Optional.
order_by	Character. Optional. Must be in the form "id[ASC]" or "id[DESC]".
per_page	Integer. Optional. Max 200, defaults to 50.

Value

A list of events returned by the Paddle API.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_get_events(
  per_page = 2
)
```

paddle_get_update_payment_transaction *Get a transaction to update payment method*

Description

Returns a transaction that can be passed to a checkout to update payment details. Only for subscriptions where 'collection_mode' is 'automatic'.

Usage

```
paddle_get_update_payment_transaction(id)
```

Arguments

id	Character. Required. Paddle ID of the subscription (e.g. "sub_abc123").
----	---

Details

Depending on subscription status: - For 'past_due', returns the most recent failed transaction. - For 'active', creates a new zero-amount transaction.

Value

A list containing transaction details for payment method update.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_get_update_payment_transaction(
  id = "sub_123"
)
```

paddle_list_credit_balances

List Credit Balances for a Customer

Description

Retrieves credit balances across all currencies for a specific customer from the Paddle API.

Usage

```
paddle_list_credit_balances(id, currency_code = NULL)
```

Arguments

id Character. Paddle customer ID (e.g., "ctm_123"). Required.

currency_code Optional character vector of ISO 4217 currency codes to filter results.

Value

A list of credit balances by currency with available, reserved, and used totals.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_list_credit_balances(
  id = "ctm_123",
  currency_code = c("USD", "EUR")
)
```

paddle_list_customers *Retrieve Paddle Customers*

Description

Fetches a paginated list of customers. By default, only active customers are returned.

Usage

```
paddle_list_customers(
  email = NULL,
  id = NULL,
  status = NULL,
  after = NULL,
  order_by = NULL,
  per_page = NULL,
  search = NULL
)
```

Arguments

email	Vector of email addresses to match exactly. Optional.
id	Character vector of Paddle customer IDs. Optional.
status	Character vector of statuses to filter by ("active" or "archived"). Optional.
after	Character. Return entities after the specified Paddle ID when working with paginated endpoints. Optional.
order_by	Character. Order results by "id[ASC]" or "id[DESC]". Optional.
per_page	Integer. Number of results per page (1–200). Optional, defaults to 50.
search	Character. Search term (one of "id", "name", "email"). Optional.

Value

A list with customer data and pagination info.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_list_customers()
```

`paddle_list_customer_addresses`*List Addresses for a Customer*

Description

Fetches a paginated list of addresses associated with a given customer from the Paddle API. By default, only active addresses are returned.

Usage

```
paddle_list_customer_addresses(  
  id,  
  address_id = NULL,  
  status = NULL,  
  after = NULL,  
  order_by = NULL,  
  per_page = NULL,  
  search = NULL  
)
```

Arguments

<code>id</code>	Character. Paddle customer ID (e.g., "ctm_abc123"). Required.
<code>address_id</code>	Character vector of address IDs (e.g., "add_123"). Optional.
<code>status</code>	Character vector. Optional. Valid: "active", "archived".
<code>after</code>	Character. Paddle ID to start listing after (for pagination). Optional.
<code>order_by</code>	Character. One of '"id[ASC]"', '"id[DESC]"'. Optional.
<code>per_page</code>	Integer. Number of results per page (max 200). Optional.
<code>search</code>	Character. Search query across most address fields. Optional. Cannot match 'status', 'created_at', or 'updated_at'.

Value

A list containing address data and pagination metadata.

Examples

```
set_paddle_mode("sandbox")  
result <- paddle_list_customer_addresses(id = "ctm_123")
```

`paddle_list_customer_businesses`*List Businesses for a Customer*

Description

Retrieves a paginated list of businesses associated with a given customer from the Paddle API. By default, only active businesses are returned.

Usage

```
paddle_list_customer_businesses(  
  id,  
  business_id = NULL,  
  status = NULL,  
  after = NULL,  
  order_by = NULL,  
  per_page = NULL,  
  search = NULL  
)
```

Arguments

<code>id</code>	Character. Paddle customer ID (e.g., "ctm_abc123"). Required.
<code>business_id</code>	Character vector of business IDs (e.g., "biz_123"). Optional.
<code>status</code>	Character. Status of the customer ("active" or "archived"). Optional.
<code>after</code>	Character. Pagination cursor: return entities after this ID. Optional.
<code>order_by</code>	Character. Must be one of "id[ASC]", "id[DESC]". Optional.
<code>per_page</code>	Integer. Max number of results (default: 50, max: 200). Optional.
<code>search</code>	Character. Optional search query across business fields and contacts (except status, created_at, updated_at).

Value

A list containing business data and pagination metadata.

Examples

```
set_paddle_mode("sandbox")  
result <- paddle_list_customer_businesses(id = "ctm_123")
```

paddle_list_discounts *Retrieve Paddle Discounts*

Description

Fetches a paginated list of discounts from the Paddle API. By default, only active discounts are returned.

Usage

```
paddle_list_discounts(  
  id = NULL,  
  code = NULL,  
  status = NULL,  
  mode = NULL,  
  after = NULL,  
  order_by = NULL,  
  per_page = NULL  
)
```

Arguments

id	Character vector of discount IDs (e.g., "dsc_xxx"). Optional.
code	Character vector of discount codes. Optional.
status	Character vector of discount statuses (one of "active", "archived"). Optional.
mode	Character. Filter discounts by mode (one of "standard", "custom"). Optional.
after	Character. Return entities after the specified Paddle ID (used for pagination). Optional.
order_by	Character. Ordering string (e.g., "id[ASC]" or "created_at[DESC]"). Optional.
per_page	Integer. Number of results per page (max 200). Optional, defaults to 50.

Value

A list containing discount data and pagination metadata.

Examples

```
set_paddle_mode("sandbox")  
result <- paddle_list_discounts()
```

paddle_list_prices	<i>Retrieve a List of Paddle Prices</i>
--------------------	---

Description

Fetches a paginated list of price entities from Paddle. Filters are available for ID, product, status, recurrence, and ordering. Optionally includes related products.

Usage

```
paddle_list_prices(  
  id = NULL,  
  product_id = NULL,  
  status = NULL,  
  include = NULL,  
  order_by = NULL,  
  per_page = NULL,  
  after = NULL,  
  recurring = NULL,  
  type = NULL  
)
```

Arguments

id	Character vector of Paddle price IDs (e.g., "pri_abc123"). Optional.
product_id	Character vector of Paddle product IDs. Optional.
status	Character vector of status filters (e.g., "active", "archived"). Optional.
include	Character vector of related entities - default 'NULL' or "product". Optional.
order_by	String specifying sort field and direction (e.g., "id[ASC]"). Valid fields for ordering: 'billing_cycle.frequency', 'billing_cycle.interval', 'id', 'product_id', 'quantity.maximum', 'quantity.minimum', 'status', 'tax_mode', 'unit_price.amount', and 'unit_price.currency_code'. Valid directions "[ASC]" and "[DESC]". Optional.
per_page	Number of results per page (max 200). Optional, defaults to 50.
after	Return entities after the specified Paddle ID when working with paginated endpoints. Optional.
recurring	Logical: TRUE to return recurring prices, FALSE for one-time. Optional.
type	Character. Type of item (one of "standard" and "custom"). Optional.

Value

A list of price entities and pagination metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_list_prices()
```

paddle_list_products *Retrieve Paddle Products*

Description

Fetches a paginated list of products from the Paddle API. By default, only active products are returned.

Usage

```
paddle_list_products(
  id = NULL,
  status = NULL,
  tax_category = NULL,
  type = NULL,
  include = NULL,
  after = NULL,
  order_by = NULL,
  per_page = NULL
)
```

Arguments

id	Character vector of product IDs to match. Optional.
status	Character vector of statuses to filter by (e.g., "active", "archived"). Optional.
tax_category	Character vector of tax categories to filter. One of "digital-goods", "ebooks", "implementation-services", "professional-services", "saas", "software-programming-services", "standard", "training-services", and "website-hosting". Optional.
type	Character. Type of product (one of "standard" and "custom"). Optional.
include	Character vector of related entities - default 'NULL' or "prices". Optional.
after	Character. Return entities after the specified Paddle ID when working with paginated endpoints.. Optional.
order_by	Character. Ordering string (e.g., "id[ASC]"). Valid fields for ordering: "created_at", "custom_data", "id", "description", "image_url", "name", "status", "tax_category", and "updated_at". Valid directions "[ASC]" and "[DESC]" Optional. Optional.
per_page	Number of products per page (max 200). Optional, defaults to 50.

Value

A list containing product data and pagination metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_list_products()
```

```
paddle_list_subscriptions
```

List subscriptions

Description

Retrieves a paginated list of subscriptions from the Paddle API.

Usage

```
paddle_list_subscriptions(
  id = NULL,
  customer_id = NULL,
  address_id = NULL,
  price_id = NULL,
  status = NULL,
  scheduled_change_action = NULL,
  collection_mode = NULL,
  after = NULL,
  order_by = NULL,
  per_page = NULL
)
```

Arguments

id, customer_id, address_id, price_id	Character vectors. Optional filters.
status	Character vector. Optional. Must be one of "active", "cancelled", "past_due", "paused", "trialing".
scheduled_change_action	Character vector. Optional. Must be one of "cancel", "pause", "resume".
collection_mode	Character. Optional. Must be one of "automatic", "manual".
after	Character. Optional. Return entities after the specified Paddle ID when working with paginated endpoints.
order_by	Character. Optional. Must be in the form "id[ASC]" or "id[DESC]".
per_page	Integer. Optional. Max 200, defaults to 50.

Value

A list with subscription data and pagination metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_list_subscriptions()
```

paddle_list_transactions

List Paddle Transactions

Description

Retrieves a paginated list of transactions from the Paddle API.

Usage

```
paddle_list_transactions(
  after = NULL,
  id = NULL,
  customer_id = NULL,
  subscription_id = NULL,
  invoice_number = NULL,
  origin = NULL,
  status = NULL,
  collection_mode = NULL,
  billed_at = NULL,
  created_at = NULL,
  updated_at = NULL,
  order_by = NULL,
  include = NULL,
  per_page = NULL
)
```

Arguments

after	Character. Paddle ID cursor for pagination. Optional.
id	Character vector of transaction IDs. Optional.
customer_id	Character vector of customer IDs. Optional.
subscription_id	Character vector of subscription IDs or "null" to fetch unlinked. Optional.
invoice_number	Character vector of invoice numbers. Optional.
origin	Character vector of origins. Optional.
status	Character vector of statuses. Must be one of "draft", "ready", "billed", "paid", "completed", "canceled", "past_due". Optional.
collection_mode	Character. Must be one of "automatic" or "manual". Optional.

billed_at	Character. RFC 3339 datetime or filter (e.g., "billed_at[LT]=2023-01-01T00:00:00Z"). Optional.
created_at	Character. Same format as billed_at. Optional.
updated_at	Character. Same format as billed_at. Optional.
order_by	Character. Must be one of 'billed_at[ASC DESC]', 'created_at[ASC DESC]', 'id[ASC DESC]', 'updated_at[ASC DESC]'. Optional.
include	Character vector. Must be one of "address", "adjustments", "adjustments_totals", "available_payment_methods", "business", "customer", "discount". Optional.
per_page	Integer. Max results per page (max 200). Optional.

Value

A list containing transactions and pagination metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_list_transactions()
```

paddle_pause_subscription
<i>Pause a subscription</i>

Description

Pauses a subscription using its ID. You can pause at the end of the billing period (default), pause immediately by setting 'effective_from = "immediately"', or set a resume date.

Usage

```
paddle_pause_subscription(
  id,
  effective_from = NULL,
  resume_at = NULL,
  on_resume = NULL
)
```

Arguments

id	Character. Required. Paddle subscription ID (e.g. "sub_abc123").
effective_from	Character or NULL. Optional. One of "next_billing_period" or "immediately". Defaults to "next_billing_period".
resume_at	Character or NULL. Optional. RFC 3339 date-time string when subscription should resume.
on_resume	Character or NULL. Optional. One of "start_new_billing_period" or "continue_billing_period".

Value

A list representing the updated subscription object.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_pause_subscription(id = "sub_123")
```

paddle_preview_one_time_charge

Preview a one-time charge for a subscription

Description

Previews a one-time charge for a subscription without billing it. Used to estimate the result of a charge for non-recurring items.

Usage

```
paddle_preview_one_time_charge(
  id,
  effective_from,
  items,
  on_payment_failure = NULL
)
```

Arguments

id	Character. Required. The Paddle subscription ID (e.g. "sub_abc123").
effective_from	Character. Required. When the one-time charge should be billed (RFC 3339 format).
items	List of item lists. Optional. Each must include 'price_id' (string) and 'quantity' (numeric). If updating an existing item and not changing the quantity, you may omit quantity.
on_payment_failure	Character. Optional. Must be one of: "prevent_change", "allow_change".

Value

A list with preview of immediate and next transactions.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_preview_one_time_charge(
  id = "sub_123",
  effective_from = "2025-07-01T00:00:00Z",
  items = list(list(price_id = "pri_123", quantity = 1))
)
```

paddle_preview_subscription_update

Preview an update to a subscription

Description

Previews an update to a subscription without applying those changes.

Usage

```
paddle_preview_subscription_update(
  id,
  customer_id = NULL,
  address_id = NULL,
  business_id = NULL,
  currency_code = NULL,
  next_billed_at = NULL,
  discount = NULL,
  collection_mode = NULL,
  billing_details = NULL,
  items = NULL,
  proration_billing_mode = NULL,
  on_payment_failure = NULL,
  custom_data = NULL,
  scheduled_change = NULL
)
```

Arguments

id	Character. Required. Paddle ID of the subscription (e.g. "sub_abc123").
customer_id	Character. Optional. Paddle customer ID.
address_id	Character. Optional. Paddle address ID.
business_id	Character or NULL. Optional.
currency_code	Character. Optional. Supported: "USD", "EUR", "GBP" (for manual collection).
next_billed_at	Character. Optional. RFC 3339 datetime string.

discount	List or NULL. Optional. Must include 'id' (string) and 'effective_from' (string, must be one of "immediately" or "next_billing_period"), or NULL to remove.
collection_mode	Character. Optional. One of: "automatic", "manual".
billing_details	List or NULL. Required if 'collection_mode' is "manual", NULL if changing collection_mode to automatic. Must include 'enable_checkout' (boolean), 'purchase_order_number' (string), 'payment_terms' (list with 'interval' ('day', 'week', 'month' or 'year') and 'frequency' (integer)) and optional 'additional_information' (string)
items	List of item lists. Optional. Each must include 'price_id' (string) and 'quantity' (numeric). If updating an existing item and not changing the quantity, you may omit quantity.
proration_billing_mode	Character. Required when making changes that impact billing. Must be one of: "prorated_immediately", "prorated_next_billing_period", "full_immediately", "full_next_billing_period" and "do_not_bill".
on_payment_failure	Character. Optional. Must be one of: "prevent_change", "allow_change".
custom_data	Named list or NULL. Optional.
scheduled_change	NULL. Set to NULL to remove a scheduled change.

Value

A list containing subscription preview and transaction impact.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_preview_subscription_update(
  id = "sub_123",
  custom_data = list(purpose = "example")
)
```

paddle_preview_transaction

Preview a transaction

Description

Sends a preview request to Paddle to simulate a transaction without creating it.

Usage

```
paddle_preview_transaction(
  items,
  customer_id = NULL,
  currency_code = NULL,
  discount_id = NULL,
  ignore_trials = NULL,
  customer_ip_address = NULL,
  address = NULL
)
```

Arguments

<code>items</code>	List of lists. Required. Each must include 'price_id' and 'quantity'.
<code>customer_id</code>	Character. Optional. Customer ID prefixed with "ctm_".
<code>currency_code</code>	Character. Optional. Must be valid ISO 4217 code.
<code>discount_id</code>	Character. Optional. Discount ID prefixed with "dsc_".
<code>ignore_trials</code>	Logical. Optional. If TRUE, disables trial discounts.
<code>customer_ip_address</code>	Character. Optional. Valid IPv4 or IPv6.
<code>address</code>	Named list. Optional. Must include 'country_code' (2-letter) and optionally 'postal_code'.

Value

A list with previewed transaction data.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_preview_transaction(
  items = list(list(price_id = "pri_123", quantity = 2)),
  customer_id = "ctm_123",
  currency_code = "EUR",
  discount_id = "dsc_123",
  ignore_trials = TRUE,
  customer_ip_address = "8.8.8.8"
)
```

`paddle_resume_subscription`*Resume a paused or scheduled-to-pause subscription*

Description

Resumes a paused subscription immediately or at a specified date. Also updates a scheduled pause if subscription is active.

Usage

```
paddle_resume_subscription(id, effective_from, on_resume = NULL)
```

Arguments

<code>id</code>	Character. Required. Paddle subscription ID (e.g. "sub_abc123").
<code>effective_from</code>	Character. Required. RFC 3339 datetime string when the resume should occur.
<code>on_resume</code>	Character or NULL. Optional. One of "start_new_billing_period" or "continue_billing_period".

Value

A list representing the updated subscription object.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_resume_subscription(
  id = "sub_123",
  effective_from = "2025-07-01T00:00:00Z"
)
```

`paddle_update_customer`*Update a Paddle Customer*

Description

Updates an existing customer using their Paddle ID.

Usage

```
paddle_update_customer(
  id,
  name = NULL,
  email = NULL,
  status = NULL,
  custom_data = NULL,
  locale = NULL
)
```

Arguments

id	Character. Paddle customer ID (required).
name	Character. Full name of the customer. Optional (can be 'NULL' to remove).
email	Character. Email address of the customer. Optional.
status	Character. Status of the customer ("active" or "archived"). Optional.
custom_data	Named list of custom metadata. Optional (can be 'NULL' to remove).
locale	Character. Locale string (IETF BCP 47). Optional.

Value

A list with the updated customer info.

Examples

```
set_paddle_mode("sandbox")
# Requires a valid id
result <- paddle_update_customer(
  id = "ctm_123",
  name = "Updated Name",
  status = "active"
)
```

paddle_update_customer_address
Update a Customer Address

Description

Updates an address for a specific customer using Paddle's API.

Usage

```
paddle_update_customer_address(  
  id,  
  address_id,  
  country_code = NULL,  
  description = NULL,  
  first_line = NULL,  
  second_line = NULL,  
  city = NULL,  
  postal_code = NULL,  
  region = NULL,  
  custom_data = NULL,  
  status = NULL  
)
```

Arguments

id	Character. Paddle customer ID (e.g., "ctm_123"). Required.
address_id	Character. Paddle address ID (e.g., "add_456"). Required.
country_code	Character. Optional. Two-letter ISO 3166-1 alpha-2 country code.
description	Character. Optional. Internal reference.
first_line	Character. Optional. First line of address.
second_line	Character. Optional. Second line of address.
city	Character. Optional. City name.
postal_code	Character. Optional. ZIP or postal code.
region	Character. Optional. Region, state, or county.
custom_data	Named list. Optional structured metadata.
status	Character. Status of the customer ("active" or "archived"). Optional.

Value

A list containing the updated address entity and metadata.

Examples

```
set_paddle_mode("sandbox")  
result <- paddle_update_customer_address(  
  id = "ctm_123",  
  address_id = "add_123",  
  city = "San Francisco",  
  region = "CA"  
)
```

`paddle_update_customer_business`*Update a Business for a Customer*

Description

Updates a business entity linked to a specific customer using the Paddle API.

Usage

```
paddle_update_customer_business(  
  id,  
  business_id,  
  name = NULL,  
  company_number = NULL,  
  tax_identifier = NULL,  
  status = NULL,  
  contacts = NULL,  
  custom_data = NULL  
)
```

Arguments

<code>id</code>	Character. Paddle customer ID (e.g., "ctm_123"). Required.
<code>business_id</code>	Character. Paddle business ID (e.g., "biz_456"). Required.
<code>name</code>	Character. Updated name of the business. Optional.
<code>company_number</code>	Character. Updated company number. Optional.
<code>tax_identifier</code>	Character. Numeric VAT/tax ID. Optional.
<code>status</code>	Character. Status of the customer (one of "active" or "archived"). Optional.
<code>contacts</code>	List of contact objects (named list with at least "email"). Optional.
<code>custom_data</code>	Named list of custom metadata. Optional.

Value

A list containing the updated business entity and metadata.

Examples

```
set_paddle_mode("sandbox")  
result <- paddle_update_customer_business(  
  id = "ctm_123",  
  business_id = "biz_123",  
  name = "Acme International"  
)
```

paddle_update_discount

Update a Paddle Discount

Description

Updates an existing discount by ID via the Paddle API.

Usage

```
paddle_update_discount(
  id,
  status = NULL,
  description = NULL,
  enabled_for_checkout = NULL,
  code = NULL,
  type = NULL,
  mode = NULL,
  amount = NULL,
  currency_code = NULL,
  recur = NULL,
  maximum_recurring_intervals = NULL,
  usage_limit = NULL,
  restrict_to = NULL,
  expires_at = NULL,
  custom_data = NULL
)
```

Arguments

id	Character. The Paddle discount ID (e.g., "dsc_123"). Required.
status	Character vector of discount statuses (one of "active", "archived"). Optional.
description	Character. Internal description. Optional.
enabled_for_checkout	Logical. Available for checkout? Optional.
code	Character. Discount code. Optional.
type	Character. Type of discount ("percentage", "flat", or "flat_per_seat"). Optional.
mode	Character. Filter discounts by mode (one of "standard", "custom"). Optional.
amount	Character. Discount amount. Optional.
currency_code	Character. Required for "flat"/"flat_per_seat". Optional.
recur	Logical. Repeating discount? Optional.
maximum_recurring_intervals	Integer. Number of repeats. Optional.

usage_limit	Integer. Max redemptions. Optional.
restrict_to	Character vector. Product or price IDs. Optional.
expires_at	Character. RFC 3339 datetime string. Optional.
custom_data	Named list. Custom metadata. Optional.

Value

A list containing the updated discount and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_update_discount(
  id = "dsc_123",
  description = "Updated description")
```

paddle_update_price	<i>Update a Paddle Price</i>
---------------------	------------------------------

Description

Updates an existing price entity in Paddle using its ID.

Usage

```
paddle_update_price(
  id,
  description = NULL,
  type = NULL,
  name = NULL,
  billing_cycle = NULL,
  trial_period = NULL,
  tax_mode = NULL,
  unit_price = NULL,
  unit_price_overrides = NULL,
  quantity = NULL,
  status = NULL,
  custom_data = NULL
)
```

Arguments

id	ID of the price. Required.
description	Internal description for your team. Required.
type	Type of item (one of "standard" and "custom"). Optional. Defaults to "standard".

name	Name of this price (shown at checkout/invoices). Optional.
billing_cycle	List with 'frequency' and 'interval' (one of "day", "week", "month", "year"). Optional. Omit for one-time pricing.
trial_period	List with 'frequency' and 'interval' (one of "day", "week", "month", "year"). Optional. Requires 'billing_cycle'.
tax_mode	One of "account_setting", "external", "internal". Optional.
unit_price	A list with 'amount' (string, lowest denomination, e.g. for 10 USD write 1000 (lowest denomination = cents)) and 'currency_code' (3-letter ISO). Required.
unit_price_overrides	A list of overrides with 'country_codes' (Supported two-letter ISO 3166-1 alpha-2 country code) and 'unit_price' (same list as in unit_price parameter). Optional.
quantity	List with 'minimum' and 'maximum' quantity limits. Optional.
status	Status of the price (one of "active", "archived"). Optional.
custom_data	Named list of custom metadata. Optional.

Value

A list representing the updated price entity.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_update_price(
  id = "pri_123",
  name = "Updated Price Name"
)
```

paddle_update_product *Update a Paddle Product*

Description

Updates an existing product using its Paddle ID. You can update any combination of fields, such as name, description, tax category, type, image URL, custom metadata, and status.

Usage

```
paddle_update_product(
  id,
  name = NULL,
  description = NULL,
  type = NULL,
  tax_category = NULL,
  image_url = NULL,
  custom_data = NULL,
  status = NULL
)
```

Arguments

<code>id</code>	The Paddle product ID (e.g., <code>"pro_abc123"</code>). Required.
<code>name</code>	Updated product name. Optional.
<code>description</code>	Updated product description. Optional (use <code>'NULL'</code> to clear).
<code>type</code>	Character. Type of product (one of <code>"standard"</code> and <code>"custom"</code>). Optional, defaults to <code>"standard"</code> .
<code>tax_category</code>	Character vector of tax categories to filter. One of <code>"digital-goods"</code> , <code>"ebooks"</code> , <code>"implementation-services"</code> , <code>"professional-services"</code> , <code>"saas"</code> , <code>"software-programming-services"</code> , <code>"standard"</code> , <code>"training-services"</code> , and <code>"website-hosting"</code> . Optional.
<code>image_url</code>	HTTPS image URL. Optional (use <code>'NULL'</code> to clear).
<code>custom_data</code>	Named list of key-value metadata. Optional (use <code>'NULL'</code> to clear).
<code>status</code>	Character vector of statuses (e.g., <code>"active"</code> , <code>"archived"</code>). Optional.

Value

A list representing the updated product.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_update_product(
  id = "pro_123",
  name = "Updated Product Name"
)
```

`paddle_update_subscription`

Update a subscription

Description

Updates a subscription using its ID. Required for changes like billing date, items, or proration.

Usage

```
paddle_update_subscription(
  id,
  customer_id = NULL,
  address_id = NULL,
  business_id = NULL,
  currency_code = NULL,
  next_billed_at = NULL,
  discount = NULL,
  collection_mode = NULL,
```

```

    billing_details = NULL,
    items = NULL,
    proration_billing_mode = NULL,
    on_payment_failure = NULL,
    custom_data = NULL,
    scheduled_change = NULL
)

```

Arguments

<code>id</code>	Character. Required. The ID of the subscription (e.g. "sub_abc123").
<code>customer_id</code>	Character. Optional. Paddle customer ID.
<code>address_id</code>	Character. Optional. Paddle address ID.
<code>business_id</code>	Character or NULL. Optional.
<code>currency_code</code>	Character. Optional. Supported: "USD", "EUR", "GBP" (for manual collection).
<code>next_billed_at</code>	Character. Optional. RFC 3339 datetime string.
<code>discount</code>	List or NULL. Optional. Must include 'id' (string) and 'effective_from' (string, must be one of "immediately" or "next_billing_period"), or NULL to remove.
<code>collection_mode</code>	Character. Optional. One of: "automatic", "manual".
<code>billing_details</code>	List or NULL. Required if 'collection_mode' is "manual", NULL if changing collection_mode to automatic. Must include 'enable_checkout' (boolean), 'purchase_order_number' (string), 'payment_terms' (list with 'interval' ('day', 'week', 'month' or 'year') and 'frequency' (integer)) and optional 'additional_information' (string)
<code>items</code>	List of item lists. Optional. Each must include 'price_id' (string) and 'quantity' (numeric). If updating an existing item and not changing the quantity, you may omit quantity.
<code>proration_billing_mode</code>	Character. Required when making changes that impact billing. Must be one of: "prorated_immediately", "prorated_next_billing_period", "full_immediately", "full_next_billing_period" and "do_not_bill".
<code>on_payment_failure</code>	Character. Optional. Must be one of: "prevent_change", "allow_change".
<code>custom_data</code>	Named list or NULL. Optional.
<code>scheduled_change</code>	NULL. Set to NULL to remove a scheduled change.

Value

A list with updated subscription entity and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_update_subscription(
  id = "sub_123",
  custom_data = list(purpose = "example")
)
```

paddle_update_transaction

Update a Transaction

Description

Updates a transaction by its ID. Only transactions with status ‘draft’ or ‘ready’ can be updated.

Usage

```
paddle_update_transaction(
  id,
  status = NULL,
  customer_id = NULL,
  address_id = NULL,
  business_id = NULL,
  custom_data = NULL,
  currency_code = NULL,
  collection_mode = NULL,
  discount_id = NULL,
  billing_details = NULL,
  payment_terms = NULL,
  billing_period = NULL,
  items = NULL,
  checkout_url = NULL,
  include = NULL
)
```

Arguments

id	Character. Required. Paddle ID of the transaction (e.g. “txn_abc123”).
status	Character. Optional. “billed” or “canceled”.
customer_id, address_id, business_id	Character. Optional Paddle IDs.
custom_data	Named list. Optional.
currency_code	Character. Optional. “USD”, “EUR”, or “GBP” if ‘collection_mode = “manual”’.

collection_mode	Character. Optional. "automatic" or "manual".
discount_id	Character. Optional.
billing_details	List. Optional, see API.
payment_terms	List with 'interval' and 'frequency'. Shortcut to fill 'billing_details\$payment_terms'.
billing_period	List with 'starts_at' and 'ends_at' in RFC 3339 format. Optional.
items	List of objects with 'price_id' and 'quantity'. Optional.
checkout_url	Character. Optional.
include	Character vector of related entities to return. Optional.

Value

A list with updated transaction data and metadata.

Examples

```
set_paddle_mode("sandbox")
result <- paddle_create_transaction(
  items = list(
    list(price_id = "price_123", quantity = 1)
  )
)
```

post

Make a POST request to Paddle API

Description

Make a POST request to Paddle API

Usage

```
post(link, body)
```

Arguments

link	The API endpoint URL.
body	The body of the request, typically a list to be converted to JSON.

Value

Parsed JSON response from the API.

post_excl_body	<i>Make a POST request to Paddle API without body</i>
----------------	---

Description

Make a POST request to Paddle API without body

Usage

```
post_excl_body(link)
```

Arguments

link	The API endpoint URL.
------	-----------------------

Value

Parsed JSON response from the API.

set_paddle_mode	<i>Set Paddle Mode This function sets the mode for Paddle API requests, either "sandbox" or "live".</i>
-----------------	---

Description

Set Paddle Mode This function sets the mode for Paddle API requests, either "sandbox" or "live".

Usage

```
set_paddle_mode(mode = c("live", "sandbox"))
```

Arguments

mode	A character string indicating the mode to set. Options are "sandbox" or "live". Defaults to "live".
------	---

Value

The base URL for Paddle API requests based on the selected mode.

Examples

```
set_paddle_mode("sandbox")
```

update	<i>Make a DELETE request to Paddle API</i>
--------	--

Description

Make a DELETE request to Paddle API

Usage

```
update(link, body)
```

Arguments

link	The API endpoint URL.
body	The body of the request, typically a list to be converted to JSON.

Value

Parsed JSON response from the API.

Index

drop_nulls, 3

get, 3

get_paddle_key, 4

get_paddle_mode, 4

get_paddle_url, 5

handle_paddle_response, 5

paddle_activate_trial_subscription, 6

paddle_cancel_subscription, 6

paddle_create_customer, 7

paddle_create_customer_address, 8

paddle_create_customer_business, 9

paddle_create_discount, 10

paddle_create_price, 11

paddle_create_product, 12

paddle_create_transaction, 13

paddle_generate_auth_token, 15

paddle_get_events, 16

paddle_get_update_payment_transaction, 16

paddle_list_credit_balances, 17

paddle_list_customer_addresses, 19

paddle_list_customer_businesses, 20

paddle_list_customers, 18

paddle_list_discounts, 21

paddle_list_prices, 22

paddle_list_products, 23

paddle_list_subscriptions, 24

paddle_list_transactions, 25

paddle_pause_subscription, 26

paddle_preview_one_time_charge, 27

paddle_preview_subscription_update, 28

paddle_preview_transaction, 29

paddle_resume_subscription, 31

paddle_update_customer, 31

paddle_update_customer_address, 32

paddle_update_customer_business, 34

paddle_update_discount, 35

paddle_update_price, 36

paddle_update_product, 37

paddle_update_subscription, 38

paddle_update_transaction, 40

post, 41

post_excl_body, 42

set_paddle_mode, 42

update, 43