

# Package ‘rappleads’

May 9, 2026

**Title** Get Data from 'Apple Ads Campaign Management API'

**Version** 0.1.3

**Description** Provides functions to load and manage data from Apple Ads accounts using the 'Apple Ads Campaign Management API'  
<[https://developer.apple.com/documentation/apple\\_ads](https://developer.apple.com/documentation/apple_ads)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** cli, dplyr, httr2, openssl, pbapply, purrr, rappdirs, retry, snakecase, stringr, tidyr, yaml

**Depends** R (>= 4.1.0)

**URL** <https://github.com/selesnow/rappleads>

**BugReports** <https://github.com/selesnow/rappleads/issues>

**NeedsCompilation** no

**Author** Alexey Seleznev [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-0410-7385>>),  
Netpeak [cph]

**Maintainer** Alexey Seleznev <[selesnow@gmail.com](mailto:selesnow@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-08-27 16:30:12 UTC

## Contents

apl_auth . . . . .	2
apl_get_access_token . . . . .	4
apl_get_ads . . . . .	4
apl_get_ad_groups . . . . .	5
apl_get_ad_group_report . . . . .	5
apl_get_ad_report . . . . .	6
apl_get_auth_account_list . . . . .	7

apl_get_budget_orders . . . . .	7
apl_get_campaigns . . . . .	8
apl_get_campaign_report . . . . .	8
apl_get_client_secret . . . . .	9
apl_get_creatives . . . . .	10
apl_get_keyword_report . . . . .	10
apl_get_me_details . . . . .	11
apl_get_search_term_report . . . . .	12
apl_get_user_acl . . . . .	13
apl_make_request . . . . .	13
apl_set_account_name . . . . .	14
apl_set_client_id . . . . .	14
apl_set_key_id . . . . .	15
apl_set_private_key_path . . . . .	15
apl_set_team_id . . . . .	16
apl_set_timezone . . . . .	16
make_selector . . . . .	17

## Index 18

---

apl_auth	<i>Apple Ads Authorization</i>
----------	--------------------------------

---

### Description

Apple Ads Authorization

### Usage

```
apl_auth(
  client_id = Sys.getenv("APL_CLIENT_ID"),
  team_id = Sys.getenv("APL_TEAM_ID"),
  key_id = Sys.getenv("APL_KEY_ID"),
  private_key_path = Sys.getenv("APL_PRIVATE_KEY_PATH"),
  account_name = Sys.getenv("APL_ACCOUNT_NAME"),
  cache_path = rappdirs::site_data_dir("rappleads")
)
```

### Arguments

client_id	You receive your clientId when you upload a public key.
team_id	The client secret is a JWT that you create and sign with your private key.
key_id	The value is your keyId that returns when you upload a public key.
private_key_path	Path to the .pem file containing your private key.
account_name	Your apple ads account name
cache_path	Path to the directory where cached authentication data will be stored.

## Details

This implementation process guides you through the following steps:

1. Invite users with API permissions.
2. Generate a private-public key pair.
3. Extract a public key from your persisted private key.
4. Upload a public key.
5. Set system environments variables.
6. Request an access token.

### Generate a Private Key:

API users need to create a private key. If you're using MacOS or a UNIX-like operating system, OpenSSL works natively. If you're on a Windows platform, you need to download [OpenSSL](#).  
`openssl ecparam -genkey -name prime256v1 -noout -out private-key.pem`

### Extract a Public Key:

`openssl ec -in private-key.pem -pubout -out public-key.pem` Open the public-key.pem file in a text editor and copy the public key, including the begin and end lines.

### Upload a Public Key:

Follow these steps to upload your public key:

1. From the Ads UI, choose Account Settings > API. Paste the key created in the above section into the Public Key field.
2. Click Save. A group of credentials displays as a code block above the public key field. Use your clientId, teamId, and keyId to create a client secret.

### Set system environments variables:

Run `usethis::edit_r_environ()` and set variables:

- APL\_CLIENT\_ID
- APL\_TEAM\_ID
- APL\_KEY\_ID
- APL\_PRIVATE\_KEY\_PATH
- APL\_ACCOUNT\_NAME

### Request an access token.:

Once the environment variables listed above are set, no further action is required from you — any function from the package will automatically request and refresh the access token when executed.

For more information see [API OAuth documentation](#).

## Value

character with access\_token

---

apl\_get\_access\_token    *Get access\_token*

---

### Description

Get access\_token

### Usage

```
apl_get_access_token(
  client_id,
  jwt_data,
  account_name,
  cache_path = rappdirs::site_data_dir("rappleads")
)
```

### Arguments

client_id	You receive your clientId when you upload a public key.
jwt_data	JWT object apl_get_client_secret()
account_name	Your apple ads account name
cache_path	Path to the directory where cached authentication data will be stored.

### Value

access\_token object

---

apl\_get\_ads            *Get Ads*

---

### Description

Get Ads

### Usage

```
apl_get_ads(org_id = apl_get_me_details()$parentOrgId)
```

### Arguments

org_id	The value is your orgId.
--------	--------------------------

### Value

tibble with ads metadata

### Examples

```
## Not run:  
ad <- apl_get_ads()  
  
## End(Not run)
```

---

apl\_get\_ad\_groups      *Get Ad Groups*

---

### Description

Get Ad Groups

### Usage

```
apl_get_ad_groups(org_id = apl_get_me_details())$parentOrgId)
```

### Arguments

org\_id              The value is your orgId.

### Value

tibble with ad group metadata

### Examples

```
## Not run:  
ad_group <- apl_get_ad_groups()  
  
## End(Not run)
```

---

apl\_get\_ad\_group\_report      *Get Ad Group-Level Reports*

---

### Description

Get Ad Group-Level Reports

**Usage**

```
apl_get_ad_group_report(
  org_id = apl_get_me_details()$parentOrgId,
  campaign_id = apl_get_campaigns()$id,
  start_date = Sys.Date() - 8,
  end_date = Sys.Date() - 1,
  group_by = NULL,
  granularity = c("DAILY", "HOURLY", "WEEKLY", "MONTHLY")
)
```

**Arguments**

org_id	The value is your orgId.
campaign_id	The unique identifier for the campaign.
start_date	Start reporting date
end_date	End reporting date
group_by	Use the groupBy field to group responses by selected dimensions. If groupBy specifies age, gender, and geodimensions
granularity	The report data organized by hour, day, week, and month.

**Value**

tibble with ad group report

**Examples**

```
## Not run:
adgr_rep <- apl_get_ad_group_report()

## End(Not run)
```

---

apl_get_ad_report	<i>Get Ad-Level Reports</i>
-------------------	-----------------------------

---

**Description**

Get Ad-Level Reports

**Usage**

```
apl_get_ad_report(
  org_id = apl_get_me_details()$parentOrgId,
  campaign_id = apl_get_campaigns()$id,
  start_date = Sys.Date() - 8,
  end_date = Sys.Date() - 1,
  group_by = NULL,
  granularity = c("DAILY", "HOURLY", "WEEKLY", "MONTHLY")
)
```

**Arguments**

org_id	The value is your orgId.
campaign_id	The unique identifier for the campaign.
start_date	Start reporting date
end_date	End reporting date
group_by	Use the groupBy field to group responses by selected dimensions. If groupBy specifies age, gender, and geodimensions
granularity	The report data organized by hour, day, week, and month.

**Value**

tibble ad report

---

apl\_get\_auth\_account\_list  
*Get List of Auth Cached Accounts*

---

**Description**

Get List of Auth Cached Accounts

**Usage**

```
apl_get_auth_account_list()
```

**Value**

character vector with accounts name

---

apl\_get\_budget\_orders *Get All Budget Orders*

---

**Description**

Get All Budget Orders

**Usage**

```
apl_get_budget_orders(org_id = apl_get_me_details()$parentOrgId)
```

**Arguments**

org_id	The value is your orgId.
--------	--------------------------

**Value**

tibble with budget data

---

apl\_get\_campaigns      *Get All Campaigns*

---

**Description**

Get All Campaigns

**Usage**

```
apl_get_campaigns(org_id = apl_get_me_details())$parentOrgId)
```

**Arguments**

org\_id              The value is your orgId.

**Value**

tibble with campaigns

**Examples**

```
## Not run:  
camp <- apl_get_campaigns(org_id = 2623170)  
  
## End(Not run)
```

---

apl\_get\_campaign\_report  
                          *Get Campaign-Level Reports*

---

**Description**

Get Campaign-Level Reports

**Usage**

```
apl_get_campaign_report(  
  org_id = apl_get_me_details())$parentOrgId,  
  start_date = Sys.Date() - 8,  
  end_date = Sys.Date() - 1,  
  group_by = NULL,  
  granularity = c("DAILY", "HOURLY", "WEEKLY", "MONTHLY")  
)
```

**Arguments**

org_id	The value is your orgId.
start_date	Start reporting date
end_date	End reporting date
group_by	Use the groupBy field to group responses by selected dimensions. If groupBy specifies age, gender, and geodimensions
granularity	The report data organized by hour, day, week, and month.

**Value**

tibble with report data

**Examples**

```
## Not run:
camp_rep <- apl_get_campaign_report(
  org_id      = 2623170,
  start_date  = '2025-07-20',
  end_date    = '2025-07-26',
  group_by    = 'countryOrRegion'
)

## End(Not run)
```

---

apl\_get\_client\_secret *Get client secret*

---

**Description**

Get client secret

**Usage**

```
apl_get_client_secret(
  client_id,
  team_id,
  key_id,
  private_key_path,
  account_name,
  cache_path = rappdirs::site_data_dir("rappleads")
)
```

**Arguments**

client_id	You receive your clientId when you upload a public key.
team_id	The client secret is a JWT that you create and sign with your private key.
key_id	The value is your keyId that returns when you upload a public key.
private_key_path	Path to the .pem file containing your private key.
account_name	Your apple ads account name
cache_path	Path to the directory where cached authentication data will be stored.

**Value**

jwt\_data

---

apl\_get\_creatives      *Get All Creatives*

---

**Description**

Get All Creatives

**Usage**

```
apl_get_creatives(org_id = apl_get_me_details())$parentOrgId)
```

**Arguments**

org_id	The value is your orgId.
--------	--------------------------

**Value**

tibble with campaigns

---

apl\_get\_keyword\_report  
*Get Keyword-Level Reports*

---

**Description**

Get Keyword-Level Reports

**Usage**

```
apl_get_keyword_report(  
  org_id = apl_get_me_details()$parentOrgId,  
  campaign_id = apl_get_campaigns()$id,  
  start_date = Sys.Date() - 8,  
  end_date = Sys.Date() - 1,  
  group_by = NULL,  
  granularity = c("DAILY", "HOURLY", "WEEKLY", "MONTHLY")  
)
```

**Arguments**

org_id	The value is your orgId.
campaign_id	The unique identifier for the campaign.
start_date	Start reporting date
end_date	End reporting date
group_by	Use the groupBy field to group responses by selected dimensions. If groupBy specifies age, gender, and geodimensions
granularity	The report data organized by hour, day, week, and month.

**Value**

tibble with keyword report

**Examples**

```
## Not run:  
keyw_rep <- apl_get_keyword_report()  
  
## End(Not run)
```

---

apl\_get\_me\_details      *Get Me Details*

---

**Description**

Get Me Details

**Usage**

```
apl_get_me_details()
```

**Value**

list with userId and parentOrgId

## Examples

```
## Not run:  
me <- apl_get_me_details()  
  
## End(Not run)
```

---

apl\_get\_search\_term\_report  
*Get Search Term-Level Reports*

---

## Description

Get Search Term-Level Reports

## Usage

```
apl_get_search_term_report(  
  org_id = apl_get_me_details()$parentOrgId,  
  campaign_id = apl_get_campaigns()$id,  
  start_date = Sys.Date() - 8,  
  end_date = Sys.Date() - 1,  
  group_by = NULL,  
  granularity = c("DAILY", "HOURLY", "WEEKLY", "MONTHLY")  
)
```

## Arguments

org_id	The value is your orgId.
campaign_id	The unique identifier for the campaign.
start_date	Start reporting date
end_date	End reporting date
group_by	Use the groupBy field to group responses by selected dimensions. If groupBy specifies age, gender, and geodimensions
granularity	The report data organized by hour, day, week, and month.

## Value

tibble search term report

## Examples

```
## Not run:  
ster_rep <- apl_get_search_term_report(start_date = '2025-06-01')  
  
## End(Not run)
```

---

apl\_get\_user\_acl      *Get User Access Control List*

---

**Description**

Get User Access Control List

**Usage**

```
apl_get_user_acl()
```

**Value**

tibble

**Examples**

```
## Not run:  
acs <- apl_get_user_acl()  
  
## End(Not run)
```

---

apl\_make\_request      *Make API request*

---

**Description**

Make API request

**Usage**

```
apl_make_request(endpoint, parser, org_id = NULL, selector = NULL)
```

**Arguments**

endpoint	API endpoint
parser	Parser function
org_id	Organization ID
selector	<b>Selector</b> objects define what data the API returns when fetching resources.

**Value**

API parsed response

---

apl\_set\_account\_name    *Set Apple Ads Account Name*

---

**Description**

Set Apple Ads Account Name

**Usage**

apl\_set\_account\_name(account\_name)

**Arguments**

account\_name    Your apple ads account name

**Value**

nothing

---

apl\_set\_client\_id    *Set Client ID*

---

**Description**

Set Client ID

**Usage**

apl\_set\_client\_id(client\_id)

**Arguments**

client\_id    You receive your clientId when you upload a public key.

**Value**

nothing

---

apl_set_key_id	<i>Set KeyID</i>
----------------	------------------

---

**Description**

Set KeyID

**Usage**

```
apl_set_key_id(key_id)
```

**Arguments**

key_id	The value is your keyId that returns when you upload a public key.
--------	--

**Value**

nothing

---

apl_set_private_key_path	<i>Set Privat Key Path</i>
--------------------------	----------------------------

---

**Description**

Set Privat Key Path

**Usage**

```
apl_set_private_key_path(private_key_path)
```

**Arguments**

private_key_path	Path to the .pem file containing your private key.
------------------	--

**Value**

nothing

---

apl_set_team_id	<i>Set Team ID</i>
-----------------	--------------------

---

**Description**

Set Team ID

**Usage**

```
apl_set_team_id(team_id)
```

**Arguments**

team_id	The client secret is a JWT that you create and sign with your private key.
---------	--

**Value**

nothing

---

apl_set_timezone	<i>Set Tinezone</i>
------------------	---------------------

---

**Description**

Set Tinezone

**Usage**

```
apl_set_timezone(timezone)
```

**Arguments**

timezone	You set the default timeZone during account creation through the Apple Search Ads UI. ORTZ (organization time zone) is the default. Possible Values: ORTZ, UTC
----------	--

**Value**

nothing

---

make_selector	<i>Make request selector</i>
---------------	------------------------------

---

## Description

Make request selector

## Usage

```
make_selector(  
  start_date = Sys.Date() - 8,  
  end_date = Sys.Date() - 1,  
  granularity = c("DAILY", "HOURLY", "WEEKLY", "MONTHLY"),  
  sort_field = "startTime",  
  time_zone = getOption("apl.time_zone"),  
  group_by = NULL,  
  part = NULL  
)
```

## Arguments

start_date	start report date
end_date	end report date
granularity	The report data organized by hour, day, week, and month.
sort_field	Specify a field name and grouping to sort the records by ASCENDING or DESCENDING.
time_zone	You set the default timeZone during account creation through the Apple Search Ads UI. ORTZ
group_by	Use the groupBy field to group responses by selected dimensions. If groupBy specifies age, gender, and geodimensions
part	Part of selector

## Details

See [API Documentation](#)

# Index

[apl\\_auth](#), [2](#)  
[apl\\_get\\_access\\_token](#), [4](#)  
[apl\\_get\\_ad\\_group\\_report](#), [5](#)  
[apl\\_get\\_ad\\_groups](#), [5](#)  
[apl\\_get\\_ad\\_report](#), [6](#)  
[apl\\_get\\_ads](#), [4](#)  
[apl\\_get\\_auth\\_account\\_list](#), [7](#)  
[apl\\_get\\_budget\\_orders](#), [7](#)  
[apl\\_get\\_campaign\\_report](#), [8](#)  
[apl\\_get\\_campaigns](#), [8](#)  
[apl\\_get\\_client\\_secret](#), [9](#)  
[apl\\_get\\_creatives](#), [10](#)  
[apl\\_get\\_keyword\\_report](#), [10](#)  
[apl\\_get\\_me\\_details](#), [11](#)  
[apl\\_get\\_search\\_term\\_report](#), [12](#)  
[apl\\_get\\_user\\_acl](#), [13](#)  
[apl\\_make\\_request](#), [13](#)  
[apl\\_set\\_account\\_name](#), [14](#)  
[apl\\_set\\_client\\_id](#), [14](#)  
[apl\\_set\\_key\\_id](#), [15](#)  
[apl\\_set\\_private\\_key\\_path](#), [15](#)  
[apl\\_set\\_team\\_id](#), [16](#)  
[apl\\_set\\_timezone](#), [16](#)  
  
[make\\_selector](#), [17](#)