

# Package ‘rtiktokads’

May 9, 2026

**Title** Load Data from 'tiktok Marketing API'

**Version** 0.3.4

**Description**

Loading data from 'tiktok Marketing API' <<https://business-api.tiktok.com/portal>> by business centers, advertisers, budgets and reports.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** cli, dplyr, httr2, jsonlite, purrr, rappdirs, stringr, tidyr, utils

**Depends** R (>= 4.1.0)

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

**BugReports** <https://github.com/selesnow/rtiktokads/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** 2025-09-22 13:30:03 UTC

## Contents

tik_auth . . . . .	2
tik_get_ads . . . . .	2
tik_get_advertiser_accounts . . . . .	3
tik_get_advertiser_balance . . . . .	3
tik_get_advertiser_info . . . . .	4
tik_get_ad_groups . . . . .	5
tik_get_bc_partners . . . . .	6

tik_get_business_centers . . . . .	6
tik_get_campaigns . . . . .	7
tik_get_report . . . . .	7
tik_get_token_path . . . . .	9
tik_get_username . . . . .	10
tik_set_token_path . . . . .	10
tik_set_username . . . . .	11

## Index 12

---

tik\_auth *Authorization in 'TikTok Marketing API'*

---

### Description

Authorization in 'TikTok Marketing API'

### Usage

```
tik_auth(username = tik_get_username(), token_path = tik_get_token_path())
```

### Arguments

username	TikTok login
token_path	auth cache folder

### Value

tik\_access\_token object

---

tik\_get\_ads *Get Ads*

---

### Description

Get Ads

### Usage

```
tik_get_ads(advertiser_id, fields = NULL)
```

### Arguments

advertiser_id	Advertiser ID.
fields	Fields that you want to get.

### Value

tibble with Ads metadata

---

```
tik_get_advertiser_accounts
```

*Get Authorized Ad Accounts*

---

**Description**

Get Authorized Ad Accounts

**Usage**

```
tik_get_advertiser_accounts()
```

**Value**

tibble with advertisers name and id

**Examples**

```
## Not run:
advertisers <- tik_get_advertiser_accounts()

## End(Not run)
```

---

```
tik_get_advertiser_balance
```

*Get the Balance and Budget of Ad Accounts*

---

**Description**

Use this endpoint to obtain the balance of ad accounts in the Business Center. You can also use this endpoint to obtain the budget of the ad accounts owned by the Business Center in auto-allocation mode.

**Usage**

```
tik_get_advertiser_balance(bc_id, fields = NULL)
```

**Arguments**

bc_id	Business Center ID.
fields	A list of additional fields to return in the response. Supported values: <ul style="list-style-type: none"><li>• budget_remaining: The remaining budget.</li><li>• budget_frequency_restriction: Restrictions on the number of budget changes allowed for the current day.</li><li>• budget_amount_restriction: Restrictions on the minimum amount that can be changed for the budget.</li><li>• min_transferable_amount: Details of the minimal amount that you can transfer to the ad account.</li></ul>

**Value**

tibble with balance and budget info

**Examples**

```
## Not run:  
bc <- tik_get_business_centers()  
acs_balance <- tik_get_advertiser_balance(bc$bc_id)  
  
## End(Not run)
```

---

tik\_get\_advertiser\_info

*Get Ad Account Details*

---

**Description**

Get Ad Account Details

**Usage**

```
tik_get_advertiser_info(advertiser_ids, fields = NULL)
```

**Arguments**

**advertiser\_ids** List of advertiser IDs to query. You can obtain Advertiser IDs through the `tik_get_advertiser_accounts()` function

**fields** A list of information to be returned. Supported values:

- telephone\_number
- contacter
- currency
- cellphone\_number
- timezone
- advertiser\_id
- role
- company
- status
- description
- rejection\_reason
- address
- name
- language
- industry
- license\_no

- email
- license\_url
- country
- balance
- create\_time
- display\_timezone
- owner\_bc\_id
- company\_name\_editable.

**Value**

tibble with advertisers onfo

**Examples**

```
## Not run:  
advertisers <- tik_get_advertiser_accounts()  
advertisers_info <- tik_get_advertiser_info(advertisers$advertiser_id)  
  
## End(Not run)
```

---

tik_get_ad_groups	<i>Get Ad Groups</i>
-------------------	----------------------

---

**Description**

Get Ad Groups

**Usage**

```
tik_get_ad_groups(advertiser_id, fields = NULL)
```

**Arguments**

advertiser\_id    Advertiser ID.  
fields            Fields that you want to get.

**Value**

tibble with AdGroup meta data

tik\_get\_bc\_partners    *Get the Partners of a BC*

---

**Description**

Get the Partners of a BC

**Usage**

```
tik_get_bc_partners(bc_id)
```

**Arguments**

bc\_id            Business Center ID.

**Value**

tibble with list of partners

---

tik\_get\_business\_centers  
                          *Get Business Centers*

---

**Description**

Get Business Centers

**Usage**

```
tik_get_business_centers()
```

**Value**

tibble with business centers info

**Examples**

```
## Not run:  
bc <- tik_get_business_centers()  
  
## End(Not run)
```

---

tik_get_campaigns	<i>Get Campaigns</i>
-------------------	----------------------

---

**Description**

Get Campaigns

**Usage**

```
tik_get_campaigns(advertiser_id, fields = NULL)
```

**Arguments**

advertiser\_id Advertiser ID.  
fields Fields that you want to get.

**Value**

tibble with campaigns metadata

---

tik_get_report	<i>Get Marketing Report</i>
----------------	-----------------------------

---

**Description**

Use this function to create a synchronous report task. This function can currently return the reporting data of up to 20,000 advertisements. If your number of advertisements exceeds 20,000, please use campaign\_ids / adgroup\_ids / ad\_ids as a filter to obtain the reporting data of all advertisements in batches. If you use campaign\_ids / adgroup\_ids / ad\_ids as a filter, you can pass in up to 100 IDs at a time.

**Usage**

```
tik_get_report(  
  advertiser_id = NULL,  
  advertiser_ids = NULL,  
  bc_id = NULL,  
  service_type = NULL,  
  report_type = "BASIC",  
  data_level = "AUCTION_ADVERTISER",  
  dimensions = c("advertiser_id", "stat_time_day"),  
  metrics = "spend",  
  enable_total_metrics = FALSE,  
  start_date = Sys.Date() - 7,  
  end_date = Sys.Date() - 1,  
)
```

```

query_lifetime = FALSE,
multi_adv_report_in_utc_time = FALSE,
order_field = NULL,
order_type = NULL
)

```

## Arguments

advertiser_id	Advertiser ID. You need to pass in either advertiser_id/advertiser_ids or bc_id. <ul style="list-style-type: none"> <li>When report_type is set to BASIC or AUDIENCE, pass in advertiser_id or advertiser_ids. If you pass in both advertiser_id and advertiser_ids, advertiser_id will be ignored.</li> <li>When report_type is set to PLAYABLE_MATERIAL, CATALOG, or TT_SHOP, pass in advertiser_id.</li> <li>When report_type is set to BC, pass in bc_id.</li> </ul>
advertiser_ids	A list of advertiser IDs.
bc_id	ID of a Business Center that you have access to.
service_type	Ad service type. Not supported when report_type is BC. <ul style="list-style-type: none"> <li>AUCTION: auction ads, or both auction ads and reservation ads.</li> <li>RESERVATION (deprecated) : reservation ads.</li> </ul>
report_type	Report type: <ul style="list-style-type: none"> <li>BASIC: basic report.</li> <li>AUDIENCE: audience report.</li> <li>PLAYABLE_MATERIAL: playable ads report.</li> <li>CATALOG: DSA report.</li> <li>BC: Business Center report.</li> <li>TT_SHOP: GMV max ads report.</li> </ul>
data_level	The data level that you'd like to query in reports. <ul style="list-style-type: none"> <li>AUCTION_AD: auction ads or both auction ads and reservation ads, ad level.</li> <li>AUCTION_ADGROUP: auction ads or both auction ads and reservation ads, ad group level.</li> <li>AUCTION_CAMPAIGN: auction ads or both auction ads and reservation ads, campaign level.</li> <li>AUCTION_ADVERTISER: auction ads or both auction ads and reservation ads, advertiser level.</li> </ul>
dimensions	Grouping conditions. For example: ("campaign_id", "stat_time_day") indicates that both campaign_id and stat_time_day (days) are grouped. Different report types support different dimensions. <ul style="list-style-type: none"> <li>Supported dimensions in <b>basic reports</b></li> <li>Supported dimensions in <b>audience reports</b></li> <li>Supported dimensions in <b>playable ad reports</b></li> <li>Supported dimensions in <b>DSA reports</b></li> </ul>

- Supported dimensions in [Business Center reports](#)
- Supported dimensions in [GMV max ads reports](#)

metrics	Metrics to query. Different report types support different metrics. For supported metrics for each report type, see the corresponding articles under <a href="#">Report types</a> .
enable_total_metrics	Whether to enable the total added-up data for your requested metrics. When enable_total_metrics is enabled, we will provide the aggregate data for all pages as you query different pages. Under this condition, you only need to specify this field when requesting data for the first page.
start_date	Query start date (closed interval) in the format of YYYY-MM-DD. The date is based on the ad account time zone.
end_date	Query end date (closed interval) in the format of YYYY-MM-DD. The date is based on the ad account time zone.
query_lifetime	Whether to request the lifetime metrics. Default value: False. If query_lifetime = True, the start_date and end_date parameters will be ignored. The lifetime metric name is the same as the normal one.
multi_adv_report_in_utc_time	Whether to set the returned metrics in the local timezone of each respective advertiser.
order_field	Sorting field.
order_type	Sorting order. ASC or DESC.

**Details**

To take more details see [API documentation](#)

**Value**

tibble with report data

**Examples**

```
## Not run:
report <- tik_get_report(advertiser_id = '7499750467069771792')

## End(Not run)
```

---

tik_get_token_path	<i>Get Auth Cache Folder</i>
--------------------	------------------------------

---

**Description**

Get Auth Cache Folder

**Usage**

```
tik_get_token_path()
```

**Value**

character, cache folder path

---

tik_get_username	<i>Get Auth Username</i>
------------------	--------------------------

---

**Description**

Get Auth Username

**Usage**

```
tik_get_username()
```

**Value**

character, current auth username

---

tik_set_token_path	<i>Change Session Token Path</i>
--------------------	----------------------------------

---

**Description**

Change Session Token Path

**Usage**

```
tik_set_token_path(
  token_path = ifelse(getOption("tiktok.auth_cache_mode") == "global",
    rappdirs::site_data_dir("rtiktokads"), rappdirs::user_cache_dir("rtiktokads"))
)
```

**Arguments**

token\_path      auth cache folder

**Value**

using for side effect, no return value

---

*tik\_set\_username*      *Set Session Auth Username*

---

**Description**

Set Session Auth Username

**Usage**

`tik_set_username(username)`

**Arguments**

username      TikTok login

**Value**

using for side effect, no return value

# Index

tik\_auth, [2](#)  
tik\_get\_ad\_groups, [5](#)  
tik\_get\_ads, [2](#)  
tik\_get\_advertiser\_accounts, [3](#)  
tik\_get\_advertiser\_balance, [3](#)  
tik\_get\_advertiser\_info, [4](#)  
tik\_get\_bc\_partners, [6](#)  
tik\_get\_business\_centers, [6](#)  
tik\_get\_campaigns, [7](#)  
tik\_get\_report, [7](#)  
tik\_get\_token\_path, [9](#)  
tik\_get\_username, [10](#)  
tik\_set\_token\_path, [10](#)  
tik\_set\_username, [11](#)