autodoc-example Documentation

Release 0.4.3

Eric Beahan

Sep 01, 2023

CONTENTS

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

CHAPTER

ONE

INSTALLATION

You can install using pip

```
pip install python-amazon-ad-api
```

CHAPTER

TWO

CREDENTIALS

You can pass your credentials multiple ways, use one of them.

2.1 Config File

An example config file is provided in this repository, it supports multiple accounts. The program looks for a file called credentials.yml[1]

The config is parsed by confused[2], see their docs for more in depth information. Search paths are:

```
macOS: ~/.config/python-sp-api
Other Unix: ~/.config/python-sp-api
Windows: %APPDATA%\python-sp-api where the APPDATA environment variable falls back to
↪%HOME%\AppData\Roaming if undefined
```

If you're only using one account, place it under default. You can pass the account's name to the client to use any other account used in the credentials.yml[1] file.

```yaml
version: '1.0'

default:
  refresh_token: ''
  client_id: ''
  client_secret: ''
  profile_id: ''

another_account:
  refresh_token: ''
  client_id: ''
  client_secret: ''
  profile_id: ''
```

[1] https://github.com/denisneuf/python-amazon-ad-api/#credentials
[2] https://confuse.readthedocs.io/en/latest/usage.html#search-paths

2.1.1 Usage with default account

```
Campaigns().list_campaigns()
```

2.1.2 Usage with another_account

You can use every account's name from the config file for account

```
Campaigns(account="another_account", marketplace=Marketplaces.ES).list_campaigns()
```

2.1.3 References

2.2 Environment Variables

2.2.1 Use with environment variables

ENVIRONMENT VARIABLE	DESCRIPTION
AD_API_REFRESH_TOKEN	The refresh token used obtained via authorization
AD_API_CLIENT_ID	Your login with amazon app id
AD_API_CLIENT_SECRET	Your login with amazon client secret
AD_API_PROFILE_ID	Your profile id as Amazon Ad

To set environment variables in your python script, use

```
import os

os.environ.setdefault('AD_API_REFRESH_TOKEN', 'Your-Token-Here')
os.environ.setdefault('AD_API_CLIENT_ID', 'Your-Client_Id-Here')
os.environ.setdefault('AD_API_CLIENT_SECRET', 'Your-Client_Secret-Here')
os.environ.setdefault('AD_API_PROFILE_ID', 'Your-Profile_Id-Here')
```

2.2.2 Note

You still need create the .env file or by default the mode configuration will be sandbox for testing

```
# environment variables defined inside a .env file
# AWS_ENV=SANDBOX
AWS_ENV=PRODUCTION
```

2.3 From Code

You can override/set credentials from code by passing a `dict` to the client.

If you pass a value in credentials, other credentials from env variables or from a config file will be ignored.

Required fields:

```
credentials = dict(
    refresh_token='your-refresh_token',
    client_id='your-client_id',
    client_secret='your-client_secret',
    profile_id='your-profile_id',
)
```

2.3.1 Usage

```
import logging
from ad_api.base import AdvertisingApiException
from ad_api.api import sponsored_products

credentials = dict(
    refresh_token='your-refresh_token',
    client_id='your-client_id',
    client_secret='your-client_secret',
    profile_id='your-profile_id',
)

try:

    status = 'enabled'

    result=sponsored_products.Campaigns(credentials=credentials, debug=True).list_
↪campaigns(
        stateFilter=status
    )

    payload = result.payload

    logging.info(payload)

except AdvertisingApiException as error:

    logging.info(error)
```

CHAPTER

THREE

COMMON RESOURCES

3.1 Profiles

https://d3a0d0y2hgofx6.cloudfront.net/openapi/en-us/profiles/3-0/openapi.yaml

Profiles represent an advertiser and their account's marketplace, and are used in all subsequent API calls via a management scope, Amazon-Advertising-API-Scope. Reports and all entity management operations are associated with a single profile. Advertisers cannot have more than one profile for each marketplace.

Advertisers who operate in more than one marketplace (for example, Amazon.com, Amazon.co.uk, Amazon.co.jp) will have only one profile associated with each marketplace. See this link for a list of marketplaces associated with each endpoint.

To retrieve your profile IDs, call the listProfiles operation, and include a valid authorization access token in the header. Use a profileId from the returned list as the value for the management scope (Amazon-Advertising-API-Scope) in the headers for subsequent API calls.

class ad_api.api.**Profiles**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_profiles(*self*, ***kwargs*) → ApiResponse

Gets a list of profiles.

query **apiProgram**:*string* | Optional. Filters response to include profiles that have permissions for the specified Advertising API program only. Available values : billing, campaign, paymentMethod, store, report, account, posts

query **accessLevel**:*string* | Optional. Filters response to include profiles that have specified permissions for the specified Advertising API program only. Available values : edit, view

query **profileTypeFilter**:*string* | Optional. Filters response to include profiles that are of the specified types in the comma-delimited list. Available values : seller, vendor, agency

query **validPaymentMethodFilter**:*string* | Optional. Filter response to include profiles that have valid payment methods. Available values : true, false

Returns:

ApiResponse

Example getting a list of profiles

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException
```

(continues on next page)

(continued from previous page)

```
def list_profiles(**kwargs):

    logging.info("---------------------------------------")
    logging.info("Profiles > list_profiles(%s)" % kwargs)
    logging.info("---------------------------------------")

    try:

        result = Profiles(account=store, debug=True).list_profiles(
            **kwargs
        )
        logging.info(result)

        accounts_info = result.payload

        for account_info in accounts_info:
            logging.info(account_info)

    except AdvertisingApiException as error:
        logging.info(error)

list_profiles()
# list_profiles(profileTypeFilter="seller")
# list_profiles(profileTypeFilter="vendor")
# list_profiles(profileTypeFilter="agency")
# list_profiles(accessLevel="edit")
# list_profiles(apiProgram="store")
# list_profiles(validPaymentMethodFilter="true")
```

update_single_profile_assistant(*profile_id: int*, *daily_budget: int*, ***kwargs*) → ApiResponse

Update the daily budget for one or more profiles. Note that this operation is only used for Sellers using Sponsored Products.

'**profile_id**': *integer($int64)* | required {'description': 'The identifier of the profile.'}

'**daily_budget**': *number*, | required {'description': 'Note that this field applies to Sponsored Product campaigns for seller type accounts only. Not supported for vendor type accounts.'}

'****kwargs**': You can add other keyword args like the original method (countryCode, currencyCode, timezone, accountInfo{}) but as they are read-only if you try to modify will get INVALID_ARGUMENT: Cannot modify "value" for profile Returns:

ApiResponse

Example updating the budget of a single profile

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException

def update_single_profile_assistant(account_profile_id: int, account_daily_budget:␣
↪float, **kwargs):
```

(continues on next page)

(continued from previous page)

```
    logging.info("-------------------------------------")
    logging.info("Profiles > update_single_profile_assistant({}, {}, {})".
↪format(str(account_profile_id), str(account_daily_budget), str(kwargs)))
    logging.info("-------------------------------------")


    try:

        result = Profiles(account=store, debug=True).update_single_profile_
↪assistant(
            profile_id=account_profile_id,
            daily_budget=account_daily_budget,
            **kwargs

        )
        logging.info(result)
        payload = result.payload
        logging.info(payload)

    except AdvertisingApiException as error:
        logging.info(error)

amz_profile_id = 1495806522428699
amz_daily_budget = 21.50
update_single_profile_assistant(amz_profile_id, amz_daily_budget)
# update_single_profile_assistant(amz_profile_id, amz_daily_budget, countryCode="ES
↪", timezone="Europe/London")
```

update_profile(*body: dict, list, str*) → ApiResponse

Update the daily budget for one or more profiles. Note that this operation is only used for Sellers using Sponsored Products.

body: | REQUIRED {'description': 'An array of ad groups.}'

'**profileId**': *integer($int64)* | required {'description': 'The identifier of the profile.'}

'**countryCode**': *string* | readOnly {'description': 'The countryCode for a given country'}

'**currencyCode**': *string* | readOnly {'description': 'The currency used for all monetary values for entities under this profile.'}

'**dailyBudget**': *number* | required {'description': 'Note that this field applies to Sponsored Product campaigns for seller type accounts only. Not supported for vendor type accounts.'}

'**timezone**': *string* | readOnly {'description': 'The time zone used for all date-based campaign management and reporting.'}

'**accountInfo**': *AccountInfoAccountInfo* | readOnly {}

Returns:

ApiResponse

Example updating the budget of one or more profiles

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException
```

(continues on next page)

(continued from previous page)

```
def update_profile(data: (dict, list, str)):

    logging.info("-------------------------------------")
    logging.info("Profiles > update_profile(%s)" % str(data))
    logging.info("-------------------------------------")

    try:
        result = Profiles(account=store, debug=True).update_profile(
            body=data
        )
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


amz_profile_id = 1495806522428699
amz_daily_budget = 21.50

# Using a dict wrapped in a list. Note: countryCode, currencyCode, timezone and␣
↪accountInfo are read only
# try to update this fields will lead to INVALID ARGUMENT so in my opinion better␣
↪no need to send unless
# you store the data and you retrieve all the fields

update_data_list = \
[
    {
        'profileId': amz_profile_id,
        'dailyBudget': amz_daily_budget,
        'countryCode': 'ES',
        'currencyCode': 'EUR',
        'timezone': 'Europe/Paris',
        'accountInfo': {
            'id': 'A30E1B2F3U4K5Y',
             'marketplaceStringId': 'A1RKKUPIHCS9HS',
             'type': 'seller'
        }
    }
]

update_profile(update_data_list)


# The most simple and useful way to update the budget of a single profile
update_data_dict = \
{
    'profileId': amz_profile_id,
    'dailyBudget': amz_daily_budget,
}

update_profile(update_data_dict)
```

(continues on next page)

(continued from previous page)

```
# The most simple and useful way to update the budget of a bunch of profiles

update_profiles_list = \
[
    {
        'profileId': 1495806522428699,
        'dailyBudget': 10.5,
    },
    {
        'profileId': 1495806522428698,
        'dailyBudget': 20,
    }
]

update_profile(update_profiles_list)
```

get_profile(*self*, *profileId*, ***kwargs*) → ApiResponse

Gets a profile specified by identifier.

path **profileId**:*number* | Required. The identifier of an existing profile Id.

Returns:

ApiResponse

Example getting a profiles by profileId

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException

def get_profile(profile_id: int):

    logging.info("---------------------------------------")
    logging.info("Profiles > get_profile(%s)" % profile_id)
    logging.info("---------------------------------------")

    try:

        result = Profiles(account=store, debug=True).get_profile(
            profileId=profile_id
        )
        logging.info(result)

        profile_info = result.payload

    except AdvertisingApiException as error:
        logging.info(error)


amz_profile_id = 1495806522428699
get_profile(amz_profile_id)
```

> **Warning:** Note that this **register_brand_assistant** operation is only used for SANDBOX test environment.

register_brand_assistant(*country_code: str*, *brand: str*) → ApiResponse

SANDBOX ONLY - Create a vendor profile for sandbox.

'**country_code**': *string*, {'description': 'The countryCode for a given country'}
'**brand**': *string*, {'description': 'The brand for the vendor account'}

Returns:

ApiResponse

Example creating a Vendor profile for sandbox

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException

def register_brand_assistant(country_code: str, brand: str):

    logging.info("---------------------------------------")
    logging.info("Profiles > register_brand_assistant({},{})".format(country_code,␣
↪brand))
    logging.info("---------------------------------------")

    try:

        result = Profiles(account=store, debug=True).register_brand_assistant(
            country_code=country_code,
            brand=brand
        )
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

amz_country_code = "UK"
vendor_brand = "LEADTECH"

register_brand_assistant(amz_country_code, vendor_brand)
```

> **Warning:** Note that this **register_brand** operation is only used for SANDBOX test environment.

register_brand(*body: dict*) → ApiResponse

SANDBOX ONLY - Create a vendor profile for sandbox.

body: | REQUIRED

{
'**countryCode**': *string*, {'description': 'The countryCode for a given country'}
'**brand**': *string*, {'description': 'The brand for the vendor account'}
}

Returns:

ApiResponse

Example creating a Vendor profile for sandbox with a regular dict

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException

def register_brand(dictionary: dict):

logging.info("-------------------------------------")
logging.info("Profiles > register_brand(%s)" % str(dictionary))
logging.info("-------------------------------------")

try:

    result = Profiles(account=store, debug=True).register_brand(
        body=dictionary
    )
    logging.info(result)

except AdvertisingApiException as error:
    logging.info(error)

amz_dict_brand = \
{
    'countryCode': amz_country_code,
    'brand': vendor_brand
}

register_brand(amz_dict_brand)
```

Warning: Note that this **register_assistant** is only used for SANDBOX test environment. Using in PRODUCTION will get a AdvertisingApiException: {'status_code': 404, 'code': 'NOT_FOUND', 'details': 'HTTP 404 Not Found', 'requestId': 'HEMG4AR2NRF5HCW8BH8S'}

Note: You only can create one profile per country, If you try to create again the same country will get a Payload: {'profileId': 81562378459925, 'status': 'SUCCESS', 'statusDetails': 'Merchant is already registered'}

register_assistant(*country_code: str*) → ApiResponse

SANDBOX ONLY - Create a seller profile for sandbox.

'**countryCode**': *string*, {'description': 'The countryCode for a given country: [US, CA, MX, UK, DE, FR, ES, IT, NL, JP, AU, AE, SE, PL, TR]'}

Returns:

ApiResponse

Example creating a profiles por a specific country

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException

def register_assistant(value: str):

    logging.info("-------------------------------------")
    logging.info("Profiles > register_assistant(%s)" % value)
    logging.info("-------------------------------------")

    try:

        result = Profiles(account=store, debug=True).register_assistant(
            country_code=value
        )
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

amz_country_code = "DE"
register_assistant(amz_country_code)
```

Payload

```
{
    'registerProfileId': '82465ad1-e8cd-4d24-beb0-5cb8423f4260DE',
    'status': 'IN_PROGRESS',
    'statusDetails': 'Registration workflow has been started'
}
```

Warning: Note that this **register** operation is only used for SANDBOX test environment.

register(*body: dict, str*) → ApiResponse

SANDBOX ONLY - Create a seller profile for sandbox.

body: | REQUIRED

{
'**countryCode**': *string*, {'description': 'The countryCode for a given country [US, CA, MX, UK, DE, FR, ES, IT, NL, JP, AU, AE, SE, PL, TR]'}
}

Example creating a profiles por a specific country with a dict

```
import logging
from ad_api.api import Profiles
from ad_api.base import AdvertisingApiException

def register(dictionary: dict):

    logging.info("-------------------------------------")
```

(continues on next page)

(continued from previous page)

```
    logging.info("Profiles > register(%s)" % str(dictionary))
    logging.info("------------------------------------")

    try:

        result = Profiles(account=store, debug=True).register(
            body=dictionary
        )
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

    amz_country_code = "DE"
    amz_dict_country_code = \
        {
            "countryCode": amz_country_code
        }

    register(amz_dict_country_code)
```

3.2 Manager Account

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/ManagerAccount_prod_3p.json

A Manager Account lets you manage a group of Amazon Advertising accounts.

class ad_api.api.**ManagerAccounts**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_manager_accounts() → ApiResponse

Returns all Manager accounts that a given Amazon Advertising user has access to.

Example getting a list of manager accounts

```
import logging
from ad_api.api import ManagerAccounts
from ad_api.base import AdvertisingApiException

def list_manager_accounts():

    try:
        result = ManagerAccounts(account="bestq", debug=True).list_manager_
↪accounts()
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':
```

(continues on next page)

(continued from previous page)

```
    list_manager_accounts()
```

create_manager_account(*body: dict, str*) → ApiResponse

Creates a new Amazon Advertising Manager account.

body: | REQUIRED

{

'**managerAccountName**': *string* Name of the Manager account.

'**managerAccountType**': *string* Type of the Manager account, which indicates how the Manager account will be used. Use Advertiser if the Manager account will be used for your own products and services, or Agency if you are managing accounts on behalf of your clients. Enum: [Advertiser, Agency]

}

Example creating a manager account

```
import logging
from ad_api.api import ManagerAccounts
from ad_api.base import AdvertisingApiException


def create_manager_account(data: dict or str):
    try:
        result = ManagerAccounts(debug=True).create_manager_account(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    manager_account_name = 'ManagerMyAccount'
    manager_account_type = 'Agency'  # Agency Advertiser

    account = \
        {
            'managerAccountName': manager_account_name,
            'managerAccountType': manager_account_type,
        }

    create_manager_account(account)
```

associate_manager_accounts(*managerAccountId: str, body: dict, str*) → ApiResponse

Link Amazon Advertising accounts or advertisers with a Manager Account.

path **managerAccountId**:string | Required. Id of the Manager Account.

body: | REQUIRED

{
"accounts": A list of Advertising accounts or advertisers to link/unlink with Manager Account. User can pass a list with a maximum of 20 accounts/advertisers using any mix of identifiers.
[
{
"roles": "list", The types of role that will exist with the Amazon Advertising account. Depending on account type, the default role will be ENTITY_USER or SELLER_USER. Only one role at a time is currently supported [ENTITY_OWNER, ENTITY_USER, ENTITY_VIEWER, SELLER_USER]
"id": "string", Id of the Amazon Advertising account.
"type": The type of the Id, Enum: [ACCOUNT_ID, DSP_ADVERTISER_ID]
}
]
}

Example associating a manager account with and advertiser account

```
import logging
from ad_api.api import ManagerAccounts
from ad_api.base import AdvertisingApiException

def associate_manager_accounts(manager_account_id: str, data: dict or str):
    try:
        result = ManagerAccounts(debug=True).associate_manager_accounts(
            managerAccountId=manager_account_id,
            body=data,
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    amz_manager_account_id = 'amzn1.ads1.ma1.ewpohpn123456789987654321'
    account_id = 'ENTITY1123456789012'
    roles = ['ENTITY_VIEWER']
    type_account = 'ACCOUNT_ID'

    association = \
        {
            'accounts':
                [
                    {
                        'roles': roles,
                        'id': account_id,
                        'type': type_account
                    }
                ]
```

(continues on next page)

(continued from previous page)

```
        }

    associate_manager_accounts(amz_manager_account_id, association)
```

disassociate_manager_accounts(*managerAccountId: str*, *body: dict, str*) → ApiResponse

Unlink Amazon Advertising accounts or advertisers with a Manager Account.

path **managerAccountId**:string | Required. Id of the Manager Account.

body: | REQUIRED

{
"accounts": A list of Advertising accounts or advertisers to link/unlink with Manager Account. User can pass a list with a maximum of 20 accounts/advertisers using any mix of identifiers.
[
{
"roles": "list", The types of role that will exist with the Amazon Advertising account. Depending on account type, the default role will be ENTITY_USER or SELLER_USER. Only one role at a time is currently supported [ENTITY_OWNER, ENTITY_USER, ENTITY_VIEWER, SELLER_USER]
"id": "string", Id of the Amazon Advertising account.
"type": The type of the Id, Enum: [ACCOUNT_ID, DSP_ADVERTISER_ID]
}
]
}

Example unlinking a manager account with and advertiser account

```
import logging
from ad_api.api import ManagerAccounts
from ad_api.base import AdvertisingApiException

def disassociate_manager_accounts(manager_account_id: str, data: dict or str):
    try:
        result = ManagerAccounts(debug=True).disassociate_manager_accounts(
            managerAccountId=manager_account_id,
            body=data,
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    amz_manager_account_id = 'amzn1.ads1.ma1.ewpohpn123456789987654321'
    account_id = 'ENTITY1123456789012'
    type_account = 'ACCOUNT_ID'
```

(continues on next page)

(continued from previous page)

```
    disassociation = \
        {
            'accounts':
                [
                    {
                        'id': account_id,
                        'type': type_account
                    }
                ]
        }

    disassociate_manager_accounts(amz_manager_account_id, disassociation)
```

3.3 Invoices

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/Billing_prod_3p.json

Get invoice data by invoice ID

class ad_api.api.**Invoices**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_invoices(***kwargs*) → ApiResponse

Get invoices for advertiser. Requires one of these permissions: ["nemo_transactions_view","nemo_transactions_edit"]

query **invoiceStatuses**:*string* | Optional. Available values : ISSUED, PAID_IN_PART, PAID_IN_FULL, WRITTEN_OFF. (Not documented: ACCUMULATING)

query **count**:*string* | Optional. Number of records to include in the paged response. Defaults to 100. Cannot be combined with the cursor parameter.

query **cursor**:*string* | Optional. A cursor representing how far into a result set this query should begin. In the absence of a cursor the request will default to start index of 0 and page size of 100.

Returns:

ApiResponse

Example getting a list of invoices

```
import logging
from ad_api.api import Invoices
from ad_api.base import AdvertisingApiException

def list_invoices(**kwargs):

    try:

        result = Invoices(account=store, debug=True).list_invoices(
            **kwargs
```

(continues on next page)

(continued from previous page)

```
        )
        res = result.payload
        payload = res.get("payload")
        invoice_summaries = payload.get("invoiceSummaries")
        for invoice in invoice_summaries:
            logging.info(invoice)
    except AdvertisingApiException as error:
        logging.info(error)

# list_invoices()
list_invoices(invoiceStatuses="PAID_IN_FULL", count=5)
```

Note: Here is an example how to get all the invoices full payed using a decorator in Utils @Utils.load_all_pages

```
import logging
from ad_api.api import Invoices
from ad_api.base import Utils

@Utils.load_all_pages(throttle_by_seconds=1, next_token_param="cursor")
def get_all_invoices(**kwargs):
    return Invoices(account=store, debug=True).list_invoices(**kwargs)

bill_status = 'PAID_IN_FULL'

for page in get_all_invoices(invoiceStatuses=bill_status):
    res = page.payload
    payload = res.get("payload")
    invoice_summaries = payload.get("invoiceSummaries")
    for invoice in invoice_summaries:
        logging.info(invoice)
```

Will Output the invoices

```
{'id': 'DR0012TTY-75', 'status': 'PAID_IN_FULL', 'fromDate': '20210613', 'toDate':
↪'20210625', 'invoiceDate': '20210624', 'amountDue': {'amount': 500.04,
↪'currencyCode': 'EUR'}, 'taxAmountDue': {'amount': 0.0, 'currencyCode': 'EUR'},
↪'remainingAmountDue': {'amount': 0.0, 'currencyCode': 'EUR'},
↪'remainingTaxAmountDue': {'amount': 0.0, 'currencyCode': 'EUR'}}
```

```
{'id': 'DR0012TTY-76', 'status': 'PAID_IN_FULL', 'fromDate': '20210624', 'toDate':
↪'20210703', 'invoiceDate': '20210702', 'amountDue': {'amount': 332.2,
↪'currencyCode': 'EUR'}, 'taxAmountDue': {'amount': 0.0, 'currencyCode': 'EUR'},
↪'remainingAmountDue': {'amount': 0.0, 'currencyCode': 'EUR'},
↪'remainingTaxAmountDue': {'amount': 0.0, 'currencyCode': 'EUR'}}
```

get_invoice(*invoiceId: str*) → ApiResponse

Get invoice data by invoice ID. Requires one of these permissions: ["nemo_transactions_view","nemo_transactions_edit"]

path **invoiceId**:*string* | required. ID of invoice to fetch

Returns:

ApiResponse

Note: You could get a specific invoice by invoiceId which is the string provided

```
import logging
from ad_api.api import Invoices
from ad_api.base import AdvertisingApiException

def get_invoice(invoice_id: str):

    try:

        result = Invoices(account=store, debug=True).get_invoice(
            invoiceId=invoice_id
        )
        logging.info(result)
    except AdvertisingApiException as error:
        logging.info(error)

amz_invoice_id = 'DR0012TTY-76'
get_invoice(amz_invoice_id)
```

Warning: This API cannot be used in sandbox mode and will return AdvertisingApiException

```
{
    'status_code': 404,
    'code': 'NOT_FOUND',
    'details': 'Could not find resource for full path: https://advertising-api-test.
↪amazon.com/invoices/DR0012TTY-76',
    'requestId': '1A83K31WYACG6YFG0S17'
}
```

3.4 Billing

class ad_api.api.**Billing**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_billing_notifications(***kwargs*)

Get the billing notifications for a list advertising accounts.

Request Body

Bulk Get Billing Notifications Request Body {
The properties needed to get the billing notifications for a set of advertisers.
advertiserMarketplaces* (array) minItems: 0 maxItems: 100 [
advertiserMarketplace {
marketplaceId* (string)

advertiserId* (string)
}
]
locale (string) Locale. Enum: [‘ar_AE’, ‘cs_CZ’, ‘de_DE’, ‘en_AU’, ‘en_CA’, ‘en_GB’, ‘en_IN’, ‘en_SG’, ‘es_ES’, ‘es_MX’, ‘fr_CA’, ‘fr_FR’, ‘he_IL’, ‘hi_IN’, ‘it_IT’, ‘ja_JP’, ‘ko_KR’, ‘nl_NL’, ‘pl_PL’, ‘pt_BR’, ‘sv_SE’, ‘ta_IN’, ‘tr_TR’, ‘zh_CN’, ‘zh_TW’]
}

Returns
ApiResponse

`list_billing_status`(***kwargs*)

Get the billing status for a list of advertising accounts.

Request Body

Bulk Get Billing Statuses Request Body {
The properties needed to get the billing statuses for a set of advertisers.
advertiserMarketplaces* (array) minItems: 0 maxItems: 100 [
advertiserMarketplace {
marketplaceId* (string)
advertiserId* (string)
}
]
locale (string) Locale. Enum: [‘ar_AE’, ‘cs_CZ’, ‘de_DE’, ‘en_AU’, ‘en_CA’, ‘en_GB’, ‘en_IN’, ‘en_SG’, ‘es_ES’, ‘es_MX’, ‘fr_CA’, ‘fr_FR’, ‘he_IL’, ‘hi_IN’, ‘it_IT’, ‘ja_JP’, ‘ko_KR’, ‘nl_NL’, ‘pl_PL’, ‘pt_BR’, ‘sv_SE’, ‘ta_IN’, ‘tr_TR’, ‘zh_CN’, ‘zh_TW’]
}

Returns
ApiResponse

3.5 Product Selector

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/ProductSelector_prod_3p.json

The Amazon Product Selector API allows integrators to receive product metadata such as inventory status, price, eligibility status and product details for SKUS or ASINs in their Product Catalog in order to launch, manage or optimize Sponsored Product, Sponsored Brands or Sponsored Display advertising campaigns. The Product Selector API is available to Sellers, Vendors, and Authors.

`class ad_api.api.Metadata`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

`get_products_metadata`(*body: dict, str*) → ApiResponse

Returns product metadata for the advertiser.

body: | REQUIRED

‘**asins**’: *list>string*, {‘description’: ‘Specific asins to search for in the advertiser’s inventory. Cannot use together with skus or searchStr input types.’}

'**checkItemDetails**': *boolean*, {'description': 'Whether item details such as name, image, and price is required. default: false'}

'**cursorToken**': *string*, {'description': 'Pagination token used for the suggested sort type'}

'**adType**': *string*, {'description': 'Program type. Required if checks advertising eligibility. Enum: [SP, SB, SD]'}

'**skus**': *list>string*, {'description': 'Specific skus to search for in the advertiser's inventory. Currently only support SP program type for sellers. Cannot use together with asins or searchStr input types'}

'**checkEligibility**': *boolean*, {'description': 'Whether advertising eligibility info is required. default: false'}

'**searchStr**': *string*, {'description': 'Specific string in the item title to search for in the advertiser's inventory. Case insensitive. Cannot use together with asins or skus input types'}

'**pageIndex**': *integer($int32)*, {'description*': 'Index of the page to be returned'}

'**sortOrder**': *string*, {'description': 'Sort order (has to be DESC for the suggested sort type). default: DESC. Enum [ASC, DESC]'}

'**pageSize**': *integer($int32)*, {'description*': 'Number of items to be returned on this page index (max 100 for author)'}

'**sortBy**': *string*, {'description': 'Sort option for the result. Currently only support SP program type for sellers. Enum [SUGGESTED, CREATED_DATE]'}

Returns:

ApiResponse

Example getting the metadata of a search string

```
import logging
from ad_api.api import Metadata
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)


def get_products_metadata(data: (str, dict)):

    try:

        result = Metadata(debug=True).get_products_metadata(
            body=data
        )

        logging.info(result)

        product_metadata_list = result.payload.get("ProductMetadataList")

        logging.info(len(product_metadata_list))
```

(continues on next page)

(continued from previous page)

```
        for product_metadata in product_metadata_list:
            logging.info(product_metadata)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    search_dict = \
        {
            'checkItemDetails': True,
            'adType': 'SP',
            'checkEligibility': True,
            'searchStr': 'obd2',
            'pageIndex': 1,
            'pageSize': 20,
            'sortBy': 'CREATED_DATE',
            'locale': 'es_ES'
        }

    get_products_metadata(search_dict)
```

3.6 Eligibility

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/Eligibility_prod_3p.json

Check advertising eligibility of products.

class ad_api.api.**Eligibility**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

get_eligibility_assistant(*asin_list: list*, *sku_list: list = None*, *ad_type: str = 'sp'*, *locale: str = 'en-GB'*) → ApiResponse

Gets a list of advertising eligibility objects for a set of products. Requests are permitted only for products sold by the merchant associated with the profile. Note that the request object is a list of ASINs, but multiple SKUs are returned if there is more than one SKU associated with an ASIN. If a product is not eligible for advertising, the response includes an object describing the reasons for ineligibility.

asin_list: 'A list of ASIN: An Amazon product identifier.'

sku_list: 'A list of SKU: A seller product identifier'

adType: *string*, {'description': 'Set to 'sp' to check product eligibility for Sponsored Products advertisements. Set to 'sb' to check product eligibility for Sponsored Brands advertisements. default: sp. [sp, sb]'}

locale': *string*, {'description': 'Set to the locale string in the table below to specify the language in which the response is returned. default: en_GB'}

Returns:

ApiResponse

Example getting the elegibility of asin and asin/sku

```
import logging
from ad_api.api import Eligibility
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)


def get_eligibility_assistant(**kwargs):

    try:

        result = Eligibility(debug=True).get_eligibility_assistant(
            **kwargs
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)



if __name__ == '__main__':

    d_asin_list = ['B08C1KN5J2', 'B08XM9C8P6']
    d_sku_list = ['SKU-OF-B08C1KN5J2', 'SKU-OF-B08XM9C8P6']
    type_ad = 'sb'
    lang = 'es_ES'

    get_eligibility_assistant(asin_list=d_asin_list, sku_list=d_sku_list, ad_
↪type=type_ad, locale=lang)

    # only by ASIN will return all available sku and show the eligibility
    get_eligibility_assistant(asin_list=d_asin_list, locale=lang)
    # the minimal call must send at least a list with one ASIN. Default: (adType="sp
↪", locale="en-GB")
    get_eligibility_assistant(asin_list=['B08C1KN5J2'])
```

get_eligibility(*body: dict, str*) → ApiResponse

Gets a list of advertising eligibility objects for a set of products. Requests are permitted only for products sold by the merchant associated with the profile. Note that the request object is a list of ASINs, but multiple SKUs are returned if there is more than one SKU associated with an ASIN. If a product is not eligible for advertising, the response includes an object describing the reasons for ineligibility.

body: | REQUIRED

'**adType**': *string*, {'description': 'Set to 'sp' to check product eligibility for Sponsored Products advertisements. Set to 'sb' to check product eligibility for Sponsored Brands advertisements. default: sp. [sp, sb]'}

'**productDetailsList**': *dict*, {'asin*': 'An Amazon product identifier.', 'sku': 'A seller product identifier'}

'**locale**': *string*, {'description': 'Set to the locale string in the table below to specify the language in which the response is returned.'}

Returns:

ApiResponse

Example getting the elegibility in the regular way sending a dict

```
import logging
from ad_api.api import Eligibility
from ad_api.base import AdvertisingApiException

def get_eligibility(data: (str, dict)):

    try:

        result = Eligibility(debug=True).get_eligibility(
            body=data
        )

        logging.info(result)


    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    dict_asin = \
        {
            'adType': 'sp',
            'productDetailsList': [
                {
                    'asin': 'B08C1KN5J2'
                }
            ],
            'locale': 'es-ES'
        }

    get_eligibility(dict_asin)

    dict_asin_sku = \
        {
            'adType': 'sp',
            'productDetailsList': [
                {
                    'asin': 'B08C1KN5J2',
                    'sku': 'SKU-OF-B08C1KN5J2'
```

(continues on next page)

(continued from previous page)

```
                }
            ],
            'locale': 'es-ES'
        }

    get_eligibility(dict_asin_sku)

    dict_multiple_asin_sku = \
        {
            'adType': 'sp',
            'productDetailsList': [
                {
                    'asin': 'B08C1KN5J2',
                    'sku': 'SKU-OF-B08C1KN5J2'
                },
                {
                    'asin': 'B08XM9C8P6',
                    'sku': 'SKU-OF-B08XM9C8P6'
                }

            ],
            'locale': 'es-ES'
        }

    get_eligibility(dict_multiple_asin_sku)
```

3.7 Change History open beta

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/Eligibility_prod_3p.json

Provides information about changes made to campaigns, adgroups, ads, etc

class ad_api.api.**History**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

get_history(*body: dict, str, file*) → ApiResponse

Returns history of changes for provided event sources that match the filters and time ranges specified. Only events that belong to the authenticated Advertiser can be queried. All times will be in UTC Epoch format. This API accepts identifiers in either the alphamumeric format (default), or the numeric format. If numeric IDs are supplied, then numeric IDs will be returned otherwise, alphanumeric IDs are returned.

body: | REQUIRED {'description': 'A HistoryQuery}'

from_date | int | Max 90 days of history.

to_date | int |

event_types | HistoryEventType

next_token | str | token from previous response to get next set of data. | [optional]

page_offset | int | Mutually exclusive with 'nextToken'. Max results with pageOffset is 10000. Use nextToken instead for more results. | [optional]

count | int | Requested number of results. Default 100. Minimum 50. Maximum 200. | [optional]

sort | HistorySortParameter

any string name | bool, date, datetime, dict, float, int, list, str, none_type | any string name can be used but the value must be the correct type | [optional]

Example python

```
import logging
from ad_api.api import History
from ad_api.base import AdvertisingApiException

def get_history(data: (str, dict)):

    try:

        result = History(debug=True).get_history(
            body=json.dumps(data)
        )
        payload = result.payload
        events = payload.get("events")
        for event in events:
            logging.info(event)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':


    # fromDate cannot be more than 90 days ago
    from_date = datetime(2022, 2, 1, 0, 0, 0).strftime('%s%f')[:-3]
    to_date = datetime.now().strftime('%s%f')[:-3]

    request = \
        {
            "fromDate": int(from_date),
            "toDate": int(to_date),
            "eventTypes": {
                "CAMPAIGN": {
                    "filters": [
                        "BUDGET_AMOUNT",
                        "STATUS"
                    ],
                    "eventTypeIds": [
                        "45662011530311"
                    ]
                }
            }
        }

    get_history(request)
```

Example query.json

Download `json` the file to use:

```
{
  "fromDate": 1626739199000,
  "toDate": 1632095999000,
  "pageOffset": 10,
  "count": 60,
  "sort": {
    "key": "DATE",
    "direction": "DESC"
  },
  "eventTypes": {
    "CAMPAIGN": {
      "filters": [
        "BUDGET_AMOUNT"
      ],
      "eventTypeIds": [
        "137359064782313"
      ]
    }
  }
}
```

3.8 Creative Asset Library beta

https://d3a0d0y2hgofx6.cloudfront.net/openapi/en-us/creative-asset-library/creative-asset-library-openapi.yaml

Advertisers can use creative assets to store, organize and reuse brand content, such as logos, images, etc. Stored content can be used for Amazon Ads and on Amazon shopping pages. Creative assets enables brands to provide a consistent shopping experience by easily applying brand content across Amazon.

class ad_api.api.**CreativeAssets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

search_assets(*body: dict, str, file*) → ApiResponse

Search the creative asset library.

body: | REQUIRED {'description': 'A caSearchRequestCommon}'

text | string | The text used for searching assets, it matches asset name, asset name prefix, tags and ASINs associated with the assets

filterCriteria Optional this is used to filter results, we support two types of filters, valueFilter and rangeFilter

valueFilters | **list** | Filter for certain values of asset attributes

values | list |

valueField | string | [TAG, ASIN, CAMPAIGN_NAME, CAMPAIGN_ID, PROGRAM, ASSET_TYPE, ASSET_SUB_TYPE, APPROVED_AD_POLICY, ASSET_EXTENSION]

rangeFilters

range | list | Filter assets which have certain ranges of asset attributes. For example, filter assets which have file size in the range of [10,20] or [40,50].

start | string |

end | string |

sortCriteria Optional this is used to get sorted results

field | **string** | Enum [CREATED_TIME, SIZE, NAME, IMAGE_HEIGHT, IMAGE_WIDTH, EXTENSION]

order | **string** | Enum [ASC, DESC]

pageCriteria Optional this is used for pagination when searching for the first page, no need to put anything, otherwise, use the token returned from previous search call

identifier | **dict** |

pageNumber | int |

token | string |

size | **int** | default: 25 minimum: 1 maximum: 500

Example python

```
import logging
from ad_api.api import CreativeAssets
from ad_api.base import AdvertisingApiException

def search_assets(data: dict or str):
    try:
        result = CreativeAssets(debug=True).search_assets(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':
```

(continues on next page)

(continued from previous page)

```
    search = \
        {
            "text": "Logo",
            "filterCriteria": {
                "valueFilters": [
                    {
                        "values": [
                            "B08XW4FDJV"
                        ],
                        "valueField": "ASIN"
                    }
                ]
            },
            "sortCriteria": {
                "field": "CREATED_TIME",
                "order": "ASC"
            }
        }

# If you send a empty query search = {} it will return all the assets in library
search_assets(search)
```

Example search.json

Download `json` template

```
{
  "text": "string",
  "filterCriteria": {
    "valueFilters": [
      {
        "values": [
          "string"
        ],
        "valueField": "TAG"
      }
    ],
    "rangeFilters": [
      {
        "range": [
          {
            "start": "string",
            "end": "string"
          }
        ]
      }
    ]
  },
  "sortCriteria": {
    "field": "CREATED_TIME",
    "order": "ASC"
  },
```

(continues on next page)

(continued from previous page)

```
  "pageCriteria": {
    "identifier": {
      "pageNumber": 0,
      "token": "string"
    },
    "size": 0
  }
}
```

get_asset(*assetId: str*, *version: str*) → ApiResponse

Retrieves an asset along with the metadata

query **assetId**:*string* | Required. The assetId

query **version**:*string* | Optional. The versionId of the asset, if not included all versions will return.

Example python

```
import logging
from ad_api.api import CreativeAssets
from ad_api.base import AdvertisingApiException


def get_asset(**kwargs):
    try:
        result = CreativeAssets(debug=True).get_asset(
            **kwargs
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    amz_asset_id = "amzn1.assetlibrary.asset1.c2867a8671670fc7a5d0bf1efa295d599"
    amz_version = "version_v3"
    # get a specific version of the asset
    get_asset(assetId=amz_asset_id, version=amz_version)
    # get all versions of the asset
    get_asset(assetId=amz_asset_id)
```

upload_asset(*body: dict, str, file*) → ApiResponse

body: | REQUIRED

{

‘**fileName**’: *string* The fileName of the asset. pattern: [w]+.jpg|png|mp4|mov|wmv|avi

}

Example python

```
import logging
from ad_api.api import CreativeAssets
from ad_api.base import AdvertisingApiException


def upload_asset(data: dict or str):
    try:
        result = CreativeAssets(debug=True).upload_asset(
            body=data
        )

        url = result.payload.get("url")
        logging.info(url)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    file_name = "Sample-1000x1000.jpeg"

    asset = \
        {
            "fileName": file_name
        }

    upload_asset(asset)
```

Warning: After upload the assets it will return a url is the url location to which you will be uploading your asset

---

Note: This is not part of the Creative Assets api is just an example to upload the file using requests in python if your response.status_code is 200 everything is fine

---

```
from requests import request


def upload_file(_method, _url, _img):

    response = request(
        _method,
        _url,
        data=open(_img, 'rb')
    )

    logging.info(response.status_code)
    logging.info(response.headers)
    logging.info(response.content)
    logging.info(response.raw)
```

(continues on next page)

(continued from previous page)

```
if __name__ == '__main__':

    file_name = "Sample-1200x1200.jpeg"
    url = "https://al-eu-726f4d26-7fdb.s3-accelerate.amazonaws.com/037764c3-6b70-
↪4da9-b9af-8ef6ed136def.jpeg?x-amz-meta-filename=Sample-1200x1200.jpeg&X-Amz-
↪Security-Token=token&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20220421T035013Z&
↪X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=credential%2Fus-east-
↪1%2Fs3%2Faws4_request&X-Amz-
↪Signature=a9cbeb0bd3d31e0c1a71c4ad27be23cd7725438581cc4e8a1f9f3f8d"
    method = "PUT"
    upload_file(method, url, file_name)
```

register_asset(*body: dict, str, file*) → ApiResponse

Registers an uploaded asset with the creative assets library with optional contextual and tagging information. The API should be called once the asset is uploaded to the location provided by the /asset/upload API endpoint.

body: | REQUIRED {'description': 'A caSearchRequestCommon}'

url | string | Required The url to upload the asset. The url expires in 15 minutes.

name | string | Required The name to be given to the asset being registered.

asinList | list | Optional Tagging assets with ASIN, promotes asset discoverability downstream. If ASIN is provided at the time of upload/during asset registration, it is applied as a tag on that asset. This allows for that asset to be searchable using that ASIN#. For e.g., An advertiser may want to search for assets tagged with ASIN BC10001, so they can create a store spotlight ad with product images for that ASIN.

assetType | string | The asset type you are registering [IMAGE]

assetSubTypeList | list | For assetType IMAGE acceptable assetSubTypes are LOGO, PRODUCT_IMAGE, AUTHOR_IMAGE, LIFESTYLE_IMAGE, OTHER_IMAGE.

versionInfo | string | The asset type you are registering [IMAGE]

linkedAssetId | **string |** The registering asset will be created as a new version of this linkedAssetId.

versionNotes | **string |** The version notes that client can associate to the asset.Versioning enables users to update an old asset, so that you can ensure the latest asset is being used. You can upload a new version of an existing asset along with version notes. Any tags/ASINs from previous version, will be retained on the new version too.

tags | list |

registrationContext | caRegistrationContext | This is used on registration of an asset, to associate DSP assets to a specific advertiser. This is required for assets being uploaded for use in DSP. | associatedPrograms | **caAssociatedProgramcaAssociatedProgram |**

metadata | string | Include key-value pairs related to the asset. For DSP use "dspAdvertiserId" = "ID". Include program as AMAZON_DSP.

programName | string | Use this field to specify which program you are uploading

an asset for. Currently, the accepted value here on registration is to associate an asset with a DSP advertiser. [AMAZON_DSP]

associatedSubEntityList | caAssociatedSubEntityList | This field is required for sellers, but not required for vendors. The brandEntityId is required for sellers uploading assets for use in Sponsored Brands. As a best practice, ensure to include brandEntityId when uploading assets for sellers.

caAssociatedSubEntity |

brandEntityId | string | The entity id of brand, which can be retrieved using GET /brands.

skipAssetSubTypesDetection | boolean | Select true if you want to set an asset to a specific assetSubType, if this is not included the system may reclassify your asset based on specifications.

Example python

```
import logging
from ad_api.api import CreativeAssets
from ad_api.base import AdvertisingApiException


def register_asset(data: dict or str):
    try:
        result = CreativeAssets(debug=True).register_asset(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    # the url was obtained with upload_asset method and was used to upload the file␣
↪and finally register
    url = "https://al-eu-726f4d26-7fdb.s3-accelerate.amazonaws.com/037764c3-6b70-
↪4da9-b9af-8ef6ed136def.jpeg?x-amz-meta-filename=Sample-1200x1200.jpeg&X-Amz-
↪Security-Token=token&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20220421T035013Z&
↪X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=credential%2Fus-east-
↪1%2Fs3%2Faws4_request&X-Amz-
↪Signature=a9cbeb0bd3d31e0c1a71c4ad27be23cd7725438581cc4e8a1f9f3f8d"

    # register and assets already uploaded as a new asset
    register = \
        {
```

(continues on next page)

(continued from previous page)

```
            "url": url,
            "name": "PRODUCT-IMAGE-NAME",
            "assetType": "IMAGE",
            "assetSubTypeList": [
                "PRODUCT_IMAGE"
            ],
            "associatedSubEntityList": [
                {
                    "brandEntityId": "ENTITY288756GCCQ6CF"
                }
            ],
            "skipAssetSubTypesDetection": True
        }


    # register and assets already uploaded as a new version of an existing assets

    register = \
        {
            "url": url,
            "name": "PRODUCT-IMAGE-NAME-2",
            "assetType": "IMAGE",
            "assetSubTypeList": [
                "PRODUCT_IMAGE"
            ],
            "versionInfo": {
                "linkedAssetId": "amzn1.assetlibrary.asset1.
↪c2867a8671670fc7a5d0bf1efa295d599",
                "versionNotes": "version v2 of an existing asset"
            },
            "associatedSubEntityList": [
                {
                    "brandEntityId": "ENTITY1234567890123"
                }
            ],
            "skipAssetSubTypesDetection": True
        }


    register_asset(register)
```

3.9 Localization

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/Localization_prod_3p.json.

This API provides operations to localize data used when creating advertising campaigns. Depending on the type of data, localization may entail translating text, converting monetary amounts, or mapping an entity in a source marketplace to an analogous entity in one or more target marketplaces.

class ad_api.api.**Localization**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

get_currency_extended(*body: dict, str*, *version: int = 1*) → ApiResponse

Gets an array of localized currencies extended with currency code and the country_code for better understanding in their target marketplaces, with the advertiser ID and source marketplace ID passed in through the header and body

'**version**': *int* [optional] Will use content body "application/vnd.currencylocalization.v"+str(version)+"+json"

body: | REQUIRED | { | '**localizeCurrencyRequests**': *list* | [| **'LocalizationCurrencyRequest'**: *dict* | { | **'currency'**: LocalizationCurrency: *dict* | { | **"amount"**: *int* | } | } |] | } | '**targetCountryCodes**': *list*, A list of two-letter country codes. When both marketplaceId and countryCode are present, countryCode is ignored. Please refer to the table above for a list of supported country codes. | '**sourceCountryCode**': *string* | '**sourceMarketplaceId**': *string* | '**targetMarketplaces**': *list* }

Warning: This method **get_currency_extended** is a helper that will extend the response and return country codes abd currency to contextualize better the results

Note: If you do not provide any filter or the filters are wide it will return tons the audiences

Example creating a dictionary and using MarketplacesIds Enum to simplify the task

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException, MarketplacesIds

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_currency_extended(data: (str, dict), version: int = 1):
    try:

        result = Localization(debug=True).get_currency_extended(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    source_country_code = "DE"
    source_marketplace_id = MarketplacesIds[source_country_code].value
    target_country_codes = ["GB", "US", "JP"]
    amount_1 = 10
    amount_2 = 15
```

(continues on next page)

(continued from previous page)

```
    request = \
    {
        'localizeCurrencyRequests': [
            {
                'currency': {
                    'amount': amount_1
                }
            },
            {
                'currency': {
                    'amount': amount_2
                }
            }

        ],
        'targetCountryCodes': target_country_codes,
        'sourceCountryCode': source_country_code,
        'sourceMarketplaceId': source_marketplace_id,
        'targetMarketplaces': [MarketplacesIds[target_country_code].value for␣
↪target_country_code in target_country_codes]
    }

    get_currency_extended(request, version=1) # You could submit version=2
```

Static .json file in case want to use as example

```
{
    "localizeCurrencyRequests": [
        {"currency": {"amount": 10}},
        {"currency": {"amount": 15}}
    ],
    "targetCountryCodes": ["GB", "US", "JP"],
    "sourceCountryCode": "DE",
    "sourceMarketplaceId": "A1PA6795UKMFR9",
    "targetMarketplaces": ["A1F83G8C2ARO7P", "ATVPDKIKX0DER", "A1VC38T7YXB528"]
}
```

Download `json` the file to use:

Example using the above file as request (MarketplacesIds not needed)

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_currency_extended(data: (str, dict), version: int = 1):
try:
```

(continues on next page)

(continued from previous page)

```
    result = Localization(debug=True).get_currency_extended(
        version=version,
        body=data
    )

    logging.info(result)

except AdvertisingApiException as error:
    logging.info(error)

if __name__ == '__main__':

    file_name = "../test/localizations/request_currency.json"
    get_currency_extended(file_name) # No version include will get version 1
    # get_currency_extended(file_name, version=2)
```

Result using version=1 application/vnd.currencylocalization.v1+json


```
{
    "localizedCurrencyResponses": [
        {
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {
                    "amount": 8.29,
                    "currency": "GBP",
                    "country_code": "UK"
                },
                "ATVPDKIKX0DER": {
                    "amount": 10.83,
                    "currency": "USD",
                    "country_code": "US"
                },
                "A1VC38T7YXB528": {
                    "amount": 1363.0,
                    "currency": "JPY",
                    "country_code": "JP"
                },
            },
            "status": "SUCCESS",
            "sourceCurrency": {
                "amount": 10,
                "country_code": "DE",
                "currency": "EUR",
                "source_marketplace_id": "A1PA6795UKMFR9"
            },
        },
        {
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {
                    "amount": 12.43,
                    "currency": "GBP",
```


(continues on next page)

(continued from previous page)


```
                    "country_code": "UK"
                },
                "ATVPDKIKX0DER": {
                    "amount": 16.25,
                    "currency": "USD",
                    "country_code": "US"
                },
                "A1VC38T7YXB528": {
                    "amount": 2045.0,
                    "currency": "JPY",
                    "country_code": "JP"
                },
            },
            "status": "SUCCESS",
            "sourceCurrency": {
                "amount": 15,
                "country_code": "DE",
                "currency": "EUR",
                "source_marketplace_id": "A1PA6795UKMFR9"
            }
        }
    ]
}
```


Result using version=2 application/vnd.currencylocalization.v2+json


```
{
    "localizedCurrencyResponses": [
        {
            "sourceCurrency": {
                "amount": 10.0,
                "country_code": "DE",
                "currency": "EUR",
                "source_marketplace_id": "A1PA6795UKMFR9"
            },
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {
                    "amount": 8.29,
                    "currency": "GBP",
                    "country_code": "UK"
                },
                "ATVPDKIKX0DER": {
                    "amount": 10.83,
                    "currency": "USD",
                    "country_code": "US"
                },
                "A1VC38T7YXB528": {
                    "amount": 1363.0,
                    "currency": "JPY",
                    "country_code": "JP"
                },
            },
```


(continues on next page)

(continued from previous page)


```
            "localizedCurrency": {
                "A1F83G8C2ARO7P": {
                    "amount": 8.29,
                    "currency": "GBP",
                    "country_code": "UK"
                },
                "ATVPDKIKX0DER": {
                    "amount": 10.83,
                    "currency": "USD",
                    "country_code": "US"
                },
                "A1VC38T7YXB528": {
                    "amount": 1363.0,
                    "currency": "JPY",
                    "country_code": "JP"
                },
            },
            "localizedCurrencyResults": {
                "A1F83G8C2ARO7P": {
                    "localizedCurrency": {
                        "amount": 8.29,
                        "currency": "GBP",
                        "country_code": "UK"
                    },
                    "status": "SUCCESS"
                },
                "ATVPDKIKX0DER": {
                    "localizedCurrency": {
                        "amount": 10.83,
                        "currency": "USD",
                        "country_code": "US"
                    },
                    "status": "SUCCESS"
                },
                "A1VC38T7YXB528": {
                    "localizedCurrency": {
                        "amount": 1363.0,
                        "currency": "JPY",
                        "country_code": "JP"
                    },
                    "status": "SUCCESS"
                },
            },
            "status": "SUCCESS"
        },
        {
            "sourceCurrency": {
                "amount": 15.0,
                "country_code": "DE",
                "currency": "EUR",
                "source_marketplace_id": "A1PA6795UKMFR9"
            },
```


(continues on next page)

(continued from previous page)


```
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {
                    "amount": 12.43,
                    "currency": "GBP",
                    "country_code": "UK"
                },
                "ATVPDKIKX0DER": {
                    "amount": 16.25,
                    "currency": "USD",
                    "country_code": "US"
                },
                "A1VC38T7YXB528": {
                    "amount": 2045.0,
                    "currency": "JPY",
                    "country_code": "JP"
                },
            },
            "localizedCurrency": {
                "A1F83G8C2ARO7P": {
                    "amount": 12.43,
                    "currency": "GBP",
                    "country_code": "UK"
                },
                "ATVPDKIKX0DER": {
                    "amount": 16.25,
                    "currency": "USD",
                    "country_code": "US"
                },
                "A1VC38T7YXB528": {
                    "amount": 2045.0,
                    "currency": "JPY",
                    "country_code": "JP"
                },
            },
            "localizedCurrencyResults": {
                "A1F83G8C2ARO7P": {
                    "localizedCurrency": {
                        "amount": 12.43,
                        "currency": "GBP",
                        "country_code": "UK"
                    },
                    "status": "SUCCESS"
                },
                "ATVPDKIKX0DER": {
                    "localizedCurrency": {
                        "amount": 16.25,
                        "currency": "USD",
                        "country_code": "US"
                    },
                    "status": "SUCCESS"
                },
                "A1VC38T7YXB528": {
```


(continues on next page)

(continued from previous page)

```
                    "localizedCurrency": {
                        "amount": 2045.0,
                        "currency": "JPY",
                        "country_code": "JP"
                    },
                    "status": "SUCCESS"
                }
            },
            "status": "SUCCESS"
        }
    ]
}
```

get_currency(*body: dict, str*, *version: int = 1*, ***kwargs*) → ApiResponse

Gets an array of localized currencies in their target marketplaces, with the advertiser ID and source marketplace ID passed in through the header and body

Returns localized currencies within specified marketplaces.

Requires one of these permissions: ["advertiser_campaign_edit","advertiser_campaign_view"]

'**version**': *int* [optional] Will use content body "application/vnd.currencylocalization.v"+str(version)+"+json"

body: | REQUIRED | { | '**localizeCurrencyRequests**': *list* | [| **'LocalizationCurrencyRequest'**: *dict* | { | **'currency'**: LocalizationCurrency: *dict* | { | **"amount"**: *int* | } | } |] | } | '**targetCountryCodes**': *list*, A list of two-letter country codes. When both marketplaceId and countryCode are present, countryCode is ignored. Please refer to the table above for a list of supported country codes. | '**sourceCountryCode**': *string* | '**sourceMarketplaceId**': *string* | '**targetMarketplaces**': *list* }

Same json file to use

```
{
    "localizeCurrencyRequests": [
        {"currency": {"amount": 10}},
        {"currency": {"amount": 15}}
    ],
    "targetCountryCodes": ["GB", "US", "JP"],
    "sourceCountryCode": "DE",
    "sourceMarketplaceId": "A1PA6795UKMFR9",
    "targetMarketplaces": ["A1F83G8C2ARO7P", "ATVPDKIKX0DER", "A1VC38T7YXB528"]
}
```

Download `json` the file to use:

Example using the above file as request (MarketplacesIds not needed)

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)
```

(continues on next page)

(continued from previous page)

```
def get_currency(data: (str, dict), version: int = 1):

    try:

        result = Localization(debug=True).get_currency(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    file_name = "../test/localizations/request_currency.json"
    try:

        with open(file_name, mode="r", encoding="utf-8") as file:
            get_currency(file_name, version=1)
            file.close()

    except FileNotFoundError as e:
        logging.info(e)
```

Result using version=1 application/vnd.currencylocalization.v1+json

```
{
    "localizedCurrencyResponses": [
        {
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {"amount": 8.29},
                "ATVPDKIKX0DER": {"amount": 10.83},
                "A1VC38T7YXB528": {"amount": 1363.0}
            },
            "status": "SUCCESS"
        },
        {
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {"amount": 12.43},
                "ATVPDKIKX0DER": {"amount": 16.25},
                "A1VC38T7YXB528": {"amount": 2045.0}
            },
            "status": "SUCCESS"
        },
    ]
}
```

Result using version=2 application/vnd.currencylocalization.v2+json

```
{
```

(continues on next page)

(continued from previous page)


```
    "localizedCurrencyResponses": [
        {
            "sourceCurrency": {"amount": 10.0},
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {"amount": 8.29},
                "ATVPDKIKX0DER": {"amount": 10.83},
                "A1VC38T7YXB528": {"amount": 1363.0}
            },
            "localizedCurrency": {
                "A1F83G8C2ARO7P": {"amount": 8.29},
                "ATVPDKIKX0DER": {"amount": 10.83},
                "A1VC38T7YXB528": {"amount": 1363.0}
            },
            "localizedCurrencyResults": {
                "A1F83G8C2ARO7P": {
                    "localizedCurrency": {"amount": 8.29},
                    "status": "SUCCESS"
                },
                "ATVPDKIKX0DER": {
                    "localizedCurrency": {"amount": 10.83},
                    "status": "SUCCESS"
                },
                "A1VC38T7YXB528": {
                    "localizedCurrency": {"amount": 1363.0},
                    "status": "SUCCESS"
                },
            },
            "status": "SUCCESS"
        },
        {
            "sourceCurrency": {"amount": 15.0},
            "localizedCurrencies": {
                "A1F83G8C2ARO7P": {"amount": 12.43},
                "ATVPDKIKX0DER": {"amount": 16.25},
                "A1VC38T7YXB528": {"amount": 2045.0}
            },
            "localizedCurrency": {
                "A1F83G8C2ARO7P": {"amount": 12.43},
                "ATVPDKIKX0DER": {"amount": 16.25},
                "A1VC38T7YXB528": {"amount": 2045.0}
            },
            "localizedCurrencyResults": {
                "A1F83G8C2ARO7P": {
                    "localizedCurrency": {"amount": 12.43},
                    "status": "SUCCESS"
                },
                "ATVPDKIKX0DER": {
                    "localizedCurrency": {"amount": 16.25},
                    "status": "SUCCESS"
                },
                "A1VC38T7YXB528": {
                    "localizedCurrency": {"amount": 2045.0},
```


(continues on next page)

(continued from previous page)

```
                    "status": "SUCCESS"
                }
            },
            "status": "SUCCESS"
        }
    ]
}
```

get_products(*body: dict, str*, *version: int = 1*, ***kwargs*) → ApiResponse

Localizes (maps) products from a source marketplace to one or more target marketplaces. The localization process succeeds for a given target marketplace if a product matching the source product can be found there and the advertiser is eligible to advertise it. Seller requests have an additional condition: the SKU of a localized product must match the SKU of the source product.

Requires one of these permissions: ["advertiser_campaign_edit","advertiser_campaign_view"]

'**version**': *int* [optional] Will use content body "application/vnd.productlocalization.v"+str(version)+"+json"

body: | REQUIRED | { | '**localizeProductRequests**': *list* | [| **'LocalizationProductRequest'**: *dict* | { | **'product'**: LocalizationProduct: *dict* | { | **"asin"**: *string*, The product's Amazon Standard Identification Number. Required for entityType=VENDOR. If caller's entityType is SELLER, this field is optional and can yield better localization results if included. | **"sku"**: *string*, The product's Stock Keeping Unit. Required for entityType=SELLER. If caller's entityType is VENDOR, this field is ignored. | } | } |] | } | '**adType**': *string*, Used to confirm that the caller is eligible to advertise localized products. Currently, only Sponsored Products advertising is supported. [SPONSORED_PRODUCTS] | '**sourceCountryCode**': *string* A two-letter country code. When both marketplaceId and countryCode are present, countryCode is ignored. Please refer to the table above for a list of supported country codes. | '**entityType**': *string* [required] The type of the advertiser accounts for which IDs are specified elsewhere in the request. [SELLER, VENDOR] | '**sourceMarketplaceId**': *string* The ID of the source marketplace. Please see the table within the description of the target details object for supported values. | '**targetDetails**': *list* | '**LocalizationProductTargetDetails**': *dict* The target details for the LocalizationProductRequests. There must be only one target details object per marketplace ID. The advertiser ID may be repeated across target details objects. The order of target details objects is irrelevant. The following marketplaces are supported for product localization: | { | **"marketplaceId"**: *string*, The ID of a target marketplace (a marketplace in which the caller wishes to localize the specified products). For example, if the caller is an advertiser based in the UK (marketplace ID A1F83G8C2ARO7P) and wishes to localize a product to Germany (marketplace ID A1PA6795UKMFR9), the target marketplace ID is that of Germany, i.e., A1PA6795UKMFR9. The following marketplaces are supported for product localization: | **"countryCode"**: *string*, A two-letter country code. When both marketplaceId and countryCode are present, countryCode is ignored. Please refer to the table above for a list of supported country codes. | **"advertiserId"**: *string*, [required] The advertiser ID of the caller in the associated target marketplace. The ID of the source advertiser account. This may be either a marketplace-specific obfuscated ID (AD9EUOBWMS33M), an entity ID (ENTITYYXZDK86N86HG), or a global account ID (amzn1.ads-account.g.e0kbzpoe2gkpai1pqeaca59k8). This is the advertiser ID returned by the Profiles API. An entity ID (one starting with "ENTITY"), or a global account advertiser ID may be provided instead. | } }

Note: The **Product Localization** supports 2 different types of content (Content-Type) so is possible get 2 different responses Just pass an optional parameter (2) as int if you want different option, default is version 1.

Example with a dictionary using marketplaceId for source and target and version 1

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_products(data: (str, dict), version: int = 1):

    try:

        result = Localization(debug=True).get_products(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    product_request = \
    {
        "localizeProductRequests": [
            {
                "product": {
                    "asin": "B000000000",
                    "sku": "SKU-OF-B000000000"
                }
            }
        ],
        "adType": "SPONSORED_PRODUCTS",
        "entityType": "SELLER",
        "sourceMarketplaceId": "A1RKKUPIHCS9HS",
        "sourceAdvertiserId": "AD9EUOBWMS33M",
        "targetDetails": [
            {
                "marketplaceId": "A1F83G8C2ARO7P",
                "advertiserId": "AD9EUOBWMS33M"
            }
        ]
    }

    get_products(product_request)
```

Example with a dictionary using countryCode for source and target and version 2

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_products(data: (str, dict), version: int = 1):

    try:

        result = Localization(debug=True).get_products(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    product_request = \
    {
        "localizeProductRequests": [
            {
                "product": {
                    "asin": "B000000000",
                    "sku": "SKU-OF-B000000000"
                }
            }
        ],
        "adType": "SPONSORED_PRODUCTS",
        "entityType": "SELLER",
        "sourceCountryCode": "ES",
        "sourceAdvertiserId": "AD9EUOBWMS33M",
        "targetDetails": [
            {
                "countryCode": "FR",
                "advertiserId": "AD9EUOBWMS33M"
            }
        ]
    }

    get_products(product_request, version=2)
```

Result using version=1 application/vnd.productlocalization.v1+json

```
{
    "localizedProductResponses": [
        {
            "localizedProducts": {
                "A1F83G8C2ARO7P": {
                    "asin": "B000000000",
                    "sku": "SKU-OF-B000000000"
                }
            },
            "status": "SUCCESS"
        }
    ]
}
```

Result when the product can't be localize version=1

```
{
    "localizedProductResponses": [
        {"errorCode": "INTERNAL_ERROR", "localizedProducts": {}, "status": "FAILURE
↪"}
    ]
}
```

Result using version=2 application/vnd.productlocalization.v2+json

```
{
    "localizedProductResponses": [
        {
            "localizedProductResults": {
                "FR": {
                    "localizedProduct": {
                        "asin": "B000000000",
                        "sku": "SKU-OF-B000000000"
                    },
                    "status": "SUCCESS"
                }
            },
            "localizedProducts": {
                "FR": {
                    "asin": "B000000000",
                    "sku": "SKU-OF-B000000000"
                }
            },
            "sourceProduct": {
                "asin": "B000000000",
                "sku": "SKU-OF-B000000000"
            },
            "status": "SUCCESS"
        }
    ]
}
```

Result when the product can't be localize version=2

```
{
    "localizedProductResponses": [
        {
            "errorCode": "INTERNAL_ERROR",
            "localizedProductResults": {
                "BR": {
                    "errorCode": "INTERNAL_ERROR",
                    "messages": [
                        "Product(asin=B000000000, sku=SKU-OF-B000000000) failed to␣
↪localize from A1RKKUPIHCS9HS to BR"
                    ],
                    "status": "FAILURE"
                }
            },
            "localizedProducts": {},
            "sourceProduct": {
                "asin": "B000000000",
                "sku": "SKU-OF-B000000000"
            },
            "status": "FAILURE"
        }
    ]
}
```

Example Request as json by Country Code

Download `json` the file to use:

```
{
    "localizeProductRequests": [
        {
            "product": {
                "asin": "B000000000",
                "sku": "SKU-OF-B000000000"
            }
        }
    ],
    "adType": "SPONSORED_PRODUCTS",
    "entityType": "SELLER",
    "sourceCountryCode": "ES",
    "sourceAdvertiserId": "AD9EUOBWMS33M",
    "targetDetails": [{"countryCode": "FR", "advertiserId": "AD9EUOBWMS33M"}]
}
```

Example Request as json by Marketplace ID

Download `json` the file to use:

```
{
    "localizeProductRequests": [
        {
            "product": {
                "asin": "B000000000",
                "sku": "SKU-OF-B000000000"
```

(continues on next page)

(continued from previous page)

```
            }
        }
    ],
    "adType": "SPONSORED_PRODUCTS",
    "entityType": "SELLER",
    "sourceMarketplaceId": "A1RKKUPIHCS9HS",
    "sourceAdvertiserId": "AD9EUOBWMS33M",
    "targetDetails": [
        {"marketplaceId": "A2Q3Y263D00KWC", "advertiserId": "AD9EUOBWMS33M"}
    ]
}
```

get_keywords(*body: dict, str*, *version: int = 1*, ***kwargs*) → ApiResponse

Returns localized keywords within specified marketplaces or locales.

Requires one of these permissions: ["advertiser_campaign_edit","advertiser_campaign_view"]

'**version**': *int* [optional] Will use content body "application/vnd.keywordlocalization.v"+str(version)+"+json"

body: | REQUIRED | { | '**localizeKeywordRequests**': *list* List of LocalizationKeywordRequests. The order will be maintained in the response maxItems: 1000 | [| **'LocalizationKeywordRequest'**: *dict* A LocalizationKeywordRequest object. Contains information needed about the keyword to be localized. | **'localizationKeyword'**: *dict* An object containing information about a keyword. | { | **'keyword'**: *str* The keyword string. | } |] | '**sourceDetails**': *dict* | { | **'marketplaceId'**: *string* | **'countryCode'**: *string* | **'locale'**: *string* | } | } | '**targetDetails**': *dict* | { | **'marketplaceIds'**: *string* | **'countryCodes'**: *string* | **'locales'**: *string* | } | } | '**targetCountryCodes**': *list*, A list of two-letter country codes. When both marketplaceId and countryCode are present, countryCode is ignored. Please refer to the table above for a list of supported country codes. | '**sourceCountryCode**': *string* | '**sourceMarketplaceId**': *string* | '**targetMarketplaces**': *list* }

Note: The **Keyword Localization** supports 2 different types of content (Content-Type) so is possible get 2 different responses Just pass an optional parameter (2) as int if you want different option, default is version 1.

Example creating a dynamic dict based on a list of keywords using countryCode for source and Locale to retrieve the locales of targets

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException, Locales

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_keywords(data: (str, dict), version: int = 1):
    try:

        result = Localization(debug=True).get_keywords(
            version=version,
            body=data
        )
```

(continues on next page)

(continued from previous page)

```
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)
    except KeyboardInterrupt as error:
        logging.info(error)


if __name__ == '__main__':

    keywords = ['máquina', 'diagnosis', 'diagnosis multimarca', 'coche', 'automóvil
↪', 'frenos', 'presión', 'neumáticos']
    source_country_code = "ES"
    target_country_codes = ["GB", "FR", "IT", "DE", "CN", "NL", "SE"]

    keywords_request = \
        {
            "localizeKeywordRequests": [{"localizationKeyword": {"keyword": keyword}
↪} for keyword in keywords],
            "sourceDetails": {
                "countryCode": source_country_code,
            },
            "targetDetails": {
                "locales": [Locales[target_country_code].value for target_country_
↪code in target_country_codes]
            }
        }

    get_keywords(keywords_request)
    # get_keywords(keywords_request, version=2)
```

Example static request_keywords_locale.json

```
{
    "localizeKeywordRequests": [
        {"localizationKeyword": {"keyword": "máquina"}},
        {"localizationKeyword": {"keyword": "diagnosis"}},
        {"localizationKeyword": {"keyword": "diagnosis multimarca"}},
        {"localizationKeyword": {"keyword": "coche"}},
        {"localizationKeyword": {"keyword": "automóvil"}},
        {"localizationKeyword": {"keyword": "frenos"}},
        {"localizationKeyword": {"keyword": "presión"}},
        {"localizationKeyword": {"keyword": "neumáticos"}}
    ],
    "sourceDetails": {"countryCode": "ES"},
    "targetDetails": {
        "locales": ["en_GB", "fr_FR", "it_IT", "de_DE", "nl_NL", "sv_SE"]
    }
}
```

Download `json` the file to use

Example result_keyword_locale.json

```
{
    "localizedKeywordResponses": [
        {
            "sourceKeyword": {"keyword": "máquina"},
            "localizedKeywords": {
                "it_IT": {"keyword": "macchina"},
                "en_GB": {"keyword": "machine"},
                "sv_SE": {"keyword": "maskin"},
                "fr_FR": {"keyword": "machine"},
                "de_DE": {"keyword": "Maschine"},
                "nl_NL": {"keyword": "machine"}
            },
            "localizedKeywordResults": {
                "it_IT": {
                    "localizedKeyword": {"keyword": "macchina"},
                    "status": "SUCCESS"
                },
                "en_GB": {
                    "localizedKeyword": {"keyword": "machine"},
                    "status": "SUCCESS"
                },
                "sv_SE": {
                    "localizedKeyword": {"keyword": "maskin"},
                    "status": "SUCCESS"
                },
                "fr_FR": {
                    "localizedKeyword": {"keyword": "machine"},
                    "status": "SUCCESS"
                },
                "de_DE": {
                    "localizedKeyword": {"keyword": "Maschine"},
                    "status": "SUCCESS"
                },
                "nl_NL": {
                    "localizedKeyword": {"keyword": "machine"},
                    "status": "SUCCESS"
                },
            },
            "status": "SUCCESS"
        },
        {...}
    ]
}
```

Example creating a dynamic dict based on a list of keywords using Locale for source and Country Codes for targets

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException, Locales

logging.basicConfig(
```

(continues on next page)

(continued from previous page)

```
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_keywords(data: (str, dict), version: int = 1):
    try:

        result = Localization(debug=True).get_keywords(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)
    except KeyboardInterrupt as error:
        logging.info(error)


if __name__ == '__main__':

    keywords = ['máquina', 'diagnosis', 'diagnosis multimarca', 'coche', 'automóvil
↪', 'frenos', 'presión', 'neumáticos']
    source_country_code = "ES"
    source_locale = Locales[source_country_code].value
    target_country_codes = ["GB", "FR", "IT", "DE", "CN", "NL", "SE"]

    keywords_request = \
        {
            "localizeKeywordRequests": [{"localizationKeyword": {"keyword": keyword}
↪} for keyword in keywords],
            "sourceDetails": {
                "locale": source_locale
            },
            "targetDetails": {
                "countryCodes": target_country_codes,
            }
        }

    get_keywords(keywords_request)
    # get_keywords(keywords_request, version=2)
```

Example static request_keywords_locales.json

```
{
    "localizeKeywordRequests": [
        {"localizationKeyword": {"keyword": "máquina"}},
        {"localizationKeyword": {"keyword": "diagnosis"}},
        {"localizationKeyword": {"keyword": "diagnosis multimarca"}},
        {"localizationKeyword": {"keyword": "coche"}},
        {"localizationKeyword": {"keyword": "automóvil"}},
```

(continues on next page)

(continued from previous page)

```
        {"localizationKeyword": {"keyword": "frenos"}},
        {"localizationKeyword": {"keyword": "presión"}},
        {"localizationKeyword": {"keyword": "neumáticos"}}
    ],
    "sourceDetails": {"locale": "es_ES"},
    "targetDetails": {"countryCodes": ["GB", "FR", "IT", "DE", "NL", "SE"]}
}
```

Download `json` the file to use

Example result keyword GB >> DE, SE, IT, FR, ES, NL

```
{
    "localizedKeywordResponses": [
        {
            "sourceKeyword": {"keyword": "anti-block system"},
            "localizedKeywords": {
                "DE": {"keyword": "Antiblockiersystem"},
                "SE": {"keyword": "antiblocksystem"},
                "IT": {"keyword": "sistema anti-blocco"},
                "FR": {"keyword": "système anti-blocage"},
                "ES": {"keyword": "sistema antibloqueo"},
                "NL": {"keyword": "anti-blokkeersysteem"}
            },
            "localizedKeywordResults": {
                "DE": {
                    "localizedKeyword": {"keyword": "Antiblockiersystem"},
                    "status": "SUCCESS"
                },
                "SE": {
                    "localizedKeyword": {"keyword": "antiblocksystem"},
                    "status": "SUCCESS"
                },
                "IT": {
                    "localizedKeyword": {"keyword": "sistema anti-blocco"},
                    "status": "SUCCESS"
                },
                "FR": {
                    "localizedKeyword": {"keyword": "système anti-blocage"},
                    "status": "SUCCESS"
                },
                "ES": {
                    "localizedKeyword": {"keyword": "sistema antibloqueo"},
                    "status": "SUCCESS"
                },
                "NL": {
                    "localizedKeyword": {"keyword": "anti-blokkeersysteem"},
                    "status": "SUCCESS"
                }
            },
            "status": "SUCCESS"
        }
    ]
```

(continues on next page)

(continued from previous page)

```
}
```

Example creating a dynamic dict based on a list of keywords using MarketplacesIds for both source and targets

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException, MarketplacesIds

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_keywords(data: (str, dict), version: int = 1):
    try:

        result = Localization(debug=True).get_keywords(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)
    except KeyboardInterrupt as error:
        logging.info(error)


if __name__ == '__main__':

    keywords = ['máquina', 'diagnosis', 'diagnosis multimarca', 'coche', 'automóvil
↪', 'frenos', 'presión', 'neumáticos']
    source_country_code = "ES"
    source_marketplace_id = MarketplacesIds[source_country_code].value
    target_country_codes = ["GB", "FR", "IT", "DE", "NL", "SE"]

    keywords_request = \
        {
            "localizeKeywordRequests": [{"localizationKeyword": {"keyword": keyword}
↪} for keyword in keywords],
            "sourceDetails": {
                "marketplaceId": source_marketplace_id
            },
            "targetDetails": {
                "marketplaceIds": [MarketplacesIds[target_country_code].value for␣
↪target_country_code in target_country_codes],
            }
        }

    get_keywords(keywords_request)
```

(continues on next page)

(continued from previous page)

```
    # get_keywords(keywords_request, version=2)
```

Example static request_keywords_marketplace_id.json

```
{
    "localizeKeywordRequests": [
        {"localizationKeyword": {"keyword": "máquina"}},
        {"localizationKeyword": {"keyword": "diagnosis"}},
        {"localizationKeyword": {"keyword": "diagnosis multimarca"}},
        {"localizationKeyword": {"keyword": "coche"}},
        {"localizationKeyword": {"keyword": "automóvil"}},
        {"localizationKeyword": {"keyword": "frenos"}},
        {"localizationKeyword": {"keyword": "presión"}},
        {"localizationKeyword": {"keyword": "neumáticos"}}
    ],
    "sourceDetails": {"marketplaceId": "A1RKKUPIHCS9HS"},
    "targetDetails": {
        "marketplaceIds": [
            "A1F83G8C2ARO7P",
            "A13V1IB3VIYZZH",
            "APJ6JRA9NG5V4",
            "A1PA6795UKMFR9",
            "A1805IZSGTT6HS",
            "A2NODRKZP88ZB9"
        ]
    },
}
```

Download `json` the file to use

Example result keyword FR >> IT, ES, GB

```
{
    "localizedKeywordResponses": [
        {
            "sourceKeyword": {"keyword": "rondelle d'essuie-glace"},
            "localizedKeywords": {
                "APJ6JRA9NG5V4": {"keyword": "tergicristallo"},
                "A1RKKUPIHCS9HS": {"keyword": "limpiaparabrisas"},
                "A1F83G8C2ARO7P": {"keyword": "windscreen wiper washer"}
            },
            "localizedKeywordResults": {
                "APJ6JRA9NG5V4": {
                    "localizedKeyword": {"keyword": "tergicristallo"},
                    "status": "SUCCESS"
                },
                "A1RKKUPIHCS9HS": {
                    "localizedKeyword": {"keyword": "limpiaparabrisas"},
                    "status": "SUCCESS"
                },
                "A1F83G8C2ARO7P": {
                    "localizedKeyword": {"keyword": "windscreen wiper washer"},
```

(continues on next page)

(continued from previous page)

```
                    "status": "SUCCESS"
                },
            },
            "status": "SUCCESS"
        }
    ]
}
```

get_targeting_expression(**args*, ***kwargs*)

Localizes targeting expressions used for advertising targeting.

Localizes (maps) targeting expressions from a source marketplace to one or more target marketplaces.

Requires one of these permissions: ["advertiser_campaign_edit","advertiser_campaign_view"]

'**version**': *int* [optional] Will use content body "application/vnd.targetingexpressionlocalization.v"+str(version)+"+json"

body: | REQUIRED | { | '**targetingExpressionLocalizationRequest**': *list* | [| **'targetDetailsList'**: *dict* LocalizationTargetingTargetDetails | { | **'marketplaceId'**: *string* The ID of the target marketplace. For example, when mapping data from the UK (marketplace ID A1F83G8C2ARO7P) to Germany (marketplace ID A1PA6795UKMFR9), the target marketplace ID is that of the UK, i.e., A1F83G8C2ARO7P. | **'countryCode'**: *string* The ID of the target marketplace. For example, when mapping data from the UK (marketplace ID A1F83G8C2ARO7P) to Germany (marketplace ID A1PA6795UKMFR9), the target marketplace ID is that of the UK, i.e., A1F83G8C2ARO7P. | } |] | '**requests**': *list* LocalizationTargetingExpressionRequest | [| **'LocalizationTargetingExpressionRequest'**: *dict* | { | **'isForNegativeTargeting'**: *bool* Specifies whether the expression is for positive targeting (false) or negative targeting (true). | **'expression'**: LocalizationTargetingExpressionPredicate: *dict* | { | **'type'**: LocalizationTargetingExpressionPredicateType: *string* Targeting predicate type. The following predicate types are supported [asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, asinIsPrimeShippingEligible, asinAgeRangeSameAs, asinGenreSameAs] | **'value'**: *string* The value of the predicate. Targeting expression syntax, including examples of predicates and the values they support, is documented here (https://advertising.amazon.com/API/docs/en-us/bulksheets/sp/sp-general-info/sp-product-attribute-targeting). Only predicates using the following types of data will be localized: | } | } |] | '**sourceDetails**': *dict*, LocalizationTargetingSourceDetails | { | **'marketplaceId'**: *string* The ID of the source marketplace. For example, when mapping data from the UK (marketplace ID A1F83G8C2ARO7P) to Germany (marketplace ID A1PA6795UKMFR9), the source marketplace ID is that of the UK, i.e., A1F83G8C2ARO7P. | **'countryCode'**: *string* A two-letter country code. Please refer to the table above for a list of supported country codes. | } }

Example creating a dynamic dict based using MarketplacesIds for both source and target and predicate asinSameAs and version 1

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException, MarketplacesIds

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)
```

(continues on next page)

(continued from previous page)

```
def get_targeting_expression(data: (str, dict), version: int = 1):
    try:

        result = Localization(debug=True).get_targeting_expression(
            version=version,
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    source_country_code = "GB"
    source_marketplace_id = MarketplacesIds[source_country_code].value

    target_country_code = "ES"
    target_marketplace_id = MarketplacesIds[target_country_code].value

    expression_request = \
        {
            "targetDetailsList": [
                {
                    "marketplaceId": target_marketplace_id,
                }
            ],
            "requests": [
                {
                    "targetingExpression": {
                        "isForNegativeTargeting": True,
                        "expression": [
                            {
                                "type": "asinSameAs",
                                "value": "B08SWH2KP4"
                            }
                        ]
                    }
                }
            ],
            "sourceDetails": {
                "marketplaceId": source_marketplace_id,
            }
        }

    get_targeting_expression(expression_request, version=1)
```

Example result result_targeting_expression_v1.json

```
{
```

(continues on next page)

(continued from previous page)

```
    "responses": [
        {
            "sourceTargetingExpression": {
                "expression": [{"value": "B08SWH2KP4", "type": "asinSameAs"}],
                "isForNegativeTargeting": true
            },
            "localizedTargetingExpressions": {
                "A1RKKUPIHCS9HS": {
                    "expression": [{"value": "B08SWH2KP4", "type": "asinSameAs"}],
                    "isForNegativeTargeting": true
                }
            },
            "localizedTargetingExpressionResults": {
                "A1RKKUPIHCS9HS": {
                    "localizedTargetingExpression": {
                        "expression": [{"value": "B08SWH2KP4", "type": "asinSameAs"}
↪],
                        "isForNegativeTargeting": true
                    },
                    "status": "SUCCESS"
                }
            },
            "status": "SUCCESS"
        }
    ]
}
```

Download `json` the file to use

Example when expression is not available (ASIN in this specific case)

```
{
    "responses": [
        {
            "sourceTargetingExpression": {
                "expression": [{"value": "B000000000", "type": "asinSameAs"}],
                "isForNegativeTargeting": true
            },
            "localizedTargetingExpressions": {},
            "localizedTargetingExpressionResults": {
                "A1RKKUPIHCS9HS": {
                    "status": "FAILURE",
                    "messages": [
                        "No match for ASIN B000000000 could be found in marketplace␣
↪A1RKKUPIHCS9HS."
                    ],
                    "errorCode": "INTERNAL_ERROR"
                }
            },
            "status": "FAILURE",
            "errorCode": "INTERNAL_ERROR"
        }
    ]
```

(continues on next page)

(continued from previous page)

```
}
```

Example creating a dynamic dict based using countryCode for both source and target and predicate asinCategorySameAs and version 2

```
import logging
from ad_api.api import Localization
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)


def get_targeting_expression(data: (str, dict), version: int = 1):
    try:

        result = Localization(debug=True).get_targeting_expression(
            version=version,
            body=data
        )

        logging.info(result)
        print (result.payload)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    source_country_code = "GB"
    target_country_code = "ES"

    expression_request = \
        {
            "targetDetailsList": [
                {
                    "countryCode": target_country_code,
                }
            ],
            "requests": [
                {
                    "targetingExpression": {
                        "isForNegativeTargeting": False,
                        "expression": [
                            {
                                "type": "asinCategorySameAs",
                                "value": "1730635031"
                            }
                        ]
                    }
```

(continues on next page)

(continued from previous page)

```
                }
            ],
            "sourceDetails": {
                "countryCode": source_country_code,
            }
        }

    get_targeting_expression(expression_request, version=2)
```

Example static result_targeting_expression_v2.json

```
{
    "responses": [
        {
            "sourceTargetingExpression": {
                "expression": [{"value": "1730635031", "type": "asinCategorySameAs"}
↪],
                "isForNegativeTargeting": false
            },
            "localizedTargetingExpressions": {
                "ES": {
                    "expression": [
                        {"value": "1909320031", "type": "asinCategorySameAs"}
                    ],
                    "isForNegativeTargeting": false
                }
            },
            "localizedTargetingExpressionResults": {
                "ES": {
                    "localizedTargetingExpression": {
                        "expression": [
                            {"value": "1909320031", "type": "asinCategorySameAs"}
                        ],
                        "isForNegativeTargeting": false
                    },
                    "status": "SUCCESS"
                }
            },
            "status": "SUCCESS"
        }
    ]
}
```

Download `json` the file to use

Example when expression is not available (category in this specific case)

```
{
    "responses": [
        {
            "sourceTargetingExpression": {
                "expression": [{"value": "1730635034", "type": "asinCategorySameAs"}
↪],
```

(continues on next page)

(continued from previous page)


```
                "isForNegativeTargeting": false
            },
            "localizedTargetingExpressions": {},
            "localizedTargetingExpressionResults": {
                "ES": {
                    "status": "FAILURE",
                    "messages": [
                        "There is no category node with the ID 1730635034 in the␣
↪source marketplace A1F83G8C2ARO7P."
                    ],
                    "errorCode": "INTERNAL_ERROR"
                }
            },
            "status": "FAILURE",
            "errorCode": "INTERNAL_ERROR"
        }
    ]
}
```


3.10 Audiences

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/Audiences_prod_3p.json.

Audience Discovery API

class ad_api.api.**Audiences**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_audiences_taxonomy(*body: dict or str*, ***kwargs*) → ApiResponse:

Returns a list of audience categories for a given category path

Requires one of these permissions: [“advertiser_campaign_edit”,”advertiser_campaign_view”]

Kwargs:

advertiserId: *string* (query) The advertiser associated with the advertising account. This parameter is required for the DSP adType, but optional for the SD adType.

nextToken: *string* (query) Token from a previous request. Use in conjunction with the maxResults parameter to control pagination of the returned array.

maxResults: *integer* (query) Sets the maximum number of categories in the returned array. Use in conjunction with the nextToken parameter to control pagination. For example, supplying maxResults=20 with a previously returned token will fetch up to the next 20 items. In some cases, fewer items may be returned. Default value : 250

body: *dict or str* (data) REQUIRED {

“adType”: **string** Enum [DSP, SD]

“categoryPath”:

[

string

]

}

Returns ApiResponse

Example python

```
import logging
from ad_api.api import Audiences
from ad_api.base import AdvertisingApiException, Utils

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)


def list_audiences_taxonomy(data: (str, dict)):
    try:

        result = Audiences(debug=True).list_audiences_taxonomy(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    category_path = "Lifestyle" # In-market

    request = \
        {
            "adType": "SD",
            "categoryPath": [
                category_path
            ]
        }

    list_audiences_taxonomy(request)
```

Example query_taxonomy.json

```
{
  "adType": "SD",
  "categoryPath": [
    "In-market"
  ]
}
```

Download `json` the file to use

Result using the request query above

```
{
    "categoryPath": ["Lifestyle"],
    "categories": [
        {"category": "Automotive Ownership", "audienceCount": 65},
        {"category": "Business & Industry", "audienceCount": 19},
        {"category": "New to Category", "audienceCount": 11},
        {"category": "Entertainment", "audienceCount": 8},
        {"category": "Shoppers", "audienceCount": 4},
        {"category": "Students & Professionals", "audienceCount": 3},
        {"category": "Characteristics", "audienceCount": 2},
        {"category": "Conscious Consumption", "audienceCount": 1},
        {"category": "Family", "audienceCount": 1},
        {"category": "Subscriptions", "audienceCount": 1},
    ],
}
```

list_audiences(*body: dict or str*, ***kwargs*) → ApiResponse:

Returns a list of audience segments for an advertiser. The result set can be filtered by providing an array of Filter objects. Each item in the resulting set will match all specified filters.

Requires one of these permissions: [“advertiser_campaign_edit”,”advertiser_campaign_view”]

Kwargs:

advertiserId: *string* (query) The advertiser associated with the advertising account. This parameter is required for the DSP adType, but optional for the SD adType.

nextToken: *string* (query) Token from a previous request. Use in conjunction with the maxResults parameter to control pagination of the returned array.

maxResults: *integer* (query) Sets the maximum number of categories in the returned array. Use in conjunction with the nextToken parameter to control pagination. For example, supplying maxResults=20 with a previously returned token will fetch up to the next 20 items. In some cases, fewer items may be returned. Default value : 10

body: *dict or str* (data) REQUIRED {

“adType”: **string** Enum [DSP, SD]

“filters”: [

{

“field”: **string** Field to filter by. Supported enums are ‘audienceName’, ‘category’, ‘categoryPath’ and ‘audienceId’. The ‘category’ enum returns all audiences under a high-level category, whereas the ‘categoryPath’ enum expects a path of nodes in the taxonomy tree and returns audiences attached directly to the node at the specified path.

“values”: [

string

]

}

]

}

Returns ApiResponse

Example python

```
import logging
from ad_api.api import Audiences
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)


def list_audiences(**kwargs):
    try:

        result = Audiences(debug=True).list_audiences(
            **kwargs
        )

        logging.info(result)
        print (result.payload)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    audience_name = "Automotive Ownership"

    request = \
        {
            "adType": "SD",
            "filters": [
                {
                    "field": "audienceName",
                    "values": [
                        audience_name
                    ]
                }
            ]
        }

    list_audiences(body=request, maxResults=2)
```

Result using the request query above

```
{
    "audiences": [
        {
            "audienceId": "381539216349606208",
```

(continues on next page)

(continued from previous page)


```
            "audienceName": "LS - Vehicle Owners",
            "description": "People who own vehicles",
            "category": "Lifestyle",
            "createDate": "2019-06-27T13:19:12.656Z",
            "updateDate": "2021-03-10T06:52:21Z",
            "status": "Active",
            "forecasts": {
                "inventoryForecasts": {
                    "all": {
                        "dailyReach": {
                            "lowerBoundInclusive": 2000000,
                            "upperBoundExclusive": 2500000,
                        }
                    }
                }
            },
        },
        {
            "audienceId": "423091775444044780",
            "audienceName": "IM - Coche y moto",
            "description": "Personas cuyas actividades de compras\xa0 indican que␣
↪estan inclinadas a comprar productos de Coche y moto",
            "category": "In-market",
            "createDate": "2013-04-19T00:00:00Z",
            "updateDate": "2017-09-19T23:36:00.964Z",
            "status": "Active",
            "forecasts": {
                "inventoryForecasts": {
                    "all": {
                        "dailyReach": {
                            "lowerBoundInclusive": 1500000,
                            "upperBoundExclusive": 2000000
                        }
                    }
                }
            },
        },
    ],
    "matchCount": 44,
    "nextToken": "RlYDS3X-i8zaLNutSzVIJIQ7RJZ2WYnHyi17DVEHBlZ-skWA-
↪ubGpcNeuoUKas4uKXRen2DYddavk_QtvVRgl8swN6v5KSo86n6tuiy95u4R5OJcmKZ5_7ExV19KnFZIf-
↪aEOnSslt9Kp84HJQjfFffTJRTzf3nIAa61kFPdVuGeW-
↪fEuZvAgdZgowSuW5NhA76bR5wPL1jPcxATKqr4b2dZFYb36r9AYVzpRgEOcI2K6PxECaq4sVDjIztIgHf1BxyLO0ppZcuGFgp
↪l_Bv5zNcWHLqo--"
}
```


Note: If you do not provide any filter it will return all the audiences

Example python retrieving all the audiences with Utils a decorator in sets of 20 with a delay of 5 seconds betwwen query

```
import logging
from ad_api.api import Audiences
from ad_api.base import AdvertisingApiException, Utils

@Utils.load_all_categories(throttle_by_seconds=5, next_token_param="nextToken")
def get_all_categories(**kwargs):
    return Audiences(debug=True).list_audiences(**kwargs)

if __name__ == '__main__':

    request = \
        {
            "adType": "SD",
        }

    for page in get_all_categories(maxResults=20,body=request):
        result = page.payload
        audiences = result.get("audiences")
        for audience in audiences:
            logging.info(audience)
```

Example query_general.json

```
{
  "adType": "SD"
}
```

3.11 Portfolios

https://d3a0d0y2hgofx6.cloudfront.net/openapi/en-us/portfolios/openapi.yaml

Portfolios consist of campaigns that are grouped together and linked to a distinct Advertiser Account. The term 'advertiser' refers to a brand, entity, account identifier, or claim identifier. Multiple portfolios are supported within an Advertiser Account.

class ad_api.api.**Portfolios**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Note: This API **Portfolios** can be used as sandbox environment for testing

list_portfolios(***kwargs*) → ApiResponse

Retrieves a list of portfolios, optionally filtered by identifier, name, or state. Note that this operation returns a maximum of 100 portfolios.

query **portfolioIdFilter**:string | Optional. The returned list includes portfolios with identifiers matching those in the specified comma-delimited list. There is a maximum of 100 identifiers allowed

query **portfolioNameFilter**:string | Optional. The returned list includes portfolios with identifiers matching those in the specified comma-delimited list. There is a maximum of 100 identifiers allowed

query **portfolioStateFilter**:string | Optional. The returned list includes portfolios with states matching those in the specified comma-delimited list. Available values : enabled, paused, archived

Example listing portfolios

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def list_portfolios(**kwargs):
    try:
        result = Portfolios(debug=True).list_portfolios(
            **kwargs
        )

        if result.payload:
            payload = result.payload
            for portfolio in payload:
                logging.info(portfolio)
        else:
            logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    # list_portfolios()
    # list_portfolios(portfolioStateFilter="enabled")
    list_portfolios(portfolioIdFilter="84653842194444,183826157455614,
↪92453628442009,31260007270996")
    # list_portfolios(portfolioNameFilter="Apple,Huawei,Sony")
```

list_portfolios_extended(***kwargs*) → ApiResponse

Retrieves a list of portfolios, optionally filtered by identifier, name, or state. Note that this operation returns a maximum of 100 portfolios.

query **portfolioIdFilter**:string | Optional. The returned list includes portfolios with identifiers matching those in the specified comma-delimited list. There is a maximum of 100 identifiers allowed

query **portfolioNameFilter**:string | Optional. The returned list includes portfolios with identifiers matching those in the specified comma-delimited list. There is a maximum of 100 identifiers allowed

query **portfolioStateFilter**:string | Optional. The returned list includes portfolios with states matching those in the specified comma-delimited list. Available values : enabled, paused, archived

Example listing portfolios extended

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException

```

(continues on next page)

(continued from previous page)

```
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def list_portfolios_extended(**kwargs):
    try:
        result = Portfolios(debug=True).list_portfolios_extended(
            **kwargs
        )

        if result.payload:
            payload = result.payload
            for portfolio in payload:
                logging.info(portfolio)
        else:
            logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':

    # list_portfolios_extended()
    # list_portfolios_extended(portfolioStateFilter="enabled")
    list_portfolios_extended(portfolioIdFilter="84653842194444,183826157455614,
↪92453628442009,31260007270996")
    # list_portfolios_extended(portfolioNameFilter="Apple,Huawei,Sony")
```

get_portfolio(*portfolioId*) → ApiResponse

Retrieves a portfolio data with the portfolioId identifier provided

query **portfolioId**:number | Required. The identifier of an existing portfolio.

Example getting a portfolio by portfolioId

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_portfolio(portfolio_id: int):
    try:
        result = Portfolios(debug=True).get_portfolio(
            portfolioId=portfolio_id
        )
        payload = result.payload
```

(continues on next page)

(continued from previous page)

```
        logging.info(payload)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    id_portfolio = 214026257044134
    get_portfolio(id_portfolio)
```

get_portfolio_extended(*portfolioId*) → ApiResponse

Gets an extended set of properties for a portfolio specified by identifier.

query **portfolioId**:number | Required. The identifier of an existing portfolio.

Example getting a portfolio extended by portfolioId

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def get_portfolio_extended(portfolio_id: int):
    try:
        result = Portfolios(debug=True).get_portfolio_extended(
            portfolioId=portfolio_id
        )
        payload = result.payload
        logging.info(payload)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    id_portfolio = 214026257044134
    get_portfolio_extended(id_portfolio)
```

create_portfolios(*body: list, str, dict*) → ApiResponse

Creates one or more portfolios.

body: | REQUIRED {'description': 'A list of portfolio resources with updated values.}'

name | string | The portfolio name.
budget | dict |

amount | number | The budget amount associated with the portfolio. Cannot be null.

currencyCode | string | The currency used for all monetary values for entities under this profile. Cannot be null.

policy | string | The budget policy. Set to dateRange to specify a budget for a specific period of time. Set to monthlyRecurring to specify a budget that is automatically renewed at the beginning of each month. Cannot be null. Enum: [dateRange, monthlyRecurring]

startDate | string | The starting date in YYYYMMDD format to which the budget is applied. Required if policy is set to dateRange. Not specified if policy is set to monthlyRecurring. Note that the starting date for monthlyRecurring is the date when the policy is set.

endDate | string | The end date after which the budget is no longer applied. Optional if policy is set to dateRange or monthlyRecurring.

inBudget | boolean | Indicates the current budget status of the portfolio. Set to true if the portfolio is in budget, set to false if the portfolio is out of budget.

state | string | The current state of the portfolio. Enum: [enabled, paused, archived]

Example creating a portfolio from a list or from a dict

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def create_portfolios(data: (list, str, dict)):
    try:
        result = Portfolios(debug=True).create_portfolios(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    request_portfolios = \
        [
            {
                "name": "Apple",
                "budget": {
                    "amount": 100,
                    "currencyCode": "EUR",
                    "policy": "monthlyRecurring",
```

(continues on next page)

(continued from previous page)

```
                    "startDate": "20220418"
                },
                "inBudget": False,
                "state": "enabled"
            },
            {
                "name": "Huawei",
                "budget": {
                    "amount": 120,
                    "currencyCode": "EUR",
                    "policy": "dateRange",
                    "startDate": "20220418"
                },
                "inBudget": False,
                "state": "enabled"
            },
            {
                "name": "Sony",
                "budget": {
                    "amount": 120,
                    "currencyCode": "EUR",
                    "policy": "dateRange",
                    "startDate": "20220418"
                },
                "inBudget": False,
                "state": "enabled"
            }
        ]

    create_single_dict_portfolio = \
        {
            "name": "Pioneer",
            "budget": {
                "amount": 44.5,
                "policy": "monthlyRecurring",
                "startDate": "20220418"
            },
            "state": "enabled"
        }

    create_portfolios(request_portfolio)
    # create_portfolios(create_single_dict_portfolio)
```

Example creating a portfolio from a static json file

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
```

(continues on next page)

(continued from previous page)

```
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def create_portfolios(data: (list, str, dict)):
    try:
        result = Portfolios(debug=True).create_portfolios(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    filename = "../test/portfolios/create.json"
    create_portfolios(filename)
```

Static .json file in case want to use as example

```
[
    {
        "name": "Apple",
        "budget": {
            "amount": 100,
            "currencyCode": "EUR",
            "policy": "monthlyRecurring",
            "startDate": "20220418"
        },
        "inBudget": true,
        "state": "enabled"
    },
    {
        "name": "Huawei",
        "budget": {
            "amount": 120,
            "currencyCode": "EUR",
            "policy": "dateRange",
            "startDate": "20220418"
        },
        "inBudget": true,
        "state": "enabled"
    },
    {
        "name": "Sony",
        "budget": {
            "amount": 120,
            "currencyCode": "EUR",
            "policy": "dateRange",
            "startDate": "20220418"
```

(continues on next page)

(continued from previous page)

```
        },
        "inBudget": true,
        "state": "enabled"
    }
]
```

Download `json` the file to use:

edit_portfolios(*body: list, str, dict*) → ApiResponse

Updates one or more portfolios.

body: | REQUIRED {'description': 'A list of portfolio resources with updated values.}'

portfolioId | number | The portfolio identifier.

name | string | The portfolio name.

budget | dict |

amount | number | The budget amount associated with the portfolio. Cannot be null.

currencyCode | string | The currency used for all monetary values for entities under this profile. Cannot be null.

policy | string | The budget policy. Set to dateRange to specify a budget for a specific period of time. Set to monthlyRecurring to specify a budget that is automatically renewed at the beginning of each month. Cannot be null. Enum: [dateRange, monthlyRecurring]

startDate | string | The starting date in YYYYMMDD format to which the budget is applied. Required if policy is set to dateRange. Not specified if policy is set to monthlyRecurring. Note that the starting date for monthlyRecurring is the date when the policy is set.

endDate | string | The end date after which the budget is no longer applied. Optional if policy is set to dateRange or monthlyRecurring.

inBudget | boolean | Indicates the current budget status of the portfolio. Set to true if the portfolio is in budget, set to false if the portfolio is out of budget.

state | string | The current state of the portfolio. Enum: [enabled, paused, archived]

Example editing a portfolio from a list or from a dict

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def edit_portfolios(data: (list, str, dict)):
    try:
        result = Portfolios(debug=True).edit_portfolios(
```

(continues on next page)

(continued from previous page)

```
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    update_portfolios = \
        [
            {
                "portfolioId": 183826157455614,
                "name": "Apple-Macbook",
                "budget": {
                    "amount": 77.5,
                    "currencyCode": "EUR",
                    "policy": "monthlyRecurring",
                    "startDate": "20220418"
                },
                "inBudget": False,
                "state": "enabled"
            },
            {
                "portfolioId": 84653842194444,
                "name": "Huawei-Phone",
                "budget": {
                    "amount": 25,
                    "currencyCode": "EUR",
                    "policy": "dateRange",
                    "startDate": "20220418"
                },
                "inBudget": False,
                "state": "enabled"
            },
            {
                "portfolioId": 924536284420O9,
                "name": "Sony-Headphones",
                "budget": {
                    "amount": 55,
                    "currencyCode": "EUR",
                    "policy": "dateRange",
                    "startDate": "20220418"
                },
                "inBudget": False,
                "state": "enabled"
            }
        ]
```

(continues on next page)

(continued from previous page)

```
    update_single_portfolio = \
        [
            {
                "portfolioId": 183826157455614,
                "name": "Apple-iMac",
                "budget": {
                    "amount": 30.5,
                    "policy": "monthlyRecurring",
                    "startDate": "20220418"
                },
                "inBudget": True,
                "state": "enabled"
            }
        ]

    update_single_dict_portfolio = \
        {
            "portfolioId": 183826157455614,
            "name": "Apple-All",
            "budget": {
                "amount": 80.5,
                "policy": "monthlyRecurring",
                "startDate": "20220418"
            },
            "inBudget": True,
            "state": "enabled"
        }

    # edit_portfolios(update_portfolios)
    # edit_portfolios(update_single_portfolio)
    edit_portfolios(update_single_dict_portfolio)
```

Example editing a portfolio from a static json file

```
import logging
from ad_api.api import Portfolios
from ad_api.base import AdvertisingApiException


logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def edit_portfolios(data: (list, str, dict)):
    try:
        result = Portfolios(debug=True).edit_portfolios(
            body=data
        )

        logging.info(result)
```

(continues on next page)

(continued from previous page)

```
    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    filename = "../test/portfolios/edit.json"
    edit_portfolios(filename)
```

Static .json file in case want to use as example

```
[
    {
        "portfolioId": 183826157455614,
        "name": "Apple-iMac",
        "budget": {
            "amount": 80.5,
            "policy": "monthlyRecurring",
            "startDate": "20220418"
        },
        "inBudget": true,
        "state": "enabled"
    }
]
```

Download `json` the file to use:

3.12 Insights

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/Insights_prod_3p.json

class ad_api.api.**Insights**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

get_insights(*audienceId: str*, *version: int = 1*, ***kwargs*) → ApiResponse

Retrieves the top audiences that overlap with the provided audience.

Requires one of these permissions: [“advertiser_campaign_edit”,”advertiser_campaign_view”]

path **audienceId**:string | Required. The identifier of an audience.

internal **version**:int | Optional. The version of the overlapping audiences accept ‘application/vnd.insightsaudiencesoverlap.v’+str(version)+’+json’. Available values : 1, 2 Default: 1

query **adType**:string | Required. The advertising program. Available values : DSP, SD

query **advertiserId**:string | Optional. The identifier of the advertiser you’d like to retrieve overlapping audiences for. This parameter is required for the DSP adType, but is optional for the SD adType.

query **minimumAudienceSize**:number | Optional. If specified, the sizes of all returned overlapping audiences will be at least the provided size. This parameter is supported only for request to return application/vnd.insightsaudiencesoverlap.v1+json.

query **maximumAudienceSize**:number | Optional. If specified, the sizes of all returned overlapping audiences will be at most the provided size. This parameter is supported only for request to return application/vnd.insightsaudiencesoverlap.v1+json.

query **minimumOverlapAffinity**:number | Optional. If specified, the affinities of all returned overlapping audiences will be at least the provided affinity.

query **maximumOverlapAffinity**:number | Optional. If specified, the affinities of all returned overlapping audiences will be at most the provided affinity.

query **audienceCategory**:array[string] | Optional. f specified, the categories of all returned overlapping audiences will be one of the provided categories.

query **maxResults**:integer | Optional. Sets the maximum number of overlapping audiences in the response. This parameter is supported only for request to return application/vnd.insightsaudiencesoverlap.v2+json. Default value : 30

query **nextToken**:string | Optional. TToken to be used to request additional overlapping audiences. If not provided, the top 30 overlapping audiences are returned. Note: subsequent calls must be made using the same parameters as used in previous requests.

Note: This **Insights** supports 2 different types of content (Accept) so is possible get 2 different responses Just pass an optional parameter (2) as int if you want different option, default is version 1.

Example getting overlapping categories using keyword arguments

```
import logging
from ad_api.api import Insights
from ad_api.base import AdvertisingApiException


def get_insights(audience_id: str, ad_type: str, version:int = 1, **kwargs):

    try:

        result = Insights(debug=True).get_insights(
            audienceId=audience_id,
            adType=ad_type,
            version=version,
            **kwargs
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

if __name__ == '__main__':
    id_audience = "427339506193361161"
    ad_type = "SD"

    get_insights(id_audience, ad_type)
    # With max and min minimumAudienceSize and maximumAudienceSize (version 1)
    # get_insights(id_audience, ad_type, minimumAudienceSize=7,␣
↪maximumAudienceSize=8)
```

(continues on next page)

(continued from previous page)

```
    # With minimumOverlapAffinity (version 2)
    # get_insights(id_audience, ad_type, 2, minimumOverlapAffinity=10)
    # With maximumOverlapAffinity (version 2)
    # get_insights(id_audience, ad_type, 2, maximumOverlapAffinity=5)
    # With Category Lifestyle (version 2)
    # get_insights(id_audience, ad_type, 2, audienceCategory="Lifestyle")
    # With Category Interest (version 2)
    # get_insights(id_audience, ad_type, 2, audienceCategory="Interest")
    # With a maxResults filter (version 2)
    # get_insights(id_audience, ad_type, 2, maxResults=10)
```

Note: If you do not provide any filter it will return all the audiences

Example python retrieving all the audiences with Utils a decorator in sets of 10 with a delay of 5 seconds between query and query

```
import logging
from ad_api.api import Insights
from ad_api.base import AdvertisingApiException, Utils

@Utils.load_all_categories(throttle_by_seconds=5, next_token_param="nextToken")
def get_all_categories(**kwargs):
    return Insights(debug=True).get_insights(**kwargs)

if __name__ == '__main__':

    id_audience = "427339506193361161"
    ad_type = "SD"

    for page in get_all_categories(audienceId=id_audience, adType=ad_type,␣
↪version=2, maxResults=10):
        result = page.payload
        overlapping_audiences = result.get("overlappingAudiences")
        for audience in overlapping_audiences:
            logging.info(audience.get("affinity"))
            logging.info(audience.get("audienceMetadata"))
```

3.13 Reports

class ad_api.api.**Reports**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Sponsored Products Reports Version 3

Documentation: https://advertising.amazon.com/API/docs/en-us/offline-report-prod-3p

Creates a report request. Use this operation to request the creation of a new report for Amazon Advertising Products. Use adProduct to specify the Advertising Product of the report.

post_report(*self*, ***kwargs*) → ApiResponse:

Requests a Sponsored Products report.

Request the creation of a performance report for all entities of a single type which have performance data to report. Record types can be one of campaigns, adGroups, keywords, productAds, asins, and targets. Note that for asin reports, the report currently can not include metrics associated with both keywords and targets. If the targetingId value is set in the request, the report filters on targets and does not return sales associated with keywords. If the targetingId value is not set in the request, the report filters on keywords and does not return sales associated with targets. Therefore, the default behavior filters the report on keywords. Also note that if both keywordId and targetingId values are passed, the report filters on targets only and does not return keywords.

Request body

name (string): [optional] The name of the report.
startDate (string): [required] The maximum lookback window supported depends on the selection of reportTypeId. Most report types support 95 days as lookback window. YYYY-MM-DD format.
endDate (string): [required] The maximum lookback window supported depends on the selection of reportTypeId. Most report types support 95 days as lookback window. YYYY-MM-DD format.
configuration (AsyncReportConfigurationAsyncReportConfiguration): [required] | **adProduct** (string): [required] Enum: The advertising product such as SPONSORED_PRODUCTS or SPONSORED_BRANDS. | **columns** (list > string): [required] The list of columns to be used for report. The availability of columns depends on the selection of reportTypeId. This list cannot be null or empty. | **reportTypeId** (string): [required] The identifier of the Report Type to be generated. | **format** (string): [required] Enum: The report file format. [GZIP_JSON] | **groupBy** (list > string): [required] This field determines the aggregation level of the report data and also makes additional fields available for selection. This field cannot be null or empty. | **filters** (AsyncReportFilterAsyncReportFilter): [optional] | **field** (string): The field name of the filter. | **values** (list > string): The values to be filtered by. | **timeUnit** (string): [required] Enum: The aggregation level of report data. If the timeUnit is set to SUMMARY, the report data is aggregated at the time period specified. The availability of time unit breakdowns depends on the selection of reportTypeId. [SUMMARY, DAILY]

Returns:
ApiResponse

Example python

```python
from ad_api.api.reports import Reports

with open("advertised_product.json", "r", encoding="utf-8") as f:
    data = f.read()

result = Reports().post_report(body=data)
payload = result.payload
report_id = payload.get('reportId')
```

Example json

Open this `json` file to see the result:

```
{
    "startDate": "2022-11-01",
    "endDate": "2022-11-01",
     "configuration": {
```

(continues on next page)

(continued from previous page)

```
        "adProduct": "SPONSORED_PRODUCTS",
        "groupBy": ["advertiser"],
        "columns": ["impressions", "clicks", "cost", "campaignStatus",
↪"advertisedAsin", "date"],
        "reportTypeId": "spAdvertisedProduct",
        "timeUnit": "DAILY",
        "format": "GZIP_JSON"
    }
}
```

get_report(*self*, *reportId*, ***kwargs*) → ApiResponse:

Gets a generation status of a report by id. Uses the reportId value from the response of previously requested report via POST /reporting/reports operation. When status is set to COMPLETED, the report will be available to be downloaded at url. Report generation can take as long as 3 hours. Repeated calls to check report status may generate a 429 response, indicating that your requests have been throttled. To retrieve reports programmatically, your application logic should institute a delay between requests.

Keyword Args

path **reportId** (number): The report identifier. [required]

Returns:
ApiResponse

Example python

```
from ad_api.api.reports import Reports

# this report_id is obtained from post_report method
report_id = '8877234e-bdaa-165d-21ee-45e2e4d8b746'

result = Reports().get_report(reportId=report_id)
```

Result json

```
{
  'configuration':
  {
    'adProduct': 'SPONSORED_PRODUCTS',
    'columns': ['impressions', 'clicks', 'cost', 'campaignStatus', 'advertisedAsin',
↪ 'date'],
    'filters': None,
    'format': 'GZIP_JSON',
    'groupBy': ['advertiser'],
    'reportTypeId': 'spAdvertisedProduct',
    'timeUnit': 'DAILY'
  },
  'createdAt': '2022-11-21T12:00:52.528Z',
  'endDate': '2022-11-01',
  'failureReason': None,
  'fileSize': 51199,
  'generatedAt': '2022-11-21T12:01:53.255Z',
  'name': None,
```

(continues on next page)

(continued from previous page)

```
  'reportId': '8877234e-bdaa-165d-21ee-45e2e4d8b746',
  'startDate': '2022-11-01',
  'status': 'COMPLETED',
  'updatedAt': '2022-11-21T12:01:53.255Z',
  'url': 'https://offline-report-storage-eu-west-1-prod.s3.eu-west-1.amazonaws.com/
↪8877234e-bdaa-165d-21ee-45e2e4d8b746/report-8877234e-bdaa-165d-21ee-45e2e4d8b746.
↪json.gz?X-Amz-Security-Token=****************&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-
↪Amz-Date=20221121T120227Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-
↪Credential=**********************&X-Amz-
↪Signature=************************************************',
  'urlExpiresAt': '2022-11-21T13:02:27.915577Z'
}
```

download_report(*self*, ***kwargs*) → ApiResponse:

Downloads the report previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the report). Take in mind that a direct download of location returned in get_report will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the report (this url has a expiration time)

Keyword Args

url (string): The location obatined from get_report [required]

file (string): The path to save the file if mode is download json, zip or gzip. [optional]

format (string): The mode to download the report: data (list), raw, url, json, zip, gzip. Default (url) [optional]

Returns:

ApiResponse

Example python

```
from ad_api.api.reports import Reports

# the url=url is obtained from get_report method need to in stay 'status': 'SUCCESS'␣
↪if is 'IN_PROGRESS' the report cannot be downloaded
url = 'https://offline-report-storage-eu-west-1-prod.s3.eu-west-1.amazonaws.com/
↪8877234e-bdaa-165d-21ee-45e2e4d8b746/report-8877234e-bdaa-165d-21ee-45e2e4d8b746.
↪json.gz?X-Amz-Security-Token=****************&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-
↪Amz-Date=20221121T120227Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-
↪Credential=**********************&X-Amz-
↪Signature=************************************************'

# path = '/Users/your-profile/Downloads/report_name'
# mode = "data"  # "data (list), raw, url, json, zip, gzip default is url"

result = Reports().download_report(
    url=url,
    # file=path,
```

(continues on next page)

(continued from previous page)

```
    # format=mode
)
```

3.14 Validation Configuration

class ad_api.api.**ValidationConfigurations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Validation Configurations API Version 3

Documentation: https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/ValidationConfigurationsAPI_prod_3p.json

This API allows users to retrieve campaign validation configuration values per marketplace, entity type, and program type.

retrieve_validation_campaigns(*self*, ***kwargs*) → ApiResponse:

Retrieves the campaign configuration values used for campaign validation for the requested marketplace, entityType, and programType context/contexts.

Request body

countryCodesList (Array): [optional] The list of countryCode enums defining the marketplaces whose configuration values are requested. When null is selected and no value is attributed to this key, all countryCode options are selected. [US, CA, MX, BR, UK, DE, FR, ES, IN, IT, NL, AE, SA, SE, TR, PL, EG, JP, AU, SG, CN]

entityTypesList (Array): [optional] The list of entityType enums defining the marketplaces whose configuration values are requested. When null is selected and no value is attributed to this key, all countryCode options are selected. [SELLER, VENDOR]

programTypesList (Array): [optional] The list of programType enums defining the marketplaces whose configuration values are requested. When null is selected and no value is attributed to this key, all countryCode options are selected. [SB, SP, SD]

Returns:
ApiResponse

Example python

```
from ad_api.api import ValidationConfigurations

dictionary = \
{
    "countryCodesList": [
        "ES"
    ],
    "entityTypesList": [
        "SELLER"
    ],
    "programTypesList": [
        "SB"
```

(continues on next page)

(continued from previous page)

```
    ]
}

result = ValidationConfigurations().retrieve_validation_campaigns(
    body=dictionary
)
payload = result.payload
```

ad_api.api.ValidationConfigurations.**retrieve_validation_targeting_clauses**(*self*, ***kwargs*) → ApiResponse:

Retrieves the configuration values used in targeting clause validation for the requested inputted marketplace, entityType, and programType context/contexts.

Request body

countryCodesList (Array): [optional] The list of countryCode enums defining the marketplaces whose configuration values are requested. When null is selected and no value is attributed to this key, all countryCode options are selected. [US, CA, MX, BR, UK, DE, FR, ES, IN, IT, NL, AE, SA, SE, TR, PL, EG, JP, AU, SG, CN]

entityTypesList (Array): [optional] The list of entityType enums defining the marketplaces whose configuration values are requested. When null is selected and no value is attributed to this key, all countryCode options are selected. [SELLER, VENDOR]

programTypesList (Array): [optional] The list of programType enums defining the marketplaces whose configuration values are requested. When null is selected and no value is attributed to this key, all countryCode options are selected. [SB, SP, SD]

Returns:
ApiResponse

Example python

```
from ad_api.api import ValidationConfigurations

dictionary = \
{
    "countryCodesList": [
        "ES"
    ],
    "entityTypesList": [
        "VENDOR"
    ],
    "programTypesList": [
        "SP"
    ]
}

result = ValidationConfigurations().retrieve_validation_targeting_clauses(
    body=dictionary
)
payload = result.payload
```

3.15 Tactical recommendations beta

class ad_api.api.**Recommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

apply_recommendations(***kwargs*)

Applies one or more recommendations.

Request Body (required)

ApplyRecommendationsRequest {
recommendationIds* (array) minItems: 1 maxItems: 100 [
(string) Recommendation identifier.
]
}

Returns
ApiResponse

list_recommendations(***kwargs*)

Retrieves a list of recommendations.

Request Body (required)

ListRecommendationsRequest {
maxResults (integer) default: 500 minimum: 1 maximum: 500
nextToken (string) Token to retrieve the next page of results.
filters* (array) minItems: 1 maxItems: 10 [
ListRecommendationsFilter {
include (boolean): Flag to specify if the filter should be included or excluded. default: true
field* FilterField (string): Field to filter by. Enum: [‘RECOMMENDATION_ID’, ‘AD_PRODUCT’, ‘RECOMMENDATION_TYPE’, ‘STATUS’, ‘GROUPING_TYPE’, ‘CAMPAIGN_ID’]
values* (array) minItems: 1 maxItems: 500 [
string
]
operator FilterOperator (string): Operator to filter field by. Enum: [‘EXACT’]
}
]
}

Returns
ApiResponse

update_recommendation(***kwargs*)

Updates a recommendation.

path **recommendationId**:*string* | Required. The identifier of the recommendation.

Request Body (required)

UpdateRecommendationRequest {
ruleBasedBidding UpdateRuleBasedBidding {
Can only be updated for recommendations with recommendationType NEW_CAMPAIGN_BIDDING_RULE or CAMPAIGN_BIDDING_RULE.
recommendedRuleRoas* (number)
}
recommendedValue (string) Recommended value of the recommendation. Type of data expected for each recommendation type:
budgetRule UpdateBudgetRule {
ruleDetails* UpdateBudgetRuleDetails {
duration UpdateBudgetRuleDuration {
dateRangeTypeDuration UpdateBudgetRuleDurationDateRange {
endDate (string): End date of the budget rule in YYYY-MM-DD format. The end date is inclusive.
}
budgetIncreaseBy BudgetRuleIncreaseBy {
value*(number): Budget of the rule.
}
ruleName* (string): Name of the budget rule. Required to be unique within a campaign. minLength: 1 maxLength: 355
performanceMeasureCondition BudgetRulePerformanceMeasureCondition {
threshold*(number): Threshold of the performance metric.
}
}
}
}

Returns
ApiResponse

CHAPTER

FOUR

AMAZON ATTRIBUTION API OPEN BETA

https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/AmazonAttribution_prod_3p.json

Amazon Attribution

Amazon Attribution is an advertising measurement product that enables advertisers to understand the impact that their non-Amazon ads (i.e. Google Ads, Facebook, Microsoft Ads) have in driving shopping activity on Amazon. Measuring ads using Amazon Attribution is done through implementing Attribution tags on non-Amazon ads. Amazon Attribution is currently available in beta for US, CA, UK, DE, FR, IT, and ES vendors and professional sellers enrolled in Brand Registry.

Amazon Attribution API

The Amazon Attribution API enables agencies and integrators to easily retrieve their advertiser client's non-Amazon publisher attribution tags to automate tag implementation on their non-Amazon ads that link to an Amazon product or Stores page. The API also enables agencies and integrators to create and retrieve reporting on behalf of their advertiser clients to better understand Amazon conversion performance on their campaigns.

Note that you must pass a header named Amazon-Advertising-Api-Scope with each call to an Amazon Attribution API URI, including GET /advertisers. The value for this header is the profileId available from the Profiles resource (/v2/profiles).

`class ad_api.api.Attribution(`*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*`)`

`get_advertisers(`**args*, ***kwargs*`)`

Gets a list of advertisers associated with an Amazon Attribution account.

`get_publishers(`**args*, ***kwargs*`)`

Gets a list of all available publishers.

`post_report(`**args*, ***kwargs*`)`

Gets an attribution report for a specified list of advertisers.

body: | REQUIRED

'**reportType**' (string): The type of report. Either *PERFORMANCE* or *PRODUCTS*. It is an optional parameter. If not used in request body, default reportType is *PERFORMANCE*. Each report type is aggregated at different levels. See below table for list of dimensions available within each report type.

'**advertiserIds**' (string): One or more advertiser Ids to filter reporting by. If requesting reporting for multiple advertiser Ids, input via a comma-delimited list.

'**endDate**' (string): The end date for the report, form as "YYYYMMDD"

'**count**' (integer): maximum: 10000, minimum:1, The number of entries to include in the report.

'**metrics**' (string):

'**startDate**' (string): The start date for the report, in "YYYYMMDD" format. For reportType PRODUCTS, startDate can only be within last 90 days from current date.

'**cursorId**' (string): The value of cursorId must be set to null without "", or set to "" for the first request. For each following request, the value of cursorId from the previous response must be included in the current request. Note that for the cursorId values the " character must be escaped with .

Returns:
ApiResponse

`get_macro_tag`(**args*, ***kwargs*)

Gets a list of attribution tags for third-party publisher campaigns that support macros.

Third-party publishers, such as Google Ads, Facebook, Microsoft Ads, and Pinterest support tags that include macro parameters. Using macro parameters, campaign tracking information is dynamically inserted into the click-through URL when an ad is clicked. This resource is a tag pre-populated with campaign, ad group, and ad level publisher macros with the values associated with your campaign.

For example, a Google Ads macro tag is "?maas=maas_adg_api_123456789_1_99&ref_=aa_maas&tag=maas&aa_campaign
{creative}_{targetid}_dev-{device}_ext-{feeditemid}"

Args:
'**publisherIds**' (array[string]): required. a list of publisher identifiers for which to request tags.

'**advertiserIds**' (array[string]: Optional. List of advertiser identifiers for which to request tags. If no values are passed, all advertisers are returned.

Returns:
ApiResponse

`get_non_macro_template_tag`(**args*, ***kwargs*)

Gets a list of attribution tags for third-party publisher campaigns that do not support macros.

Some third-party publishers do not support tags that include macro parameters. In this case, the attribution tag includes a set of 'insertValue' placeholder values. Replace these placeholder values with your campaign, ad group, and ad identifiers to create unique ad-level tags. For example: "?maas=maas_adg_api_123456789_static_9_99&ref_=aa_maas&tag=maas&aa_campaignid={insertCampaignId}&aa_adg
An example of an integrator nonMacro tag with filled campaign, ad group, and ad ID values is "?maas=maas_adg_api_123456789_static_9_99&ref_=aa_maas&tag=maas&aa_campaignid=12345&aa_adgroupid=5678

Args:
'**publisherIds**' (array[string]): required. a list of publisher identifiers for which to request tags.

'**advertiserIds**' (array[string]: Optional. List of advertiser identifiers for which to request tags. If no values are passed, all advertisers are returned.

Returns:
ApiResponse

CHAPTER

FIVE

BRAND METRICS API OPEN BETA

class ad_api.api.**BrandMetrics**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Brand Metrics provides a new measurement solution that quantifies opportunities for your brand at each stage of the customer journey on Amazon, and helps brands understand the value of different shopping engagements that impact stages of that journey. You can now access Awareness and Consideration indices that compare your performance to peers using models predictive of consideration and sales. Brand Metrics quantifies the number of customers in the awareness and consideration marketing funnel stages and is built at scale to measure all shopping engagements with your brand on Amazon, not just ad-attributed engagements. Additionally, BM breaks out key shopping engagements at each stage of the shopping journey, along with the Return on Engagement, so you can measure the historical sales following a consideration event or purchase.

post_report(**args*, ***kwargs*)

Requests a Sponsored Brands report.

Generates the Brand Metrics report in CSV or JSON format. Customize the report by passing a specific categoryNodeTreeName, categoryNodePath, brandName, reportStartDate, reportEndDate, lookbackPeriod, format or a list of metrics from the available metrics in the metrics field. If an empty request body is passed, report for the latest available report date in JSON format will get generated with all the available brands and metrics for an advertiser. The report may or may not contain the Brand Metrics data for one or more brands depending on data availability.
Request body

categoryNodeTreeName (string): [optional] The node at the top of a browse tree. It is the start node of a tree. example: us-home

categoryNodePath (string): [optional] The hierarchical path that leads to a node starting with the root node. If no Category Node Name is passed, then all data available for all brands belonging to the entity are retrieved. example: List [“Categories”, “Snack Food”, “Granola & Snack Bars”]]

brandName (string): [optional] Brand Name. If no Brand Name is passed, then all data available for all brands belonging to the entity are retrieved. example: Mattress Mogul

reportStartDate (string($date)): [optional] Retrieves metrics with metricsComputationDate between reportStartDate and reportEndDate (inclusive). The date will be in the Coordinated Universal Time (UTC) timezone in YYYY-MM-DD format. If no date is passed in reportStartDate, all available metrics with metricsComputationDate till the reportEndDate will be provided. If no date is passed for either reportStartDate or reportEndDate, the metrics with the most receont metricsComputationDate will be returned. example: 2021-01-01

lookBackPeriod (string) [optional] Currently supported values: “1w” (one week), “1m” (one month) and “1cm” (one calendar month). This defines the period of time used to determine the number of shoppers in the metrics computation [1m, 1w, 1cm]. default: 1w

format (string) [optional] Format of the report. [JSON, CSV]. default: JSON

metrics (list) [optional] Specify an array of string of metrics field names to include in the report. If no metric field names are specified, all metrics are returned.

reportEndDate (string($date)) [optional] Retrieves metrics with metricsComputationDate between reportStartDate and reportEndDate (inclusive). The date will be in the Coordinated Universal Time (UTC) timezone in YYYY-MM-DD format. If no date is passed in reportEndDate, all available metrics with metricsComputationDate from the reportStartDate will be provided. If no date is passed for either reportStartDate or reportEndDate, the metrics with the most receont metricsComputationDate will be returned. example: 2021-01-30

Returns:

ApiResponse

Example python

```
from ad_api.api import BrandMetrics
from ad_api.base import Marketplaces, AdvertisingApiException
import json

dictionary = {
    "categoryNodeTreeName": "es-automotive",
    "brandName": "BMW",
    "reportStartDate": "2022-03-01",
    "lookBackPeriod": "1w",
    "format": "CSV",
    "metrics": [
        "engagedShopperRateLowerBound",
        "customerConversionRate",
        "newToBrandCustomerRate"
    ],
    "reportEndDate": "2022-03-05"
}

try:

    result = BrandMetrics(account=store, marketplace=marketplace, debug=True).post_
↪report(
        body=json.dumps(data)
    )

    print(result)

except AdvertisingApiException as error:

    print(error)
```

Example results

```
{
    "expiration": 300000,
     "format": "CSV",
     "location": "",
     "reportId": "820882fa-ff22-4772-99e1-6889e3ae97f8",
     "status": "IN_PROGRESS",
     "statusDetails": "Generation of the report is in progress!"
}
```

get_report(**args*, ***kwargs*)

Gets a previously requested report specified by identifier.

Keyword Args

path **reportId** (string): [required] The report Id to be fetched.

Returns:

ApiResponse

Example python

```
from ad_api.api import BrandMetrics
from ad_api.base import Marketplaces, AdvertisingApiException

report_id = "820882fa-ff22-4772-99e1-6889e3ae97f8"

try:

    result = BrandMetrics(account=store, marketplace=marketplace, debug=True).get_
↪report(
        reportId=report_id
    )

    print(result)

except AdvertisingApiException as error:

    print(error)
```

Result to get the location

```
{
 "brandsInfo":
 [
   {
      "id": "BRAND|BRAND_AID|34cf8a28-77f0-44dd-kl98-c419a6203257",
      "name": "BMW"
   }
 ],
  "expiration": 300000,
  "format": "CSV",
  "location": "https://infrastack-prod-eu-eu-we-generatedreportsbucket49-
↪96329mbigajd.s3.eu-west-1.amazonaws.com/[...]/[...]",
  "reportId": "820882fa-ff22-4772-99e1-6889e3ae97f8",
  "status": "SUCCESSFUL",
  "statusDetails": "Generation of the report was successful!"
}
```

download_report(*self*, ***kwargs*) → ApiResponse

Downloads the report previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the report).

In this API the url is self-authorized so to avoid double authorization we pass a simple header to ovrewrite the API auth headers that will lead to an error.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path

with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the report (this url has a expiration time)

Keyword Args

url (string): [required] The location obatined from get_report.

file (string): [optional] The path to save the file if mode is download *json*, *zip* or *gzip*.

format (string): [optional] The mode to download the report: *raw*, *url*, *json*, *csv*, Default (*url*)

Returns:

ApiResponse

Example python

```
from ad_api.api import BrandMetrics
from ad_api.base import Marketplaces, AdvertisingApiException

location = "https://infrastack-prod-eu-eu-we-generatedreportsbucket49-96329mbigajd.
↪s3.eu-west-1.amazonaws.com/[...]"


try:

    result = BrandMetrics(account=store, marketplace=marketplace, debug=True).
↪download_report(
        url=location,
        format="csv"
    )

    print(result)

except AdvertisingApiException as error:

    print(error)
```

Result of the file downloaded

```
{
 "headers":{},
 "next_token": None,
 "payload": "A1RLL123456789-820882fa-ff22-4772-99e1-6889e3ae97f8.csv"
}
```

CHAPTER

SIX

SPONSORED PRODUCTS

6.1 2.0

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the migration guide[1].

Amazon Advertising API - Sponsored Products

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-products/2-0/openapi#/ Use the Amazon Advertising API for Sponsored Products for campaign, ad group, keyword, negative keyword, and product ad management operations. For more information about Sponsored Products, see the Sponsored Products Support Center. For onboarding information, see the account setup topic.

This specification is available for download from the Advertising API developer portal.

6.1.1 Campaigns

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the migration guide[1].

class ad_api.api.sp.**Campaigns**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *debug=False*)

Amazon Ads API - Sponsored Products

Warning: This method is a helper that basically will build the json body for you based on params. If you are happy building your own dictinary and passing to the api as json you could use the **create_campaigns** method who follow the Amazon api strictly accepting only the keyword argument body which is a json string and allow to create more than one campaign at the same time.

[1] https://advertising.amazon.com/API/docs/en-us/sponsored-products/v3-migration-guide

[1] https://advertising.amazon.com/API/docs/en-us/sponsored-products/v3-migration-guide

create_single_campaign_assistant(*campaign_name: str*, *targeting_type: str*, *daily_budget: int*, *start_date: str*, *end_date: str = None*, *campaign_status: str = 'enabled'*, *portfolio_id: int = None*, *po_number: str = None*, *account_manager: str = None*, *premium_bid_adjustment: bool = False*, *strategy: str = None*, *predicate: str = None*, *percentage: int = None*, ***kwargs*) → ApiResponse

Creates one campaigns and create the body based on the params provided

Kwargs:

campaign_name (string): [required] A name for the campaign

campaign_type (string): [fixed] The advertising product managed by this campaign. Value: sponsoredProducts

targeting_type (string): [required] The type of targeting for the campaign. Values: manual, auto

daily_budget (float): [required] A daily budget for the campaign

start_date (string): [required] A starting date for the campaign to go live. The format of the date is YYYYMMDD

end_date (string): [optional] An ending date for the campaign to stop running. The format of the date is YYYYMMDD

campaign_status': (string): [optional] The current resource state Values: enabled, paused, archived. Default: enabled

portfolio_id (number): [optional] The identifier of an existing portfolio to which the campaign is associated

po_number (string): [optional] 'A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers

account_manager (string): [optional] A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers

premium_bid_adjustment' (boolean) If set to true, Amazon increases the default bid for ads that are eligible to appear in this placement. See developer notes for more information.

strategy (string): [optional] The bidding strategy. 'Values': legacyForSales (Dynamic bids - down only), autoForSales (Dynamic bids - up and down), manual (Fixed bid)

predicate (string): [optional] You can enable controls to adjust your bid based on the placement location. Specify a location where you want to use bid controls. The percentage value set is the percentage of the original bid for which you want to have your bid adjustment increased. For example, a 50% adjustment on a $1.00 bid would increase the bid to $1.50 for the opportunity to win a specified placement. 'Values': placementTop, placementProductPage

percentage (float): [optional] The bid adjustment percentage value.

Returns:

ApiResponse

Note: The minimal campaign should contain at least a campaign_name, targeting_type, daily_budget and start_date. Once set-up the targeting type, auto or manual, cannot be edited.

Note: If the targeting_type="auto" it cannot combine with premium_bid_adjustment=True. INVALID_ARGUMENT: Cannot have premium bid adjustment on auto targeted campaign. Ignore the premium_bid_adjustment will result in 'premiumBidAdjustment': False.

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def create_single_campaign_assistant(**kwargs):
try:

    result = sponsored_products.Campaigns(account=store, marketplace=marketplace,␣
↪debug=True).create_single_campaign_assistant(
        **kwargs
    )

    logging.info(result)

except AdvertisingApiException as error:
    logging.info(error)


if __name__ == '__main__':

    # Create a campaign targeting auto
    create_single_campaign_assistant(
        campaign_name="Test Auto Basic",
        targeting_type="auto",
        start_date="20220415",
        daily_budget=5
        )

    # Create a campaign targeting manual
    create_single_campaign_assistant(
        campaign_name="Test Manual Basic",
        targeting_type="manual",
        start_date="20220415",
        daily_budget=7.0
        )

    # Create a campaign targeting manual with single adjustment
    create_single_campaign_assistant(
        campaign_name="Test targeting_type manual strategy + Predicate + Percentage
↪",
        targeting_type="manual",
        start_date="20220415",
        daily_budget=20.5,
        strategy="autoForSales",
        predicate="placementProductPage",
        percentage=25
    )
```

(continues on next page)

(continued from previous page)

```
    # Create a campaign targeting manual with complete adjustments as tuple
    create_single_campaign_assistant(
        campaign_name="Test targeting_type manual strategy + Tuple Predicate +␣
↪Percentage",
        targeting_type="manual",
        start_date="20220415",
        daily_budget=20.5,
        strategy="legacyForSales",
        predicate=("placementProductPage", "placementTop"),
        percentage=(10, 15)
    )

    # Note that automatically create 'premiumBidAdjustment': True,
    '''

    {
        'bidding':
            {
                'adjustments':
                    [
                        {
                            'percentage': 10,
                            'predicate': 'placementProductPage'
                        },
                        {
                            'percentage': 15,
                            'predicate': 'placementTop'
                        }
                    ],
                    'strategy': 'legacyForSales'
            },
        'campaignId': 21102815236563,
        'campaignType': 'sponsoredProducts',
        'creationDate': 1649301115000,
        'dailyBudget': 20.5,
        'lastUpdatedDate': 1649301115000,
        'name': 'Test targeting_type manual // strategy + Tuple Predicate + Percentage
↪',
        'premiumBidAdjustment': True,
        'servingStatus': 'PENDING_START_DATE',
        'startDate': '20220415',
        'state': 'enabled',
        'targetingType': 'manual'
    }

    '''

    # Create a complete campaign targeting manual with premium_bid_adjustment True
    create_single_campaign_assistant(
        campaign_name="Test manual + premium_bid_adjustment True",
        targeting_type="manual",
        campaign_status="paused",
```

(continues on next page)

(continued from previous page)

```
        start_date="20220415",
        end_date="20230415",
        daily_budget=25.00,
        portfolio_id=141331265528226,
        po_number="23774",
        account_manager="exampleAccountManager",
        premium_bid_adjustment=True,
    )

    # Note that adjustments are automaticallly created {'adjustments': [{'percentage':␣
↪50, 'predicate': 'placementTop'}], 'strategy': 'legacyForSales'}
    '''

    {
        'bidding':
            {
            'adjustments':
                [
                    {
                        'percentage': 50,
                        'predicate': 'placementTop'
                    }
                ],
                'strategy': 'legacyForSales'},
                'campaignId': 189457270490886,
                'campaignType': 'sponsoredProducts',
                'creationDate': 1649302267000,
                'dailyBudget': 25.0,
                'endDate': '20230415',
                'lastUpdatedDate': 1649302267000,
                'name': 'Test manual + premium_bid_adjustment True',
                'portfolioId': 141331265528226,
                'premiumBidAdjustment': True,
                'servingStatus': 'CAMPAIGN_PAUSED',
                'startDate': '20220415',
                'state': 'paused',
                'tags':
                    {
                        'PONumber': '23774',
                        'accountManager': 'exampleAccountManager'
                    },
                'targetingType': 'manual'
        }
    }

    '''
```

Warning: This method is a helper that basically will build the json body for you based on params. If you are happy building your own dictinary and passing to the api as json you could use the **edit_campaign** method who follow the Amazon api strictly accepting only the keyword argument body which is a json string.

`edit_single_campaign_assistant`(*campaign_id: int*, *portfolio_id: int = None*, *campaign_name: str*, *po_number: str = None*, *account_manager: str = None*, *campaign_status: str = None*, *daily_budget: int*, *start_date: str*, *end_date: str = None*, *premium_bid_adjustment: bool = None*, *strategy:str = None*, *predicate:str or tuple = None*, *percentage:int or tuple = None*, ***kwargs*) → ApiResponse

Edit one campaigns and create the body based on the params provided, at least one of the optional params need to be set or a INVALID_ARGUMENT code is thorwn
Kwargs:

campaign_id (number): [required] The identifier of an existing campaign to update.

portfolio_id (number): [optional] The identifier of an existing portfolio to which the campaign is associated

campaign_name (string): [optional] A name for the campaign

po_number (string): [optional] 'A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers

account_manager (string): [optional] A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers

campaign_status': (string): [optional] The current resource state Values: enabled, paused, archived. Default: enabled

daily_budget (float): [optional] A daily budget for the campaign

start_date (string): [optional] A starting date for the campaign to go live. The format of the date is YYYYMMDD

end_date (string): [optional] An ending date for the campaign to stop running. The format of the date is YYYYMMDD

premium_bid_adjustment' (boolean): [optional] If set to true, Amazon increases the default bid for ads that are eligible to appear in this placement. See developer notes for more information. Tip: When Campaign has been adopted to enhanced bidding premiumBidAdjustment can not be set

strategy (string): [optional] The bidding strategy. 'Values': legacyForSales (Dynamic bids - down only), autoForSales (Dynamic bids - up and down), manual (Fixed bid)

predicate (string or tuple(str)): [optional] You can enable controls to adjust your bid based on the placement location. Specify a location where you want to use bid controls. The percentage value set is the percentage of the original bid for which you want to have your bid adjustment increased. For example, a 50% adjustment on a $1.00 bid would increase the bid to $1.50 for the opportunity to win a specified placement. 'Values': placementTop (str), placementProductPage (str), ("placementTop", "placementProductPage") (tuple)

percentage (float or tuple(float)): [optional] The bid adjustment percentage value. Example: 15 (float), (15, 25) (tuple)

Returns:
ApiResponse

Note: If the targeting_type="manual" and premium_bid_adjustment=True it results in {'bidding': {'adjustments': [{'percentage': 50, 'predicate': 'placementTop'}], 'strategy': 'legacyForSales'} The premium_bid_adjustment can be edited to premium_bid_adjustment=False, even without providing, predicate and percentage. The result is {'bidding': {'adjustments': [], 'strategy': 'legacyForSales'} and is allowed to roll back to premium_bid_adjustment=True

---

Note: If you already set some adjustments automatically is set 'premiumBidAdjustment': True, so if

you try only set to False. INVALID_ARGUMENT: Campaign has been adopted to enhanced bidding controls. 'premiumBidAdjustment' can not be set any more. If you want reset the adjustments you need pass (...,strategy="legacyForSales",predicate=("placementProductPage", "placementTop"),percentage=(0, 0),...). The percentage 0 reset and clean the adjustments. Since no adjustments you can edit again and set premium_bid_adjustment=True. After that you could set premium_bid_adjustment=True or define a strategy, predicate and percent which will make premiumBidAdjustment=True but not both. INVALID_ARGUMENT: Either 'premiumBidAdjustment' or enhanced bidding controls should be specified, but not both.

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

def edit_single_campaign_assistant(**kwargs):
    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).edit_single_campaign_assistant(
            **kwargs
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


if __name__ == '__main__':

    sp_campaign_id = 21102815236563

    # Updating the budget and reset to False the premium_bid_adjustment
    edit_single_campaign_assistant(
        campaign_id=sp_campaign_id,
        daily_budget=28.5,
        strategy="legacyForSales",
        predicate=("placementProductPage", "placementTop"),
        percentage=(0, 0)
    )
```

create_campaigns(*body: dict, str, list*) → ApiResponse

Creates one or more campaigns.

body: | REQUIRED {'description': 'An array of campaigns.}'

'**portfolioId**': *number*, {'description': 'The identifier of an existing portfolio to which the campaign is associated'}

'**name**': *string*, {'description': 'A name for the campaign'}

'**tags**': *string*, {'description': 'A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50

identifiers.'}

'**campaignType**': *string*, {'description': 'The advertising product managed by this campaign', 'Enum': '[sponsoredProducts]'}

'**targetingType**': *string*, {'description': 'The type of targeting for the campaign.', 'Enum': '[manual, auto]'}

'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

'**dailyBudget**': *number($float)*, {'description': 'A daily budget for the campaign.'}

'**startDate**': *string*, {'description': 'A starting date for the campaign to go live. The format of the date is YYYYMMDD.'}

'**endDate**': *string* nullable: true, {'description': 'An ending date for the campaign to stop running. The format of the date is YYYYMMDD.'}

'**premiumBidAdjustment**': *boolean*, {'description': 'If set to true, Amazon increases the default bid for ads that are eligible to appear in this placement. See developer notes for more information.'}

'**bidding**': *Bidding*, {'strategy': 'string', 'Enum': '[legacyForSales, autoForSales, manual]', 'adjustments': '{...}'}

Returns:

ApiResponse

New in version 0.2.7: The support to pass the body as dictionary, list, path to file or content of file.

Warning: The **regular way** to create a campaign is pass a keyword argument body as JSON string but now we could support a variety of other types and will cast to feed the right string in JSON format

Example Sending a dictionary or a list

```
import logging
import json
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def create_campaigns(data: (dict, list)):

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).create_campaigns(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


# If you submit a dictionary the method create_campaigns(body=data) will check if␣
↪is a
# instance of dict and wrap and dumps in JSON string: body = json.dumps([body])
```

(continues on next page)

(continued from previous page)

```
single_dictionary = \
{
    'name': 'Campaign as a dict with no []',
    'campaignType': 'sponsoredProducts',
    'targetingType': 'auto',
    'state': 'paused',
    'dailyBudget': 9,
    'startDate': '20221230'
}

create_campaigns(single_dictionary)

# If you submit a list[{dict},{dict}] which is a right way the wrapper will check␣
↪if is a
# instance of list and dumps in JSON string: body = json.dumps(body)
# This allow you to create 1 or more campaigns at once.

list_dictionary = \
    [
        {
            'portfolioId': 214026257044134,
            'name': 'Campaign manual bid true',
            'tags': {
                'PONumber': 'examplePONumber',
                'accountManager': 'exampleAccountManager'
            },
            'campaignType': 'sponsoredProducts',
            'targetingType': 'manual',
            'state': 'enabled',
            'dailyBudget': 22.0,
            'startDate': '20220430',
            'endDate': '20420430',
            'premiumBidAdjustment': True,
        },
        {
            'portfolioId': 214026257044134,
            'name': 'Campaign manual adjustments true',
            'tags': {
              'PONumber': 'examplePONumber',
              'accountManager': 'exampleAccountManager'
            },
            'campaignType': 'sponsoredProducts',
            'targetingType': 'manual',
            'state': 'enabled',
            'dailyBudget': 22.0,
            'startDate': '20220430',
            'endDate': '20420430',
            'bidding': {
                'strategy': 'legacyForSales',
                'adjustments':
                [
                    {
```

(continues on next page)

(continued from previous page)

```
                    'predicate': 'placementTop',
                    'percentage': 15
                }
            ]
        }
    }
]

create_campaigns(list_dictionary)
# Note that you could dump by yourself the list[dict] as was the classical way
# create_campaigns(json.dumps(list_dictionary))
```

Example Sending a path to a .json file

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException, AdvertisingTypeException

def create_campaigns(data: str):

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).create_campaigns(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

    # Is possible to capture the exceptions if the json contains some errors
    # "<class 'json.decoder.JSONDecodeError'>",
    # JSONDecodeError('Expecting value: line 15 column 33 (char 431)'
    except AdvertisingTypeException as type_error:
        logging.info(type_error)

file_name = "../test/campaigns/sp-sx-create-campaigns.json"

create_campaigns(file_name)

# Or More elegant in your side catching the FileNotFoundError
'''
try:

    with open(file_name, mode="r", encoding="utf-8") as file:
        create_campaigns(file_name)
        file.close()

except FileNotFoundError as e:
    logging.info(e)
```

(continues on next page)

(continued from previous page)

```
'''
```

Example Sending the content of a .json file

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException, AdvertisingTypeException

def create_campaigns(data: str):

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).create_campaigns(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)
    except AdvertisingTypeException as type_error:
        logging.info(type_error)


file_name = "../test/campaigns/sp-sx-create-campaigns.json"

# Read the file and call the create_campaigns(f) before closing the file or
# will raise ValueError: I/O operation on closed file.

try:

    with open(file_name, mode="r", encoding="utf-8") as file:
        f = file.read()
        create_campaigns(f)
        file.close()

except FileNotFoundError as e:
    logging.info(e)
```

Example json

Download `json` the file to use:

```
[
    {
        "portfolioId": 214026257044134,
        "name": "Campaign manual bid true",
        "tags": {
            "PONumber": "203382",
            "accountManager": "Manager001"
        },
        "campaignType": "sponsoredProducts",
```

(continues on next page)

(continued from previous page)


```
        "targetingType": "manual",
        "state": "enabled",
        "dailyBudget": 22.0,
        "startDate": "20220430",
        "endDate": "20420430",
        "premiumBidAdjustment": true
    },
    {
        "portfolioId": 214026257044134,
        "name": "Campaign manual adjustments true",
        "tags": {
          "PONumber": "203382",
          "accountManager": "Manager001"
        },
        "campaignType": "sponsoredProducts",
        "targetingType": "manual",
        "state": "enabled",
        "dailyBudget": 22.0,
        "startDate": "20220430",
        "endDate": "20420430",
        "bidding": {
            "strategy": "legacyForSales",
            "adjustments":
            [
                {
                  "predicate": "placementTop",
                  "percentage": 15
                }
            ]
        }
    },
    {
        "portfolioId": 214026257044134,
        "name": "Campaign auto test from dict as 3 option",
        "tags": {
            "PONumber": "203382",
            "accountManager": "Manager001"
        },
        "campaignType": "sponsoredProducts",
        "targetingType": "auto",
        "state": "enabled",
        "dailyBudget": 12.5,
        "startDate": "20220430",
        "endDate": "20420430"
    }
]
```


edit_campaigns(*body: dict, str, list*) → ApiResponse

Updates one or more campaigns.

body: | REQUIRED {'description': 'An array of campaigns.}'

'**campaignId**': *number*, {'description': 'The identifier of an existing campaign to update.'}

'**portfolioId**': *number*, {'description': 'The identifier of an existing portfolio to which the

campaign is associated'}

'**name**': *string*, {'description': 'The name for the campaign'}

'**tags**': *CampaignTags*, {'description': 'A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers.'}

'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

'**dailyBudget**': *number($float)*, {'description': 'The daily budget for the campaign.'}

'**startDate**': *string*, {'description': 'The starting date for the campaign to go live. The format of the date is YYYYMMDD.'}

'**endDate**': *string* nullable: true, {'description': 'The ending date for the campaign to stop running. The format of the date is YYYYMMDD.'}

'**premiumBidAdjustment**': *boolean*, {'description': 'If set to true, Amazon increases the default bid for ads that are eligible to appear in this placement. See developer notes for more information.'}

'**bidding**': *Bidding*, {'strategy': 'string', 'Enum': '[legacyForSales, autoForSales, manual]', 'adjustments': '{...}'}

Returns:

ApiResponse

New in version 0.2.7: The support to pass the body as dictionary, list, path to file or content of file.

Warning: The **regular way** to create a campaign is pass a keyword argument body as JSON string but now we could support a variety of other types and will cast to feed the right string in JSON format

Example Sending a dictionary or a list

```
import logging
import json
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def edit_campaigns(data: (dict, list)):
    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).edit_campaigns(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


# If you submit a dictionary the method create_campaigns(body=data) will check if␣
↪is a
# instance of dict and wrap and dumps in JSON string: body = json.dumps([body])

single_dictionary = \
```

(continues on next page)

(continued from previous page)

```
    {
        'campaignId': 247123430252449,
        'state': 'enabled',
        'dailyBudget': 99,
    }

edit_campaigns(single_dictionary)


# If you submit a list[{dict},{dict}] which is a right way the wrapper will check␣
↪if is a
# instance of list and dumps in JSON string: body = json.dumps(body)
# This allow you to create 1 or more campaigns at once.

list_dictionary = \
    [
        {
            'campaignId': 247123430252449,
            'tags': {
                'PONumber': '203384',
                'accountManager': 'Manager003'
            },
            'state': 'enabled',
            'dailyBudget': 29.0,
        },
        {
            'campaignId': 1280722862611,
            'state': 'paused',
            'dailyBudget': 22.0,
            'bidding': {
                'strategy': 'legacyForSales',
                'adjustments':
                [
                    {
                        'predicate': 'placementTop',
                        'percentage': 20
                    },
                    {
                        'predicate': 'placementProductPage',
                        'percentage': 25
                    }
                ]
            }
        },
        {
            'campaignId': 71644473255002,
            'state': 'paused'
        }
    ]

edit_campaigns(list_dictionary)
```

Example Sending a path to a .json file

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def edit_campaigns(data: str):
    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).edit_campaigns(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

file_name = "../test/campaigns/sp-sx-edit-campaigns.json"
edit_campaigns(file_name)
```

Example Sending the content of a .json file

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def edit_campaigns(data: str):
    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).edit_campaigns(
            body=data
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)



file_name = "../test/campaigns/sp-sx-edit-campaigns.json"

# Open the file or read, then call the create_campaigns(f) before closing the file␣
↪or
# will raise ValueError: I/O operation on closed file.

try:

    with open(file_name, mode="r", encoding="utf-8") as file:

        # Is also possible to send the instance of TextIOWrapper
        # edit_campaigns(file)
```

(continues on next page)

(continued from previous page)

```
        # But recommend to read it and send the content as str
        f = file.read()
        edit_campaigns(f)
        file.close()

except FileNotFoundError as e:
    logging.info(e)
```

Example json

Download `json` the file to use:

list_campaigns(***kwargs*) → ApiResponse

Gets an array of campaigns.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **name**:*string* | Optional. Restricts results to campaigns with the specified name.

query **portfolioIdFilter**:*string* | Optional. A comma-delimited list of portfolio identifiers.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

Returns:

ApiResponse

Example getting a list of campaigns

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def sp_list_campaigns(**kwargs):

    logging.info("-----------------------------------")
    logging.info("Sponsored Products > list_campaigns(%s)" % str(kwargs))
    logging.info("-----------------------------------")

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).list_campaigns(
            **kwargs
        )

        campaigns = result.payload
        logging.info(len(campaigns))
        for campaign in campaigns:
```

(continues on next page)

(continued from previous page)

```
            logging.info(campaign)

    except AdvertisingApiException as error:
        logging.info(error)

# if no filters provided will return all campaigns
sp_list_campaigns()

# Examples using filters
# sp_list_campaigns(portfolioIdFilter="214026257044134")
# sp_list_campaigns(stateFilter="enabled,paused")
# sp_list_campaigns(stateFilter="enabled", count=10)
# sp_list_campaigns(startIndex=10, stateFilter="enabled", count=10)
# sp_list_campaigns(name="API.ES.Campaign.Manual.JSON.Edit.3")
# sp_list_campaigns(campaignIdFilter="247123430252449,1280722862611,7164447325502")
```

get_campaign(*campaignId: int*) → ApiResponse

Gets a campaign specified by identifier.

path **campaignId**:*number* | Required. The identifier of an existing campaign.

Returns:

ApiResponse

Example getting a campaign

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def sp_get_campaign(campaign_id: int):

    logging.info("---------------------------------")
    logging.info("Sponsored Products > get_campaign(%s)" % campaign_id)
    logging.info("---------------------------------")

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).get_campaign(
            campaignId=campaign_id
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

sp_campaign_id = 247123430252449
sp_get_campaign(sp_campaign_id)
```

delete_campaign(*campaignId: int*) → ApiResponse

Sets the campaign status to archived. Archived entities cannot be made active again. See developer notes for more information.

path **campaignId**:*number* | Required. The identifier of an existing campaign.

Returns:
ApiResponse

Warning: Sets the campaign status to archived. Archived entities cannot be made active again. Consider editing the campaign and setting the status to "paused".

Example deleting a campaign

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def sp_delete_campaign(campaign_id: int):

    logging.info("---------------------------------")
    logging.info("Sponsored Products > delete_campaign(%s)" % campaign_id)
    logging.info("---------------------------------")

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).delete_campaign(
            campaignId=campaign_id
        )

        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)


sp_campaign_id = 217954743666143
sp_delete_campaign(sp_campaign_id)
```

list_campaigns_extended(***kwargs*) → ApiResponse

Gets an array of campaigns with extended data fields.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **name**:*string* | Optional. Restricts results to campaigns with the specified name.

query **portfolioIdFilter**:*string* | Optional. A comma-delimited list of portfolio identifiers.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

Returns:
ApiResponse

Example getting a list of campaigns with extended data

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def sp_list_campaigns_extended(**kwargs):

    logging.info("-----------------------------------")
    logging.info("Sponsored Products > sp_list_campaigns_extended(%s)" %␣
↪str(kwargs))
    logging.info("-----------------------------------")

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).list_campaigns_extended(
            **kwargs
        )

        campaigns = result.payload
        logging.info(len(campaigns))
        for campaign in campaigns:
            logging.info(campaign)

    except AdvertisingApiException as error:
        logging.info(error)

# sp_list_campaigns_extended()
sp_list_campaigns_extended(stateFilter="paused")
```

get_campaign_extended(*campaignId: int*) → ApiResponse

Gets an array of campaigns with extended data fields.

path **campaignId**:*number* | Required. The identifier of an existing campaign.

Returns:

ApiResponse

Example getting a campaign with extended fields

```
import logging
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def sp_get_campaign_extended(campaign_id: int):

    logging.info("-----------------------------------")
    logging.info("Sponsored Products > get_campaign_extended(%s)" % campaign_id)
    logging.info("-----------------------------------")

    try:

        result = sponsored_products.Campaigns(account=store,␣
↪marketplace=marketplace, debug=True).get_campaign_extended(
            campaignId=campaign_id
        )
```

(continues on next page)

(continued from previous page)

```
        logging.info(result)

    except AdvertisingApiException as error:
        logging.info(error)

sp_campaign_id = 247123430252449
sp_get_campaign_extended(sp_campaign_id)
```

References

6.1.2 Ad Groups

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the **`migration guide`_**.

class ad_api.api.sp.**AdGroups**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_ad_groups(*self*, ***kwargs*) → ApiResponse

Creates one or more ad groups.

body: | REQUIRED {'description': 'An array of ad groups.}'
'**name**': *string*, {'description': 'A name for the ad group'}
'**campaignId**': *number*, {'description': 'An existing campaign to which the ad group is associated'}
'**defaultBid**': *number($float)*, {'description': 'A bid value for use when no bid is specified for keywords in the ad group', 'minimum': '0.02'}
'**state**': *string*, {'description': 'A name for the ad group', 'Enum': '[enabled, paused, archived]'}

Returns:
ApiResponse

delete_ad_group(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Sets the ad group status to archived. Archived entities cannot be made active again. See developer notes for more information.
path **adGroupId**:*number* | Required. The identifier of an existing ad group.

Returns:
ApiResponse

edit_ad_groups(***kwargs*)

edit_ad_group(self, **kwargs) -> ApiResponse

Updates one or more ad groups.

body: | REQUIRED {'description': 'An array of ad groups.}'
'**adGroupId**': *number*, {'description': 'The identifier of the ad group.'}
'**name**': *string*, {'description': 'The name of the ad group.'}

'**defaultBid**': *number($float)*, {'description': 'The bid value used when no bid is specified for keywords in the ad group.', 'minimum': '0.02'}

'**state**': *string*, {'description': 'The current resource state', 'Enum': '[enabled, paused, archived]'}

Returns:

ApiResponse

`get_ad_group`(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Gets an ad group specified by identifier.

path **adGroupId**:*number* | Required. The identifier of an existing ad group.

Returns:

ApiResponse

`get_ad_group_extended`(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Gets an ad group that has extended data fields.

path **adGroupId**:*number* | Required. The identifier of an existing ad group.

Returns:

ApiResponse

`list_ad_groups`(*self*, ***kwargs*) → ApiResponse

Gets an array of AdGroup objects for a requested set of Sponsored Display ad groups. Note that the AdGroup object is designed for performance, and includes a small set of commonly used fields to reduce size. If the extended set of fields is required, use the ad group operations that return the AdGroupResponseEx object.

query **startIndex**:*integer* | Optional. Sets a cursor into the requested set of campaigns. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of AdGroup objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first ten ad groups set startIndex=0 and count=10. To return the next ten ad groups, set startIndex=10 and count=10, and so on. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived

query **campaignIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups associated with the campaign identifiers in the specified comma-delimited list.

query **adGroupIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups with an identifier specified in the comma-delimited list.

query **name**:*string* | Optional. The returned array includes only ad groups with the specified name.

Returns:

ApiResponse

`list_ad_groups_extended`(*self*, ***kwargs*) → ApiResponse

Gets an array of AdGroup objects for a requested set of Sponsored Display ad groups. Note that the AdGroup object is designed for performance, and includes a small set of commonly used fields to reduce size. If the extended set of fields is required, use the ad group operations that return the AdGroupResponseEx object.

query: **startIndex**:*integer* | Optional. Sets a cursor into the requested set of campaigns. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of AdGroup objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first ten ad groups set startIndex=0 and count=10. To return the next ten ad groups, set startIndex=10 and count=10, and so on. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived

query **campaignIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups associated with the campaign identifiers in the specified comma-delimited list.

query **adGroupIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups with an identifier specified in the comma-delimited list.

query **name**:*string* | Optional. The returned array includes only ad groups with the specified name.

Returns:
ApiResponse

Campaigns explanation goes here.

6.1.3 Bid Recommendations

class ad_api.api.sp.**BidRecommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

get_ad_group_bid_recommendations(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Gets a bid recommendation for an ad group.
path **adGroupId**:*number* | Required. The identifier of an existing ad group.
Returns:
ApiResponse

get_keyword_bid_recommendations(***kwargs*)

get_ad_group_bid_recommendations(self, adGroupId, **kwargs) -> ApiResponse

Gets a bid recommendation for a keyword.
path **keywordId**:*number* | Required. The identifier of an existing keyword.
Returns:
ApiResponse

get_keywords_bid_recommendations(*self*, ***kwargs*) → ApiResponse:

Gets bid recommendations for keywords.

body: | REQUIRED {'description': 'An array of ad groups.}'
'**adGroupId**': *number*, {'description': 'The identifier of the ad group.'}
keywords {
'**keywords**': *string*, {'description': 'The keyword text.'}
'**matchType**': *string*, {'description': 'The type of match', 'Enum': '[exact, phrase, broad]'}
}
Returns:
ApiResponse

get_targets_bid_recommendations(*self*, ***kwargs*) → ApiResponse:

Gets a list of bid recommendations for keyword, product, or auto targeting expressions.

body: | REQUIRED {'description': 'An array of ad groups.}'
'**adGroupId**': *number*, {'description': 'The ad group identifier.'}
expressions {
'**value**': *string*, {'description': 'The expression value.'}
'**type**': *string*, {'description': 'The type of targeting expression', 'Enum': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}
}

Returns:
ApiResponse

Campaigns explanation goes here.

6.1.4 Keywords

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the **`migration guide`_**.

class ad_api.api.sp.**Keywords**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_keywords(*self*, ***kwargs*) → ApiResponse:

Creates one or more keywords.

body: | REQUIRED {'description': 'An array of keyword objects.}'
'**campaignId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}
'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this keyword is associated.'}
'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}
'**keywordText**': *string*, {'description': 'The keyword text.'}
'**nativeLanguageKeyword**': *string*, {'description': 'The unlocalized keyword text in the preferred locale of the advertiser.'}
'**nativeLanguageLocale**': *string*, {'description': 'The locale preference of the advertiser.'}
'**matchType**': *string*, {'description': 'The type of match.' , 'Enum': '[exact, phrase, broad]'}
'**bid**': *number($float)* {'description': 'Bid associated with this keyword. Applicable to biddable match types only.'}

Returns:
ApiResponse

`delete_keyword`(*self*, *keywordId*, ***kwargs*) → ApiResponse:

Archives a keyword.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

`edit_keywords`(*self*, ***kwargs*) → ApiResponse:

Updates one or more keywords.

body: | REQUIRED {'description': 'An array of keyword objects.}'

'**keywordId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}

'**bid**': *number($float)* {'description': 'Bid associated with this keyword. Applicable to biddable match types only.'}

Returns:

ApiResponse

`get_keyword`(*self*, *keywordId*, ***kwargs*) → ApiResponse

Gets a keyword specified by identifier.

path **keywordId**:*number* | Required. The identifier of an existing keyword. query **locale**:*number* | Optional. The locale preference of the advertiser.

Returns:

ApiResponse

`get_keyword_extended`(*self*, *keywordId*, ***kwargs*) → ApiResponse

Gets a keyword with extended data fields.

path **keywordId**:*number* | Required. The identifier of an existing keyword. query **locale**:*number* | Optional. The locale preference of the advertiser.

Returns:

ApiResponse

`list_keywords`(*self*, ***kwargs*) → ApiResponse

Gets a list of keywords.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list.. Available values : broad, phrase, exact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list..

query **locale**:*string* | Optional. Restricts results to keywords associated with locale.

Returns:

ApiResponse

`list_keywords_extended`(*self*, ***kwargs*) → ApiResponse

Gets a list of keywords that have extended data fields.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list.. Available values : broad, phrase, exact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list..

query **locale**:*string* | Optional. Restricts results to keywords associated with locale.

Returns:

ApiResponse

Campaigns explanation goes here.

6.1.5 Negative Keywords

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the **`migration guide`_**.

`class ad_api.api.sp.NegativeKeywords`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

`create_negative_keywords`(*self*, ***kwargs*) → ApiResponse:

Creates one or more campaign negative keywords.

body: | REQUIRED {'description': 'An array of keyword objects.}'

'**campaignId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled]'}

'**keywordText**': *string*, {'description': 'The text of the expression to match against a search query.'}

'**matchType**': *string*, {'description': 'The type of match.' , 'Enum': '[negativeExact, negativePhrase]'}

Returns:

ApiResponse

`delete_negative_keyword`(*self*, *keywordId*, ***kwargs*) → ApiResponse

Archives a campaign negative keyword.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

`edit_negative_keywords`(*self*, ***kwargs*) → ApiResponse:

Updates one or more campaign negative keywords.

body: | REQUIRED {'description': 'An array of campaign negative keywords with updated values.'}

'**keywordId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}

Returns:

ApiResponse

`get_negative_keyword`(*self*, *keywordId*, ***kwargs*) → ApiResponse

Gets a campaign negative keyword specified by identifier.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

`get_negative_keyword_extended`(*self*, *keywordId*, ***kwargs*) → ApiResponse

Gets a campaign negative keyword that has extended data fields.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

`list_negative_keywords`(*self*, ***kwargs*) → ApiResponse

Gets a list of negative keyword objects.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list. Available values : negativePhrase, negativeExact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list..

Returns:

ApiResponse

`list_negative_keywords_extended`(*self*, ***kwargs*) → ApiResponse

Gets a list of negative keywords that have extended data fields.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list. Available values : negativePhrase, negativeExact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list.

Returns:

ApiResponse

Campaigns explanation goes here.

6.1.6 Campaign Negative Keywords

`class` ad_api.api.sp.`CampaignNegativeKeywords`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

`create_campaign_negative_keywords`(*self*, ***kwargs*) → ApiResponse:

Creates one or more campaign negative keywords.

body: | REQUIRED {'description': 'An array of keyword objects.}'

'**campaignId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled]'}

'**keywordText**': *string*, {'description': 'The text of the expression to match against a search query.'}

'**matchType**': *string*, {'description': 'The type of match.' , 'Enum': '[negativeExact, negativePhrase]'}

Returns:

ApiResponse

delete_campaign_negative_keyword(*self*, *keywordId*, ***kwargs*) → ApiResponse

Archives a campaign negative keyword.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

edit_campaign_negative_keywords(*self*, ***kwargs*) → ApiResponse:

Updates one or more campaign negative keywords.

body: | REQUIRED {'description': 'An array of campaign negative keywords with updated values.'}

'**keywordId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[deleted]'}

Returns:

ApiResponse

get_campaign_negative_keyword(*self*, *keywordId*, ***kwargs*) → ApiResponse

Gets a campaign negative keyword specified by identifier.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

get_campaign_negative_keyword_extended(*self*, *keywordId*, ***kwargs*) → ApiResponse

Gets a campaign negative keyword that has extended data fields.

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns:

ApiResponse

list_campaign_negative_keywords(*self*, ***kwargs*) → ApiResponse

Gets a list of campaign negative keywords.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list. Available values : negativePhrase, negativeExact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list.

Returns:

ApiResponse

list_campaign_negative_keywords_extended(*self*, ***kwargs*) → ApiResponse

Gets a list of campaign negative keywords that have extended data fields.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list. Available values : negativePhrase, negativeExact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list.

Returns:

ApiResponse

6.1.7 Suggested Keywords

class ad_api.api.sp.**SuggestedKeywords**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

get_asin_keywords(*self*, *asinValue*, ***kwargs*) → ApiResponse

Gets suggested keywords for the specified ad group.
path asinValue:*string* | Required. An ASIN.

query maxNumSuggestions:*integer* | Optional. The maxiumum number of suggested keywords for the response. Default value 100.

Returns:
ApiResponse

get_asins_keywords(*self*, ***kwargs*) → ApiResponse:

Gets suggested keyword for a specified list of ASINs.

body: | REQUIRED {'description': 'An object with ASINs.}'
'asins': *string*, {'description': 'A list of ASINs.'}
'maxNumSuggestions': *integer*, {'description': 'The maximum number of suggested keywords in the response. minItems: 1. maxItems: 1000. default: 100'}

Returns:
ApiResponse

get_keywords(***kwargs*)

get_keywords_request(self, adGroupId, **kwargs) -> ApiResponse

Gets suggested keywords for the specified ad group.
path adGroupId:*number* | Required. The identifier of a valid ad group.

query maxNumSuggestions:*integer* | Optional. The maxiumum number of suggested keywords for the response. Default value 100.

query adStateFilter:*string* | Optional. Filters results to ad groups with state matching the comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived.

Returns:
ApiResponse

get_keywords_extended(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Gets suggested keywords with extended data for the specified ad group.

path adGroupId:*number* | Required. The identifier of a valid ad group.

query maxNumSuggestions:*integer* | Optional. The maxiumum number of suggested keywords for the response. Default value 100.

query suggestBids:*string* | Optional. Set to yes to include a suggest bid for the suggested keyword in the response. Otherwise, set to no. Available values : yes, no. Default value : no.

query adStateFilter:*string* | Optional. Filters results to ad groups with state matching the comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived.

Returns:

ApiResponse

Campaigns explanation goes here.

6.1.8 Product Ads

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the **`migration guide`_**.

class ad_api.api.sp.**ProductAds**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_product_ads(*self*, ***kwargs*) → ApiResponse

Creates one or more product ads.

body: | REQUIRED {'description': 'A list of product ads for creation. Note that the SKU field is used by sellers and the ASIN field is used by vendors.'}

'**campaignId**': *number*, {'description': 'The campaign identifier.'}

'**adGroupId**': *number*, {'description': 'The ad group identifier.'}

'**sku**': *string*, {'description': 'The SKU associated with the product. Defined for seller accounts only.'}

'**asin**': *string*, {'description': 'The ASIN associated with the product. Defined for vendors only.'}

'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

Returns:

ApiResponse

delete_product_ad(*self*, *adId*, ***kwargs*) → ApiResponse

Sets the state of a specified product ad to archived. Note that once the state is set to archived it cannot be changed.

path **adId**:*number* | Required. A product ad identifier.

Returns:

ApiResponse

edit_product_ads(*self*, ***kwargs*) → ApiResponse

Updates one or more product ads specified by identifier.

body: | REQUIRED {'description': 'A list of product ad objects with updated values for the state field.}'

'**adId**': *number*, {'description': 'The identifier of an existing campaign to update.'}

'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

Returns:
ApiResponse

get_product_ad(***kwargs*)

get_product_ad_request(self, adId, **kwargs) -> ApiResponse

Gets a product ad specified by identifier.
path **adId**:*number* | Required. A product ad identifier.

Returns:
ApiResponse

get_product_ad_extended(*self*, *adId*, ***kwargs*) → ApiResponse

Gets extended data for a product ad specified by identifier.
path **adId**:*number* | Required. A product ad identifier.

Returns:
ApiResponse

list_product_ads(*self*, ***kwargs*) → ApiResponse

Gets an array of campaigns.
query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **adIdFilter**:*string* | Optional. Restricts results to product ads associated with the product ad identifiers specified in the comma-delimited list.

Returns:
ApiResponse

list_product_ads_extended(***kwargs*)

list_product_ads_extended_request(self, **kwargs) -> ApiResponse

Gets extended data for a list of product ads filtered by specified criteria.
query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **adIdFilter**:*string* | Optional. Restricts results to product ads associated with the product ad identifiers specified in the comma-delimited list.

Returns:

ApiResponse

Campaigns explanation goes here.

6.1.9 Product Targeting

Warning: Sponsored Product v3.0 is not available for Sandbox endpoint while v2.0 it is

Note: This API contains version 2.0 and some endpoints of 3.0 for older version compatibility please migrate to v3.0

class ad_api.api.sp.**Targets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon Advertising API - Sponsored Products - Product Targetting

Use the Amazon Advertising API for Sponsored Products for campaign, ad group, keyword, negative keyword, and product ad management operations. For more information about Sponsored Products, see the Sponsored Products Support Center. For onboarding information, see the account setup topic.

Version 2.0

Doc: https://advertising.amazon.com/API/docs/en-us/sponsored-products/2-0/openapi#/Product%20targeting This specification is available for download from the Advertising API Sponsored Products 2.0.

Version 3.0

Doc: https://advertising.amazon.com/API/docs/en-us/sponsored-products/3-0/openapi/prod#/Product%20Targeting This specification is available for download from the Advertising API Sponsored Products 3.0.

Endpoints available version 2.0

Method	Endpoint	Description
POST	/v2/sp/targets	Creates one or more targeting expressions.
PUT	/v2/sp/targets	Updates one or more targeting clauses.
GET	/v2/sp/targets	Gets a list of targeting clauses filtered by specified criteria.
GET	/v2/sp/targets/{targetId}	Get a targeting clause specified by identifier.
DELETE	/v2/sp/targets/{targetId}	Archives a targeting clause.
GET	/v2/sp/targets/extended	Gets a list of targeting clauses filtered by specified criteria.
GET	/v2/sp/targets/extended/{targetId}	Get a targeting clause specified by identifier.
POST	/v2/sp/targets/productRecommendations	Gets a list of recommended products for targeting.
GET	/v2/sp/targets/brands	Get recommended brands for Sponsored Products.

Endpoints available version 3.0

Method	Endpoint	Description
POST	/sp/targets/categories/recommendations	Returns a list of category recommendations for the input list of ASINs.
GET	/sp/negativeTargets/brands/recommendations	Returns brands recommended for negative targeting.
POST	/sp/targets/products/count	Get number of targetable asins based on refinements provided by the user.
GET	/sp/targets/categories	Returns all targetable categories.
POST	/sp/negativeTargets/brands/search	Returns brands related to keyword input for negative targeting.
GET	/sp/targets/category/{categoryId}/refinements	Returns refinements according to category input.

`create_products_targets`(*self*, ***kwargs*) → ApiResponse:

Creates one or more targeting expressions.

body: | REQUIRED {'description': 'An array of asins objects.}'

'**campaignId**': *number*, {'description': 'The number or recommendations returned in a single page.'}

'**adGroupId**': *number*, {'description': 'The page number in the result set to return.'}

'**expression**'

'**value**': *string*, {'description': 'The expression value.'}

'**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}

'**resolvedExpression**'

'**value**': *string*, {'description': 'The expression value.'}

'**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}

'**expressionType**': *string*, {'description': '[auto, manual]'}

'**bid**': *number*, {'description': 'The bid for ads sourced using the target. Min / Max 0.02 / 1000'}

Returns:

ApiResponse

`edit_products_targets`(*self*, ***kwargs*) → ApiResponse:

Updates one or more targeting clauses.

body: | REQUIRED {'description': 'An array of asins objects.}'

'**targetId**': *number*, {'description': 'The target id.'}

'**state**': *string*, {'description': '[enabled, paused, archived]'}

'**expression**'

'**value**': *string*, {'description': 'The expression value.'}

'**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan,

asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}

'**resolvedExpression**'

'**value**': *string*, {'description': 'The expression value.'}

'**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}

'**expressionType**': *string*, {'description': '[auto, manual]'}

'**bid**': *number*, {'description': 'The bid for ads sourced using the target. Min / Max 0.02 / 1000'}

Returns:

ApiResponse

`list_products_targets`(*self*, ***kwargs*) → ApiResponse:

Gets a list of targeting clauses filtered by specified criteria.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers.

Returns:

ApiResponse

`get_products_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Get a targeting clause specified by identifier.

path **targetId**:*number* | Required. The target identifier.

Returns:

ApiResponse

`delete_products_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Archives a targeting clause.

path **targetId**:*number* | Required. The target identifier.

Returns:

ApiResponse

`list_products_targets_extended`(*self*, ***kwargs*) → ApiResponse:

Gets a list of targeting clauses filtered by specified criteria.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers.

Returns:
ApiResponse

`get_products_target_extended`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Get a targeting clause specified by identifier.
path **targetId**:*number* | Required. The target identifier.

Returns:
ApiResponse

`get_products_targets_recommendations`(*self*, ***kwargs*) → ApiResponse:

Gets a list of recommended products for targeting.

body: | REQUIRED {'description': 'An array of asins objects.}'
'**pageSize**': *number*, {'description': 'The number or recommendations returned in a single page.'}
'**pageNumber**': *number*, {'description': 'The page number in the result set to return.'}
'**asins**': list>*string*, {'description': 'A list of ASINs.'}

Returns:
ApiResponse

`get_brand_targets`(*self*, ***kwargs*) → ApiResponse:

Gets a list of recommended products for targeting.
path **keyword**:*string* | Optional Unique exclude categoryId. A keyword for which to get recommended brands. path **categoryId**:*number* | Optional Unique exclude keyword. Gets the top 50 brands for the specified category identifier.

Returns:
ApiResponse

`list_products_targets_categories_recommendations`(*self*, ***kwargs*) → ApiResponse:

Returns a list of category recommendations for the input list of ASINs. Use this API to discover relevant categories to target. To find ASINs, either use the Product Metadata API or browse the Amazon Retail Website.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

body: | REQUIRED {'description': 'An array of asins objects.}'
'**asins**': list>*string*, {'description': 'List of input ASINs. This API does not check if the ASINs are valid ASINs. maxItems: 10000.'}

'**includeAncestor**': *boolean*, {'description': 'Enable this if you would like to retrieve categories which are ancestor nodes of the original recommended categories. This may increase the number of categories returned, but decrease the relevancy of those categories.'}

Returns:

ApiResponse

`list_negative_targets_brands_recommendations`(*self*, ***kwargs*) → ApiResponse:

Returns brands recommended for negative targeting. Only available for Sellers and Vendors. These recommendations include your own brands because targeting your own brands usually results in lower performance than targeting competitors' brands.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

`get_products_targets_count`(*self*, ***kwargs*) → ApiResponse:

Get number of targetable asins based on refinements provided by the user. Please use the GetTargetableCategories API or the GetCategoryRecommendationsForASINs API to retrieve the category ID. Please use the GetRefinementsByCategory API to retrieve refinements data for a category.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

`list_targets_categories`(*self*, ***kwargs*) → ApiResponse:

Returns all targetable categories. This API returns a large JSON string containing a tree of category nodes. Each category node has the fields - category id, category name, and child categories.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

`list_negative_targets_brands_search`(*self*, ***kwargs*) → ApiResponse:

Returns brands related to keyword input for negative targeting.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

`list_products_targets_category_refinements`(*self*, *categoryId*, ***kwargs*) → ApiResponse:

Get a targeting clause specified by identifier.

path **categoryId**:*string* | Required. The target identifier.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

6.1.10 Negative Product Targeting

Deprecated since version 4.0.2.

Warning: There is a new version 3 of Sponsored Product API, please check the **`migration guide`_**.

class ad_api.api.sp.**NegativeTargets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_negative_targets(***kwargs*)

create_products_targets(self, **kwargs) -> ApiResponse:

Creates one or more targeting expressions.

body: | REQUIRED {'description': 'An array of asins objects.}'
'**campaignId**': *number*, {'description': 'The identifier of the campaign to which this negative target is associated.'}
'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this negative target is associated.'}
'**state**': *number*, {'description': 'The current resource state. [enabled, paused, archived]'}
'**expression**'
'**value**': *string*, {'description': 'The expression value. '}
'**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}
'**expressionType**': *string*, {'description': '[auto, manual]'}

Returns:
ApiResponse

delete_negative_targets(*self*, *targetId*, ***kwargs*) → ApiResponse

Archives a negative targeting clause.
path **targetId**:*number* | Required. The target identifier.

Returns:
ApiResponse

edit_negative_targets(*self*, ***kwargs*) → ApiResponse:

Updates one or more negative targeting clauses.

body: | REQUIRED {'description': 'An array of asins objects.}'
'**campaignId**': *number*, {'description': 'The identifier of the campaign to which this negative target is associated.'}
'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this negative target is associated.'}
'**state**': *number*, {'description': 'The current resource state. [enabled, paused, archived]'}
'**expression**'
'**value**': *string*, {'description': 'The expression value. '}
'**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan,

asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]' }

'**expressionType**': *string*, { 'description': '[auto, manual]' }

Returns:
ApiResponse

`get_negative_target`(***kwargs*)

get_negative_targets(self, targetId, **kwargs) -> ApiResponse

Get a negative targeting clause specified by identifier.
path **targetId**:*number* | Required. The target identifier.

Returns:
ApiResponse

`get_negative_target_extended`(*self*, *targetId*, ***kwargs*) → ApiResponse

Get a negative targeting clause specified by identifier.
path **targetId**:*number* | Required. The target identifier.

Returns:
ApiResponse

`list_negative_targets`(*self*, ***kwargs*) → ApiResponse

Gets a list of negative targeting clauses filtered by specified criteria.
query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers.

Returns:
ApiResponse

`list_negative_targets_extended`(*self*, ***kwargs*) → ApiResponse

Gets a list of negative targeting clauses filtered by specified criteria.
query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers.

Returns:

ApiResponse

6.1.11 Reports

class ad_api.api.sp.**Reports**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Sponsored Products Reports

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-products/2-0/openapi#/Reports

Use the Amazon Advertising API for Sponsored Products for campaign, ad group, keyword, negative keyword, and product ad management operations. For more information about Sponsored Products, see the Sponsored Products Support Center. For onboarding information, see the account setup topic.

post_report(*self*, *recordType*, ***kwargs*) → ApiResponse:

Requests a Sponsored Products report.

Request the creation of a performance report for all entities of a single type which have performance data to report. Record types can be one of campaigns, adGroups, keywords, productAds, asins, and targets. Note that for asin reports, the report currently can not include metrics associated with both keywords and targets. If the targetingId value is set in the request, the report filters on targets and does not return sales associated with keywords. If the targetingId value is not set in the request, the report filters on keywords and does not return sales associated with targets. Therefore, the default behavior filters the report on keywords. Also note that if both keywordId and targetingId values are passed, the report filters on targets only and does not return keywords.

Keyword Args

path **recordType** (integer): The type of entity for which the report should be generated. Available values : campaigns, adGroups, keywords, productAds, asins, targets [required]

Request body

stateFilter (string): [optional] Filters the response to include reports with state set to one of the values in the comma-delimited list. Note that this filter is only valid for reports of the following type and segment. Asins and targets report types are not supported. Enum [enabled, paused, archived].

campaignType (list > string): [required] Enum: The type of campaign. Only required for asins report - don't use with other report types. [sponsoredProducts]

segment (string) Dimension on which the report is segmented. Note that Search-terms report for auto-targeted campaigns created before 11/14/2018 can be accessed from the /v2/sp/keywords/report resource. Search-terms report for auto-targeted campaigns generated on-and-after 11/14/2018 can be accessed from the /v2/sp/targets/report resource. Also, keyword search terms reports only return search terms that have generated at least one click or one sale. Enum [query, placement].

reportDate (string): [optional] The date for which to retrieve the performance report in YYYYMMDD format. The time zone is specified by the profile used to request the report. If this date is today, then the performance report may contain partial information. Reports are not available for data older than 60 days. For details on data latency, see the Service Guarantees in the developer notes section.

metrics (string) [optional] A comma-separated list of the metrics to be included in the report. The following tables summarize report metrics which can be requested via the reports interface. Different report types can use different metrics. Note that ASIN reports only return data for either keywords or targets, but not both.

Returns:

ApiResponse

Example python

```
from ad_api.api.sp.reports import Reports

file = open("ad_groups.json")
data = file.read()
file.close()

# Available values : campaigns, adGroups, keywords, productAds, asins, targets
record_type = 'adGroups'

result = Reports().post_report(
    recordType=record_type,
    body=data
)

payload = result.payload
report_id = payload.get('reportId')
```

Example json

Open this `json` file to see the result:

```
{
  "stateFilter": "enabled",
  "reportDate": "20210917",
  "metrics": "campaignName,campaignId,adGroupName,adGroupId,impressions,clicks,
↪attributedConversions30d"
}
```

get_report(*self*, *reportId*, ***kwargs*) → ApiResponse:

Gets a previously requested report specified by identifier.

Keyword Args

path **reportId** (number): The report identifier. [required]

Returns:

ApiResponse

Example python

```
from ad_api.api.sp.reports import Reports

# this report_id is obtained from post_report method
report_id = 'amzn1.clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-
↪fc1fecdb165c'

result = Reports().get_report(
    reportId=report_id
)
```

Result json

```
{
    "expiration": 1640736000000,
    "fileSize": 6546,
    "location": "https://advertising-api-eu.amazon.com/v1/reports/amzn1.
↪clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c/download",
    "reportId": "amzn1.clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-
↪fc1fecdb165c",
    "status": "SUCCESS",
    "statusDetails": "Report has been successfully generated."
}
```

download_report(*self*, ***kwargs*) → ApiResponse:

Downloads the report previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the report). Take in mind that a direct download of location returned in get_report will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the report (this url has a expiration time)

Keyword Args

url (string): The location obatined from get_report [required]

file (string): The path to save the file if mode is download json, zip or gzip. [optional]

format (string): The mode to download the report: data (list), raw, url, json, zip, gzip. Default (url) [optional]

Returns:

ApiResponse

> **Warning:** This method is not a part of the Amazon Advertising Api.

Example python

```
from ad_api.api.sp.reports import Reports

# the url=location is obtained from get_report method need to in stay 'status':
↪'SUCCESS' if is 'IN_PROGRESS' the report cannot be downloaded
location = 'https://advertising-api-eu.amazon.com/v1/reports/amzn1.clicksAPI.
↪v1.p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c/download'

# path = '/Users/your-profile/Downloads/report_name'
# mode = "data"  # "data (list), raw, url, json, zip, gzip default is url"

result = Reports().download_report(
    url=location,
    # file=path,
    # format=mode
)
```

Tip: Just provide the url keyword arg with the location and you will get a response with the url ready to download

6.1.12 Snapshots

class ad_api.api.sp.**Snapshots**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Products for campaign, ad group, keyword, negative keyword, and product ad management operations. For more information about Sponsored Products, see the Sponsored Products Support Center. For onboarding information, see the account setup topic.

post_snapshot(*self*, *recordType*, ***kwargs*) → ApiResponse:

Request a file-based snapshot of all entities of the specified type in the account satisfying the filtering criteria.

Keyword Args

path **recordType** (integer): The type of entity for which the snapshot is generated. Available values : campaigns, adGroups, keywords, negativeKeywords, campaignNegativeKeywords, productAds, targets, negativeTargets [required]

Request body

stateFilter (string): [required] [enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived].

Returns:

ApiResponse

Example python

```
from ad_api.api.sp.snapshots import Snapshots

file = open("request.json")
data = file.read()
file.close()

# Available values : campaigns, adGroups, keywords, negativeKeywords,␣
↪campaignNegativeKeywords, productAds, targets, negativeTargets

record_type = 'campaigns'

result = Snapshots().post_snapshot(
    recordType=record_type,
    body=data
)
```

Example json

Open this `json` file to see the result:

```
{
  "stateFilter": "enabled"
}
```

get_snapshot(*self*, *reportId*, ***kwargs*) → ApiResponse:

Gets the status of a requested snapshot.
Keyword Args

path **snapshotId** (number): The snapshot identifier. [required]
Returns:
ApiResponse
Example python

```
from ad_api.api.sp.snapshots import Snapshots

# this snapshot_id is obtained from post_snapshot method
snapshot_id = "amzn1.clicksAPI.v1.p44551.614D9309.84477233-ccc8-4591-80f2-
↪1f96b7ea9c7e"

result = Snapshots().get_snapshot(
    snapshotId=snapshot_id
)
```

Result json

```
{'expiration': 1640304000000,
 'fileSize': 1241,
 'location': 'https://advertising-api-eu.amazon.com/v1/snapshots/amzn1.
↪clicksAPI.v1.p44551.614D9309.84477233-ccc8-4591-80f2-1f96b7ea9c7e/download',
 'snapshotId': 'amzn1.clicksAPI.v1.p44551.614D9309.84477233-ccc8-4591-80f2-
↪1f96b7ea9c7e',
 'status': 'SUCCESS',
 'statusDetails': 'Snapshot has been successfully generated.'}}
```

download_snapshot(*self*, ***kwargs*) → ApiResponse:

Downloads the snapshot previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the snapshot). Take in mind that a direct download of location returned in get_snapshot will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the snapshot (this url has a expiration time)
Keyword Args

url (string): The location obatined from get_snapshot [required]

file (string): The path to save the file if mode is download json, zip or gzip. [optional]

format (string): The mode to download the snapshot: data (list), raw, url, json, zip, gzip. Default (url) [optional]

Returns:
ApiResponse

> **Warning:** This method is not a part of the Amazon Advertising Api.

Example python

```
from ad_api.api.sp.snapshots import Snapshots

# the url=location is obtained from get_snapshot method need to in stay 'status
↪': 'SUCCESS' if is 'IN_PROGRESS' the snapshot cannot be downloaded
location = 'https://advertising-api-eu.amazon.com/v1/snapshots/amzn1.clicksAPI.
↪v1.p44551.614D9309.84477233-ccc8-4591-80f2-1f96b7ea9c7e/download'

# path = '/Users/your-profile/Downloads/report_name'
# mode = "data"  # "data (list), raw, url, json, zip, gzip default is url"

result = Reports().download_report(
    url=location,
    # file=path,
    # format=mode
)
```

6.1.13 References

6.2 3.0

Sponsored Products

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-products/3-0/openapi/prod#/

This specification is available for download from the Advertising API developer portal.

6.2.1 Campaigns

class ad_api.api.sp.**CampaignsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon Ads API - Sponsored Products

> **Warning:** This replaces the version 2 of Campaigns

create_campaigns(*body: dict, str, list*) → ApiResponse

Request Body [Required]

name (string): [required] The name of the campaign. This name must be unique to the Amazon Advertising account to which the campaign is associated. Maximum length of the string is 128 characters.

state (State > string): [required] Enum: [enabled, paused, archived]

portfolio_id (int) [optional] The identifier of the portfolio to which the campaign is associated.

budget (float): [required] | **budgetType** (BudgetType > String) : Enum [DAILY] | **budget** (float) : The budget amount associated with the campaign.

targetingType (Targeting > string) : [required] Enum : [AUTO, MANUAL]

dynamicBidding ({}) [optional] | **placementBidding** (string) : You can enable controls to adjust your bid based on the placement location. Specify a location where you want to use bid controls. | **strategy** (BiddingStrategy > String) : Enum [LEGACY_FOR_SALES, AUTO_FOR_SALES, MANUAL, RULE_BASED]

startDate [optional] (string) : A starting date for the campaign to go live. The format of the date is YYYYMMDD.

endDate [optional] (string) : An ending date for the campaign to stop running. The format of the date is YYYYMMDD.

tags ({string}) : A list of advertiser-specified custom identifiers for the campaign.

Returns

ApiResponse

`edit_campaigns`(*body: dict, str, list*) → ApiResponse

Update existing Sponsored Product Campaigns.
Request Body [Required]

name (string): [optional] The name of the campaign. This name must be unique to the Amazon Advertising account to which the campaign is associated. Maximum length of the string is 128 characters.

state (State > string): [optional] Enum: [enabled, paused, archived]

portfolio_id (int) [optional] The identifier of the portfolio to which the campaign is associated.

budget (float): [required] | **budgetType** (BudgetType > String) : Enum [DAILY] | **budget** (float) : The budget amount associated with the campaign.

targetingType (Targeting > string) : [optional] Enum : [AUTO, MANUAL]

dynamicBidding ({}) [optional] | **placementBidding** (string) : You can enable controls to adjust your bid based on the placement location. Specify a location where you want to use bid controls. | **strategy** (BiddingStrategy > String) [required] : Enum [LEGACY_FOR_SALES, AUTO_FOR_SALES, MANUAL, RULE_BASED]

campaignId (string) : the entity object identifier

startDate [optional] (string) : A starting date for the campaign to go live. The format of the date is YYYYMMDD.

endDate [optional] (string) : An ending date for the campaign to stop running. The format of the date is YYYYMMDD.

tags ({string}) : A list of advertiser-specified custom identifiers for the campaign.

Returns

ApiResponse

`list_campaigns`(*body: dict, str, list*) → ApiResponse

Request Body (optional)

[Include the body for specific filtering, or leave empty to get all campaigns.]

state_filter (State): The returned array is filtered to include only campaigns with state set to one of the values in the specified comma-delimited list. Defaults to *enabled* and *paused*. Note that Campaigns rejected during moderation have state set to *archived*. Available values : enabled, paused, archived[optional]

nameFilter (str): The returned array includes only campaigns with the specified name.. [optional] | queryTermMatchType (MatchType > string) Enum : [BROAD_MATCH, EXACT_MATCH]

portfolio_id_filter (str): The returned array includes only campaigns associated with Portfolio identifiers matching those specified in the comma-delimited string.. [optional]

campaign_id_filter (str): The returned array includes only campaigns with identifiers matching those specified in the comma-delimited string.. [optional]

ad_format_filter (AdFormat): The returned array includes only campaigns with ad format matching those specified in the comma-delimited adFormats. Returns all campaigns if not specified. Available values : productCollection, video[optional]

max_results (int) Number of records to include in the paginated response. Defaults to max page size for given API. Minimum 10 and a Maximum of 100 [optional]

next_token (string) Token value allowing to navigate to the next response page. [optional]

includeExtendedDataFields (boolean) Setting to true will slow down performance because the API needs to retrieve extra information for each campaign. [optional]

Returns

ApiResponse

`delete_campaigns`(*body: dict, str, list*) → ApiResponse

Deletes Sponsored Products campaigns.
Request body (required)

campaignIdFilter (ObjectIdFilter): The identifier of an existing campaign. [required] | **include** (list>str): Entity object identifier. [required] minItems: 0 maxItems: 1000

Returns

ApiResponse

6.2.2 Ad Groups

`class` ad_api.api.sp.**AdGroupsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Version 3 of Sponsored Products API

> **Warning:** This replaces the version 2 of Ad Groups

`create_ad_groups`(**args*, ***kwargs*)

Creates one or more ad groups.
Request body: (required) An array of ad groups

name (*string*) : A name for the ad group

campaignId: (*string*) : An existing campaign to which the ad group is associated.

defaultBid: (*float*) A bid value for use when no bid is specified for keywords in the ad group

state: *string* A name for the ad group, Enum: [enabled, paused, archived]

Returns:

ApiResponse

`edit_ad_groups`(**args*, ***kwargs*)

Creates one or more ad groups.
Request body: (required) An array of ad groups

name (*string*) : A name for the ad group

adGroupId (*string*) : The identifier of the ad group.

defaultBid: (*float*) A bid value for use when no bid is specified for keywords in the ad group

state: *string* A name for the ad group, Enum: [enabled, paused, archived]

Returns:

ApiResponse

`delete_ad_groups`(**args*, ***kwargs*)

Deletes Sponsored Products ad groups by changing it's state to "ARCHIVED".

Request body (required)

adGroupIdFilter (ObjectIdFilter): The identifier of an existing ad group. [required] | **include** (list>str): Entity object identifier. [required] minItems: 0 maxItems: 1000

Returns

ApiResponse

`list_ad_groups`(**args*, ***kwargs*)

Lists Sponsored Products campaigns.

Request Body (optional)

[Include the body for specific filtering, or leave empty to get all ad groups.]

state_filter (State): The returned array is filtered to include only campaigns with state set to one of the values in the specified comma-delimited list. Defaults to *enabled* and *paused*. Note that Campaigns rejected during moderation have state set to *archived*. Available values : enabled, paused, archived[optional]

nameFilter (str): The returned array includes only campaigns with the specified name.. [optional] | queryTermMatchType (MatchType > string) Enum : [BROAD_MATCH, EXACT_MATCH]

portfolio_id_filter (str): The returned array includes only campaigns associated with Portfolio identifiers matching those specified in the comma-delimited string.. [optional]

campaign_id_filter (str): The returned array includes only campaigns with identifiers matching those specified in the comma-delimited string.. [optional]

includeExtendedDataFields (boolean) Whether to get entity with extended data fields such as creationDate, lastUpdateDate, servingStatus [optional]

max_results (int) Number of records to include in the paginated response. Defaults to max page size for given API. Minimum 10 and a Maximum of 100 [optional]

next_token (string) Token value allowing to navigate to the next response page. [optional]

campaignTargetingTypeFilter (TargetingType > String) Enum : [AUTO, MANUAL]

Returns

ApiResponse

6.2.3 Bid Recommendations

> **Warning:** This replaces the version 2 of SP BidRecommendations

`class` ad_api.api.sp.`BidRecommendationsV3`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

`get_bid_recommendations`(***kwargs*)

Gets theme-based bid recommendations for new or existing ad groups.

Request Body (oneOf)

AdGroupThemeBasedBidRecommendationRequest {

targetingExpressions* TargetingExpressionList [

example: [OrderedMap {'type': 'CLOSE_MATCH'}, OrderedMap {'type': 'LOOSE_MATCH'}, OrderedMap {'type': 'SUBSTITUTES'}, OrderedMap {'type': 'COMPLEMENTS'}]

maxItems: 100. The list of targeting expressions. Maximum of 100 per request, use pagination for more if needed.

TargetingExpression {

The targeting expression. The *type* property specifies the targeting option. Use *CLOSE_MATCH* to match your auto targeting ads closely to the specified value. Use *LOOSE_MATCH* to match your auto targeting ads broadly to the specified value. Use *SUBSTITUTES* to display your auto targeting ads along with substitutable products. Use *COMPLEMENTS* to display your auto targeting ads along with affiliated products. Use *KEYWORD_BROAD_MATCH* to broadly match your keyword targeting ads with search queries. Use *KEYWORD_EXACT_MATCH* to exactly match your keyword targeting ads with search queries. Use *KEYWORD_PHRASE_MATCH* to match your keyword targeting ads with search phrases.

example: {'type': 'CLOSE_MATCH'}

type*(string) Enum: ['CLOSE_MATCH', 'LOOSE_MATCH', 'SUBSTITUTES', 'COMPLEMENTS', 'KEYWORD_BROAD_MATCH', 'KEYWORD_EXACT_MATCH', 'KEYWORD_PHRASE_MATCH']

value(string): The targeting expression value.

}

]

campaignId*(string): The campaign identifier.

recommendationType*(string): The bid recommendation type. Enum: ['BIDS_FOR_EXISTING_AD_GROUP']

adGroupId*(string): The ad group identifier.

}

AsinsThemeBasedBidRecommendationRequest {

asins*(array): maxItems: 50. The list of ad ASINs in the ad group.

targetingExpressions* TargetingExpressionList [

example: [OrderedMap {'type': 'CLOSE_MATCH'}, OrderedMap {'type': 'LOOSE_MATCH'}, OrderedMap {'type': 'SUBSTITUTES'}, OrderedMap {'type': 'COMPLEMENTS'}]

maxItems: 100. The list of targeting expressions. Maximum of 100 per request, use pagination for more if needed.

TargetingExpression {

The targeting expression. The *type* property specifies the targeting option. Use *CLOSE_MATCH* to match your auto targeting ads closely to the specified value. Use *LOOSE_MATCH* to match your auto targeting ads broadly to the specified value. Use *SUBSTITUTES* to display your auto targeting ads along with substitutable products. Use *COMPLEMENTS* to display your auto targeting ads along with affiliated products. Use *KEYWORD_BROAD_MATCH* to broadly match your keyword targeting ads with search queries. Use *KEYWORD_EXACT_MATCH* to exactly match your keyword targeting ads with search queries. Use *KEYWORD_PHRASE_MATCH* to match your keyword targeting ads with search phrases.

example: {'type': 'CLOSE_MATCH'}

type*(string) Enum: ['CLOSE_MATCH', 'LOOSE_MATCH', 'SUBSTITUTES', 'COMPLEMENTS', 'KEYWORD_BROAD_MATCH', 'KEYWORD_EXACT_MATCH', 'KEYWORD_PHRASE_MATCH']

value(string): The targeting expression value.
}
]
bidding* Bidding control configuration for the campaign.
adjustments (array) maxItems: 2. Placement adjustment configuration for the campaign.
PlacementAdjustment {
Specifies bid adjustments based on the placement location. Use *PLACEMENT_TOP* for the top of the search page. Use *PLACEMENT_PRODUCT_PAGE* for a product page.
example: {'predicate': 'PLACEMENT_TOP', 'percentage': '100'}
predicate (string) Enum: ['PLACEMENT_TOP', 'PLACEMENT_PRODUCT_PAGE']
percentage (integer) maximum: 900 minimum: 0
}
strategy* BiddingStrategy (string): The bidding strategy selected for the campaign. Use *LEGACY_FOR_SALES* to lower your bid in real time when your ad may be less likely to convert to a sale. Use *AUTO_FOR_SALES* to increase your bid in real time when your ad may be more likely to convert to a sale or lower your bid when less likely to convert to a sale. Use *MANUAL* to use your exact bid along with any manual adjustments. Enum: ['LEGACY_FOR_SALES', 'AUTO_FOR_SALES', 'MANUAL', 'RULE_BASED']
recommendationType*(string): The bid recommendation type. Enum: ['BIDS_FOR_EXISTING_AD_GROUP']
}

Returns
ApiResponse

6.2.4 Product Ads

`class` ad_api.api.sp.**ProductAdsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Version 3 of the Sponsored Products API

> **Warning:** This replaces the version 2 of Product Ads

list_product_ads(**args*, ***kwargs*)

Lists Sponsored Products Ads.
Request Body (optional): Include the body for specific filtering, or leave empty to get all ad groups.

startIndex:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0
stateFilter:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.
campaignIdFilter:*string* | Optional. A comma-delimited list of campaign identifiers.

adGroupIdFilter:*string* | Optional. Restrict results to keywords associated with ad groups specified by identifier in the comma-delimited list.

adIdFilter:*string* | Optional. Restrict results to product ads associated with the product ad identifiers specified in the comma-delimited list. | **next_token** (string) Token value allowing to navigate to the next response page. [optional]

max_results (int) Number of records to include in the paginated response. Defaults to max page size for given API. Minimum 10 and a Maximum of 100 [optional]

includeExtendedDataFields (boolean) Whether to get entity with extended data fields such as creationDate, lastUpdateDate, servingStatus [optional]

Returns:

ApiResponse

`create_product_ads`(**args*, ***kwargs*)

Create one or more SP Product Ads.

Request body: (Required) A list of product ads for creation. Note that the SKU field is used by sellers and the ASIN field is used by vendors.

'**campaignId**': (*string*) The campaign identifier.

customText (*string*) : The custom text to use for creating a custom text ad for the associated ASIN. Defined only for KDP Authors and Book Vendors in US marketplace.

'**adGroupId**': (*string*) The ad group identifier.

'**sku**': (*string*) The SKU associated with the product. Defined for seller accounts only.

'**asin**': (*string*) The ASIN associated with the product. Defined for vendors only.

'**state**': (*string*) The current resource state. 'Enum':[enabled, paused, archived]

Prefer (header) : You can use the prefer header to specify whether you want the full object representation as part of the response. If you don't specify the prefer header, you will see just the ID of the created entity in the response.

Returns:

ApiResponse

`edit_product_ads`(**args*, ***kwargs*)

Updates one or more product ads specified by identifier.

Request Body: (REQUIRED) A list of product ad objects with updated values for the state field.

adId: *string* The identifier of an existing product ad to update.

state: *string* The current resource state.', 'Enum': [enabled, paused, archived]

Prefer (header) : You can use the prefer header to specify whether you want the full object representation as part of the response. If you don't specify the prefer header, you will see just the ID of the created entity in the response.

Returns:

ApiResponse

`delete_product_ads`(**args*, ***kwargs*)

Delete one or multiple sponsored product ads.

Request Body (required) : a dictionary to filter with ad ids that should be deleted. Sets the state of a specified product ad to archived

adIdFilter ({}) | include [string] : string list of ad object identifier.

Returns

ApiResponse

6.2.5 Budget Rules

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**BudgetRules**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
GET	/sp/campaigns/{campaignId}/budgetRules/budgetHistory	Gets the budget history for a campaign specified by identifier.
POST	/sp/budgetRules	Creates one or more budget rules.
GET	/sp/budgetRules	Get all budget rules created by an advertiser
PUT	/sp/budgetRules	Updates one or more budget rules.
GET	/sp/budgetRules/{budgetRuleId}	Gets a budget rule specified by identifier.
GET	/sp/budgetRules/{budgetRuleId}/campaigns	Gets all the campaigns associated with a budget rule
POST	/sp/campaigns/{campaignId}/budgetRules	Associates one or more budget rules to a campaign specified by identifer.
GET	/sp/campaigns/{campaignId}/budgetRules	Gets a list of budget rules associated to a campaign specified by identifier.
DELETE	/sp/campaigns/{campaignId}/budgetRules/{budgetRuleId}	Disassociates a budget rule specified by identifier from a campaign specified by identifier.

get_budget_history(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Gets the budget history for a campaign specified by identifier.
Param:

path **campaignId*** (number). The campaign identifier.

query **nextToken** (string). To retrieve the next page of results, call the same operation and specify this token in the request. If the nextToken field is empty, there are no further results.

query **pageSize*** (number). Sets a limit on the number of results returned. Maximum limit of pageSize is 30.

query **startDate*** (string). The start date of the budget history in YYYYMMDD format.

query **endDate*** (string). The end date of the budget history in YYYYMMDD format.

Returns:
ApiResponse

create_budget_rules(*self*, ***kwargs*) → ApiResponse:

Creates one or more budget rules.
Request Body

CreateSPBudgetRulesRequest {
budgetRulesDetails (array) [
maxItems: 25.

A list of budget rule details.
SPBudgetRuleDetails {
Object representing details of a budget rule for SP campaign
duration RuleDuration {
eventTypeRuleDuration EventTypeRuleDuration {
Object representing event type rule duration.
eventId*(string): The event identifier. This value is available from the budget rules recommendation API.
endDate(string): The event end date in YYYYMMDD format. Read-only.
eventName(string): The event name. Read-only.
startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.
}
dateRangeTypeRuleDuration DateRangeTypeRuleDuration {
Object representing date range type rule duration.
endDate(string): The end date of the budget rule in YYYYMMDD format. The end date is inclusive. Required to be equal or greater than *startDate*.
startDate*(string): The start date of the budget rule in YYYYMMDD format. The start date is inclusive. Required to be greater than or equal to current date.
eventName(string): The event name. Read-only.
startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.
}
}
recurrence Recurrence {
type (string): The frequency of the rule application. Enum: [‘DAILY’]
daysOfWeek (array). Object representing days of the week for weekly type rule. It is not required for daily recurrence type
DayOfWeek [
DayOfWeek(string): The day of the week. Enum: [‘MONDAY’, ‘TUESDAY’, ‘WEDNESDAY’, ‘THURSDAY’, ‘FRIDAY’, ‘SATURDAY’, ‘SUNDAY’]
]
}
ruleType* SPRuleType (string): The type of budget rule. SCHEDULE: A budget rule based on a start and end date. PERFORMANCE: A budget rule based on advertising performance criteria. Enum: [‘SCHEDULE’, ‘PERFORMANCE’]
budgetIncreaseBy budgetIncreaseBy {
type* BudgetChangeType (string): The value by which to update the budget of the budget rule. Enum: [‘PERCENT’]
value* number($double): The budget value.
}
name (string): The budget rule name. Required to be unique within a campaign. maxLength: 355
performanceMeasureCondition* PerformanceMeasureCondition {
metricName* PerformanceMetrics (string): The advertising performance metric. Enum: [ACOS, CTR, CVR, ROAS]
comparisonOperator* ComparisonOperator (string): The comparison operator. Enum: [GREATER_THAN, LESS_THAN, EQUAL_TO, LESS_THAN_OR_EQUAL_TO, GREATER_THAN_OR_EQUAL_TO]
threshold* number($double): The performance threshold value.

}
]
}
Returns:
ApiResponse

`list_budget_rules`(*self*, ***kwargs*) → ApiResponse:

Get all budget rules created by an advertiser
Param:

query **nextToken** (string). To retrieve the next page of results, call the same operation and specify this token in the request. If the nextToken field is empty, there are no further results.
query **pageSize*** (number). Sets a limit on the number of results returned. Maximum limit of pageSize is 30.
Returns:
ApiResponse

`edit_budget_rules`(*self*, ***kwargs*) → ApiResponse:

Updates one or more budget rules.
Request Body

UpdateSPBudgetRulesRequest {
budgetRulesDetails (array) [
maxItems: 25.
A list of budget rule details.
SPBudgetRule {
ruleState state (string): The budget rule state. Enum: [‘ACTIVE’, ‘PAUSED’]
lastUpdatedDate number($int64): Epoch time of budget rule update. Read-only.
createdDate number($int64): Epoch time of budget rule creation. Read-only.
ruleDetails SPBudgetRuleDetails {
Object representing details of a budget rule for SP campaign
duration RuleDuration {
eventTypeRuleDuration EventTypeRuleDuration {
Object representing event type rule duration.
eventId*(string): The event identifier. This value is available from the budget rules recommendation API.
endDate(string): The event end date in YYYYMMDD format. Read-only.
eventName(string): The event name. Read-only.
startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.
}
dateRangeTypeRuleDuration DateRangeTypeRuleDuration {
Object representing date range type rule duration.
endDate(string): The end date of the budget rule in YYYYMMDD format. The end date is inclusive. Required to be equal or greater than *startDate*.
startDate*(string): The start date of the budget rule in YYYYMMDD format. The start date is inclusive. Required to be

greater than or equal to current date.
eventName(string): The event name. Read-only.
startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.
}
}
recurrence Recurrence {
type (string): The frequency of the rule application. Enum: ['DAILY']
daysOfWeek (array). Object representing days of the week for weekly type rule. It is not required for daily recurrence type
DayOfWeek [
DayOfWeek(string): The day of the week. Enum: ['MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY', 'SUNDAY']
]
}
ruleType* SPRuleType (string): The type of budget rule. SCHEDULE: A budget rule based on a start and end date. PERFORMANCE: A budget rule based on advertising performance criteria. Enum: ['SCHEDULE', 'PERFORMANCE']
budgetIncreaseBy budgetIncreaseBy {
type* BudgetChangeType (string): The value by which to update the budget of the budget rule. Enum: ['PERCENT']
value* number($double): The budget value.
}
name (string): The budget rule name. Required to be unique within a campaign. maxLength: 355
performanceMeasureCondition* PerformanceMeasureCondition {
metricName* PerformanceMetrics (string): The advertising performance metric. Enum: [ACOS, CTR, CVR, ROAS]
comparisonOperator* ComparisonOperator (string): The comparison operator. Enum: [GREATER_THAN, LESS_THAN, EQUAL_TO, LESS_THAN_OR_EQUAL_TO, GREATER_THAN_OR_EQUAL_TO]
threshold* number($double): The performance threshold value.
}
ruleId* (string): The budget rule identifier.
ruleStatus (string): he budget rule status. Read-only.
}
]
}
Returns:
ApiResponse

`get_budget_rule`(*self*, *budgetRuleId*, ***kwargs*) → ApiResponse:

Gets a budget rule specified by identifier.
Param:

path **budgetRuleId*** (string). The budget rule identifier.
Returns:

ApiResponse

`get_campaigns_budget_rule`(*self*, *budgetRuleId*, ***kwargs*) → ApiResponse:

Gets all the campaigns associated with a budget rule
Param:

path **budgetRuleId*** (string). The budget rule identifier.
query **nextToken** (string). To retrieve the next page of results, call the same operation and specify this token in the request. If the nextToken field is empty, there are no further results.
query **pageSize*** (number). Sets a limit on the number of results returned. Maximum limit of pageSize is 30.

Returns:
ApiResponse

`create_campaign_budget_rules`(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Associates one or more budget rules to a campaign specified by identifer.
Param:

path **campaignId*** (number). The campaign identifier.

Returns:
ApiResponse

`get_budget_rules_campaign`(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Gets a list of budget rules associated to a campaign specified by identifier.
Param:

path **campaignId*** (number). The campaign identifier.

Returns:
ApiResponse

`delete_budget_rule_campaign`(*self*, *campaignId*, *budgetRuleId*, ***kwargs*) → ApiResponse:

Disassociates a budget rule specified by identifier from a campaign specified by identifier.
Param:

path **campaignId*** (number). The campaign identifier.
path **budgetRuleId*** (string). The budget rule identifier.

Returns:
ApiResponse

6.2.6 Campaign Optimization Rules

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

Note: This API requires separated access from Amazon Advertising Support

```
{
'code': 403,
'message': 'Client does not have access to this API',
'responseBody': None,
'responseHeaders': None
}
```

class ad_api.api.sp.**CampaignOptimization**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Specification: https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/SponsoredProducts_prod_3p.json

Endpoints available

Method	Endpoint	Description
POST	/sp/rules/campaignOptimization/eligibility	Gets a campaign optimization rule recommendation for SP campaigns.
GET	/sp/rules/campaignOptimization/{campaignOptimizationId}	Gets a campaign optimization rule specified by identifier.
DELETE	/sp/rules/campaignOptimization/{campaignOptimizationId}	Deletes a campaign optimization rule specified by identifier.
POST	/sp/rules/campaignOptimization	Creates a campaign optimization rule.
PUT	/sp/rules/campaignOptimization	Updates a campaign optimization rule.
POST	/sp/rules/campaignOptimization/state	Gets campaign optimization rule state. Recommended refresh frequency is once a day.

list_campaigns_optimization_eligibility(*self*, ***kwargs*) → ApiResponse:

Gets a campaign optimization rule recommendation for SP campaigns.
Request Body

SPCampaignOptimizationRecommendationsAPIRequest {
campaignIds* (array) maxItems: 100. A list of campaign ids [
RuleCampaignId (string): campaignId
]
}

```
{
  "campaignIds": [
    "string"
  ]
}
```

get_budget_campaign_optimization(*self*, *campaignOptimizationId*, ***kwargs*) → ApiResponse:

Gets a campaign optimization rule specified by identifier.
Param:

path **campaignOptimizationId*** (string). The sp campaign optimization rule identifier.
Returns:
ApiResponse

delete_budget_campaign_optimization(*self*, *campaignOptimizationId*, ***kwargs*) → ApiResponse:

Deletes a campaign optimization rule specified by identifier.
Param:

path **campaignOptimizationId*** (string). The sp campaign optimization rule identifier.

Returns:

ApiResponse

create_budget_campaign_optimization(*self*, ***kwargs*) → ApiResponse:

Creates a campaign optimization rule.
Request Body

CreateSPCampaignOptimizationRulesRequest {
recurrence* RecurrenceType (string): The frequency of the rule application. Enum: [‘DAILY’]
ruleAction* RuleAction (string): The action taken when the campaign optimization rule is enabled. Defaults to adopt Enum: [‘ADOPT’]
ruleCondition* RuleConditionList [
maxItems: 3
RuleCondition {
metricName* RuleConditionMetric (string): The advertising performance metric. ROAS is the only supported metric. Enum: [‘ROAS’, ‘AVERAGE_BID’]
comparisonOperator* ComparisonOperator (string): The comparison operator. Enum: [‘GREATER_THAN’, ‘LESS_THAN’, ‘EQUAL_TO’, ‘LESS_THAN_OR_EQUAL_TO’, ‘GREATER_THAN_OR_EQUAL_TO’]
threshold* number (double): The performance threshold value.
}
]
ruleType* RuleType (string): The type of the campaign optimization rule. Only Support BID as of now Enum: [‘BID’, ‘KEYWORD’, ‘PRODUCT’]
ruleName RuleName (string): The campaign optimization rule name. maxLength: 355
campaignIds* (array) maxItems: 20. A list of campaign ids [
RuleCampaignId (string): campaignId
]

Returns:

ApiResponse

```
{
  "recurrence": "DAILY",
  "ruleAction": "ADOPT",
  "ruleCondition": [
    {
      "metricName": "ROAS",
      "comparisonOperator": "GREATER_THAN",
      "threshold": "4"
    }
  ],
  "ruleType": "BID",
  "ruleName": "RuleROAS4",
  "campaignIds": [
    "123784",
```

(continues on next page)

(continued from previous page)

```
    "1223785"
  ]
}
```

edit_budget_campaign_optimization(*self*, ***kwargs*) → ApiResponse:

Updates a campaign optimization rule.
Request Body

UpdateSPCampaignOptimizationRulesRequest {
recurrence* RecurrenceType (string): The frequency of the rule application. Enum: [‘DAILY’]
ruleAction* RuleAction (string): The action taken when the campaign optimization rule is enabled. Defaults to adopt Enum: [‘ADOPT’]
campaignOptimizationId* campaignOptimizationId (string): The persistent rule identifier. maxLength: 355
ruleCondition* RuleConditionList [
maxItems: 3
RuleCondition {
metricName* RuleConditionMetric (string): The advertising performance metric. ROAS is the only supported metric. Enum: [‘ROAS’, ‘AVERAGE_BID’]
comparisonOperator* ComparisonOperator (string): The comparison operator. Enum: [‘GREATER_THAN’, ‘LESS_THAN’, ‘EQUAL_TO’, ‘LESS_THAN_OR_EQUAL_TO’, ‘GREATER_THAN_OR_EQUAL_TO’]
threshold* number (double): The performance threshold value.
}
]
ruleType* RuleType (string): The type of the campaign optimization rule. Only Support BID as of now Enum: [‘BID’, ‘KEYWORD’, ‘PRODUCT’]
ruleName RuleName (string): The campaign optimization rule name. maxLength: 355
campaignIds* (array) maxItems: 20. A list of campaign ids [
RuleCampaignId (string): campaignId
]

Returns:
ApiResponse

```
{
  "recurrence": "DAILY",
  "ruleAction": "ADOPT",
  "campaignOptimizationId": "10001",
  "ruleCondition": [
    {
      "metricName": "ROAS",
      "comparisonOperator": "GREATER_THAN",
      "threshold": "7"
    }
  ],
  "ruleType": "BID",
  "ruleName": "RuleROAS4",
  "campaignIds": [
    "123784",
```

(continues on next page)

(continued from previous page)

```
    "1223785"
  ]
}
```

get_state_budget_campaign_optimization(*self*, ***kwargs*) → ApiResponse:

Gets campaign optimization rule state. Recommended refresh frequency is once a day.
Request Body

SPCampaignOptimizationNotificationAPIRequest {
campaignIds* (array) maxItems: 100. A list of campaign ids [
RuleCampaignId (string): campaignId
]
}
Returns:
ApiResponse

6.2.7 Campaign Consolidated Recommendations

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

Note: This API is not clear which marketplaces are supported

```
{
"code":400,
"responseHeaders":null,
"responseBody":null,
"message":"Marketplace is not supported"
}
```

class ad_api.api.sp.**CampaignsRecommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
GET	/sp/campaign/recommendations	Gets the top consolidated recommendations.

list_campaigns_recommendations(*self*, ***kwargs*) → ApiResponse:

Gets the top consolidated recommendations across bid, budget, targeting for SP campaigns given an advertiser profile id. The recommendations are refreshed everyday.

Param:

query **nextToken** (string). Optional. Token to retrieve subsequent page of results.

query **maxResults** (string). Optional. Limits the number of items to return in the response.

Returns:

ApiResponse

6.2.8 Get Ranked Keywords Recommendations

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**RankedKeywordsRecommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Specification: https://dtrnk0o2zy01c.cloudfront.net/openapi/en-us/dest/SponsoredProducts_prod_3p.json

Endpoint available

Method	Endpoint	Description
POST	/sp/targets/keywords/recommendations	Get ranked keywords recommendations.

list_ranked_keywords_recommendations(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

Get keyword recommendations

The POST /sp/targets/keywords/recommendations endpoint returns recommended keyword targets given either A) a list of ad ASINs or B) a campaign ID and ad group ID. Please use the recommendationType field to specify if you want to use option A or option B. This endpoint will also return recommended bids along with each recommendation keyword target.

Ranking The keyword recommendations will be ranked in descending order of clicks or impressions, depending on the sortDimension field provided by the user. You may also input your own keyword targets to be ranked alongside the keyword recommendations by using the targets array.

Localization Use the locale field to get keywords in your specified locale. Supported marketplace to locale mappings can be found at the POST /keywords/localize endpoint.

Version 5.0

New Features

Version 5.0 utilizes the new theme-based bid recommendations, which can be retrieved at the endpoint /sp/targets/bid/recommendations, to return improved bid recommendations for each keyword. Theme-based bid recommendations provide "themes" and "impact metrics" along with each bid suggestion to help you choose the right bid for your keyword target.

Themes

We now may return multiple bid suggestions for each keyword target. Each suggestion will have a theme to express the business objective of the bid. Available themes are:

- CONVERSION_OPPORTUNITIES - The default theme which aims to maximize number of conversions.
- SPECIAL_DAYS - A theme available during high sales events such as Prime Day, to anticipate an increase in sales and competition.

Impact Metrics

We have added impact metrics which provide insight on the number of clicks and conversions you will receive for targeting a keyword at a certain bid.

Bidding Strategy

You may now specify your bidding strategy in the KEYWORDS_BY_ASINS request to get bid suggestions tailored to your bidding strategy. For KEYWORDS_BY_ADGROUP requests, you will not specify a bidding strategy, because the bidding strategy of the ad group is used. The three bidding strategies are:

- LEGACY_FOR_SALES - Dynamic bids (down only)
- AUTO_FOR_SALES - Dynamic bids (up and down)
- MANUAL - Fixed bids

Availability

Version 5.0 is only available in the following marketplaces: US, CA, UK, DE, FR, ES, IN, JP.

Request Body (oneOf)

This request type is used to retrieve recommended keyword targets for an existing ad group. Set the recommendationType to KEYWORD_FOR_ADGROUP to use this request type.

RankedKeywordTargetsForAdGroupRequest {

maxRecommendations (number): The max size of recommended target. Set it to 0 if you only want to rank user-defined keywords. default: 200 maximum: 200 minimum: 0

sortDimension (string): The ranking metric value. Supported values: CLICKS, CONVERSIONS, DEFAULT. DEFAULT will be applied if no value passed in. Enum: [‘CLICKS’, ‘CONVERSIONS’, ‘DEFAULT’]

locale (string): Translations are for readability and do not affect the targeting of ads. Supported marketplace to locale mappings can be found at the POST /keywords/localize endpoint. Note: Translations will be null if locale is unsupported. Enum: [‘ar_EG’, ‘de_DE’, ‘en_AE’, ‘en_AU’, ‘en_CA’, ‘en_GB’, ‘en_IN’, ‘en_SA’, ‘en_SG’, ‘en_US’, ‘es_ES’, ‘es_MX’, ‘fr_FR’, ‘it_IT’, ‘ja_JP’, ‘nl_NL’, ‘pl_PL’, ‘pt_BR’, ‘sv_SE’, ‘tr_TR’, ‘zh_CN’]

targets [

minItems: 0. maxItems: 100. A list of targets that need to be ranked.

{

matchType (string): Keyword match type. The default value will be BROAD. Enum: [‘BROAD’, ‘EXACT’, ‘PHRASE’]

keyword (string): The keyword value

bid number (double): The bid value for the keyword. The default value will be the suggested bid.

userSelectedKeyword (boolean): Flag that tells if keyword was selected by the user or was recommended by KRS

}

]

campaignId*(string): The identifier of the campaign

recommendationType*(string): The recommendationType to retrieve recommended keyword

targets for an existing ad group. Enum: [‘KEYWORDS_FOR_ADGROUP’]
bidsEnabled(boolean): Set this parameter to false if you do not want to retrieve bid suggestions for your keyword targets. Defaults to true. default: true
dGroupId*(string): The identifier of the ad group
}

This request type is used to retrieve recommended keyword targets for ASINs. Set the recommendationType to KEYWORD_FOR_ASINS to use this request type.

RankedKeywordTargetsForAsinsRequest {
maxRecommendations (number): The max size of recommended target. Set it to 0 if you only want to rank user-defined keywords. default: 200 maximum: 200 minimum: 0
sortDimension (string): The ranking metric value. Supported values: CLICKS, CONVERSIONS, DEFAULT. DEFAULT will be applied if no value passed in. Enum: [‘CLICKS’, ‘CONVERSIONS’, ‘DEFAULT’]
locale (string): Translations are for readability and do not affect the targeting of ads. Supported marketplace to locale mappings can be found at the POST /keywords/localize endpoint. Note: Translations will be null if locale is unsupported. Enum: [‘ar_EG’, ‘de_DE’, ‘en_AE’, ‘en_AU’, ‘en_CA’, ‘en_GB’, ‘en_IN’, ‘en_SA’, ‘en_SG’, ‘en_US’, ‘es_ES’, ‘es_MX’, ‘fr_FR’, ‘it_IT’, ‘ja_JP’, ‘nl_NL’, ‘pl_PL’, ‘pt_BR’, ‘sv_SE’, ‘tr_TR’, ‘zh_CN’]
targets [
minItems: 0. maxItems: 100. A list of targets that need to be ranked.
{
matchType (string): Keyword match type. The default value will be BROAD. Enum: [‘BROAD’, ‘EXACT’, ‘PHRASE’]
keyword (string): The keyword value
bid number (double): The bid value for the keyword. The default value will be the suggested bid.
userSelectedKeyword (boolean): Flag that tells if keyword was selected by the user or was recommended by KRS
}
asins* (array) maxItems: 50. An array list of Asin.
biddingStrategy (string) The bid recommendations returned will depend on the bidding strategy. LEGACY_FOR_SALES - Dynamic Bids (Down only) AUTO_FOR_SALES - Dynamic Bids (Up or down) MANUAL - Fixed Bids. Enum: [LEGACY_FOR_SALES, AUTO_FOR_SALES, MANUAL, RULE_BASED] default: LEGACY_FOR_SALES
recommendationType* (string): The recommendationType to retrieve recommended keyword targets for a list of ASINs. Enum: [‘KEYWORDS_FOR_ASINS’]
bidsEnabled(boolean): Set this parameter to false if you do not want to retrieve bid suggestions for your keyword targets. Defaults to true. default: true
}

Version 4.0

New Features

Version 4.0 allows users to retrieve recommended keyword targets which are sorted in descending order of clicks or conversions. The default sort dimension, if not specified, ranks recommendations by our

interal ranking mechanism. We have also have added search term metrics. **Search term impression share** indicates the percentage share of all ad-attributed impressions you received on that keyword in the last 30 days. This metric helps advertisers identify potential opportunities based on their share on relevant keywords. **Search term impression rank** indicates your ranking among all advertisers for the keyword by ad impressions in a marketplace. It tells an advertiser how many advertisers had higher share of ad impressions. Search term information is only available for keywords the advertiser targeted with ad impressions.

Availability

Version 4.0 is available in all marketplaces.
Request Body (oneOf)

This request type is used to retrieve recommended keyword targets for an existing ad group. Set the recommendationType to KEYWORD_FOR_ADGROUP to use this request type.

AdGroupKeywordTargetRankRecommendationRequest {
maxRecommendations (number): The max size of recommended target. Set it to 0 if you only want to rank user-defined keywords. default: 200 maximum: 200 minimum: 0
sortDimension (string): The ranking metric value. Supported values: CLICKS, CONVERSIONS, DEFAULT. DEFAULT will be applied if no value passed in. Enum: [‘CLICKS’, ‘CONVERSIONS’, ‘DEFAULT’]
locale (string): Translations are for readability and do not affect the targeting of ads. Supported marketplace to locale mappings can be found at the POST /keywords/localize endpoint. Note: Translations will be null if locale is unsupported. Enum: [‘ar_EG’, ‘de_DE’, ‘en_AE’, ‘en_AU’, ‘en_CA’, ‘en_GB’, ‘en_IN’, ‘en_SA’, ‘en_SG’, ‘en_US’, ‘es_ES’, ‘es_MX’, ‘fr_FR’, ‘it_IT’, ‘ja_JP’, ‘nl_NL’, ‘pl_PL’, ‘pt_BR’, ‘sv_SE’, ‘tr_TR’, ‘zh_CN’]
targets [
minItems: 0. maxItems: 100. A list of targets that need to be ranked.
{
matchType (string): Keyword match type. The default value will be BROAD. Enum: [‘BROAD’, ‘EXACT’, ‘PHRASE’]
keyword (string): The keyword value
bid number (double): The bid value for the keyword. The default value will be the suggested bid.
userSelectedKeyword (boolean): Flag that tells if keyword was selected by the user or was recommended by KRS
}
]
campaignId*(string): The identifier of the campaign
recommendationType*(string): The recommendationType to retrieve recommended keyword targets for an existing ad group. Enum: [‘KEYWORDS_FOR_ADGROUP’]
dGroupId*(string): The identifier of the ad group
}

This request type is used to retrieve recommended keyword targets for ASINs. Set the recommendationType to KEYWORD_FOR_ASINS to use this request type.

AsinsKeywordTargetRankRecommendationRequest {

maxRecommendations (number): The max size of recommended target. Set it to 0 if you only want to rank user-defined keywords. default: 200 maximum: 200 minimum: 0

sortDimension (string): The ranking metric value. Supported values: CLICKS, CONVERSIONS, DEFAULT. DEFAULT will be applied if no value passed in. Enum: ['CLICKS', 'CONVERSIONS', 'DEFAULT']

locale (string): Translations are for readability and do not affect the targeting of ads. Supported marketplace to locale mappings can be found at the POST /keywords/localize endpoint. Note: Translations will be null if locale is unsupported. Enum: ['ar_EG', 'de_DE', 'en_AE', 'en_AU', 'en_CA', 'en_GB', 'en_IN', 'en_SA', 'en_SG', 'en_US', 'es_ES', 'es_MX', 'fr_FR', 'it_IT', 'ja_JP', 'nl_NL', 'pl_PL', 'pt_BR', 'sv_SE', 'tr_TR', 'zh_CN']

targets [

minItems: 0. maxItems: 100. A list of targets that need to be ranked.

{

matchType (string): Keyword match type. The default value will be BROAD. Enum: ['BROAD', 'EXACT', 'PHRASE']

keyword (string): The keyword value

bid number (double): The bid value for the keyword. The default value will be the suggested bid.

userSelectedKeyword (boolean): Flag that tells if keyword was selected by the user or was recommended by KRS

}

asins* (array) maxItems: 50. An array list of Asin.

recommendationType* (string): The recommendationType to retrieve recommended keyword targets for a list of ASINs. Enum: ['KEYWORDS_FOR_ASINS']

}

Example Using the endpoint with version

```
import logging
import json
from ad_api.api import sponsored_products
from ad_api.base import AdvertisingApiException

def sp_get_ranked_keywords_recommendations_asin(version:int):

    dictionary = \
        {
            "asins": [
                "B08C1KN5J2"
            ],
            "recommendationType": "KEYWORDS_FOR_ASINS"
        }

    data = json.dumps(dictionary)


    try:
```

(continues on next page)

(continued from previous page)

```
        result = sponsored_products.RankedKeywordsRecommendations(debug=True).list_
↪ranked_keywords_recommendations(
            version=version,
            body=data
        )

    except AdvertisingApiException as error:
        logging.error(error)


    logging.info("version {}".format(str(version)))
    logging.info(result.payload)

def sp_get_ranked_keywords_recommendations_targets(version:int):

    dictionary = \
        {
            "maxRecommendations": 10,
            "sortDimension": "CLICKS",
            "locale": "zh_CN",
            "campaignId": 199522597016061,
            "recommendationType": "KEYWORDS_FOR_ADGROUP",
            "adGroupId": 261060438275923
        }

    try:

        result = sponsored_products.RankedKeywordsRecommendations(debug=True).list_
↪ranked_keywords_recommendations(
            version=version,
            body=dictionary
        )

    except AdvertisingApiException as error:
        logging.error(error)

    logging.info("version {}".format(str(version)))
    logging.info(result.payload)


if __name__ == '__main__':

    # sp_get_ranked_keywords_recommendations_targets(version=3) # default api␣
↪endpoint version=3
    sp_get_ranked_keywords_recommendations_targets(version=4)
    sp_get_ranked_keywords_recommendations_targets(version=5)

    # sp_get_ranked_keywords_recommendations_asin(version=3) # default api endpoint␣
↪version=3
    sp_get_ranked_keywords_recommendations_asin(version=4)
    sp_get_ranked_keywords_recommendations_asin(version=5)
```

6.2.9 Keywords

Warning: This replaces the version 2 of SP Keywords

class ad_api.api.sp.**KeywordsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_keyword(***kwargs*)

Creating product keywords.
Request Body (required)

nativeLanguageKeyword : (*string*), The unlocalized keyword text in the preferred locale of the advertiser

nativeLanguageLocale : (*string*), The locale preference of the advertiser.

campaignId: *string*, The identifer of the campaign to which the keyword is associated.

adGroupId: *string*, The identifier of the ad group to which this keyword is associated

state: *string*, The current resource state.' , 'Enum': '[enabled]

keywordText: *string*, The text of the expression to match against a search query.

matchType: *string*, 'The type of match.' , 'Enum': '[EXACT, PHRASE, BROAD]

Returns
ApiResponse

delete_keywords(***kwargs*)

Deleting product keywords.
Request Body (required)

keywordIdFilter {} : Filter keywords by the list of objectIds include [string] : list of keywordsIds as String to be used as filter. MinItems : 0, MaxItems :1000

Returns
ApiResponse

edit_keyword(***kwargs*)

Updating product keywords.
Request Body (required)

'**keywordId**': *string*, (required) {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}

'**bid**': *float* {'description': 'Bid associated with this keyword. Applicable to biddable match types only.'}

Returns
ApiResponse

list_keywords(***kwargs*)

Listing product keywords.

Request Body (optional)
Returns
ApiResponse

6.2.10 Negative Keywords

Warning: This replaces the version 2 of Negative Keywords

class ad_api.api.sp.**NegativeKeywordsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_negative_keyword(***kwargs*)

Creating negative product keywords.
Request Body (required)

nativeLanguageKeyword : (*string*), The unlocalized keyword text in the preferred locale of the advertiser
nativeLanguageLocale : (*string*), The locale preference of the advertiser.
campaignId: *string*, The identifer of the campaign to which the keyword is associated.
adGroupId: *string*, The identifier of the ad group to which this keyword is associated
state: *string*, The current resource state.' , 'Enum': '[enabled]
keywordText: *string*, The text of the expression to match against a search query.
matchType: *string*, 'The type of match.' , 'Enum': '[NEGATIVE_EXACT, NEGATIVE_PHRASE, NEGATIVE_BROAD]

Returns
ApiResponse

delete_negative_keywords(***kwargs*)

Deleting negative product keywords.
Request Body (required)

keywordIdFilter {} : Filter negative keywords by the list of objectIds include [string] : list of negativeKeywordsIds as String to be used as filter. MinItems : 0, MaxItems :1000

Returns
ApiResponse

edit_negative_keyword(***kwargs*)

Updating negative product keywords.
Request Body (required) :

'**keywordId**': *string*, (required) {'description': 'The identifer of the campaign to which the keyword is associated.'}
'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}

Returns
ApiResponse

list_negative_keywords(***kwargs*)

Listing negative product keywords.

Request Body (optional)
Returns
ApiResponse

6.2.11 Negative Product Targeting

Warning: This replaces the version 2 of Negative Product Targeting

class ad_api.api.sp.**NegativeTargetsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_negative_product_targets(***kwargs*)

Creating negative product targets.
Request Body (required)

'**campaignId**': *number*, {'description': 'The identifier of the campaign to which this negative target is associated.'}
'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this negative target is associated.'}
'**state**': *number*, {'description': 'The current resource state. [enabled, paused, archived]'}
'**expression**' | '**value**': *string*, {'description': 'The expression value. '} | '**type**': *string*, {'description': The type of negative targeting expression. You can only specify values for the following predicates: 'Enum : [ASIN_BRAND_SAME_AS, ASIN_SAME_AS]'}

Returns
ApiResponse

delete_negative_product_targets(***kwargs*)

Deleting negative product targets.
Request Body (required)

keywordIdFilter {} : Filter negative targets by the list of objectIds include [string] : list of negativeTargetIds as String to be used as filter. MinItems : 0, MaxItems :1000

Returns
ApiResponse

edit_negative_product_targets(***kwargs*)

Updating negative product targets.
Request Body (required)

targetId : *string*, The target identifier
'**state**': *string*, The current resource state. [enabled, paused, archived]
'**expression**'
'**value**': *string*, The expression value.
'**type**': *string*, The type of nagative targeting expression. You can only specify values for the following predicates: Enum : [ASIN_BRAND_SAME_AS, ASIN_SAME_AS]

Returns
ApiResponse

list_negative_product_targets(***kwargs*)

Listing negative product targets.

Request Body (optional)
Returns
ApiResponse

list_negative_targets_brands_recommendations(***kwargs*)

Returns brands recommended for negative targeting. Only available for Sellers and Vendors. These recommendations include your own brands because targeting your own brands usually results in lower performance than targeting competitors' brands.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

list_negative_targets_brands_search(***kwargs*)

Returns brands related to keyword input for negative targeting.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Returns:

ApiResponse

6.2.12 Product Targeting

Warning: Sponsored Product v3.0 is not available for Sandbox endpoint

Note: This documentation API 3.0

Endpoints available version 3.0

Method	Endpoint	Description
POST	/sp/targets/list	Listing product targets.
POST	/sp/targets	Creating product targets.
PUT	/sp/targets	Creating product targets.
POST	/sp/targets/delete	Deleting product targets.
POST	/sp/targets/categories/recommendations	Returns a list of category recommendations for the input list of ASINs.
POST	/sp/targets/products/count	Get number of targetable asins based on refinements provided by the user.
GET	/sp/targets/categories	Returns all targetable categories.
GET	/sp/targets/category/{categoryId}/refinements	Returns refinements according to category input.

class ad_api.api.sp.**TargetsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_product_targets(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

Listing product targets.

Request Body (optional)

Returns

ApiResponse

create_product_targets(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Creating product targets.

Request Body (required)

'**campaignId**': *string*, {'description': 'The number or recommendations returned in a single page.'}

'**adGroupId**': *string*, {'description': 'The page number in the result set to return.'}

'**expression**' | '**value**': *string*, {'description': 'The expression value.'} | '**type**': *string*, {'description': '[queryBroadMatches, queryPhraseMatches, queryExactMatches, asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs, queryBroadRelMatches, queryHighRelMatches, asinSubstituteRelated, asinAccessoryRelated, asinAgeRangeSameAs, asinGenreSameAs, asinIsPrimeShippingEligible]'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}

'**expressionType**': *string*, {'description': '[auto, manual]'}

'**bid**': *float*, {'description': 'The bid for ads sourced using the target. Min / Max 0.02 / 1000'}

Returns

ApiResponse

`edit_product_targets`(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Updating product targets.

Request Body (required)

'**targetId**': *string*, (required) {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled, paused, archived]'}

'**bid**': *float* {'description': 'Bid associated with this keyword. Applicable to biddable match types only.'}

'**expression**'

'**value**': *string*, The expression value.

'**type**': *string*, The type of nagative targeting expression. You can only specify values for the following predicates: Enum : [ASIN_BRAND_SAME_AS, ASIN_SAME_AS]

'**expressionType**' Enum : [AUTO, MANUAL]

Returns

ApiResponse

`delete_product_targets`(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

Deleting product targets

Request Body (required)

targetIdFilter {} : Filter product targets by the list of objectIds include [string] : list of productTargetIds as String to be used as filter. MinItems : 0, MaxItems :1000

Returns

ApiResponse

`list_products_targets_categories_recommendations`(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Returns a list of category recommendations for the input list of ASINs. Use this API to discover relevant categories to target. To find ASINs, either use the Product Metadata API or browse the Amazon Retail Website.

header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

body: | REQUIRED {'description': 'An array of asins objects.}'
'**asins**': list>*string*, {'description': 'List of input ASINs. This API does not check if the ASINs are valid ASINs. maxItems: 10000.'}
'**includeAncestor**': *boolean*, {'description': 'Enable this if you would like to retrieve categories which are ancestor nodes of the original recommended categories. This may increase the number of categories returned, but decrease the relevancy of those categories.'}
Returns:
ApiResponse

`get_products_targets_count`(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Get number of targetable asins based on refinements provided by the user. Please use the GetTargetableCategories API or the GetCategoryRecommendationsForASINs API to retrieve the category ID. Please use the GetRefinementsByCategory API to retrieve refinements data for a category.
header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.
Returns:
ApiResponse

`list_targets_categories`(*self*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Returns all targetable categories. This API returns a large JSON string containing a tree of category nodes. Each category node has the fields - category id, category name, and child categories.
header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.
Returns:
ApiResponse

`list_products_targets_category_refinements`(*self*, *categoryId*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Get a targeting clause specified by identifier.
path **categoryId**:*string* | Required. The target identifier.
header **Prefer**:*string* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.
Returns:
ApiResponse

6.2.13 Product Recommendation Service

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

`class ad_api.api.sp.ProductRecommendations`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoint available

Method	Endpoint	Description
POST	/sp/targets/products/recommendations	Suggested target ASINs for your advertised product.

list_products_recommendations(*self*, ***kwargs*) → ApiResponse:

6.2.14 Campaign Negative Targeting Clauses

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**CampaignNegativeTargets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
DELETE	/sp/campaignNegativeTargets/delete	Deletes Campaign Negative Targeting Clauses
POST	/sp/campaignNegativeTargets	Create Campaign Negative Targeting Clauses
PUT	/sp/campaignNegativeTargets	Updates Campaign Negative Targeting Clauses
POST	/sp/campaignNegativeTargets/list	List Campaign Negative Targeting Clauses

delete_campaign_negative_targets(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

Deleting campaign negative targets.
Request Body (required)

SponsoredProductsDeleteSponsoredProductsCampaignNegativeTargetingClausesRequestContent
{
campaignNegativeTargetIdFilter* SponsoredProductsObjectIdFilter {
Filter entities by the list of objectIds
include* (array) minItems: 0 maxItems: 1000 [
string entity object identifier
]
}
}

Returns
ApiResponse

create_campaign_negative_targets(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Create campaign negative targets.

Param:
header **Prefer*** (string). The “Prefer” header, as defined in [RFC7240], allows clients to request certain behavior from the service. The service ignores preference values that are

either not supported or not known by the service. Either multiple Prefer headers are passed or single one with comma separated values, both forms are equivalent Supported preferences: return=representation - return the full object when doing create/update/delete operations instead of ids

Request Body (required)

SponsoredProductsCreateSponsoredProductsCampaignNegativeTargetingClausesRequestContent {
campaignNegativeTargetingClauses* SponsoredProductsCreateCampaignNegativeTargetingClause {
expression* (array) The NegativeTargeting expression. minItems: 0 maxItems: 1000 [
SponsoredProductsCreateOrUpdateNegativeTargetingExpressionPredicate {
type* (string) SponsoredProductsCreateOrUpdateNegativeTargetingExpressionPredicateType. The type of negative targeting expression. Enum: ['ASIN_BRAND_SAME_AS', 'ASIN_SAME_AS'].
value (string): The expression value
}
]
campaignId* (string) The identifier of the campaign to which this target is associated.
state* (string) SponsoredProductsCreateOrUpdateEntityState. Entity state for create or update operation. Enum: [ENABLED, PAUSED]
}
}

Request Body (optional)

Returns

ApiResponse

`edit_negative_product_targets`(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Update campaign negative targets.

Param:

header **Prefer*** (string). The "Prefer" header, as defined in [RFC7240], allows clients to request certain behavior from the service. The service ignores preference values that are either not supported or not known by the service. Either multiple Prefer headers are passed or single one with comma separated values, both forms are equivalent Supported preferences: return=representation - return the full object when doing create/update/delete operations instead of ids

Request Body (required)

SponsoredProductsUpdateSponsoredProductsCampaignNegativeTargetingClausesRequestContent {
campaignNegativeTargetingClauses* SponsoredProductsUpdateCampaignNegativeTargetingClause {
expression* (array) The NegativeTargeting expression. minItems: 0 maxItems: 1000 [
SponsoredProductsCreateOrUpdateNegativeTargetingExpressionPredicate {
type* (string) SponsoredProductsCreateOrUpdateNegativeTargetingExpressionPredicateType. The type of negative targeting expression. Enum: ['ASIN_BRAND_SAME_AS', 'ASIN_SAME_AS'].
value (string): The expression value
}

]
targetId* (string) The target identifier
state* (string) SponsoredProductsCreateOrUpdateEntityState. Entity state for create or update operation. Enum: [ENABLED, PAUSED]
}
}

Request Body (optional)
Returns
ApiResponse

`list_campaign_negative_targets`(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

List campaign negative targets.
Request Body (required)

SponsoredProductsListSponsoredProductsCampaignNegativeTargetingClausesRequestContent {
campaignIdFilter SponsoredProductsReducedObjectIdFilter {
Filter entities by the list of objectIds
include* (array) minItems: 0 maxItems: 100 [
string entity object identifier
]
}
campaignNegativeTargetIdFilter SponsoredProductsObjectIdFilter {
Filter entities by the list of objectIds
include* (array) minItems: 0 maxItems: 1000 [
string entity object identifier
]
}
stateFilter SponsoredProductsEntityStateFilter {
Filter entities by state
include* (array) minItems: 0 maxItems: 10 [
SponsoredProductsEntityState (string) The current resource state. Enum: [ENABLED, PAUSED, ARCHIVED, ENABLING, USER_DELETED, OTHER]
]
}
maxResults integer($int32) Number of records to include in the paginated response. Defaults to max page size for given API
nextToken (string) token value allowing to navigate to the next response page
asinFilter SponsoredProductsAsinFilter {
queryTermMatchType SponsoredProductsQueryTermMatchType (string): Match type for query filters. Enum: [BROAD_MATCH, EXACT_MATCH]
include (array) maxItems: 100 [
string
]
}
includeExtendedDataFields (boolean) Whether to get entity with extended data fields such as creationDate, lastUpdateDate, servingStatus
}

Request Body (optional)
Returns
ApiResponse

6.2.15 Budget Rules Recommendations

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**BudgetRulesRecommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoint available

Method	Endpoint	Description
POST	/sp/campaigns/budgetRules/recommendations	Gets a list of special events with suggested date range and suggested budget increase for a campaign specified by identifier.

list_campaigns_budget_rules_recommendations(*self*, ***kwargs*) → ApiResponse:

Gets a list of special events with suggested date range and suggested budget increase for a campaign specified by identifier.
Request Body (required)

SPBudgetRulesRecommendationEventRequest {
campaignId* (string) The campaign identifier.
}
Returns
ApiResponse

6.2.16 Budget Recommendations and Missed Opportunities

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**BudgetRecommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoint available

Method	Endpoint	Description
POST	/sp/campaigns/budgetRecommendations	Get recommended daily budget and estimated missed opportunities for campaigns.

list_campaigns_budget_recommendations(*self*, ***kwargs*) → ApiResponse:

Get recommended daily budget and estimated missed opportunities for campaigns.
Request Body (required)

BudgetRecommendationRequest {
campaignIds* (array) The campaign identifier. minItems: 1. maxItems: 100 [
List of campaigns. (string)
]
}

Returns
ApiResponse

6.2.17 Campaigns Budget Usage

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**CampaignsBudgetUsage**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
POST	/sp/campaigns/budget/usage	Budget usage API for SP campaigns

list_campaigns_budget_usage(*self*, *version: int = 1*, ***kwargs*) → ApiResponse:

Budget usage API for SP campaigns
Request Body

BudgetUsageCampaignRequest {
campaignIds* (array) maxItems: 100 [
A list of campaign IDs (string)
]
}

Returns
ApiResponse

6.2.18 Campaign Negative Keywords

Warning: Sponsored Product v3 is not available for Sandbox endpoint

Note: This API is version 3.0

class ad_api.api.sp.**CampaignNegativeKeywordsV3**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
POST	/sp/campaignNegativeKeywords/delete	Delete campaigns negative keywords
POST	/sp/campaignNegativeKeywords/list	List campaigns negative keywords
POST	/sp/campaignNegativeKeywords	Create campaigns negative keywords
PUT	/sp/campaignNegativeKeywords	Update campaigns negative keywords

delete_campaign_negative_keyword(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

Delete Campaign negative keywords
Request Body

SponsoredProductsDeleteSponsoredProductsCampaignNegativeKeywordsRequestContent {
campaignNegativeKeywordIdFilter* SponsoredProductsObjectIdFilter {
Filter entities by the list of objectIds
include* (array). minItems: 0. maxItems: 1000 [
entity object identifier (string)
]
}
}
Returns
ApiResponse

list_campaign_negative_keywords(*self*, *version: int = 3*, ***kwargs*) → ApiResponse:

List Campaign negative keywords
Request Body

SponsoredProductsListSponsoredProductsCampaignNegativeKeywordsRequestContent {
campaignIdFilter* SponsoredProductsReducedObjectIdFilter {
Filter entities by the list of objectIds
include* (array). minItems: 0. maxItems: 1000 [
entity object identifier (string)
]
}
campaignNegativeKeywordIdFilter* SponsoredProductsObjectIdFilter {
Filter entities by the list of objectIds

include* (array). minItems: 0. maxItems: 1000 [
entity object identifier (string)
]
}
maxResults integer($int32) Number of records to include in the paginated response. Defaults to max page size for given API
nextToken (string) token value allowing to navigate to the next response page
includeExtendedDataFields (boolean) Whether to get entity with extended data fields such as creationDate, lastUpdateDate, servingStatus. Enum: [BROAD_MATCH, EXACT_MATCH]
campaignNegativeKeywordTextFilter* SponsoredProductsKeywordTextFilter {
Filter by keywordText
queryTermMatchType SponsoredProductsQueryTermMatchType (string): Match type for query filters.
include (array). minItems: 0. maxItems: 1000 [
entity object identifier (string)
]
}
matchTypeFilter* (array). [
Restricts results to resources with the selected matchType.
SponsoredProductsNegativeMatchType (string) Enum: ['NEGATIVE_EXACT', 'NEGATIVE_PHRASE', 'NEGATIVE_BROAD', 'OTHER']
]
}

Returns
ApiResponse

`create_campaign_negative_keywords`(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Create Campaign negative keywords

header **Prefer**:*boolean* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Request Body

SponsoredProductsCreateSponsoredProductsCampaignNegativeKeywordsRequestContent {
campaignNegativeKeywords* (array) minItems: 0 maxItems: 1000 [
SponsoredProductsCreateCampaignNegativeKeyword {
campaignId* (string): The identifier of the campaign to which the keyword is associated.
matchType* SponsoredProductsCreateOrUpdateNegativeMatchType (string) Enum: ['NEGATIVE_EXACT', 'NEGATIVE_PHRASE', 'NEGATIVE_BROAD']
state* SponsoredProductsCreateOrUpdateEntityState (string): Entity state for create or update operation Enum: ['ENABLED', 'PAUSED']
keywordText* (string): The keyword text.
}
]
}

Returns
ApiResponse

`edit_campaign_negative_keywords`(*self*, *version: int = 3*, *prefer: bool = False*, ***kwargs*) → ApiResponse:

Update Campaign negative keywords

header **Prefer**:*boolean* | Used to indicate the behavior preferred by the client but is not required for successful completion of the request. Supported values will be updated in the future.

Request Body

SponsoredProductsUpdateSponsoredProductsCampaignNegativeKeywordsRequestContent {
campaignNegativeKeywords* (array) minItems: 0 maxItems: 1000 [
SponsoredProductsUpdateCampaignNegativeKeyword {
keywordId* (string): entity object identifier.
state* SponsoredProductsCreateOrUpdateEntityState (string): Entity state for create or update operation Enum: [‘ENABLED’, ‘PAUSED’]
}
]
}

Returns
ApiResponse

Warning: Version 2 is planned deprecation on 6/30/2023. There is a new version 3 of Sponsored Product API, please check the migration guide.

CHAPTER

SEVEN

SPONSORED BRANDS

7.1 3.0

7.1.1 Campaigns

class ad_api.api.sb.**Campaigns**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

Deprecated since version 4.0.2.

list_campaigns(*self*, ***kwargs*) → ApiResponse:

Gets an array of all campaigns associated with the client identifier passed in the authorization header, filtered by specified criteria.

Returns both productCollection and video campaigns. Use either 'adFormatFilter' or 'creativeType' to filter campaigns by ad formats (productCollection, video).
Keyword Args

start_index (int): Sets a zero-based offset into the requested set of campaigns. Use in conjunction with the *count* parameter to control pagination of the returned array.. [optional] if omitted the server will use the default value of 0. Default value : 0

count (int): Sets the number of campaigns in the returned array. Use in conjunction with the *startIndex* parameter to control pagination. For example, to return the first ten campaigns set *startIndex=0* and *count=10*. To return the next ten campaigns, set *startIndex=10* and *count=10*, and so on. Default value : max page size[optional]

state_filter (State): The returned array is filtered to include only campaigns with state set to one of the values in the specified comma-delimited list. Defaults to *enabled* and *paused*. Note that Campaigns rejected during moderation have state set to *archived*. Available values : enabled, paused, archived[optional]

name (str): The returned array includes only campaigns with the specified name.. [optional]

portfolio_id_filter (str): The returned array includes only campaigns associated with Portfolio identifiers matching those specified in the comma-delimited string.. [optional]

campaign_id_filter (str): The returned array includes only campaigns with identifiers matching those specified in the comma-delimited string.. [optional]

ad_format_filter (AdFormat): The returned array includes only campaigns with ad format matching those specified in the comma-delimited adFormats. Returns all campaigns if not

specified. Available values : productCollection, video[optional]

creative_type (CreativeType): Filter by the type of creative the campaign is associated with. To get non-video campaigns specify 'productCollection'. To get video campaigns, this must be set to 'video'. Returns all campaigns if not specified. Available values : productCollection, video[optional]

Example python

```
from ad_api.api.sb.campaigns import Campaigns

res = Campaigns().list_campaigns()

print(result)
```

Deprecated since version 4.0.2.

create_campaigns(*self*, ***kwargs*) → ApiResponse:

Creates one or more new Campaigns.

Request body

name (string): [optional] The name of the campaign. This name must be unique to the Amazon Advertising account to which the campaign is associated. Maximum length of the string is 128 characters.

tags (CampaignTags > string): [optional] A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers.

budget (float): [optional] The budget amount associated with the campaign.

budget_type (BudgetType > string) [optional] Note that for the lifetime budget type, *startDate* and *endDate* must be specified. The lifetime budget range is from 100 to 20,000,000 and daily budget range is 1 to 1,000,000 by default for most marketplaces. For the JP marketplace, the lifetime budget range is fromt 10,000 to 2,000,000,000, and the daily budget range is 100 to 21,000,000.., must be one of ["lifetime", "daily",]

start_date (StartDate > string) [optional] The YYYYMMDD start date of the campaign. Must be equal to or greater than the current date. If this property is not included in the request, the startDate value is not updated. If set to null, startDate is set to the current date. [nullable: true] [pattern: ^d{8}$]

end_date (EndDate > string) [optional] The YYYYMMDD end date of the campaign. Must be greater than the value specified in the startDate field. If this property is not included in the request, the endDate value is not updated. If set to null, endDate is deleted from the draft campaign. [nullable: true] [pattern: ^d{8}$]

ad_format (AdFormat > string) [optional] The type of ad format. Enum: [productCollection, video]

state (State > string): [optional] Enum: [enabled, paused, archived]

brand_entity_id (str, writeOnly: true) [optional] The brand entity identifier. Note that this field is required for sellers. For more information, see the [Stores reference](https://advertising.amazon.com/API/docs/v2/reference/stores) or [Brands reference](https://advertising.amazon.com/API/docs/v3/reference/SponsoredBrands/Brands).

bid_optimization (bool) [optional] Set to *true* to allow Amazon to automatically optimize bids for placements below top of search if omitted the server will use the default value of True

bid_multiplier (float minimum: -99 maximum: 99) [optional] A bid multiplier. Note that this field can only be set when 'bidOptimization' is set to false. Value is a percentage to two decimal

places. Example: If set to -40.00 for a $5.00 bid, the resulting bid is $3.00.

portfolio_id (int) [optional] The identifier of the portfolio to which the campaign is associated.

creative (SBCreative): [optional] | **brand_name** (str) A brand name. Maximum length is 30 characters.

brand_logo_asset_id (str) The identifier of the brand logo image from the Store assets library. See [listAssets](https://advertising.amazon.com/API/docs/v3/reference/SponsoredBrands/Stores) for more information. Note that for campaigns created in the Amazon Advertising console prior to release of the Store assets library, responses will not include a value for the brandLogoAssetID field.

brand_logo_url (str readOnly: true) The address of the hosted image.

headline (str) The headline text. Maximum length of the string is 50 characters for all marketplaces other than Japan, which has a maximum length of 35 characters.. [optional]

asins (str) An array of ASINs associated with the creative.

NOTE: do not pass an empty array, this results in an error.

should_optimize_asins (bool) default: false

NOTE: Starting on March 25th, 2021, this property will no longer be supported. This feature is currently available in the US and UK. Existing Sponsored Brands campaigns with product optimization enabled will no longer have the products in the creative automatically optimized. Campaigns with product optimization enabled will be converted to standard Sponsored Brands product collection campaigns with the default selected products showing in the creative. For POST and PUT operations, setting this property to *true* will not have any effect. The value returned in the response will always be *false*. For the GET operation, the value of this field will always be *false*. And starting on September 25th, 2021, this property will be removed completely. . [optional] if omitted the server will use the default value of False

landing_page (SBLandingPage): [optional]

asins ([str]) An array of ASINs used to generate a simple landing page. The response includes the URL of the generated simple landing page. Do not include this property in the request if the *url* property is also included, these properties are mutually exclusive.

url (str) URL of an existing simple landing page or Store page. Vendors may also specify the URL of a custom landing page. If a custom URL is specified, the landing page must include the ASINs of at least three products that are advertised as part of the campaign. Do not include this property in the request if the *asins* property is also included, these properties are mutually exclusive.

keywords ([SBCommonKeywordsKeywords]): An array of keywords associated with the campaign. [optional]

keyword_text (str) The keyword text. Maximum of 10 words.. [optional]

native_language_keyword (str) The unlocalized keyword text in the preferred locale of the advertiser.. [optional]

native_language_locale (str) The locale preference of the advertiser. For example, if the advertiser's preferred language is Simplified Chinese, set the locale to *zh_CN*. Supported locales include: Simplified Chinese (locale: zh_CN) for US, UK and CA. English (locale: en_GB) for DE, FR, IT and ES.. [optional]

match_type (MatchType) [optional] The match type. For more information, see match types in the Amazon Advertising support center. Enum: [broad, exact, phrase]

bid (float) The associated bid. Note that this value must be less than the budget associated with the Advertiser account. For more information, see [supported features](https://advertising.amazon.com/API/docs/v2/guides/supported_features) [optional]

negative_keywords ([SBUpdateDraftCampaignRequestWithKeywordsAllOfNegativeKeywords]): An array of negative keywords associated with the campaign.[optional]

keyword_text (str): The keyword text. Maximum of 10 words. [optional]

match_type (NegativeMatchType): [optional] The negative match type. For more information, see negative keyword match types in the Amazon Advertising support center. Enum:[negativeExact, negativePhrase]

Example python

```python
from ad_api.api.sb.campaigns import Campaigns

file = open("SBCreateCampaignWithKeywords.json")
data = file.read()
file.close()

result = Campaigns().create_campaigns(
    body=data
)
print(result)
```

Example json

```
[
  {
    "name": "SB.Campaign.Keyword.Api.001",
    "budget": 100,
    "budgetType": "lifetime",
    "startDate": "20240131",
    "endDate": "20250131",
    "adFormat": "productCollection",
    "brandEntityId": "ENTITY218756GCCQ6CF",
    "bidOptimization": true,
    "creative": {
      "brandName": "Apple",
      "brandLogoAssetID": "AW0_z87632Ghstax",
      "headline": "Apple Iphone XS",
```

(continues on next page)

(continued from previous page)

```
      "asins": [
        "B08N5WRTN2", "B081G9YQ73", "B008ATNJNS"
      ],
      "shouldOptimizeAsins": false
    },
    "landingPage": {
      "asins": [
        "B08N5WRTN2", "B081G9YQ73", "B008ATNJNS"
      ]
    },
    "keywords": [
      {
        "keywordText": "limpieza professional",
        "nativeLanguageKeyword": "cleaning profesional",
        "nativeLanguageLocale": "en_GB",
        "matchType": "broad",
        "bid": 0.1
      }
    ],
    "negativeKeywords": [
      {
        "keywordText": "Huawei",
        "matchType": "negativeExact"
      }
    ]
  }
]
```

Deprecated since version 4.0.2.

edit_campaigns(*self*, ***kwargs*) → ApiResponse:

Updates one or more campaigns.
Mutable fields:

name
state
portfolioId
budget
bidOptimization
bidMultiplier
endDate

Request body

name (string): [optional] The name of the campaign. This name must be unique to the Amazon Advertising account to which the campaign is associated. Maximum length of the string is 128 characters.

state (State > string): [optional] Enum: [enabled, paused, archived]

portfolio_id (int) [optional] The identifier of the portfolio to which the campaign is associated.

budget (float): [optional] The budget amount associated with the campaign.

bid_optimization (bool) [optional] Set to *true* to allow Amazon to automatically optimize bids for

placements below top of search if omitted the server will use the default value of True

bid_multiplier (float minimum: -99 maximum: 99) [optional] A bid multiplier. Note that this field can only be set when 'bidOptimization' is set to false. Value is a percentage to two decimal places. Example: If set to -40.00 for a $5.00 bid, the resulting bid is $3.00.

end_date (EndDate > string) [optional] The YYYYMMDD end date of the campaign. Must be greater than the value specified in the startDate field. If this property is not included in the request, the endDate value is not updated. If set to null, endDate is deleted from the draft campaign. [nullable: true] [pattern: ^d{8}$]

Example json

```
[
  {
    "campaignId": 144329325594765093,
    "name": "SB.Campaign.Keyword.Api.Edit",
    "budget": 110,
    "endDate": "20290131"
  }
]
```

get_campaign(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Gets a campaign specified by identifier.

Keyword Args

query **campaignId** (integer): The identifier of an existing campaign. [required]

query **locale** (string): The returned array includes only keywords associated with locale matching those specified by identifier. [optional]

Example python

```
from ad_api.api.sb.campaigns import Campaigns

campaign_id = 144329325594765093

result = Campaigns().get_campaign(
    campaignId=campaign_id
)
print(result)
```

Deprecated since version 4.0.2.

delete_campaign(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Archives a campaign specified by identifier.

This operation is equivalent to an update operation that sets the status field to 'archived'. Note that setting the status field to 'archived' is permanent and can't be undone. See Developer Notes for more information

Keyword Args

query **campaignId** (integer): The identifier of an existing campaign. [required]

Returns

ApiResponse

```
from ad_api.api.sb.campaigns import Campaigns
```

(continues on next page)

(continued from previous page)

```
campaign_id = 144329325594765093

result = Campaigns().delete_campaign(
    campaignId=campaign_id
)
print(result)
```

NOT AVAILABLE SANDBOX [INTERNAL_ERROR - HTTP 500 Internal Server Error]

7.1.2 Ad Groups

class ad_api.api.sb.**AdGroups**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_ad_groups(*self*, ***kwargs*) → ApiResponse:

Gets an array of ad groups associated with the client identifier passed in the authorization header, filtered by specified criteria.
Keyword Args

query **startIndex**:*integer* | Optional. Sets a cursor into the requested set of campaigns. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of AdGroup objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first ten ad groups set startIndex=0 and count=10. To return the next ten ad groups, set startIndex=10 and count=10, and so on. Defaults to max page size.

query **name**:*string* | Optional. The returned array includes only ad groups with the specified name.

query **adGroupIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups with an identifier specified in the comma-delimited list.

query **campaignIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups associated with the campaign identifiers in the specified comma-delimited list.

query **creativeType**:*string* | Optional. Filter by the type of creative the campaign is associated with. To get ad groups associated with non-video campaigns specify 'productCollection'. To get ad groups associated with video campaigns, this must be set to 'video'. Returns all ad groups if not specified. Available values : productCollection, video

Returns:

ApiResponse

Example python

```
from ad_api.api.sb.ad_groups import AdGroups

res = AdGroups().list_ad_groups()

print(result)
```

get_ad_group(*self*, *adGroupId*, ***kwargs*) → ApiResponse:

get_ad_group(self, adGroupId, **kwargs) -> ApiResponse

Gets an ad group specified by identifier.
Keyword Args

path **adGroupId**:*number* | Required. The identifier of an existing ad group.
Returns:

ApiResponse

Example python

```
from ad_api.api.sb.ad_groups import AdGroups

ad_group_id = 144356535815171236

res = AdGroups().get_ad_group(
    adGroupId=ad_group_id
)

print(res)
```

7.1.3 Keywords

class ad_api.api.sb.**Keywords**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_keywords(*self*, ***kwargs*) → ApiResponse:

Gets an array of keywords, filtered by optional criteria.
Keyword Args

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0
query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list.. Available values : broad, phrase, exact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list..

query **creativeType**:*string* | Optional. Filter by the type of creative the campaign is associated with. To get ad groups associated with non-video campaigns specify 'productCollection'. To get ad groups associated with video campaigns, this must be set to 'video'. Returns all ad groups if not specified. Available values : productCollection, video

query **locale**:*string* | Optional. Restricts results to keywords associated with locale.

Returns:

ApiResponse

Example python

```
from ad_api.api.sb.keywords import Keywords

result = Keywords().list_keywords()
print(result)
```

edit_keywords(*self*, ***kwargs*) → ApiResponse:

Updates one or more keywords.

Keywords submitted for update may have state set to pending for moderation review. Moderation may take up to 72 hours. Note that keywords can be updated on campaigns where serving status is not one of archived, terminated, rejected, or ended. Note that this operation supports a maximum list size of 100 keywords.

Request body

keywordId (integer($int64)): [required] The identifier of the keyword.

adGroupId (integer($int64)): [required] The identifier of an existing ad group to which the keyword is associated.

campaignId (integer($int64)) [required] The identifier of an existing campaign to which the keyword is associated.

state (string): [optional] Newly created SB keywords are in a default state of 'draft' before transitioning to a 'pending' state for moderation. After moderation, the keyword will be in an enabled state. Enum: [enabled, paused, pending, archived, draft]

bid (number) [optional] The bid associated with the keyword. Note that this value must be less than the budget associated with the Advertiser account. For more information, see the Keyword bid constraints by marketplace section of the supported features article

Returns:

ApiResponse

Example python

```
from ad_api.api.sb.keywords import Keywords

file = open("edit.json")
data = file.read()
file.close()
result = Keywords().edit_keywords(
    body=data
)
print(result)
```

Example json

```
[
  {
    "keywordId": 144351213232803965,
    "adGroupId": 144344914935156713,
    "campaignId": 144371273320927424,
    "state": "enabled",
    "bid": 0.51
  }
]
```

create_keywords(*self*, ***kwargs*) → ApiResponse:

Creates one or more keywords.

Note that state can't be set at keyword creation. Keywords submitted for creation have state set to pending while under moderation review. Moderation review may take up to 72 hours.

Note that keywords can be created on campaigns where serving status is not one of archived, terminated, rejected, or ended.

Note that this operation supports a maximum list size of 100 keywords.
Request body

adGroupId (integer($int64)) The identifier of an existing ad group to which the keyword is associated.

campaignId (integer($int64)) The identifier of an existing campaign to which the keyword is associated.

keywordText (string) The keyword text. The maximum number of words for this string is 10.

nativeLanguageKeyword (string) The unlocalized keyword text in the preferred locale of the advertiser.

nativeLanguageLocale (string) The locale preference of the advertiser. For example, if the advertiser's preferred language is Simplified Chinese, set the locale to zh_CN. Supported locales include: Simplified Chinese (locale: zh_CN) for US, UK and CA. English (locale: en_GB) for DE, FR, IT and ES.

matchType (string) The match type. For more information, see match types in the Amazon Advertising support center. Enum: [broad, exact, phrase]

bid (number) [optional] The bid associated with the keyword. Note that this value must be less than the budget associated with the Advertiser account. For more information, see the Keyword bid constraints by marketplace section of the supported features article.

Returns:

ApiResponse
Example python

```
from ad_api.api.sb.keywords import Keywords

file = open("create.json")
data = file.read()
file.close()
result = Keywords().create_keywords(
    body=data
)
logging.info(result)
```

Example json

```
[
  {
    "adGroupId": 144356535815171236,
    "campaignId": 144329324494765093,
    "keywordText": "frenos",
    "nativeLanguageKeyword": "brakes",
    "nativeLanguageLocale": "en_GB",
    "matchType": "exact",
    "bid": 0.4
  }
]
```

get_keyword(*self*, *keywordId*, ***kwargs*) → ApiResponse:

Gets a keyword specified by identifier.
Keyword Args

path **keywordId** (integer): The identifier of an existing keyword. [required]
query **locale** (string): The returned array includes only keywords associated with locale matching those specified by identifier. [optional]
Returns:

ApiResponse
Example python

```
from ad_api.api.sb.keywords import Keywords

keyword_id = 144148613757234151
result = Keywords().get_keyword(
    keywordId=keyword_id
)
print(result)
```

delete_keyword(*self*, *keywordId*, ***kwargs*) → ApiResponse:

Archives a keyword specified by identifier.

This operation is equivalent to an update operation that sets the status field to 'archived'. Note that setting the status field to 'archived' is permanent and can't be undone. See Developer Notes for more information.
Keyword Args

path **keywordId** (integer): The identifier of an existing keyword. [required]

Returns:

ApiResponse
Example python

```
from ad_api.api.sb.keywords import Keywords

keyword_id = 144148613757234151
result = Keywords().delete_keyword(
    keywordId=keyword_id
)
print(result)
```

7.1.4 Negative Keywords

class ad_api.api.sb.**NegativeKeywords**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_negative_keywords(*self*, ***kwargs*) → ApiResponse:

Gets an array of negative keywords, filtered by optional criteria.
Keyword Args

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **matchTypeFilter**:*string* | Optional. Restricts results to keywords with match types within the specified comma-separated list. Available values : negativePhrase, negativeExact.

query **keywordText**:*string* | Optional. Restricts results to keywords that match the specified text exactly.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **keywordIdFilter**:*string* | Optional. Restricts results to keywords associated with campaigns specified by identifier in the comma-delimited list.

query **creativeType**:*string* | Optional. Filter by the type of creative the campaign is associated with. To get negative keywords associated with non-video campaigns specify 'productCollection'. To get negative keywords associated with video campaigns, this must be set to 'video'. Returns all negative keywords if not specified. Available values : productCollection, video

Returns:

ApiResponse

`edit_negative_keywords`(*self*, ***kwargs*) → ApiResponse:

Updates one or more negative keywords.

Negative keywords submitted for update may have state set to pending for moderation review. Moderation may take up to 72 hours. Note that negative keywords can be updated on campaigns where serving status is not one of archived, terminated, rejected, or ended. Note that this operation supports a maximum list size of 100 negative keywords.

Request Body

'**keywordId**': *number*, {'description': 'The identifier of the negative keyword.'}

'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which the negative keyword is associated.'}

'**campaignId**': *number*, {'description': 'The identifier of the campaign to which the negative keyword is associated.'}

'**state**': *string*, {'description': 'The current state of the negative keyword. Newly created SB negative keywords are in a default state of 'draft' before transitioning to a 'pending' state for moderation review. 'enabled' refers to negative keywords that are active. 'archived' refers to negative keywords that are permanently inactive and cannot be returned to the 'enabled' state.' , 'Enum': 'enabled, pending, archived, draft'}

Returns:

ApiResponse

`create_negative_keywords`(*self*, ***kwargs*) → ApiResponse:

Creates one or more negative keywords.

Note that bid and state can't be set at negative keyword creation.

Note that Negative keywords submitted for creation have state set to pending while under moderation review. Moderation review may take up to 72 hours.

Note that negative keywords can be created on campaigns one where serving status is not one of archived, terminated, rejected, or ended.

Note that this operation supports a maximum list size of 100 negative keywords.

Request Body

'**campaignId**': *number*, {'description': 'The identifer of the campaign to which the keyword is associated.'}

'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this keyword is associated.'}

'**state**': *string*, {'description': 'The current resource state.' , 'Enum': '[enabled]'}

'**keywordText**': *string*, {'description': 'The text of the expression to match against a search query.'}

'**matchType**': *string*, {'description': 'The type of match.' , 'Enum': '[negativeExact, negativePhrase]'}

Returns:

ApiResponse

`get_negative_keyword`(*self*, *keywordId*, ***kwargs*) → ApiResponse:

Gets a negative keyword specified by identifier.

Keyword Args

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns

ApiResponse

delete_negative_keyword(*self*, *keywordId*, ***kwargs*) → ApiResponse:

Archives a negative keyword specified by identifier.

This operation is equivalent to an update operation that sets the status field to 'archived'. Note that setting the status field to 'archived' is permanent and can't be undone. See Developer Notes for more information.

Keyword Args

path **keywordId**:*number* | Required. The identifier of an existing keyword.

Returns

ApiResponse

7.1.5 Product Targeting

class ad_api.api.sb.**Targets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_products_targets(*self*, ***kwargs*) → ApiResponse:

Gets a list of product targets associated with the client identifier passed in the authorization header, filtered by specified criteria.

Request Body

'**nextToken**': *string*, {'description': 'Operations that return paginated results include a pagination token in this field. To retrieve the next page of results, call the same operation and specify this token in the request. If the NextToken field is empty, there are no further results.'}

'**maxResults**': *number*, {'description': 'The identifier of the ad group to which this keyword is associated.'}

'**filters**': *string*, {'Restricts results to targets with the specified filters. Filters are inclusive. Filters are joined using 'and' logic. Specify one type of each filter. Specifying multiples of the same type of filter results in an error.'}

'**filterType**': *string*, {'CREATIVE_TYPE': '[CREATIVE_TYPE]'}

'**filterType**': *string*, {'TARGETING_STATE': '[archived, paused, pending, enabled]'}

'**filterType**': *string*, {'CAMPAIGN_ID': '[CAMPAIGN_ID]'}

'**filterType**': *string*, {'AD_GROUP_ID': '[SBAdGroupId]'}

Returns:

ApiResponse

edit_products_targets(*self*, ***kwargs*) → ApiResponse:

Updates one or more targets.

Request Body

'**targetId**': *integer($int64)*, {'description': 'The identifier of the target.'}

'**adGroupId**': *integer($int64)*, {'description': 'The identifier of the ad group to which the target is associated.'}

'**campaignId**': *integer($int64)*, {'description': 'The identifier of the campaign to which the target is associated.'}

'**state**': *string*, {'values': '[enabled, paused, pending, archived, draft]'}

'**bid**': *number*, {'description': 'The associated bid. Note that this value must be less than the budget associated with the Advertiser account. For more information, see supported features.'}

Returns:

ApiResponse

`create_products_targets`(*self*, ***kwargs*) → ApiResponse:

Create one or more targets.

Request Body

'**adGroupId**': *integer($int64)*, {'description': 'The identifier of the ad group to which the target is associated.'}

'**campaignId**': *integer($int64)*, {'description': 'The identifier of the campaign to which the target is associated.'}

'**expressions**': *SBExpression*, {'type': 'asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs', 'values': 'The text of the targeting expression. The - token defines a range. For example, 2-4 defines a range of 2, 3, and 4.'}

'**bid**': *number*, {'description': 'The associated bid. Note that this value must be less than the budget associated with the Advertiser account. For more information, see supported features.'}

Returns:

ApiResponse

`get_products_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Gets a target specified by identifier.

Keyword Args

path **targetId**:*number* | Required. The identifier of an existing target.

Returns

ApiResponse

`delete_products_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Archives a target specified by identifier. Note that archiving is permanent, and once a target has been archived it can't be made active again.

Keyword Args

path **targetId**:*number* | Required. The identifier of an existing target.

Returns

ApiResponse

7.1.6 Negative Product Targeting

class ad_api.api.sb.**NegativeTargets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_negative_targets(*self*, ***kwargs*) → ApiResponse:

Gets a list of product negative targets associated with the client identifier passed in the authorization header, filtered by specified criteria.
Request Body

'**nextToken**': *string*, {'description': 'Operations that return paginated results include a pagination token in this field. To retrieve the next page of results, call the same operation and specify this token in the request. If the NextToken field is empty, there are no further results.'}
'**maxResults**': *number*, {'description': 'The identifier of the ad group to which this keyword is associated.'}
'**filters**': *string*, {'Restricts results to targets with the specified filters. Filters are inclusive. Filters are joined using 'and' logic. Specify one type of each filter. Specifying multiples of the same type of filter results in an error.'}

'**filterType**': *string*, {'CREATIVE_TYPE': '[productCollection, video]'}

'**filterType**': *string*, {'TARGETING_STATE': '[enabled, pending, archived]'}

'**filterType**': *string*, {'CAMPAIGN_ID': '[CAMPAIGN_ID]'}

'**filterType**': *string*, {'AD_GROUP_ID': '[SBAdGroupId]'}

Returns:
ApiResponse

create_negative_targets(*self*, ***kwargs*) → ApiResponse:

Creates one or more targeting expressions.

Request Body
'**adGroupId**': *integer($int64)*, {'description': 'The identifier of the ad group to which the target is associated.'}
'**campaignId**': *integer($int64)*, {'description': 'The identifier of the campaign to which the target is associated.'}
'**expressions**': *string*, {'Restricts results to targets with the specified filters. Filters are inclusive. Filters are joined using 'and' logic. Specify one type of each filter. Specifying multiples of the same type of filter results in an error.'}

'**type**': *string*, {'values': 'asinBrandSameAs, asinSameAs'}

'**value**': *string*, {'description': 'The text of the negative expression.'}

Returns:
ApiResponse

`edit_negative_targets`(*self*, ***kwargs*) → ApiResponse:

Updates one or more negative targeting clauses.
Request Body

'**targetId**': *integer($int64)*, {'description': 'The target identifier.'}
'**adGroupId**': *integer($int64)*, {'description': 'The identifier of an existing ad group. The newly created target is associated to this ad group.'}
'**state**': *string*, {'values': '[enabled, paused, pending, archived, draft]'}

Returns:
ApiResponse

`get_negative_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Get a negative targeting clause specified by identifier.
Keyword Args
path **negativeTargetId**:*number* | Required. The identifier of an existing negative target.
Returns
ApiResponse

`delete_negative_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Archives a negative targeting clause.
Keyword Args
path **negativeTargetId**:*number* | Required. The identifier of an existing negative target.
Returns
ApiResponse

7.1.7 Targeting Recommendations

`class ad_api.api.sb.TargetsRecommendations`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

`list_products_targets`(*self*, ***kwargs*) → ApiResponse:

Gets a list of recommended categories for targeting.

Recommendations are based on the ASINs that are passed in the request.
Request Body

'**nextToken**': *string*, {'description': 'Operations that return paginated results include a pagination token in this field. To retrieve the next page of results, call the same operation and specify this token in the request. If the NextToken field is empty, there are no further results.'}
'**maxResults**': *integer*, {'description': 'Sets a limit on the number of results returned by an operation. minimum: 1, maximum: 100'}
'**filters**': *SBExpression*, {'filterType': '[ASINS]', 'values': 'A list of ASINs / An ASIN.'}

Returns:
ApiResponse

`list_category_targets`(*self*, ***kwargs*) → ApiResponse:

Gets a list of recommended categories for targeting.

Recommendations are based on the ASINs that are passed in the request.
Request Body

'**asins**': *string*, {'description': 'A list of ASINs.'}
Returns:
ApiResponse

`list_brand_targets`(*self*, ***kwargs*) → ApiResponse:

Gets a list of brand suggestions.

The Brand suggestions are based on a list of either category identifiers or keywords passed in the request. It is not valid to specify both category identifiers and keywords in the request.
Request Body (oneOf ->)

'**categoryId**': *integer($int64)*, {'description': 'The category identifier for which to get recommendations.'}
'**keyword**': *string*, {'description': 'The keyword for which to get recommendations.}
Returns:
ApiResponse

7.1.8 Bid Recommendations

`class` ad_api.api.sb.`BidRecommendations`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

`get_bid_recommendations`(***kwargs*)

Get a list of bid recommendation objects for a specified list of keywords or products.
Request Body

'**campaignId**': *integer($int64)*, {'description': 'The identifier of the campaign for which bid recommendations are created.'}
'**targets**': *integer($int64)*, {'SBExpression': 'A name value pair that defines a targeting expression. The type field defines the predicate. The value field defines the value to match for the predicate.'}

'**type**': *string*, {'values': '[asinCategorySameAs, asinBrandSameAs, asinPriceLessThan, asinPriceBetween, asinPriceGreaterThan, asinReviewRatingLessThan, asinReviewRatingBetween, asinReviewRatingGreaterThan, asinSameAs]'}

'**value**': *string*, {'description': 'The text of the targeting expression. The - token defines a range. For example, 2-4 defines a range of 2, 3, and 4.'}

'**keywords**': *string*, {'description': 'SBBidRecommendationKeyword'}

'**matchType**': *string*, {'values': '[broad, exact, phrase]'}

'**keywordText**': *string*, {'description': 'The text of the keyword. Maximum of 10 words.'}

'**adFormat**': *integer($int64)*, {'values': '[productCollection, video]'}

Returns:
ApiResponse

7.1.9 Stores

class ad_api.api.sb.**Stores**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_assets(*self*, ***kwargs*) → ApiResponse:

For sellers or vendors, gets an array of assets associated with the specified brand entity identifier. Vendors are not required to specify a brand entity identifier, and in this case all assets associated with the vendor are returned.
Keyword Args

path **brandEntityId** (string): For sellers, this field is required. It is the Brand entity identifier of the Brand for which assets are returned. This identifier is retrieved using the getBrands operation. For vendors, this field is optional. If a vendor does not specify this field, all assets associated with the vendor are returned. For more information about the difference between a seller and a vendor, see the Amazon Advertising FAQ. [required]

query **mediaType** (string): Specifies the media types used to filter the returned array. Currently, only the brandLogo type is supported. If not specified, all media types are returned. Available values : brandLogo, image [optional]

Returns:

ApiResponse

list_stores(*self*, ***kwargs*) → ApiResponse:

List store information for all registered stores under an advertiser.
Keyword Args
None

7.1.10 Landing Page Asins

class ad_api.api.sb.**PageAsins**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

get_page_asins(*self*, ***kwargs*) → ApiResponse:

Gets ASIN information for a specified address.

Keyword Args

query **pageUrl**:*string* | Required. For sellers, the address of a Store page. Vendors may also specify the address of a custom landing page. For more information, see the [Stores section](https://advertising.amazon.com/help#GPRM3ZHEXEY5RBFZ) of the Amazon Advertising support center.

Returns

asinList

Example python

```
from ad_api.api.sb.landing_page_asins import PageAsins

page_url = 'https://www.amazon.es/stores/page/49D4CB50-9C2F-46D5-8E50-5505529C790D'

result = PageAsins().get_page_asins(
    pageUrl=page_url
)

logging.info(result)
```

Payload

```
{'asinList': ['B08N5WRTN2',
              'B081G9YQ73',
              'B008ATNJNS',
              'B08N5VXMK6',
              'B016UPAVDE',
              'B08N5S5HH5',
              'B016MUBL4U',
              'B08N5WM84C',
              'B08N5TLVQ2',
              'B0863B2L69',
              'B08N5TLB5J',
              'B081GBLPTB',
              'B081G4SK26',
              'B08N5VT5SV',
              'B0863G2M7F',
              'B07BRLMY93',
              'B08N5V4CKB',
              'B086395QZM',
```

(continues on next page)

(continued from previous page)

```
              'B081GC15CY',
              'B086395QZP'],
 'code': 'SUCCESS'}
```

NOT AVAILABLE SANDBOX [NOT_FOUND - Could not find resource for full path]

7.1.11 Media

class ad_api.api.sb.**Media**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use this interface to request and retrieve store information. This can be used for Sponsored Brands campaign creation, to pull the store URL information, and for asset registration for Stores.

create_media(*self*, ***kwargs*) → ApiResponse:

Creates an ephemeral resource (upload location) to upload Media for an Ad Program (SponsoredBrands).

The upload location is short lived and expires in 15 minutes. Once the upload is complete, /media/complete API should be used to notify that the upload is complete.

The upload location only supports PUT HTTP Method to upload the media content. If the upload location expires, API user will get 403 Forbidden response.

Request body

programType (string): [required] [SponsoredBrands].

creativeType (string): [required] [Video].

Returns:

ApiResponse

7.1.12 Brands

class ad_api.api.sb.**Brands**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Brands for campaign, ad group, keyword, negative keyword, drafts, Stores, landing pages, and Brands management operations. For more information about Sponsored Brands, see the Sponsored Brands Support Center. For onboarding information, see the account setup topic.

list_brands(*self*, ***kwargs*) → ApiResponse:

Gets an array of Brand data objects for the Brand associated with the profile ID passed in the header. For more information about Brands, see [Brand Services](https://brandservices.amazon.com/).

Keyword Args

None

Returns:

ApiResponse payload:

'**brandId**': *string*, {'description': 'The Brand identifier.'}
'**brandEntityId**': *string*, {'description': 'The Brand entity identifier.'}
'**brandRegistryName**': *string*, {'description': 'The Brand name.'}

Example python

```
from ad_api.api.sb.brands import Brands

result = Brands().list_brands()

logging.info(result)
```

Payload

```
[{'brandEntityId': 'ENTITY5ON7M22396H',
  'brandId': 'A387T2Q1UNXHJK',
  'brandRegistryName': 'Apple'},
 {'brandEntityId': 'ENTITY218756GCCQ6CF',
  'brandId': 'A36UAF6UNGFFAR',
  'brandRegistryName': 'Huawei'},
 {'brandEntityId': 'ENTITY1PRG7GD8FVXA3',
  'brandId': 'A87RK5OLHEBUU5',
  'brandRegistryName': 'Xiaomi'}]
```

NOT AVAILABLE SANDBOX [NOT_FOUND - Could not find resource for full path]

7.1.13 Moderation

class ad_api.api.sb.**Moderation**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use this interface to request and retrieve store information. This can be used for Sponsored Brands campaign creation, to pull the store URL information, and for asset registration for Stores.

get_moderation(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Gets the moderation result for a campaign specified by identifier.
Keyword Args

path **campaignId** (integer): The campaign identifier. [required]

> **Warning:** Note that this resource is only available for campaigns in the US marketplace.

###Example Python

```
campaign_id = 144329324494765093

result = Moderation().get_moderation(
    campaignId=campaign_id
)

print(result)
```

7.1.14 Reports

class ad_api.api.sb.**Reports**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Sponsored Brand Reports

Documentation: https://advertising.amazon.com/API/docs/en-us/reference/sponsored-brands/2/reports

post_report(*self*, *recordType*, ***kwargs*) → ApiResponse:

Requests a Sponsored Brands report.

Request the creation of a performance report for all entities of a single type which have performance data to report. Record types can be: *campaigns*, *adGroups*, and *keywords*.

Keyword Args

path **recordType** (string): The type of entity for which the report should be generated. Supported record types: *campaigns*, *adGroups*, and *keywords*

Request body

segment (string): [required] Dimension on which to segment the report. Available values : *placement*, *query*

reportDate (string): [required] The date for which to retrieve the performance report in *YYYYMMDD* format. The time zone is specified by the profile used to request the report. If this date is today, then the performance report may contain partial information. Reports are not available for data older than 60 days.

metrics (string) [optional] A comma-separated list of the metrics to be included in the report.

creativeType (string) [optional] Set to *video* to request a Sponsored Brands video report.

Returns:

ApiResponse

Example python

```
from ad_api.api.sb.reports import Reports

with open("campaigns.json", "r", encoding="utf-8") as f:
    data = f.read()

# Available values : campaigns, adGroups, targets and keywords
record_type = 'campaigns'

result = Reports().post_report(
    recordType=record_type,
    body=data
)

payload = result.payload
report_id = payload.get('reportId')
```

Example json

Open this `json` file to see the result:

```
{
  "segment": "placement",
  "reportDate": "20211101",
  "metrics": "campaignName,campaignId,campaignStatus,campaignBudget,
↪campaignBudgetType,campaignRuleBasedBudget"
}
```

get_report(*self*, *reportId*, ***kwargs*) → ApiResponse:

Gets a previously requested report specified by identifier.
Keyword Args

path **reportId** (string): [required] The report identifier.
Request body

creativeType (string): [optional] Set to *video* to retrieve a Sponsored Brands video report.
Returns:
ApiResponse

Example python

```
from ad_api.api.sb.reports import Reports

# this report_id is obtained from post_report method
report_id = 'amzn1.clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c
↪'

result = Reports().get_report(
    reportId=report_id
)
```

Result json

```
{'expiration': 1640736000000,
 'fileSize': 6546,
 'location': 'https://advertising-api-eu.amazon.com/v1/reports/amzn1.clicksAPI.v1.
↪p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c/download',
 'reportId': 'amzn1.clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-
↪fc1fecdb165c',
 'status': 'SUCCESS',
 'statusDetails': 'Report has been successfully generated.'}}
```

download_report(*self*, ***kwargs*) → ApiResponse:

Downloads the report previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the report). Take in mind that a direct download of location returned in get_report will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the report (this url has a expiration time)
Keyword Args

url (string): [required] The location obatined from get_report.

file (string): [optional] The path to save the file if mode is download *json*, *zip* or *gzip*.

format (string): [optional] The mode to download the report: *data* (list), *raw*, *url*, *json*, *zip*, *gzip*. Default (*url*)

Returns:

ApiResponse

Example python

```
from ad_api.api.sb.reports import Reports

# the url=location is obtained from get_report method need to in stay 'status':
↪'SUCCESS' if is 'IN_PROGRESS' the report cannot be downloaded
location = 'https://advertising-api-eu.amazon.com/v1/reports/amzn1.clicksAPI.v1.
↪p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c/download'

# path = '/Users/your-profile/Downloads/report_name'
# mode = "data"  # "data (list), raw, url, json, zip, gzip default is url"

result = Reports().download_report(
    url=location,
    # file=path,
    # format=mode
)
```

7.1.15 Snapshots

Warning: Currently, the Ads API does not support snapshots for Sponsored Brands video campaigns or campaigns created using the version 4 endpoints. Snapshots include records for version 3, non-video campaigns only.

class ad_api.api.sb.**Snapshots**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Use the Amazon Advertising API for Sponsored Products for campaign, ad group, keyword, negative keyword, and product ad management operations. For more information about Sponsored Products, see the Sponsored Products Support Center. For onboarding information, see the account setup topic.

post_snapshot(*self*, *recordType*, ***kwargs*) → ApiResponse:

Returns:

ApiResponse

get_snapshot(*self*, *reportId*, ***kwargs*) → ApiResponse:

Gets the status of a requested snapshot.

Returns:

ApiResponse

download_snapshot(*self*, ***kwargs*) → ApiResponse:

Downloads the snapshot previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the snapshot). Take in mind that a direct download of location returned in get_snapshot will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the snapshot (this url has a expiration time)
Keyword Args

url (string): The location obatined from get_snapshot [required]

file (string): The path to save the file if mode is download json, zip or gzip. [optional]

format (string): The mode to download the snapshot: data (list), raw, url, json, zip, gzip. Default (url) [optional]

Returns:
ApiResponse

7.2 4.0

7.2.1 Campaigns

class ad_api.api.sb.**CampaignsV4**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Version 4 of the SB Campaigns

create_campaigns(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Creates Sponsored Brands campaigns.
Request body (Required)
Request body | **name** (string): [optional] The name of the campaign. This name must be unique to the Amazon Advertising account to which the campaign is associated. Maximum length of the string is 128 characters. | **state** (State > string): [optional] Enum: [enabled, paused, archived] | **portfolio_id** (int) [optional] The identifier of the portfolio to which the campaign is associated. | **budget** (float): [optional] The budget amount associated with the campaign. | **bid_optimization** (bool) [optional] Set to *true* to allow Amazon to automatically optimize bids for placements below top of search if omitted the server will use the default value of True | **bid_multiplier** (float minimum: -99 maximum: 99) [optional] A bid multiplier. Note that this field can only be set when 'bidOptimization' is set to false. Value is a percentage to two decimal places. Example: If set to -40.00 for a $5.00 bid, the resulting bid is $3.00. | **end_date** (EndDate > string) [optional] The YYYYMMDD end date of the campaign. Must be greater than the value specified in the startDate field. If this property is not included in the request, the endDate value is not updated. If set to null, endDate is deleted from the draft campaign. [nullable: true] [pattern: ^d{8}$]
Returns
ApiResponse

edit_campaigns(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Update Sponsored Brand Campaigns
Request body (required)

campaignId (string) : Entity object identifier.

name (string): [optional] The name of the campaign. This name must be unique to the Amazon Advertising account to which the campaign is associated. Maximum length of the string is 128 characters.

state (State > string): [optional] Enum: [enabled, paused, archived]

portfolio_id (int) [optional] The identifier of the portfolio to which the campaign is associated.

budget (float): [optional] The budget amount associated with the campaign.

bid_optimization (bool) [optional] Set to *true* to allow Amazon to automatically optimize bids for placements below top of search if omitted the server will use the default value of True

bid_multiplier (float minimum: -99 maximum: 99) [optional] A bid multiplier. Note that this field can only be set when 'bidOptimization' is set to false. Value is a percentage to two decimal places. Example: If set to -40.00 for a $5.00 bid, the resulting bid is $3.00.

end_date (EndDate > string) [optional] The YYYYMMDD end date of the campaign. Must be greater than the value specified in the startDate field. If this property is not included in the request, the endDate value is not updated. If set to null, endDate is deleted from the draft campaign. [nullable: true] [pattern: ^d{8}$]

Returns

ApiResponse

`delete_campaigns`(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

The operation is currently not supported subject to potential change. Deletes Sponsored Brands campaigns.
Request body (required)

campaignIdFilter (ObjectIdFilter): The identifier of an existing campaign. [required] | **include** (list>str): Entity object identifier. [required] minItems: 0 maxItems: 10

Returns

ApiResponse

`list_campaigns`(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Lists Sponsored Brands campaigns.
Request Body (optional)

[Include the body for specific filtering, or leave empty to get all campaigns.]

start_index (int): Sets a zero-based offset into the requested set of campaigns. Use in conjunction with the *count* parameter to control pagination of the returned array.. [optional] if omitted the server will use the default value of 0. Default value : 0

state_filter (State): The returned array is filtered to include only campaigns with state set to one of the values in the specified comma-delimited list. Defaults to *enabled* and *paused*. Note that Campaigns rejected during moderation have state set to *archived*. Available values : enabled, paused, archived[optional]

name (str): The returned array includes only campaigns with the specified name.. [optional]

portfolio_id_filter (str): The returned array includes only campaigns associated with Portfolio identifiers matching those specified in the comma-delimited string.. [optional]

campaign_id_filter (str): The returned array includes only campaigns with identifiers matching those specified in the comma-delimited string.. [optional]

ad_format_filter (AdFormat): The returned array includes only campaigns with ad format matching those specified in the comma-delimited adFormats. Returns all campaigns if not specified. Available values : productCollection, video[optional]

max_results (int) Number of records to include in the paginated response. Defaults to max page size for given API. Minimum 10 and a Maximum of 100 [optional]

next_token (string) Token value allowing to navigate to the next response page. [optional]

includeExtendedDataFields (boolean) Setting to true will slow down performance because the API needs to retrieve extra information for each campaign. [optional]

Returns
ApiResponse

7.2.2 Ad Groups

class ad_api.api.sb.**AdGroupsV4**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Version 4 of Sponsored Brands

create_ad_groups(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Creates Sponsored Brand Ad Group.

Request Body | campaignId (string) : The identifier of the campaign to which the keyword is associated. | name (string) : The name of the ad group. | state (CreateOrUpdateEntityState > string) : Entity state for create or update operation. Enum : [ENABLED, PAUSED]
Returns
ApiResponse

update_ad_groups(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Update Sponsored Brand Ad groups.
Request Body

campaignId (string) : The identifier of the campaign to which the keyword is associated. [optional]
name (string) : The name of the ad group. [optional]
state (CreateOrUpdateEntityState > string) : Entity state for create or update operation. Enum : [ENABLED, PAUSED]

Returns
ApiResponse

list_ad_groups(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

List Sponsored Brand Ad groups.

Request Body (optional) | **campaignIdFilter** (dict) : Filter entities by the list of objectIds. | **stateFilter** (dict) : Filter entities by state. | **maxResults** (int) : Number of records to include in the paginated response. Defaults to max page size for given API. | **nextToken** (string) : Token value allowing to navigate to the next response page. | **adGroupIdFilter** (dict) : Filter entities by the list of objectIds. | **includeExtendedDataFields** (boolean) Setting to true will slow down performance because the API needs to retrieve extra information for each campaign. | **nameFilter** (dict) : Filter entities by name.
Returns:

ApiResponse

delete_ad_groups(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Delete Sponsored Brands ad groups.
Request Body (optional) :

adGroupIdFilter (dict)
[Filter entities by the list of objectIds. [optional]] include (list) : Entity object identifier.

Returns
ApiResponse

7.2.3 Ads

class ad_api.api.sb.**AdsV4**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

list_ads(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Lists all Sponsored Brands ads.

Request Body (optional) | **campaignIdFilter** (dict) : Filter entities by the list of objectIds. | **stateFilter** (dict) : Filter entities by state. | **maxResults** (int) : Number of records to include in the paginated response. Defaults to max page size for given API. | **nextToken** (string) : Token value allowing to navigate to the next response page. | **adIdFilter** (dict) : Filter entities by the list of objectIds. | **adGroupIdFilter** (dict) : Filter entities by the list of objectIds. | **nameFilter** (dict) : Filter entities by name.

Returns

ApiResponse

create_video_ads(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Creates Sponsored Brand video ads.

Request Body (required) | **ads** ([{}]) :

name (string) : The name of the ad **state** (CreateOrUpdateEntityState > String) Entity state for create or update operation. Enum : [ENABLED, PAUSED] **adGroupId** (string) : The adGroup identifier. **creative** (dict) :

asins ([string]) [optional] videoAssetIds ([string]) [optional]

Returns

ApiResponse

create_product_collection_ads(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Creates Sponsored Brands product collection ads.

Request Body (required) | **ads** ([{}]) :

landingPage (dict) :

asins [string] [optional] pageType (LadingPageType > String) : The type of landing page, such as store page, product list (simple landing page), custom url. [optional] url (string) : URL of an existing simple landing page or Store page. [optional]

name (string) : The name of the ad **state** (CreateOrUpdateEntityState > String) Entity state for create or update operation. Enum : [ENABLED, PAUSED] **adGroupId** (string) : The adGroup identifier. **creative** (dict) :

brandLogoCrop (string) [optional] brandName (string) [optional] customImage-AssetId (string) customImageCrop ({int}) [optional] brandLogoAssetID (string) [optional] headline (string) The headline text [optional]

Returns

ApiResponse

create_brand_video_ads(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Creates Sponsored Brands brand video ads.

Request Body (required) | **ads** ([{}]) :

landingPage (dict) :

asins [string] [optional] pageType (LadingPageType > String) : The type of landing page, such as store page, product list (simple landing page), custom url. [optional] url (string) : URL of an existing simple landing page or Store page. [optional]

name (string) : The name of the ad **state** (CreateOrUpdateEntityState > String) Entity state for create or update operation. Enum : [ENABLED, PAUSED] **adGroupId** (string) : The adGroup identifier. **creative** (dict) :

asins ([string]) brandName (string) [optional] brandLogoCrop ({}) [optional] videoAssetIds ([string]) [optional] brandLogoAssetID (string) [optional] headline (string) The headline text [optional]

Returns

ApiResponse

`create_store_spotlight_ads`(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Creates Sponsored Brands store spotlight ads.

Request Body (required) | **ads** ([{}]) :

landingPage (dict) :

asins [string] [optional] pageType (LadingPageType > String) : The type of landing page, such as store page, product list (simple landing page), custom url. [optional] url (string) : URL of an existing simple landing page or Store page. [optional]

name (string) : The name of the ad **state** (CreateOrUpdateEntityState > String) Entity state for create or update operation. Enum : [ENABLED, PAUSED] **adGroupId** (string) : The adGroup identifier. **creative** (dict) :

brandLogoCrop (string) [optional] brandName (string) [optional] subpages (string) [optional] brandLogoAssetID (string) [optional] headline (string) The headline text [optional]

Returns

ApiResponse

`update_ads`(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Updates the Sponsored Brands ads.

Request Body (required) | **ads** ([{}]) :

adsId (string) [required] : The product ad identifier. name (string) [optional] : The name of the ad state (State>String) [optional] : Entity state for create or update operation. Enum : [ENABLED, PAUSED]

Returns

ApiResponse

`delete_ads`(*self*, *version: int = 4*, ***kwargs*) → ApiResponse:

Deletes Sponsored Brands ads.

Request Body | **adIdFilter** (dict) : Filter entities by the list of objectIds.

include : list of adIds as string

CHAPTER

EIGHT

SPONSORED DISPLAY

8.1 Campaigns

class ad_api.api.sd.**Campaigns**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

create_campaigns(*self*, ***kwargs*) → ApiResponse

Creates one or more campaigns.

body: | REQUIRED {'description': 'An array of ad groups.}'

'**portfolioId**': *number*, {'description': 'The identifier of an existing portfolio to which the campaign is associated'}

'**name**': *string*, {'description': 'A name for the campaign'}

'**tags**': *string*, {'description': 'A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers.'}

'**campaignType**': *string*, {'description': 'The advertising product managed by this campaign', 'Enum': '[sponsoredProducts]'}

'**targetingType**': *string*, {'description': 'The type of targeting for the campaign.', 'Enum': '[manual, auto]'}

'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

'**dailyBudget**': *number($float)*, {'description': 'A daily budget for the campaign.'}

'**startDate**': *string*, {'description': 'A starting date for the campaign to go live. The format of the date is YYYYMMDD.'}

'**endDate**': *string* nullable: true, {'description': 'An ending date for the campaign to stop running. The format of the date is YYYYMMDD.'}

'**premiumBidAdjustment**': *boolean*, {'description': 'If set to true, Amazon increases the default bid for ads that are eligible to appear in this placement. See developer notes for more information.'}

'**bidding**': *Bidding*, {'strategy': 'string', 'Enum': '[legacyForSales, autoForSales, manual]', 'adjustments': '{...}'}

Returns:

ApiResponse

delete_campaign(*self*, *campaignId*, ***kwargs*) → ApiResponse

Sets the campaign status to archived. Archived entities cannot be made active again. See developer notes for more information.

path **campaignId**:*number* | Required. The identifier of an existing campaign.

Returns:
ApiResponse

`edit_campaigns`(*self*, ***kwargs*) → ApiResponse

Updates one or more campaigns.

body: | REQUIRED {'description': 'An array of ad groups.}'
'**campaignId**': *number*, {'description': 'The identifier of an existing campaign to update.'}
'**portfolioId**': *number*, {'description': 'The identifier of an existing portfolio to which the campaign is associated'}
'**name**': *string*, {'description': 'The name for the campaign'}
'**tags**': *CampaignTags*, {'description': 'A list of advertiser-specified custom identifiers for the campaign. Each customer identifier is a key-value pair. You can specify a maximum of 50 identifiers.'}
'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}
'**dailyBudget**': *number($float)*, {'description': 'The daily budget for the campaign.'}
'**startDate**': *string*, {'description': 'The starting date for the campaign to go live. The format of the date is YYYYMMDD.'}
'**endDate**': *string* nullable: true, {'description': 'The ending date for the campaign to stop running. The format of the date is YYYYMMDD.'}
'**premiumBidAdjustment**': *boolean*, {'description': 'If set to true, Amazon increases the default bid for ads that are eligible to appear in this placement. See developer notes for more information.'}
'**bidding**': *Bidding*, {'strategy': 'string', 'Enum': '[legacyForSales, autoForSales, manual]', 'adjustments': '{...}'}
Returns:
ApiResponse

`get_campaign`(*self*, *campaignId*, ***kwargs*) → ApiResponse

Gets a campaign specified by identifier.
path **campaignId**:*number* | Required. The identifier of an existing campaign.
Returns:
ApiResponse

`get_campaign_extended`(*self*, *campaignId*, ***kwargs*) → ApiResponse

Gets an array of campaigns with extended data fields.
path **campaignId**:*number* | Required. The identifier of an existing campaign.
Returns:
ApiResponse

`list_campaigns`(*self*, ***kwargs*) → ApiResponse

Gets an array of campaigns.
query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **name**:*string* | Optional. Restricts results to campaigns with the specified name.

query **portfolioIdFilter**:*string* | Optional. A comma-delimited list of portfolio identifiers.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

Returns:
ApiResponse

`list_campaigns_extended`(*self*, ***kwargs*) → ApiResponse

Gets an array of campaigns with extended data fields.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **name**:*string* | Optional. Restricts results to campaigns with the specified name.

query **portfolioIdFilter**:*string* | Optional. A comma-delimited list of portfolio identifiers.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

Returns:
ApiResponse

Campaigns explanation goes here.

8.2 Ad Groups

`class` ad_api.api.sd.**AdGroups**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

`create_ad_groups`(*self*, ***kwargs*) → ApiResponse

Creates one or more ad groups.

body: | REQUIRED {'description': 'An array of ad groups.}'
'**name**': *string*, {'description': 'A name for the ad group'}
'**campaignId**': *number*, {'description': 'An existing campaign to which the ad group is associated'}
'**defaultBid**': *number($float)*, {'description': 'A bid value for use when no bid is specified for keywords in the ad group', 'minimum': '0.02'}
'**state**': *string*, {'description': 'A name for the ad group', 'Enum': '[enabled, paused, archived]'}

Returns:
ApiResponse

`delete_ad_group`(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Sets the ad group status to archived. Archived entities cannot be made active again. See developer notes for more information.

path **adGroupId**:*number* | Required. The identifier of an existing ad group.

Returns:
ApiResponse

`edit_ad_groups`(***kwargs*)

edit_ad_group(self, **kwargs) -> ApiResponse

Updates one or more ad groups.

body: | REQUIRED {'description': 'An array of ad groups.}'

'**adGroupId**': *number*, {'description': 'The identifier of the ad group.'}

'**name**': *string*, {'description': 'The name of the ad group.'}

'**defaultBid**': *number($float)*, {'description': 'The bid value used when no bid is specified for keywords in the ad group.', 'minimum': '0.02'}

'**state**': *string*, {'description': 'The current resource state', 'Enum': '[enabled, paused, archived]'}

Returns:

ApiResponse

`get_ad_group`(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Gets an ad group specified by identifier.

path **adGroupId**:*number* | Required. The identifier of an existing ad group.

Returns:

ApiResponse

`get_ad_group_extended`(*self*, *adGroupId*, ***kwargs*) → ApiResponse

Gets an ad group that has extended data fields.

path **adGroupId**:*number* | Required. The identifier of an existing ad group.

Returns:

ApiResponse

`list_ad_groups`(*self*, ***kwargs*) → ApiResponse

Gets an array of AdGroup objects for a requested set of Sponsored Display ad groups. Note that the AdGroup object is designed for performance, and includes a small set of commonly used fields to reduce size. If the extended set of fields is required, use the ad group operations that return the AdGroupResponseEx object.

query **startIndex**:*integer* | Optional. Sets a cursor into the requested set of campaigns. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of AdGroup objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first ten ad groups set startIndex=0 and count=10. To return the next ten ad groups, set startIndex=10 and count=10, and so on. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived

query **campaignIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups associated with the campaign identifiers in the specified comma-delimited list.

query **adGroupIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups with an identifier specified in the comma-delimited list.

query **name**:*string* | Optional. The returned array includes only ad groups with the specified name.

Returns:

ApiResponse

list_ad_groups_extended(*self*, ***kwargs*) → ApiResponse

Gets an array of AdGroup objects for a requested set of Sponsored Display ad groups. Note that the AdGroup object is designed for performance, and includes a small set of commonly used fields to reduce size. If the extended set of fields is required, use the ad group operations that return the AdGroupResponseEx object.

query: **startIndex**:*integer* | Optional. Sets a cursor into the requested set of campaigns. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of AdGroup objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first ten ad groups set startIndex=0 and count=10. To return the next ten ad groups, set startIndex=10 and count=10, and so on. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived

query **campaignIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups associated with the campaign identifiers in the specified comma-delimited list.

query **adGroupIdFilter**:*string* | Optional. The returned array is filtered to include only ad groups with an identifier specified in the comma-delimited list.

query **name**:*string* | Optional. The returned array includes only ad groups with the specified name.

Returns:

ApiResponse

Campaigns explanation goes here.

8.3 Product Ads

Warning: Sponsored Display is not available for Sandbox endpoint

class ad_api.api.sd.**ProductAds**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Sponsored Display Product Ads

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-display/3-0/openapi#/Product%20ads/

The Sponsored Display API supports creation of reports for campaigns, ad groups, product ads, targets, and asins. Create a ReportRequest object specifying the fields corresponding to performance data metrics to include in the report. See ReportRequest object for more information about the performance data metric fields and their descriptions.

list_product_ads(*self*, ***kwargs*) → ApiResponse:

Gets a list of product ads.

Gets an array of ProductAd objects for a requested set of Sponsored Display product ads. Note that the ProductAd object is designed for performance, and includes a small set of commonly used fields to reduce

size. If the extended set of fields is required, use a product ad operation that returns the ProductAdResponseEx object.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **adIdFilter**:*string* | Optional. Restricts results to product ads associated with the product ad identifiers specified in the comma-delimited list.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

Returns:

ApiResponse

Example python 1

```
from ad_api.base import AdvertisingApiException, Marketplaces
from ad_api.api import sponsored_display

try:
    status = 'enabled'
    store = 'my_store'

    result = sponsored_display.ProductAds(account=store,
                                          marketplace=Marketplaces.ES,
                                          debug=True).list_product_ads(
        stateFilter=status
    )
    products_ads = result.payload # <class 'list'>

except AdvertisingApiException as error:
    logging.error(error)
```

Example python 2

The default config in the credentials.yml if not passed as kwargs

account='my_store',

The default marketplace is Marketplaces.EU if not passed as kwarg ex. Marketplaces.US

marketplace=Marketplaces.US,

Debug info like headers, method and url and raw response will not provided

debug=True

```
import os
from ad_api.base import AdvertisingApiException
from ad_api.api.sd.product_ads import ProductAds

```

(continues on next page)

(continued from previous page)

```
try:
    status = 'enabled'
    index = 5 # Sets a cursor into the requested set of product ads
    number = 2 # Sets the number of ProductAd objects in the returned array
    result = ProductAds().list_product_ads(
        stateFilter=status
    )

    products_ads = result.payload # <class 'list'>

except AdvertisingApiException as error:
    logging.error(error)
```

edit_product_ads(*self*, ***kwargs*) → ApiResponse:

edit_product_ads(self, **kwargs) -> ApiResponse

Updates one or more product ads specified by identifier.

body: | REQUIRED {'description': 'A list of product ad objects with updated values for the state field.}'
'**adId**': *number*, {'description': 'The identifier of an existing campaign to update.'}
'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

Returns:
ApiResponse
Example body json

An array of ProductAd objects. For each object, specify a product ad identifier and the only mutable field, state. Maximum length of the array is 100 objects.

```
[
    {
        "state": "enabled",
        "adId": 182575048323550
    }
]
```

Example python

```
import json
from ad_api.api import sponsored_display

try:

    dictionary = \
    [
        {
            "state": "enabled",
            "adId": 182575048323550
        }
    ]

    data = json.dumps(dictionary)
```

(continues on next page)

(continued from previous page)

```
    result = sponsored_display.ProductAds(debug=True).edit_product_ads(
        body=data
    )

    logging.info(result.payload)
    logging.info(json.dumps(response.payload))

except AdvertisingApiException as error:
    logging.error(error)
```

Example response.payload **<class 'list'>**

```
[{'code': 'SUCCESS', 'adId': 182575048323550}]
```

Example json.dumps(response.payload) **<class 'str'>**

```
[{"code": "SUCCESS", "adId": 182575048323550}]
```

create_product_ads(*self*, ***kwargs*) → ApiResponse:

create_product_ads(self, **kwargs) -> ApiResponse

Creates one or more product ads.

body: | REQUIRED {'description': 'An array of ProductAd objects. For each object, specify required fields and their values. Required fields are adGroupId, SKU (for sellers) or ASIN (for vendors), and state'. Maximum length of the array is 100 objects.'}

'**state**': *string*, {'description': 'The current resource state.', 'Enum': '[enabled, paused, archived]'}

'**adGroupId**': *integer($int64)*, {'description': 'The identifier of the ad group.'}

'**campaignId**': *integer($int64)*, {'description': 'The identifier of the campaign.'}

'**asin**': *string*, {'description': 'The ASIN associated with the product. Defined for vendors only.'}

'**sku**': *string*, {'description': 'The SKU associated with the product. Defined for seller accounts only.'}

Returns:

ApiResponse

get_product_ad(*self*, *adId*, ***kwargs*) → ApiResponse:

get_product_ad_request(self, adId, **kwargs) -> ApiResponse

Gets a product ad specified by identifier.

path **adId**:*number* | Required. A product ad identifier.

Returns:

ApiResponse

delete_product_ad(*self*, *adId*, ***kwargs*) → ApiResponse:

delete_product_ad(self, adId, **kwargs) -> ApiResponse

Sets the state of a specified product ad to archived. Note that once the state is set to archived it cannot be changed.

path **adId**:*number* | Required. A product ad identifier.

Returns:

ApiResponse

list_product_ads_extended(*self*, ***kwargs*) → ApiResponse:

list_product_ads_extended_request(self, **kwargs) -> ApiResponse

Gets extended data for a list of product ads filtered by specified criteria.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **adIdFilter**:*string* | Optional. Restricts results to product ads associated with the product ad identifiers specified in the comma-delimited list.

Returns:

ApiResponse

get_product_ad_extended(*self*, *adId*, ***kwargs*) → ApiResponse:

get_product_ad_extended(self, adId, **kwargs) -> ApiResponse

Gets extended data for a product ad specified by identifier.

path **adId**:*number* | Required. A product ad identifier.

Returns:

ApiResponse

8.4 Reports

class ad_api.api.sd.**Reports**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Sponsored Display Reports

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-display/3-0/openapi#/

The Sponsored Display API supports creation of reports for campaigns, ad groups, product ads, targets, and asins. Create a ReportRequest object specifying the fields corresponding to performance data metrics to include in the report. See ReportRequest object for more information about the performance data metric fields and their descriptions.

post_report(*self*, *recordType*, ***kwargs*) → ApiResponse:

Requests a Sponsored Display report.

Request the creation of a performance report for all entities of a single type which have performance data to report. Record types can be one of campaigns, adGroups, keywords, productAds, asins, and targets. Note that for asin reports, the report currently can not include metrics associated with both keywords and targets. If the targetingId value is set in the request, the report filters on targets and does not return sales associated with keywords. If the targetingId value is not set in the request, the report filters on keywords and does not return sales associated with targets. Therefore, the default behavior filters the report on keywords. Also

note that if both keywordId and targetingId values are passed, the report filters on targets only and does not return keywords.

Keyword Args

path **recordType** (string): The type of entity for which the report should be generated. Available values : campaigns, adGroups, productAds, targets, asins [required]

Request body

reportDate (string): [optional] The date for which to retrieve the performance report in YYYYMMDD format. The time zone is specified by the profile used to request the report. If this date is today, then the performance report may contain partial information. Reports are not available for data older than 60 days. For details on data latency, see the Service Guarantees in the developer notes section.

tactic (string): [optional] The advertising tactic associated with the campaign. The following table lists available tactic names: T00001, T00020, T00030, remarketing

metrics (string) [optional] A comma-separated list of the metrics to be included in the report. The following tables summarize report metrics which can be requested via the reports interface. Different report types can use different metrics. Note that ASIN reports only return data for either keywords or targets, but not both.

Returns:

ApiResponse

Example python

```python
from ad_api.api.sd.reports import Reports

with open("asins.json", "r", encoding="utf-8") as f:
    data = f.read()

# Available values : campaigns, adGroups, productAds, targets, asins
record_type = 'asins'

result = Reports().post_report(
    recordType=record_type,
    body=data
)

payload = result.payload
report_id = payload.get('reportId')
```

Example json

Open this `json` file to see the result:

```
{
  "tactic": "T00020",
  "reportDate": "20211101",
  "metrics": "campaignName,campaignId,adGroupName,adGroupId,asin"
}
```

`get_report`(*self*, *reportId*, ***kwargs*) → ApiResponse:

Gets a previously requested report specified by identifier.

Keyword Args

path **reportId** (string): [required] The report identifier.

Returns:

ApiResponse

Example python

```
from ad_api.api.sd.reports import Reports

# this report_id is obtained from post_report method
report_id = 'amzn1.clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c
↪'

result = Reports().get_report(
    reportId=report_id
)
```

Result json

```
{'expiration': 1640736000000,
 'fileSize': 6546,
 'location': 'https://advertising-api-eu.amazon.com/v1/reports/amzn1.clicksAPI.v1.
↪p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c/download',
 'reportId': 'amzn1.clicksAPI.v1.p44551.61549C5E.e4599469-7392-4624-a858-
↪fc1fecdb165c',
 'status': 'SUCCESS',
 'statusDetails': 'Report has been successfully generated.'}}
```

download_report(*self*, ***kwargs*) → ApiResponse:

Downloads the report previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the report). Take in mind that a direct download of location returned in get_report will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the report (this url has a expiration time)

Keyword Args

url (string): [required] The location obatined from get_report

file (string): [optional] The path to save the file if mode is download *json*, *zip* or *gzip*.

format (string): [optional] The mode to download the report: *data* (list), *raw*, *url*, *json*, *zip*, *gzip*. Default (*url*)

Returns:

ApiResponse

Example python

```
from ad_api.api.sd.reports import Reports

# the url=location is obtained from get_report method need to in stay 'status':
↪'SUCCESS' if is 'IN_PROGRESS' the report cannot be downloaded
location = 'https://advertising-api-eu.amazon.com/v1/reports/amzn1.clicksAPI.v1.
↪p44551.61549C5E.e4599469-7392-4624-a858-fc1fecdb165c/download'
```

(continues on next page)

(continued from previous page)

```
# path = '/Users/your-profile/Downloads/report_name'
# mode = "data"  # "data (list), raw, url, json, zip, gzip default is url"

result = Reports().download_report(
    url=location,
    # file=path,
    # format=mode
)
```

8.5 Product Targeting

Warning: Sponsored Display is not available for Sandbox endpoint

class ad_api.api.sd.**Targets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon Advertising API for Sponsored Display

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-display/3-0/openapi#/Targeting

This API enables programmatic access for campaign creation, management, and reporting for Sponsored Display campaigns. For more information on the functionality, see the Sponsored Display Support Center . For API onboarding information, see the account setup topic.

This specification is available for download from the Advertising API developer portal.

Endpoints available

Method	Endpoint	Description
GET	/sd/targets	Gets a list of targeting clauses.
PUT	/sd/targets	Updates one or more targeting clauses.
POST	/sd/targets	Creates one or more targeting clauses.
GET	/sd/targets/{targetId}	Gets a targeting clause specified by identifier.
DELETE	/sd/targets/{targetId}	Sets the *state* of a targeting clause to *archived*.
GET	/sd/targets/extended	Gets a list of targeting clause objects with extended fields.
GET	/sd/targets/extended/{targetId}	Gets extended information for a targeting clause.

list_products_targets(*self*, ***kwargs*) → ApiResponse:

Gets a list of targeting clauses objects for a requested set of Sponsored Display targets. Note that the Targeting Clause object is designed for performance, and includes a small set of commonly used fields to reduce size. If the extended set of fields is required, use the target operations that return the TargetingClauseEx object.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Defaults to 0.

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. Optional. Restricts results to those with state set to values in the specified comma-separated list. Available values : enabled, paused, archived,

enabled, paused, enabled, archived, paused, archived, enabled, paused, archived. Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Optional list of comma separated adGroupIds. Restricts results to targeting clauses with the specified adGroupId

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers. Missing in Amazon documentation.

Returns:

ApiResponse

`edit_products_targets`(*self*, ***kwargs*) → ApiResponse:

Updates one or more targeting clauses. Targeting clauses are identified using their targetId. The mutable fields are bid and state. Maximum length of the array is 100 objects.

body: | REQUIRED {'description': 'A list of up to 100 targeting clauses. Mutable fields: state, bid.}'

'**state**': *string*, {'description': '[enabled, paused, archived]'}

'**bid**': *number($float)*, {'description': 'The bid will override the adGroup bid if specified. This field is not used for negative targeting clauses. The bid must be less than the maximum allowable bid for the campaign's marketplace; for a list of maximum allowable bids, find the "Bid constraints by marketplace" table in our documentation overview.'}

'**targetId**': *integer($int64)*, {'description': 'The target id.'}

Returns:

ApiResponse

`create_products_targets`(*self*, ***kwargs*) → ApiResponse:

Creates one or more targeting expressions.

body: | REQUIRED {'description': 'A list of up to 100 targeting clauses for creation.}'

'**state**': *string*, {'description': '[enabled, paused, archived]'}

'**bid**': *number($float)*, {'description': 'The bid will override the adGroup bid if specified. This field is not used for negative targeting clauses. The bid must be less than the maximum allowable bid for the campaign's marketplace; for a list of maximum allowable bids, find the "Bid constraints by marketplace" table in our documentation overview.'}

'**adGroupId**': *integer($int64)*, {'description': 'The page number in the result set to return.'}

'**expressionType**': *string*, {'description': 'Tactic T00020 ad groups only allow manual targeting. [manual, auto]'}

'**expression**': *CreateTargetingExpression*

CreateTargetingExpression: type: array description: The targeting expression to match against.

Applicable to Product targeting (T00020)

- A 'TargetingExpression' in a Product targeting Campaign can only contain 'Targeting-Predicate' components.
- Expressions must specify either a category predicate or an ASIN predicate, but never both.
- Only one category may be specified per targeting expression.
- Only one brand may be specified per targeting expression.
- Only one asin may be specified per targeting expression.
- To exclude a brand from a targeting expression you must create a negative targeting expression in the same ad group as the positive targeting expression.

Applicable to Audience targeting (T00030)

- A 'TargetingExpression' in an Audience Campaign can only contain 'TargetingPredicateNested' components.
- Expressions must specify either auto ASIN-grain (exact products), manual ASIN-grain (similar products), or manual category-grain targeting.
- **Future** To exclude parts of an audience, specify a TargetingPredicateNested component that contains a negative TargetingPredicate type.

oneOf->

TargetingPredicate:

type: object description: A predicate to match against in the Targeting Expression (only applicable to Product targeting - T00020).

- All IDs passed for category and brand-targeting predicates must be valid IDs in the Amazon Advertising browse system.
- Brand, price, and review predicates are optional and may only be specified if category is also specified.
- Review predicates accept numbers between 0 and 5 and are inclusive.
- When using either of the 'between' strings to construct a targeting expression the format of the string is 'double-double' where the first double must be smaller than the second double. Prices are not inclusive.

'**type**': *string*, {'enum': '[asinSameAs, asinCategorySameAs, asinBrandSameAs, asinPriceBetween, asinPriceGreaterThan, asinPriceLessThan, asinReviewRatingLessThan, asinReviewRatingGreaterThan, asinReviewRatingBetween, asinIsPrimeShippingEligible, asinAgeRangeSameAs, asinGenreSameAs, similarProduct]'}

'**value**': *string*, {'description': 'The value to be targeted. example: B0123456789'}

TargetingPredicateNested:

type: object description: A behavioral event and list of targeting predicates that represents an Audience to target (only applicable to Audience targeting - T00030).

- For auto ASIN-grain targeting, the value array must contain only 'exactProduct' and 'lookback' TargetingPredicateBase components.
- For manual ASIN-grain targeting, the value array must contain only 'similarProduct' and 'lookback' TargetingPredicateBase components.
- For manual Category-grain targeting, the value array must contain a 'lookback' and 'asinCategorySameAs' TargetingPredicateBase component, which can be further refined with optional brand, price, star-rating and shipping eligibility refinements.
- For Amazon Audiences targeting, the TargetingPredicateNested type should be set to 'audience' and the value array should include one TargetingPredicateBase component with type set to 'audienceSameAs'.

- **Future** For manual Category-grain targeting, adding a 'negative' TargetingPredicateBase will exclude that TargetingPredicateNested from the overall audience.

'**type**': *string*, {'enum': '[views, audience, purchases]'}
'**value**': TargetingPredicateBase

type: object

description: A predicate to match against inside the TargetingPredicateNested component (only applicable to Audience targeting - T00030).

- All IDs passed for category and brand-targeting predicates must be valid IDs in the Amazon Advertising browse system.
- Brand, price, and review predicates are optional and may only be specified if category is also specified.
- Review predicates accept numbers between 0 and 5 and are inclusive.
- When using either of the 'between' strings to construct a targeting expression the format of the string is 'double-double' where the first double must be smaller than the second double. Prices are not inclusive.
- The exactProduct, similarProduct, and negative types do not utilize the value field.
- The only type currently applicable to Amazon Audiences targeting is 'audienceSameAs'.
- A 'relatedProduct' TargetingPredicateBase will Target an audience that has purchased a related product in the past 7,14,30,60,90,180, or 365 days.
- **Future** A 'negative' TargetingPredicateBase will exclude that TargetingPredicateNested from the overall audience.

'**type**': *string*, {'enum': '[asinCategorySameAs, asinBrandSameAs, asinPriceBetween, asinPriceGreaterThan, asinPriceLessThan, asinReviewRatingLessThan, asinReviewRatingGreaterThan, asinReviewRatingBetween, similarProduct, exactProduct, asinIsPrimeShippingEligible, asinAgeRangeSameAs, asinGenreSameAs, audienceSameAs, lookback, negative, relatedProduct]'}
'**value**': *string*, {'description': 'The value to be targeted. example: B0123456789'}

Returns:

ApiResponse

`get_products_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

This call returns the minimal set of targeting clause fields.

path **targetId**:*number* | Required. The identifier of a targeting clause.

Returns:

ApiResponse

`delete_products_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Equivalent to using the updateTargetingClauses operation to set the state property of a targeting clause to archived. See Developer Notes for more information.

path **targetId**:*number* | Required. The identifier of a targeting clause.

Returns:

ApiResponse

`list_products_targets_extended`(*self*, ***kwargs*) → ApiResponse:

Gets an array of TargetingClauseEx objects for a set of requested targets. Note that this call returns the full set of targeting clause extended fields, but is less efficient than getTargets.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Defaults to 0.

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. Optional. Restricts results to those with state set to values in the specified comma-separated list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived. Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Optional list of comma separated adGroupIds. Restricts results to targeting clauses with the specified adGroupId

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers. Missing in Amazon documentation.

Returns:

ApiResponse

`get_products_target_extended`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Gets a targeting clause object with extended fields. Note that this call returns the full set of targeting clause extended fields, but is less efficient than getTarget.

path **targetId**:*number* | Required. The identifier of a targeting clause.

Returns:

ApiResponse

8.6 Targeting Recommendations

Warning: Sponsored Display is not available for Sandbox endpoint

`class ad_api.api.sd.TargetsRecommendations`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon Advertising API for Sponsored Display

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-display/3-0/openapi#/Targeting%20Recommendations

This API enables programmatic access for campaign creation, management, and reporting for Sponsored Display campaigns. For more information on the functionality, see the Sponsored Display Support Center . For API onboarding information, see the account setup topic.

This specification is available for download from the Advertising API developer portal.

Endpoint available

Method	Endpoint	Description
POST	/sd/targets/recommendations	Returns a set of bid recommendations for targeting clauses

list_targets_recommendations(*self*, ***kwargs*) → ApiResponse:

Note that version application/vnd.sdtargetingrecommendations.v3.1+json is now supported. Provides a list of products to target based on the list of input ASINs. Allow 1 week for our systems to process data for any new ASINs listed on Amazon before using this service. Currently the API will return up to 100 recommended products and categories. The currently available tactic identifiers are:

Tactic Name	Type	Description
T00020	Product Targeting	Products: Choose individual products to show your ads in placements related to those products.
T00030	Audience Targeting	Audiences: Select individual audiences to show your ads

body: SDTargetingRecommendationsRequestV31 | REQUIRED {'description': 'Request for targeting recommendations.}'

'**tactic***': *SDTacticV31string*, {'description': 'The advertising tactic associated with the campaign.', 'Enum': '[T00020, T00030]'}

'**products**': *SDGoalProduct*, {'description': 'A list of products for which to get targeting recommendations. minItems: 1, maxItems: 10000'}

'**SDGoalProduct***', {'description': 'A product an advertisers wants to advertise. Recommendations will be made for specified goal products.}

'**asin***': *SDASINstring*, REQUIRED {'description': 'Amazon Standard Identification Number. pattern: [a-zA-Z0-9]{10}. example: B00PN11UNW}

'**typeFilter**': *SDRecommendationTypeV31string*, {'description': 'Signifies a type of recommendation', 'Enum': '[PRODUCT, CATEGORY]'}

Returns:

ApiResponse

8.7 Bid Recommendations

Warning: Sponsored Display is not available for Sandbox endpoint

class ad_api.api.sd.**BidRecommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon Advertising API for Sponsored Display

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-display/3-0/openapi#/Bid%20Recommendations

This API enables programmatic access for campaign creation, management, and reporting for Sponsored Display campaigns. For more information on the functionality, see the Sponsored Display Support Center . For API onboarding information, see the account setup topic.

This specification is available for download from the Advertising API developer portal.

Endpoint available

Method	Endpoint	Description
POST	/sd/targets/bid/recommendations	Returns a set of bid recommendations for targeting clauses

list_targets_bid_recommendations(*self*, ***kwargs*) → ApiResponse:

Provides a list of bid recommendations based on the list of input advertised ASINs and targeting clauses in the same format as the targeting API. For each targeting clause in the request a corresponding bid recommendation will be returned in the response. Currently the API will accept up to 100 targeting clauses. The recommended bids are derrived from the last 7 days of winning auction bids for the related targeting clause.

body: SDTargetingBidRecommendationsRequestV32 | REQUIRED {'description': 'A list of products to tailor bid recommendations for category and audience based targeting clauses.}'

'**products**': *SDGoalProduct*, {'description': 'A list of products for which to get targeting recommendations. minItems: 1, maxItems: 10000'}

'**bidOptimization**': *SDBidOptimizationV32 string*, {'description': 'Determines what the recommended bids will be optimized for. Note that "reach" for bid optimization is not currently supported. This note will be removed when these operations become available.', 'Enum': '[clicks, conversions, reach]'}

'**costType**': *SDCostTypeV31 string*, {'description': 'Determines what performance metric the bid recommendations will be optimized for. Note that "vcpm" for cost type is not currently supported. This note will be removed when these operations become available.', 'Enum': '[cpc, vcpm]'}

'**targetingClauses**': SDTargetingClauseV31

type: object

description: A list of targeting clauses to receive bid recommendations for.

'**expressionType**': *string*, {'description': 'Tactic T00020 ad groups only allow manual targeting.', 'Enum': '[manual, auto]'}

'**expression**': SDTargetingExpressionV31

type: object description: The targeting expression to match against.

oneOf->

TargetingPredicate:

type: object description: A predicate to match against in the Targeting Expression (only applicable to Product targeting - T00020).

- All IDs passed for category and brand-targeting predicates must be valid IDs in the Amazon Advertising browse system.
- Brand, price, and review predicates are optional and may only be specified if category is also specified.
- Review predicates accept numbers between 0 and 5 and are inclusive.
- When using either of the 'between' strings to construct a targeting expression the format of the string is 'double-double' where the first double must be smaller than the second double. Prices are not inclusive.

'**type**': *string*, {'enum': '[asinSameAs, asinCategorySameAs, asinBrandSameAs, asinPriceBetween, asinPriceGreaterThan, asinPriceLessThan, asinReviewRatingLessThan, asinReviewRatingGreaterThan, asinReviewRatingBetween, asinIsPrimeShippingEligible, asinAgeRangeSameAs, asinGenreSameAs]'}

'**value**': *string*, {'description': 'The value to be targeted. example: B0123456789'}

TargetingPredicateNested:

type: object description: A behavioral event and list of targeting predicates that represents an Audience to target (only applicable to Audience targeting - T00030).

- For manual ASIN-grain targeting, the value array must contain only, 'exactProduct', 'similarProduct', 'releatedProduct' and 'lookback' TargetingPredicateBase components. The 'lookback' is mandatory and the value should be set to '7', '14', '30', '60', '90', '180' or '365'
- For manual Category-grain targeting, the value array must contain a 'lookback' and 'asinCategorySameAs' TargetingPredicateBase component, which can be further refined with optional brand, price, star-rating and shipping eligibility refinements. The 'lookback' is mandatory and the value should be set to '7', '14', '30', '60', '90', '180' or '365'
- For manual Category-grain targeting, the value array must contain a 'lookback' and 'asinCategorySameAs'

TargetingPredicateBase component, which can be further refined with optional brand, price, star-rating and shipping eligibility refinements.

- For Amazon Audiences targeting, the TargetingPredicateNested type should be set to 'audience' and the value array should include one TargetingPredicateBase component with type set to 'audienceSameAs'.
- Future For manual Category-grain targeting, adding a 'negative' TargetingPredicateBase will exclude that TargetingPredicateNested from the overall audience.

'**type**': *string*, {'enum': '[views, audience, purchases]'}
'**value**': TargetingPredicateBase

type: object

description: A predicate to match against inside the TargetingPredicateNested component (only applicable to Audience targeting - T00030).

- All IDs passed for category and brand-targeting predicates must be valid IDs in the Amazon Advertising browse system.
- Brand, price, and review predicates are optional and may only be specified if category is also specified.
- Review predicates accept numbers between 0 and 5 and are inclusive.
- When using either of the 'between' strings to construct a targeting expression the format of the string is 'double-double' where the first double must be smaller than the second double. Prices are not inclusive.
- The exactProduct, similarProduct, and negative types do not utilize the value field.
- The only type currently applicable to Amazon Audiences targeting is 'audienceSameAs'.
- A 'relatedProduct' TargetingPredicateBase will Target an audience that has purchased a related product in the past 7,14,30,60,90,180, or 365 days.
- **Future** A 'negative' TargetingPredicateBase will exclude that TargetingPredicateNested from the overall audience.

'**type**': *string*, {'enum': '[asinCategorySameAs, asinBrandSameAs, asinPriceBetween, asinPriceGreaterThan, asinPriceLessThan,

asinReviewRatingLessThan,
asinReviewRatingGreaterThan,
asinReviewRatingBetween, similarProduct,
exactProduct, asinIsPrimeShippingEligible,
asinAgeRangeSameAs, asinGenreSameAs,
audienceSameAs, lookback]'}
'**value**': *string*, {'description': 'The value to be
targeted. example: B0123456789'}

Returns:
ApiResponse

8.8 Negative targeting

Warning: Sponsored Display is not available for Sandbox endpoint

class ad_api.api.sd.**NegativeTargets**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon Advertising API for Sponsored Display

Documentation: https://advertising.amazon.com/API/docs/en-us/sponsored-display/3-0/openapi#/Negative%20targeting

This API enables programmatic access for campaign creation, management, and reporting for Sponsored Display campaigns. For more information on the functionality, see the Sponsored Display Support Center . For API onboarding information, see the account setup topic.

This specification is available for download from the Advertising API developer portal.

Endpoints available

Method	Endpoint	Description
GET	/sd/negativeTargets	Gets a list of negative targeting clauses.
PUT	/sd/negativeTargets	Updates one or more negative targeting clauses.
POST	/sd/negativeTargets	Creates one or more negative targeting clauses.
GET	/sd/negativeTargets/{negativeTargetId}	Gets a negative targeting clause specified by identifier.
DELETE	/sd/negativeTargets/{negativeTargetId}	Sets the *state* of a negative targeting clause to *archived*.
GET	/sd/negativeTargets/extended	Gets a list of negative targeting clause objects with extended fields.
GET	/sd/negativeTargets/extended/{negativeTargetId}	Gets extended information for a negative targeting clause.

list_negative_targets(*self*, ***kwargs*) → ApiResponse:

list_negative_targets(self, **kwargs) -> ApiResponse

Gets a list of negative targeting clauses filtered by specified criteria.
query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers. Missing in official Amazon documentation

Returns:

ApiResponse

`edit_negative_targets`(*self*, ***kwargs*) → ApiResponse:

Updates one or more negative targeting clauses. Negative targeting clauses are identified using their targetId. The mutable field is state. Maximum length of the array is 100 objects.

body: | UpdateNegativeTargetingClause REQUIRED {'description': 'A list of up to 100 negative targeting clauses. Note that the only mutable field is state.}'

'**state**': *number*, {'description': 'The resource state. [enabled, paused, archived]'}

'**targetId***': *integer($int64)*, {'description': 'The identifier of the TargetId.'}

Returns:

ApiResponse

`create_negative_targets`(*self*, ***kwargs*) → ApiResponse:

create_products_targets(self, **kwargs) -> ApiResponse:

Creates one or more targeting expressions.

body: | REQUIRED {'description': 'An array of asins objects.}'

'**state**': *number*, {'description': 'The current resource state. [enabled, paused, archived]'}

'**adGroupId**': *number*, {'description': 'The identifier of the ad group to which this negative target is associated.'}

'**expression**'

'**type**': *string*, {'description': 'The intent type. See the targeting topic in the Amazon Advertising support center for more information.', 'enum': '[asinSameAs, asinBrandSameAs]'}

'**value**': *string*, {'description': 'The value to be negatively targeted. Used only in manual expressions.'}

'**expressionType**': *string*, {'description': '[auto, manual]'}

Returns:

ApiResponse

`get_negative_target`(*self*, *targetId*, ***kwargs*) → ApiResponse:

This call returns the minimal set of negative targeting clause fields, but is more efficient than getNegativeTargetsEx.

Get a negative targeting clause specified by identifier.

path **negativeTargetId**:*integer* | Required. The negative targeting clause identifier.

Returns:

ApiResponse

`delete_negative_targets`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Equivalent to using the updateNegativeTargetingClauses operation to set the state property of a targeting clause to archived. See Developer Notes for more information.

Archives a negative targeting clause.

path **negativeTargetId**:*integer* | Required. The negative targeting clause identifier.

Returns:

ApiResponse

`list_negative_targets_extended`(*self*, ***kwargs*) → ApiResponse:

Gets an array of NegativeTargetingClauseEx objects for a set of requested negative targets. Note that this call returns the full set of negative targeting clause extended fields, but is less efficient than getNegativeTargets.

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **stateFilter**:*string* | Optional. The returned array is filtered to include only ad groups with state set to one of the values in the specified comma-delimited list. Available values : enabled, paused, archived, enabled, paused, enabled, archived, paused, archived, enabled, paused, archived Default value : enabled, paused, archived.

query **campaignIdFilter**:*string* | Optional. A comma-delimited list of campaign identifiers.

query **adGroupIdFilter**:*string* | Optional. Restricts results to keywords associated with ad groups specified by identifier in the comma-delimited list.

query **targetIdFilter**:*string* | Optional. A comma-delimited list of target identifiers. Missing in official Amazon documentation

Returns:

ApiResponse

`get_negative_target_extended`(*self*, *targetId*, ***kwargs*) → ApiResponse:

Gets a negative targeting clause with extended fields. Note that this call returns the full set of negative targeting clause extended fields, but is less efficient than getNegativeTarget.

path **negativeTargetId**:*integer* | Required. The negative targeting clause identifier.

Returns:

ApiResponse

8.9 Creatives

Warning: Sponsored Display is not available for Sandbox endpoint

`class ad_api.api.sd.Creatives`(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
GET	/sd/creatives	Gets a list of creatives.
PUT	/sd/creatives	Updates one or more creatives.
POST	/sd/creatives	A POST request of one or more creatives.
POST	/sd/creatives/preview	Gets creative preview HTML.
GET	/sd/moderation/creatives	Gets a list of creative moderations

`list_creatives`(*self*, ***kwargs*) → ApiResponse:

Gets a list of creatives

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **adGroupIdFilter**:*string* | Optional. The returned array includes only creatives associated with ad group identifiers matching those specified in the comma-delimited string. Cannot be used in conjunction with the creativeIdFilter parameter.

query **creativeIdFilter**:*string* | Optional. The returned array includes only creatives with identifiers matching those specified in the comma-delimited string. Cannot be used in conjunction with the adGroupIdFilter parameter.

Returns:

ApiResponse

`edit_creatives`(*self*, ***kwargs*) → ApiResponse:

Updates one or more creatives.
Request Body (required)

[
CreativeUpdate {
creativeType (CreativeTypeInCreativeRequest > String) : The type of the creative. Enum [IMAGE, VIDEO]
properties* (CreativeProperties): anyOf
HeadlineCreativeProperties {
User-customizable properties of a creative with headline.
headline (string): A marketing phrase to display on the ad. This field is optional and mutable. Maximum number of characters allowed is 50. maxLength: 50
}
LogoCreativeProperties {
User-customizable properties of a creative with a logo.
brandLogo (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}

CustomImageCreativeProperties {
User-customizable properties of a custom image creative.
rectCustomImage (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
squareCustomImage (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
}
VideoCreativeProperties {
User-customizable properties of a video creative.
video (Video)
assetId* (string): The unique identifier of the video asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular video assetversion.
}
}
]

Returns:
ApiResponse

`create_creatives`(*self*, ***kwargs*) → ApiResponse:

A POST request of one or more creatives.
Request Body (required)

CreateCreative {
adGroupId* (number) : Unique identifier for the ad group associated with the creative.
creativeType (CreativeTypeInCreativeRequest > String) : The type of the creative. Enum [IMAGE, VIDEO]
properties* (CreativeProperties): anyOf
HeadlineCreativeProperties {
User-customizable properties of a creative with headline.
headline (string): A marketing phrase to display on the ad. This field is optional and mutable. Maximum number of characters allowed is 50. maxLength: 50
}
LogoCreativeProperties {
User-customizable properties of a creative with a logo.
brandLogo (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
CustomImageCreativeProperties {
User-customizable properties of a custom image creative.
rectCustomImage (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping

zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
squareCustomImage (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
}
VideoCreativeProperties {
User-customizable properties of a video creative.
video (Video)
assetId* (string): The unique identifier of the video asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular video assetversion.
}
}
Returns
ApiResponse

`list_moderation_creatives`(*self*, ***kwargs*) → ApiResponse:

Gets a list of creative moderations

query **language***:*string* | The language of the returned creative moderation metadata. Available values : en-US, es-MX, zh-CN, es-ES, it-IT, fr-FR, fr-CA, de-DE, ja-JP, ko-KR, en-GB, en-CA, hi-IN, en-IN, en-DE, en-ES, en-FR, en-IT, en-JP, en-AE, ar-AE

query **startIndex**:*integer* | Optional. 0-indexed record offset for the result set. Default value : 0

query **count**:*integer* | Optional. Number of records to include in the paged response. Defaults to max page size.

query **adGroupIdFilter**:*string* | Optional. The returned array includes only creatives associated with ad group identifiers matching those specified in the comma-delimited string. Cannot

be used in conjunction with the creativeIdFilter parameter.

query **creativeIdFilter**:*string* | Optional. The returned array includes only creatives with identifiers matching those specified in the comma-delimited string. Cannot be used in conjunction with the adGroupIdFilter parameter.

Returns:

ApiResponse

`show_creative_preview`(*self*, ***kwargs*) → ApiResponse:

A POST request of one or more creatives.
Request Body (required)

CreativePreviewRequest {

creative* (PreviewCreativeModel){

creativeType (CreativeTypeInCreativeRequest > String) : The type of the creative. Enum [IMAGE, VIDEO]

properties (CreativeProperties): anyOf

HeadlineCreativeProperties {

User-customizable properties of a creative with headline.

headline (string): A marketing phrase to display on the ad. This field is optional and mutable. Maximum number of characters allowed is 50. maxLength: 50

}

LogoCreativeProperties {

User-customizable properties of a creative with a logo.

brandLogo (Image)

{

assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.

assetVersion* (string): The identifier of the particular image assetversion.

croppingCoordinates

{

Optional cropping coordinates to apply to the image.

top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0

left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0

width* (integer): Pixel width of the cropping zone. minimum: 0

height* (integer): Pixel height of the cropping zone. minimum: 0

}

}

}

CustomImageCreativeProperties {

User-customizable properties of a custom image creative.

rectCustomImage (Image)

{

assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.

assetVersion* (string): The identifier of the particular image assetversion.

croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
squareCustomImage (Image)
{
assetId* (string): The unique identifier of the image asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular image assetversion.
croppingCoordinates
{
Optional cropping coordinates to apply to the image.
top* (integer): Pixel distance from the top edge of the cropping zone to the top edge of the original image. minimum: 0
left* (integer): Pixel distance from the left edge of the cropping zone to the left edge of the original image. minimum: 0
width* (integer): Pixel width of the cropping zone. minimum: 0
height* (integer): Pixel height of the cropping zone. minimum: 0
}
}
}
}
VideoCreativeProperties {
User-customizable properties of a video creative.
video (Video)
assetId* (string): The unique identifier of the video asset. This assetId comes from the Creative Asset Library.
assetVersion* (string): The identifier of the particular video assetversion.
}
}
previewConfiguration* (CreativePreviewConfiguration){
Optional configuration for creative preview.
size {
The slot dimension to render the creative. Sponsored Display creatives are responsive to a limited list of width and height pairs, including 300x250, 650x130, 245x250, 414x125, 600x160, 600x300, 728x90, 980x55, 320x50, 970x250 and 270x150.
width (integer)
height (integer)
}

products [
The products to preview. Currently only the first product is previewable.
{
asin (string): The ASIN of the product.
}
]
landingPageURL (string): This operation is a PREVIEW ONLY. The URL where customers will land after clicking on its link. Must be provided if a LandingPageType is set. Please note that if a single product ad sets the landing page url, only one product ad can be added to the ad group. Note that this field is not supported when using ASIN or SKU fields.
landingPageType (string): This operation is a PREVIEW ONLY. The type of the landingPage used. This field is completely optional and will be set in conjunction with the LandingPageURL to indicate the type of landing page that will be set. Note that this field is not supported when using ASIN or SKU fields. Enum: [STORE, MOMENT, OFF_AMAZON_LINK]
adName (string): This operation is a PREVIEW ONLY. The name of the ad. Note that this field is not supported when using ASIN or SKU fields.
isMobile (boolean): Preview the creative as if it is on a mobile environment.
isOnAmazon (boolean): Preview the creative as if it is on an amazon site or third party site. The main difference is whether the preview will contain an AdChoices icon.
}
}
Returns
ApiResponse

8.10 Brand Safety List

Warning: Sponsored Display is not available for Sandbox endpoint

`class ad_api.api.sd.BrandSafety(account='default', marketplace: Marketplaces = Marketplaces.EU, credentials=None, proxies=None, verify=True, timeout=None, debug=False, access_token=None)`

Endpoints available

Method	Endpoint	Description
GET	/sd/brandSafety/deny	Gets a list of websites/apps that are on the advertiser's Brand Safety Deny List.
POST	/sd/brandSafety/deny	Creates one or more domains to add to a Brand Safety Deny Lis
DELETE	/sd/brandSafety/deny	Archives all of the domains in the Brand Safety Deny List.
GET	/sd/brandSafety/{requestId}/results	Gets the results for the given request
GET	/sd/brandSafety/{requestId}/status	Gets the status for the given request
GET	/sd/brandSafety/status	List status of all Brand Safety List requests.

`list_brand_safety(self, **kwargs) → ApiResponse:`

Gets an array of websites/apps that are on the advertiser's Brand Safety Deny List. It can take up to 15 minutes from the time a domain is added/deleted to the time it is reflected in the deny list.

query **startIndex**:*integer* | Optional. Optional. Sets a cursor into the requested set of domains. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of domain objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first 1000 domains set startIndex=0 and count=1000. To return the next 1000 domains, set startIndex=1000 and count=1000, and so on. Defaults to max page size(1000).

Returns:

ApiResponse

`post_brand_safety`(*self*, ***kwargs*) → ApiResponse:

Creates one or more domains to add to a Brand Safety Deny List. The Brand Safety Deny List is at the advertiser level. It can take up to 15 minutes from the time a domain is added to the time it is reflected in the deny list.

Request Body

BrandSafetyPostRequest {
POST Request for Brand Safety
domains* (array) minItems: 0 maxItems: 10000 [
BrandSafetyDenyListDomain {
name* (string): The website or app identifier. This can be in the form of full domain (eg. 'example.com' or 'example.net'), or mobile app identifier (eg. 'com.example.app' for Android apps or '1234567890' for iOS apps). maxLength: 250
type* (BrandSafetyDenyListDomainType >> string): The domain type. Enum: [WEBSITE, APP]
}
]
}

Returns

ApiResponse

`delete_brand_safety`(*self*, ***kwargs*) → ApiResponse:

Archives all of the domains in the Brand Safety Deny List. It can take several hours from the time a domain is deleted to the time it is reflected in the deny list. You can check the status of the delete request by calling GET /sd/brandSafety/{requestId}/status. If the status is "COMPLETED", you can call GET /sd/brandSafety/deny to validate that your deny list has been successfully deleted.

Returns:

ApiResponse

`get_result_brand_safety_request`(*self*, *requestId*, ***kwargs*) → ApiResponse:

When a user adds domains to their Brand Safety Deny List, the request is processed asynchronously, and a requestId is provided to the user. This requestId can be used to view the request results for up to 90 days from when the request was submitted. The results provide the status of each domain in the given request. Request results may contain multiple pages. This endpoint will only be available once the request has completed processing. To see the status of the request you can call GET /sd/brandSafety/{requestId}/status. Note that this endpoint only lists the results of POST requests to /sd/brandSafety/deny - it does not reflect the results of DELETE requests.

query **requestId***:*string* | The ID of the request previously submitted.

query **startIndex**:*integer* | Optional. Sets a cursor into the requested set of domains. Use in conjunction with the count parameter to control pagination of the returned array. 0-indexed record offset for the result set, defaults to 0.

query **count**:*integer* | Optional. Sets the number of domain objects in the returned array. Use in conjunction with the startIndex parameter to control pagination. For example, to return the first 1000 domains set startIndex=0 and count=1000. To return the next 1000 domains, set startIndex=1000 and count=1000, and so on. Defaults to max page size(1000).

Returns:
ApiResponse

`get_status_brand_safety_request`(*self*, *requestId*, ***kwargs*) → ApiResponse:

When a user modifies their Brand Safety Deny List, the request is processed asynchronously, and a requestId is provided to the user. This requestId can be used to check the status of the request for up to 90 days from when the request was submitted.

query **requestId***:*string* | The ID of the request previously submitted.

Returns:
ApiResponse

`list_brand_safety_requests_history`(*self*, ***kwargs*) → ApiResponse:

List status of all Brand Safety List requests. The list will contain requests that were submitted in the past 90 days.

Returns:
ApiResponse

8.11 Budget Rules

`class` ad_api.api.sd.**BudgetRules**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
GET	/sd/campaigns/{campaignId}/budgetRules/budgetHistory	Gets the budget history for a campaign specified by identifier.
POST	/sd/budgetRules	Creates one or more budget rules.
GET	/sd/budgetRules	Get all budget rules created by an advertiser
PUT	/sd/budgetRules	Updates one or more budget rules.
GET	/sd/budgetRules/{budgetRuleId}	Gets a budget rule specified by identifier.
GET	/sd/budgetRules/{budgetRuleId}/campaigns	Gets all the campaigns associated with a budget rule
POST	/sd/campaigns/{campaignId}/budgetRules	Associates one or more budget rules to a campaign specified by identifer.
GET	/sd/campaigns/{campaignId}/budgetRules	Gets a list of budget rules associated to a campaign specified by identifier.
DELETE	/sd/campaigns/{campaignId}/budgetRules/{budgetRuleId}	Disassociates a budget rule specified by identifier from a campaign specified by identifier.

`get_budget_history`(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Gets the budget history for a campaign specified by identifier.

Param:

path **campaignId*** (number). The campaign identifier.

query **nextToken** (string). To retrieve the next page of results, call the same operation and specify this token in the request. If the nextToken field is empty, there are no further results.

query **pageSize*** (number). Sets a limit on the number of results returned. Maximum limit of pageSize is 30.

query **startDate*** (string). The start date of the budget history in YYYYMMDD format.

query **endDate*** (string). The end date of the budget history in YYYYMMDD format.

Returns:

ApiResponse

`create_budget_rules`(*self*, ***kwargs*) → ApiResponse:

Creates one or more budget rules.

Request Body

CreateSPBudgetRulesRequest {

budgetRulesDetails (array) [

maxItems: 25.

A list of budget rule details.

SPBudgetRuleDetails {

Object representing details of a budget rule for SP campaign

duration RuleDuration {

eventTypeRuleDuration EventTypeRuleDuration {

Object representing event type rule duration.

eventId*(string): The event identifier. This value is available from the budget rules recommendation API.

endDate(string): The event end date in YYYYMMDD format. Read-only.

eventName(string): The event name. Read-only.

startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.

}

dateRangeTypeRuleDuration DateRangeTypeRuleDuration {

Object representing date range type rule duration.

endDate(string): The end date of the budget rule in YYYYMMDD format. The end date is inclusive. Required to be equal or greater than *startDate*.

startDate*(string): The start date of the budget rule in YYYYMMDD format. The start date is inclusive. Required to be greater than or equal to current date.

eventName(string): The event name. Read-only.

startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.

}

}

recurrence Recurrence {

type (string): The frequency of the rule application. Enum: [‘DAILY’]

daysOfWeek (array). Object representing days of the week for weekly type rule. It is not required for daily recurrence type

DayOfWeek [

DayOfWeek(string): The day of the week. Enum: [‘MONDAY’, ‘TUESDAY’, ‘WEDNESDAY’, ‘THURSDAY’, ‘FRIDAY’, ‘SATURDAY’, ‘SUNDAY’]

]

}

ruleType* SPRuleType (string): The type of budget rule. SCHEDULE: A budget rule based on a start and end date. PERFORMANCE: A budget rule based on advertising performance criteria. Enum: [‘SCHEDULE’, ‘PERFORMANCE’]

budgetIncreaseBy budgetIncreaseBy {

type* BudgetChangeType (string): The value by which to update the budget of the budget rule. Enum: [‘PERCENT’]

value* number($double): The budget value.

}

name (string): The budget rule name. Required to be unique within a campaign. maxLength: 355

performanceMeasureCondition* PerformanceMeasureCondition {

metricName* PerformanceMetrics (string): The advertising performance metric. Enum: [ACOS, CTR, CVR, ROAS]

comparisonOperator* ComparisonOperator (string): The comparison operator. Enum: [GREATER_THAN, LESS_THAN, EQUAL_TO, LESS_THAN_OR_EQUAL_TO, GREATER_THAN_OR_EQUAL_TO]

threshold* number($double): The performance threshold value.

}

]

}

Returns:

ApiResponse

`list_budget_rules`(*self*, ***kwargs*) → ApiResponse:

Get all budget rules created by an advertiser

Param:

query **nextToken** (string). To retrieve the next page of results, call the same operation and specify this token in the request. If the nextToken field is empty, there are no further results.

query **pageSize*** (number). Sets a limit on the number of results returned. Maximum limit of pageSize is 30.

Returns:

ApiResponse

`edit_budget_rules`(*self*, ***kwargs*) → ApiResponse:

Updates one or more budget rules.

Request Body

UpdateSPBudgetRulesRequest {

budgetRulesDetails (array) [

maxItems: 25.

A list of budget rule details.

SPBudgetRule {

ruleState state (string): The budget rule state. Enum: [‘ACTIVE’, ‘PAUSED’]

lastUpdatedDate number($int64): Epoch time of budget rule update. Read-only.

createdDate number($int64): Epoch time of budget rule creation. Read-only.

ruleDetails SPBudgetRuleDetails {

Object representing details of a budget rule for SP campaign

duration RuleDuration {

eventTypeRuleDuration EventTypeRuleDuration {
Object representing event type rule duration.
eventId*(string): The event identifier. This value is available from the budget rules recommendation API.
endDate(string): The event end date in YYYYMMDD format. Read-only.
eventName(string): The event name. Read-only.
startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.
}
dateRangeTypeRuleDuration DateRangeTypeRuleDuration {
Object representing date range type rule duration.
endDate(string): The end date of the budget rule in YYYYMMDD format. The end date is inclusive. Required to be equal or greater than *startDate*.
startDate*(string): The start date of the budget rule in YYYYMMDD format. The start date is inclusive. Required to be greater than or equal to current date.
eventName(string): The event name. Read-only.
startDate(string): The event start date in YYYYMMDD format. Read-only. Note that this field is present only for announced events.
}
}
recurrence Recurrence {
type (string): The frequency of the rule application. Enum: [‘DAILY’]
daysOfWeek (array). Object representing days of the week for weekly type rule. It is not required for daily recurrence type
DayOfWeek [
DayOfWeek(string): The day of the week. Enum: [‘MONDAY’, ‘TUESDAY’, ‘WEDNESDAY’, ‘THURSDAY’, ‘FRIDAY’, ‘SATURDAY’, ‘SUNDAY’]
]
}
ruleType* SPRuleType (string): The type of budget rule. SCHEDULE: A budget rule based on a start and end date. PERFORMANCE: A budget rule based on advertising performance criteria. Enum: [‘SCHEDULE’, ‘PERFORMANCE’]
budgetIncreaseBy budgetIncreaseBy {
type* BudgetChangeType (string): The value by which to update the budget of the budget rule. Enum: [‘PERCENT’]
value* number($double): The budget value.
}
name (string): The budget rule name. Required to be unique within a campaign. maxLength: 355
performanceMeasureCondition* PerformanceMeasureCondition {
metricName* PerformanceMetrics (string): The advertising performance metric. Enum: [ACOS, CTR, CVR, ROAS]
comparisonOperator* ComparisonOperator (string): The comparison operator. Enum: [GREATER_THAN, LESS_THAN, EQUAL_TO,

LESS_THAN_OR_EQUAL_TO, GREATER_THAN_OR_EQUAL_TO
]
threshold* number($double): The performance threshold value.
}
ruleId* (string): The budget rule identifier.
ruleStatus (string): he budget rule status. Read-only.
}
]
}
Returns:
ApiResponse

`get_budget_rule`(*self*, *budgetRuleId*, ***kwargs*) → ApiResponse:

Gets a budget rule specified by identifier.
Param:

path **budgetRuleId*** (string). The budget rule identifier.
Returns:
ApiResponse

`get_campaigns_budget_rule`(*self*, *budgetRuleId*, ***kwargs*) → ApiResponse:

Gets all the campaigns associated with a budget rule
Param:

path **budgetRuleId*** (string). The budget rule identifier.
query **nextToken** (string). To retrieve the next page of results, call the same operation and specify this token in the request. If the nextToken field is empty, there are no further results.
query **pageSize*** (number). Sets a limit on the number of results returned. Maximum limit of pageSize is 30.
Returns:
ApiResponse

`create_campaign_budget_rules`(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Associates one or more budget rules to a campaign specified by identifer.
Param:

path **campaignId*** (number). The campaign identifier.
Returns:
ApiResponse

`get_budget_rules_campaign`(*self*, *campaignId*, ***kwargs*) → ApiResponse:

Gets a list of budget rules associated to a campaign specified by identifier.
Param:

path **campaignId*** (number). The campaign identifier.
Returns:
ApiResponse

`delete_budget_rule_campaign`(*self*, *campaignId*, *budgetRuleId*, ***kwargs*) → ApiResponse:

Disassociates a budget rule specified by identifier from a campaign specified by identifier.
Param:

path **campaignId*** (number). The campaign identifier.
path **budgetRuleId*** (string). The budget rule identifier.
Returns:
ApiResponse

8.12 Campaigns Budget Usage

class ad_api.api.sd.**CampaignsBudgetUsage**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
POST	/sd/campaigns/budget/usage	Budget usage API for SP campaigns

list_campaigns_budget_usage(*self*, *version: int = 1*, ***kwargs*) → ApiResponse:
Budget usage API for SD campaigns
Request Body

BudgetUsageCampaignRequest {
campaignIds* (array) maxItems: 100 [
A list of campaign IDs (string)
]
}
Returns
ApiResponse

8.13 Forecasts

class ad_api.api.sd.**Forecast**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
POST	/sd/forecasts	Return forecasts for an ad group that may or may not exist.

list_forecasts(*self*, ***kwargs*) → ApiResponse:
Return forecasts for an ad group that may or may not exist.
Request Body

SDForecastRequest {
}
Returns
ApiResponse

8.14 Recommendations

class ad_api.api.sd.**Recommendations**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Endpoints available

Method	Endpoint	Description
POST	/sd/recommendations/creative/headline	Retrieve creative headline recommendations.

list_headline_recommendations(*self*, ***kwargs*) → ApiResponse:

You can use this Sponsored Display API to retrieve creative headline recommendations from an array of ASINs.

Request Body

SDHeadlineRecommendationRequest {

Request structure of SD headline recommendation API.

asins (array) minItems: 0 maxItems: 100 [

(string) An array of ASINs associated with the creative.

]

maxNumRecommendations (array) Maximum number of recommendations that API should return. Response will [0, maxNumRecommendations] recommendations (recommendations are not guaranteed as there can be instances where the ML model can not generate policy compliant headlines for the given set of asins). maximum: 10. minimum: 1 .

adFormat (string): Enum: [SPONSORED_DISPLAY]

}

Returns

ApiResponse

CHAPTER

NINE

AMAZON DSP

Amazon Advertising API - DSP

Documentation: https://advertising.amazon.com/API/docs/en-us/dsp-reports-beta-3p/#/Reports Amazon DSP is a demand-side platform (DSP) that enables advertisers to programmatically buy display, video, and audio ads both on and off Amazon.

9.1 Reports

class ad_api.api.dsp.**Reports**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Amazon DSP Reports

Documentation: https://advertising.amazon.com/API/docs/en-us/dsp-reports-beta-3p/#/Reports

Amazon DSP is a demand-side platform (DSP) that enables advertisers to programmatically buy display, video, and audio ads both on and off Amazon. Using Amazon's DSP, you can reach audiences across the web on both Amazon sites and apps as well as through our publishing partners and third-party exchanges. Amazon DSP is available to both advertisers who sell products on Amazon and those who do not.

post_report(*self*, *dspAccountId*, *accept='application/vnd.dspcreatereports.v3+json'*, ***kwargs*) → ApiResponse:

Request reports with performance metrics for DSP campaigns.

Request creation of a report that includes metrics about your Amazon DSP campaigns. Specify the type of report and the metrics you'd like to include. Note that the value specified for the dimensions field affects the metrics included in the report. See the dimensions field description for more information.
Keyword Args

path **dspAccountId** (string): Account Identifier you use to access the DSP. This is your DSP Entity ID if you have access to all DSP advertisers within that entity, or your DSP Advertiser ID if you only have access to a specific advertiser ID. [required]

Request body

advertiserIds (list > string): [optional] List of advertisers specified by identifier to include in the report. This should not be present if accountId is advertiser.

endDate (string): [required] Date in yyyy-MM-dd format. The report contains only metrics generated on the specified date range between startDate and endDate. The maximum date range between startDate and endDate is 31 days. The endDate can be up to 90 days older from today.

format (string): [optional] Enum: The report file format. [JSON]

orderIds (list > string): [optional] List of orders specified by identifier to include in the report.

metrics (list > string): [optional] Specify a list of metrics field names to include in the report. For example: [“impressions”, “clickThroughs”, “CTR”, “eCPC”, “totalCost”, “eCPM”]. If no metric field names are specified, only the default fields and selected DIMENSION fields are included by default. Specifying default fields returns an error.

type (string): [optional] Enum: The report type. [CAMPAIGN]

startDate (string): [required] Date in yyyy-MM-dd format. The report contains only metrics generated on the specified date range between startDate and endDate. The maximum date range between startDate and endDate is 31 days. The startDate can be up to 90 days older from today.

dimensions (list > string): [optional] List of dimensions to include in the report. Specify one or many comma-delimited strings of dimensions. For example: [“ORDER”, “LINE_ITEM”, “CREATIVE”]. Adding a dimension in this array determines the aggregation level of the report data and also adds the fields for that dimension in the report. If the list is null or empty, the aggregation of the report data is at ORDER level. The allowed values can be used together in this array as an allowed value in which case the report aggregation will be at the lowest aggregation level and the report will contain the fields for all the dimensions included in the report.

timeUnit (string): [optional] Enum: Adding timeUnit determines the aggregation level (SUMMARY or DAILY) of the report data. If the timeUnit is null or empty, the aggregation of the report data is at the SUMMARY level and aggregated at the time period specified. DAILY timeUnit is not supported for AUDIENCE report type. The report will contain the fields based on timeUnit. [SUMMARY]

Returns:

ApiResponse

Example python

```
from ad_api.api.dsp.reports import Reports

with open("campaign.json", "r", encoding="utf-8") as f:
    data = f.read()

dsp_account_id = '1111111111111'

result = Reports().post_report(
    dspAccountId=dsp_account_id,
    body=data
)

payload = result.payload
report_id = payload.get('reportId')
```

Example json

Open this `json` file to see the result:

```
{
  "type": "CAMPAIGN",
  "startDate": "2022-05-02",
  "endDate": "2022-05-30"
}
```

get_report(*self*, *dspAccountId*, *reportId*, *accept='application/vnd.dspgetreports.v3+json'*, ***kwargs*) → ApiResponse:

Gets a previously requested report specified by identifier.

Keyword Args

path **dspAccountId** (string): Account Identifier for DSP. Please input DSP entity ID if you want to retrieve reports for a group of advertisers, or input DSP advertiser ID if you want to retrieve reports for a single advertiser. [required]

path **reportId** (string): The identifier of the requested report. [required]

Returns:

ApiResponse

Example python

```
from ad_api.api.dsp.reports import Reports

dsp_account_id = '1111111111111'

# this report_id is obtained from post_report method
report_id = 'fac0e6e4-0ff7-b9a1-a3a1-9f528bacce9e'

result = Reports().get_report(
    dspAccountId=dsp_account_id,
    reportId=report_id
)
```

Result json

```
{'expiration': '2022-07-04T16:18:48.753Z',
 'format': 'JSON',
 'location': 'https://corvo-reports.s3.amazonaws.com/DSP_API/2022-07-04/fac0e6e4-
↪0ff7-b9a1-a3a1-9f528bacce9e/campaign-report-1111111111111.json?X-Amz-
↪Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20220704T151848Z&X-Amz-SignedHeaders=host&X-
↪Amz-Expires=3600&X-Amz-
↪Credential=**********************************************************&X-Amz-
↪Signature=**********************************************************',
 'reportId': 'fac0e6e4-0ff7-b9a1-a3a1-9f528bacce9e',
 'status': 'SUCCESS',
 'statusDetails': 'Success',
 'type': 'CAMPAIGN'}
```

download_report(*self*, ***kwargs*) → ApiResponse:

Downloads the report previously get report specified by location (this is not part of the official Amazon Advertising API, is a helper method to download the report). Take in mind that a direct download of location returned in get_report will return 401 - Unauthorized.

kwarg parameter **file** if not provided will take the default amazon name from path download (add a path with slash / if you want a specific folder, do not add extension as the return will provide the right extension based on format choosed if needed)

kwarg parameter **format** if not provided a format will return a url to download the report (this url has a expiration time)

Keyword Args

url (string): The location obatined from get_report [required]

file (string): The path to save the file if mode is download json, zip or gzip. [optional]

format (string): The mode to download the report: data (list), raw, url, json, zip, gzip. Default (url) [optional]

Returns:

ApiResponse

Example python

```
from ad_api.api.dsp.reports import Reports

# the url=location is obtained from get_report method need to in stay 'status':
↪'SUCCESS' if is 'IN_PROGRESS' the report cannot be downloaded
location = 'https://corvo-reports.s3.amazonaws.com/DSP_API/2022-07-04/fac0e6e4-0ff7-
↪b9a1-a3a1-9f528bacce9e/campaign-report-1111111111111.json?X-Amz-Algorithm=AWS4-
↪HMAC-SHA256&X-Amz-Date=20220704T151848Z&X-Amz-SignedHeaders=host&X-Amz-
↪Expires=3600&X-Amz-
↪Credential=**********************************************************************&X-Amz-
↪Signature=**********************************************************************'

# path = '/Users/your-profile/Downloads/report_name'
# mode = "data"  # "data (list), raw, url, json, zip, gzip default is url"

result = Reports().download_report(
    url=location,
    # file=path,
    # format=mode
)
```

CHAPTER

TEN

ADVERTISING TEST ACCOUNT

class ad_api.api.**AdvertisingTestAccount**(*account='default'*, *marketplace: Marketplaces = Marketplaces.EU*, *credentials=None*, *proxies=None*, *verify=True*, *timeout=None*, *debug=False*, *access_token=None*)

Create test advertising account for 3P API integrators

create_test_account(*body: dict or str*) → ApiResponse:

API to create test accounts

Submit a account creation request. You can create up to 1 test account type per marketplace.
Request body

countryCode (string): [required] Country code of the test account. [US, CA, MX, BR, UK, DE, FR, ES, IT, CN, JP, AU, AE, SA, NL]

accountMetaData (string): | vendorCode [optional] Vendor code that needs to be associated with the vendor account. example: ABCDE

accountType (string): [required] Type of test account. [VENDOR, AUTHOR]

Returns:

ApiResponse

Example python

```
import json
from ad_api.api import AdvertisingTestAccount

data = \
{
    "countryCode": "ES",
    "accountMetaData":
        {
            "vendorCode": "ABCDE"
        },
    "accountType": "VENDOR"
}

result = AdvertisingTestAccount(account=store, marketplace=marketplace, debug=True).
↪create_test_account(
    body=json.dumps(data)
)
```

```
{
'requestId': 'A7BCDGCEVXQ1CJJ4301V'
}
```

get_test_account(*requestId: str*) → ApiResponse

Keyword Args

query **requestId** (string): [required] request id.

Returns:

ApiResponse

Example python

```
from ad_api.api import AdvertisingTestAccount

request_id = "A7BCDGCEVXQ1CJJ4301V"

result = AdvertisingTestAccount(account=store, marketplace=marketplace, debug=True).
↪get_test_account(
    requestId=request_id
)
```

```
{
"accountType": "VENDOR",
"asins": [],
"countryCode": "ES",
"id": "ENTITY1MBW4T9T7Z5PC",
"status": "COMPLETED"
}
```

CHAPTER

ELEVEN

DISCLAIMER

```
MIT License
```

Copyright (c) 2021-2023 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 0.4.3: The v4 of *Sponsored Brand* Deprecated *Sponsored Products v2* Created new endpoints in *Sponsored Display*: Brand Safety, Budget Rules, Campaigns Budget Usage, Forecast and Recommendations. Created new endpoint in *Common Resources*: Tactical recommendations beta

```
import logging
from ad_api.base import AdvertisingApiException
from ad_api.api.sp import Campaigns

logging.basicConfig(
    level=logging.DEBUG,
    format="%(asctime)s:%(levelname)s:%(message)s"
)

try:
    states = 'enabled'
    result = Campaigns().list_campaigns_extended_request(
        stateFilter=states
    )

    campaigns = result.payload
    for campaign in campaigns:
        logging.info(campaign)
```

(continues on next page)

(continued from previous page)

```
except AdvertisingApiException as e:
    logging.error(e)
```

INDEX

C

D

E

F

G

H

I

K

L

M

N

P

R

S

T

U

V