

# Package: rmytarget (via r-universe)

June 1, 2026

**Type** Package

**Title** Load Data from 'MyTarget API v2 and v3'

**Version** 2.4.0

**Date** 2021-10-20

**Maintainer** Alexey Seleznev <selesnow@gmail.com>

**Description** Allows work with 'MyTarget Statistics API v2'  
<<https://target.my.com/adv/api-marketing/doc/stat-v2>> and  
'MyTarget Statistics API v3'  
<<https://target.my.com/adv/api-marketing/doc/stat-v2#statisticsv3>>  
load data by ads, campaigns, agency clients and statistic from  
your ads account.

**URL** <https://selesnow.github.io/rmytarget/>

**Imports** httr, dplyr, lubridate, stringr, purrr, tidyr

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Language** ru

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

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

**Config/pak/sysreqs** libicu-dev libssl-dev

**Repository** <https://cranhaven.r-universe.dev>

**Date/Publication** 2026-06-01 04:02:00 UTC

**RemoteUrl** <https://github.com/cranhaven/cranhaven.r-universe.dev>

**RemoteRef** package/rmytarget

**RemoteSha** 6231779d268052c5cb1ef49f148552ab9c7b60ed

**RemoteSubdir** rmytarget

## Contents

rmytarget-package	2
myTarAuth	4
myTarGetAdList	6
myTarGetCampaignList	7
myTarGetClientList	8
myTarGetFastStats	9
myTarGetGoalList	10
myTarGetGoalsStats	10
myTarGetInAppStats	11
myTarGetOfflineConversionsStats	12
myTarGetStats	13
myTarRefreshToken	19
myTarSetLogin	19
myTarSetTokenPath	20
<b>Index</b>	<b>21</b>

---

rmytarget-package	<i>Package for load data from 'MyTarget' API.</i>
-------------------	---

---

## Description

Load data from API 'MyTarget' into R, by agency clients and campaigns

## Details

The DESCRIPTION file:

```

Package:      rmytarget
Type:         Package
Title:        Load Data from 'MyTarget API v2 and v3'
Version:      2.4.0
Date:         2021-10-20
Authors@R:    c(person(given = "Alexey", family = "Seleznev", role = c("aut", "cre"), email = "selesnow@gmail.com", co
Maintainer:   Alexey Seleznev <selesnow@gmail.com>
Description:  Allows work with 'MyTarget Statistics API v2' <https://target.my.com/adv/api-marketing/doc/stat-v2> and
URL:          https://selesnow.github.io/rmytarget/
Imports:      httr, dplyr, lubridate, stringr, purrr, tidyr
Suggests:     knitr, rmarkdown
VignetteBuilder: knitr
Depends:      R (>= 3.5.0)
License:      MIT + file LICENSE
Language:     ru
Encoding:     UTF-8
RoxygenNote: 7.1.2
Author:       Alexey Seleznev [aut, cre] (<https://orcid.org/0000-0003-0410-7385>), Netpeak [cph]

```

## Index of help topics:

myTarAuth	Authentication in 'MyTarget API'
myTarGetAdList	Get ads list from 'MyTarget'
myTarGetCampaignList	Get campaign list from 'MyTarget'
myTarGetClientList	Get Agency Clients from 'MyTarget'
myTarGetFastStats	Returns Basic Statistics in Real Time
myTarGetGoalList	Get goal list from 'MyTarget'
myTarGetGoalsStats	Get Goal Statistics from 'MyTarget'
myTarGetInAppStats	Returns statistics on mobile app events attributed with advertising impressions myTarget by campaigns and banners
myTarGetOfflineConversionsStats	Returns Statistics on Events Attributed With 'myTarget' Ad Impressions From Offline Conversion Lists for Campaigns
myTarGetStats	Get stat by clients, campaigns or banners from 'API MyTarget'
myTarRefreshToken	Refresh Access Token
myTarSetLogin	Set MyTarget login
myTarSetTokenPath	Set path to auth cache
rmytarget-package	Paxkage for load data from 'MyTarget' API.

**Author(s)**

Alexey Seleznev

**References**

1. [MyTarget API documents](#) 2. [MyTarget API statistic methods](#) 3. [Authorization](#)

**Examples**

```
## Not run:
# Get objects
## Get ads
Ads <- myTarGetAdList(login = "my_test_client")

## Get campaigns
Campaigns <- myTarGetCampaignList(login = "my_test_client")

# Get statistic
# base metrics by campaigns
base_data <- myTarGetTotalData(date_from = Sys.Date() - 7,
                               date_to   = Sys.Date(),
                               object_type = "campaigns",
                               object_id  = Campaigns$id,
                               metrics   = "base",
```

```

stat_type = "day",
login = "my_test_client",
token_path = "mytarget_token")

# all metrics by campaigns
all_data <- myTarGetTotalData(date_from = Sys.Date() - 7,
                             date_to   = Sys.Date(),
                             object_type = "campaigns",
                             object_id = Campaigns$id,
                             metrics = "all",
                             stat_type = "day",
                             login = "my_test_client",
                             token_path = "mytarget_token")

# custom set of metric by campaigns
custom_data <- myTarGetTotalData(date_from = Sys.Date() - 7,
                                 date_to   = Sys.Date(),
                                 object_type = "campaigns",
                                 object_id = Campaigns$id,
                                 metrics = c("base", "tps", "viral"),
                                 stat_type = "day",
                                 login = "my_test_client",
                                 token_path = "mytarget_token")

# if have note objects id, base metrics by ads
base_data2 <- myTarGetStats(date_from = as.Date("2013-01-01"),
                            date_to   = Sys.Date(),
                            object_type = "banners",
                            metrics = "base",
                            stat_type = "day",
                            login = "my_test_client",
                            token_path = "mytarget_token")

## End(Not run)

```

---

myTarAuth

*Authentication in 'MyTarget API'*


---

## Description

Authentication in 'MyTarget API' by Code Grant Schema or other auth schemas. For detail you can see [documentation](#).

## Usage

```

myTarAuth(
  login = getOption("rmytarget.login"),
  grant_type = "client_credentials",
  client_id = getOption("rmytarget.client_id"),

```

```
    client_secret = getOption("rmytarget.client_secret"),
    agency_client_name = NULL,
    code_grant = getOption("rmytarget.code_grant_auth"),
    token_path = myTarTokenPath()
)
```

### Arguments

login	Account name, used in file name if you save credential
grant_type	Your account grant, get one of two values, "client_credentials" or "agency_client_credentials". Not use in Code Grant Schema.
client_id	Yoyr client ID. Not use in Code Grant Schema.
client_secret	Your client secret. Not use in Code Grant Schema.
agency_client_name	Your client user name. Only for "agency_client_credentials" grant_type.
code_grant	logical, Use code gran authorise schema, <a href="#">detail</a>
token_path	Path to directory where you save credential data.

### Value

No return value, called for side effects

### Author(s)

Alexey Seleznev

### See Also

API authorization [documentation](#) by 'MyTarget'.

### Examples

```
## Not run:
# Recomendation auth by code grant schema
myTarAuth(login = "my_account_name")

## End(Not run)
```

---

myTarGetAdList	<i>Get ads list from 'MyTarget'</i>
----------------	-------------------------------------

---

### Description

Get ads list from 'MyTarget'

### Usage

```
myTarGetAdList(  
  auth = NULL,  
  login = getOption("rmytarget.login"),  
  token_path = myTarTokenPath(),  
  request_speed = 1.2  
)
```

### Arguments

auth	MyTarget Authorization R object. See <a href="#">myTarAuth</a>
login	Account name, used in file name if you save credential
token_path	Path to directory where you save credential data.
request_speed	Integer, or one of slow, normal, fast. Pause between requests to API

### Value

Data frame with banners list

### See Also

[MyTarget API Documentation](#)

### Examples

```
## Not run:  
myAds <- myTarGetAdList(login = "my_test_client")  
  
## End(Not run)
```

---

myTarGetCampaignList *Get campaign list from 'MyTarget'*

---

### Description

Get campaign list from 'MyTarget'

### Usage

```
myTarGetCampaignList(  
  auth = NULL,  
  login = getOption("rmytarget.login"),  
  token_path = myTarTokenPath(),  
  request_speed = 1.2  
)
```

### Arguments

auth	MyTarget Autherization R object. See <a href="#">myTarAuth</a>
login	Account name, used in file name if you save credential
token_path	Path to directory where you save credential data.
request_speed	Integer, or one of slow, normal, fast. Pause between requests to API

### Value

Data frame with campaigns list

### See Also

[MyTarget API ocumentation](#)

### Examples

```
## Not run:  
myTargetCampaign <- myTarGetCampaignList(myTargetAuth)  
  
## End(Not run)
```

---

myTarGetClientList     *Get Agency Clients from 'MyTarget'*

---

### Description

Get Agency Clients from 'MyTarget'

### Usage

```
myTarGetClientList(  
  auth = NULL,  
  login = getOption("rmytarget.login"),  
  token_path = myTarTokenPath()  
)
```

### Arguments

auth	MyTarget Authorization R object. See <a href="#">myTarAuth</a>
login	Account name, used in file name if you save credential
token_path	Path to directory where you save credential data.

### Value

data frame with agency clients

### See Also

MyTarget API Documenation[https://target.my.com/doc/api/detailed/#resource\\_agency\\_clients](https://target.my.com/doc/api/detailed/#resource_agency_clients)

### Examples

```
## Not run:  
myTarSetLogin("Your Login")  
clients <- myTarGetClientList()  
  
## End(Not run)
```

---

myTarGetFastStats	<i>Returns Basic Statistics in Real Time</i>
-------------------	--

---

## Description

returns basic statistics on advertising objects in real time, without taking into account the filtering of incorrect traffic. The summary statistics can vary significantly.

## Usage

```
myTarGetFastStats(  
  object_type = "campaigns",  
  object_id = NULL,  
  auth = NULL,  
  login = getOption("rmytarget.login"),  
  token_path = myTarTokenPath()  
)
```

## Arguments

object_type	API object type, character value, apply one of campaigns, banners, users
object_id	ID of API object (id campaign or any object)
auth	R auth object
login	Your login, or client name in MyTarget account
token_path	Path to directory where you save credential data

## Value

tibble with fast statistics

## Examples

```
## Not run:  
rt_stat <- myTarGetFastStats()  
  
## End(Not run)
```

myTarGetGoalList      *Get goal list from 'MyTarget'*

---

### Description

Get goal list from 'MyTarget'

### Usage

```
myTarGetGoalList(  
  auth = NULL,  
  login = getOption("rmytarget.login"),  
  token_path = myTarTokenPath()  
)
```

### Arguments

auth                  MyTarget Authorization R object. See [myTarAuth](#)  
login                 Path to directory where you save credential data  
token\_path            Your login, or client name in MyTarget account

### Value

data frame with goals parameters

### See Also

[Goals API Documentation](#)

---

myTarGetGoalsStats      *Get Goal Statistics from 'MyTarget'*

---

### Description

Get Goal Statistics from 'MyTarget'

### Usage

```
myTarGetGoalsStats(  
  date_from = Sys.Date() - 7,  
  date_to = Sys.Date(),  
  object_type = "campaigns",  
  object_id = NULL,  
  attribution = c("impression", "conversion"),  
  conversion_type = c("postview", "postclick", "total"),
```

```

    auth = NULL,
    login = getOption("rmytarget.login"),
    token_path = myTarTokenPath()
)

```

### Arguments

date_from	Start date
date_to	End date
object_type	API object type, character value, one of campaigns, banners, users.
object_id	ID of API object (id campaign or any object).
attribution	Attributing by event time or impression time. Available options: conversion, impression, default.
conversion_type	Conversion type: postclick - postclick, postview - postview, total - total.
auth	MyTarget Authorization R object. See <a href="#">myTarAuth</a>
login	Path to directory where you save credential data
token_path	Your login, or client name in MyTarget account

### Value

data frame with goal statics

### See Also

[API Documentation](#)

---

myTarGetInAppStats	<i>Returns statistics on mobile app events attributed with advertising impressions myTarget by campaigns and banners</i>
--------------------	--

---

### Description

Returns statistics on mobile app events attributed with advertising impressions myTarget by campaigns and banners

### Usage

```

myTarGetInAppStats(
  date_from = Sys.Date() - 7,
  date_to = Sys.Date(),
  object_type = "campaigns",
  object_id = NULL,
  attribution = c("conversion", "impression"),
  conversion_type = c("postview", "postclick", "total"),

```

```

  auth = NULL,
  login = getOption("rmytarget.login"),
  token_path = myTarTokenPath()
)

```

### Arguments

date_from	Start date
date_to	End date
object_type	API object type, character value, apply one of campaigns, banners, users
object_id	ID of API object (id campaign or any object)
attribution	Attributing by event time or impression time. Available options: conversion, impression
conversion_type	Conversion type: postclick - postclick, postview - postview, total - total.
auth	R auth object
login	Your login, or client name in MyTarget account
token_path	Path to directory where you save credential data

### Value

tibble with fast statistics

### Examples

```

## Not run:
rt_stat <- myTarGetFastStats()

## End(Not run)

```

---

myTarGetOfflineConversionsStats

*Returns Statistics on Events Attributed With 'myTarget' Ad Impressions From Offline Conversion Lists for Campaigns*

---

### Description

Returns Statistics on Events Attributed With 'myTarget' Ad Impressions From Offline Conversion Lists for Campaigns

**Usage**

```
myTarGetOfflineConversionsStats(
  date_from = Sys.Date() - 7,
  date_to = Sys.Date(),
  object_id = NULL,
  auth = NULL,
  login = getOption("rmytarget.login"),
  token_path = myTarTokenPath()
)
```

**Arguments**

date_from	Start date
date_to	End date
object_id	Campaigns IDs
auth	R auth object
login	Your login, or client name in MyTarget account
token_path	Path to directory where you save credential data

**Value**

tibble with offline conversions statistics

---

myTarGetStats	<i>Get stat by clients, campaigns or banners from 'API MyTarget'</i>
---------------	--

---

**Description**

Get stat by clients, campaigns or banners from 'API MyTarget'

**Usage**

```
myTarGetStats(
  date_from = Sys.Date() - 7,
  date_to = Sys.Date(),
  object_type = "campaigns",
  object_id = NULL,
  stat_type = "day",
  metrics = "base",
  package_id = NULL,
  attribution = c("conversion", "impression", "default"),
  banner_status = NULL,
  campaign_status = NULL,
  sort_by = NULL,
  sort_direction = c("asc", "desc"),
)
```

```

  auth = NULL,
  login = getOption("rmytarget.login"),
  token_path = myTarTokenPath(),
  api_version = getOption("rmytarget.stat_api_version")
)

```

### Arguments

date_from	Start date
date_to	End date
object_type	API object type, character value, one of campaigns, banners, users.
object_id	ID of API object (id campaign or any object).
stat_type	Breakdown by day, possible values: day, summary.
metrics	Set of metrics or fields, see metrics section for more details, default "base", but you can load any of all, base, events, video, viral, uniques, tps, or go this <a href="#">My-Target Stat API v2</a> or <a href="#">MyTarget Stat API v3</a> .
package_id	List of package identifiers. Available for banner statistics.
attribution	Attributing by event time or impression time. Available options: conversion, impression, default.
banner_status	Banner statuses list, available for campaign and banner statistics. Possible values: all, active, blocked, deleted.
campaign_status	List of campaign statuses, available for campaign and banner statistics. Possible values: all, active, blocked, deleted.
sort_by	The field by which identifiers of campaigns, banners or users will be sorted. Available metrics: base, events, video, viral, carousel, tps, moat, playable, romi.
sort_direction	Sorting direction. Available options: asc, desc
auth	MyTarget Authorization R object. See <a href="#">myTarAuth</a>
login	Path to directory where you save credential data
token_path	Your login, or client name in MyTarget account
api_version	MyTarget Statistic API version: v2, v3

### Value

Data frame with statistic.

### Sets of metrics

You can load different sets of metrics, for this use metrics arguments, and set a vector containing the names of the desired metric sets. For more details of v2 API go [this link](#). For more details of v3 API go [this link](#). For example: metrics = c("base", "video", "viral")

**base - based metrics.**

**shows** Number of impressions

**clicks** Number of clicks

**goals** Number of goals achieved

**spent** Cost sum

**cpm** Average cost per 1000 views

**cpc** Average cost per click

**cpa** Average cost per goals

**ctr** Percentage of clicks to views

**cr** Percentage ratio of the number of goals achieved to the number of clicks.

**events - metrics for advertised posts on social media feeds.**

**opening\_app** Number of discoveries of the advertised social networks application

**opening\_post** Number of discoveries of the advertised message in the social media feed

**moving\_into\_group** Number of transitions to the group page from the advertised message

**clicks\_on\_external\_url** Number of clicks on the external link in the advertised message

**launching\_video** Number of video launches in advertised message

**comments** Number of comments left in the advertised message

**joinings** Number of joining the group through the advertised message

**likes** Number of likes of the advertised message

**shares** Number of action "Share" for the advertised message.

**votings** Number of voting actions in the advertised message

**uniques - metrics by the number of unique users.**

**reach** Number of unique users who saw the ad for the specified period

**total** Number of unique users who saw an ad for all time

**increment** Number of new unique users who saw the ad for the specified period

**frequency** Average frequency of displaying ads to one unique user

**video - metrics for video ads.**

**started** Number of video playback starts

**paused** Number of pauses of video playback

**resumed\_after\_pause** Number of video playback after pause

**fullscreen\_on** Number of full-screen video playbacks

**fullscreen\_off** Number of shutdowns of full-screen video playback

**sound\_turned\_off** Number of video mute

**sound\_turned\_on** Number of video sound starts

**viewed\_10\_seconds** Number of views of the first 10 seconds of the video

**viewed\_25\_percent** Number of views of the first 25 percent of the video duration

**viewed\_50\_percent** Number of views of the first 50 percent of the video duration

**viewed\_75\_percent** Number of views of the first 75 of the video duration

**viewed\_100\_percent** Number of views 100 percent of the video duration

**viewed\_10\_seconds\_rate** Percentage of views with the achievement of the first 10 seconds of the video

**viewed\_25\_percent\_rate** Percentage of views with the achievement of the first 25 percent of the video duration

**viewed\_50\_percent\_rate** Percentage of views with the achievement of the first 50 percent of the video duration

**viewed\_75\_percent\_rate** Percentage of views with the achievement of the first 75 percent of the video duration

**viewed\_100\_percent\_rate** Percentage of views with the achievement of the first 100 percent of the video duration

**depth\_of\_view** Average video viewing depth (percent)

**view\_10\_seconds\_cost** Average cost of watching the first 10 seconds of a video

**viewed\_25\_percent\_cost** Average viewing cost of the first 25 percent of video length

**viewed\_50\_percent\_cost** Average viewing cost of the first 50 percent of video length

**viewed\_75\_percent\_cost** Average viewing cost of the first 75 percent of video length

**viewed\_100\_percent\_cost** Average viewing cost of the first 100 percent of video length

**viral - metrics of viral events.**

**viral\_impressions** Number of impressions of the shared advertising message in social networks

**viral\_reach** Number of unique users who saw the shared advertising message for the specified period

**viral\_total** Total number of unique users who have seen the shared advertising message for all time

**viral\_increment** Number of new unique users who saw the shared advertising message for the specified period

**viral\_frequency** Average frequency of displaying a shared advertising message to one unique user

**viral\_opening\_app** Number of openings of the advertised application from the shared advertising message

**viral\_opening\_post** Number of discoveries of the shared advertised message in the social media feed

**viral\_moving\_into\_group** number of transitions to the group page from the shared advertised message

**viral\_clicks\_on\_external\_url** Number of clicks on the external link in the shared advertised message

**viral\_launching\_video** Number of video launches in the shared advertised message

**viral\_comments** Number of comments left in the shared advertised message

**viral\_joinings** Number of joining the group through the shared advertised message

**viral\_likes** Number of likes of the shared advertised message

**viral\_shares** Number of actions "Share" for the shared advertised message

**viral\_votings** Number of voting actions in the shared advertised message

**carousel - statistics on individual slides of the advertising carousel (N - from 1 to the number of slides).**

**slide\_N\_shows** Number of N slide shows

**slide\_N\_clicks** Number of clicks on slide N

**slide\_N\_ctr** Percentage ratio of clicks to the number of views on slide N

**tps - additional write-off statistics.**

**tps** Additional charges for using the moat service

**tpd** Additional charges for using third-party data (from dmp).

**moat - statistics according to the moat service.**

**impressions** Number of impressions

**in\_view** Number of visible shows

**never\_focused** Number of impressions in the inactive tab

**never\_visible** Number of impressions out of sight

**never\_50\_perc\_visible** Additional charges for using third-party data (from dmp).

**never\_1\_sec\_visible** Number of impressions with visibility duration less than 1 second

**human\_impressions** Number of verified impressions

**impressions\_analyzed** number of impressions analyzed

**in\_view\_percent** Number of impressions analyzed

**human\_and\_viewable\_perc** Percentage of visible hits

**never\_focused\_percent** Verified impression percentage

**never\_visible\_percent** Impression percentage in inactive tab

**never\_50\_perc\_visible\_percent** The percentage of orders with a zone of visibility of the ad is less than 50 percent

**never\_1\_sec\_visible\_percent** Percentage of impressions with visibility duration less than 1 second

**in\_view\_diff\_percent** Visible impression difference

**active\_in\_view\_time** Average time the ad is in view

**attention\_quality** Engagement level

**playable - Playable Ads metrics.**

**playable\_game\_open** Opening the game

**playable\_game\_close** Closing the game

**playable\_call\_to\_action** Clicks

**romi - Playable Ads metrics.**

**value** The given value of the event

**romi** Return on investment

**adv\_cost\_share** Ad spend share

**See Also**

[MyTarget API documentation](#)

**Examples**

```
## Not run:
# base metrics by campaigns
base_data <- myTarGetStats(date_from = Sys.Date() - 7,
                           date_to   = Sys.Date(),
                           object_type = "campaigns",
                           metrics = "base",
                           stat_type = "day",
                           login = "client_login",
                           token_path = "D:\\mytarget_token")

# all metrics by campaigns
all_data <- myTarGetStats(date_from = Sys.Date() - 7,
                          date_to   = Sys.Date(),
                          object_type = "campaigns",
                          metrics = "all",
                          stat_type = "day",
                          login = "client_login",
                          token_path = "D:\\mytarget_token")

# custom set of metric by campaigns
custom_data <- myTarGetStats(date_from = Sys.Date() - 7,
                              date_to   = Sys.Date(),
                              object_type = "campaigns",
                              metrics = c("base", "tps", "viral"),
                              stat_type = "day",
                              login = "client_login",
                              token_path = "D:\\mytarget_token")

# if have note objects id, base metrics by ads
base_data2 <- myTarGetStats(date_from = as.Date("2013-01-01"),
                             date_to   = Sys.Date(),
                             object_type = "banners",
                             metrics = "base",
                             stat_type = "day",
                             login = "client_login",
                             token_path = "D:\\mytarget_token")

# get all stats group by clients, only for agency account
client_stat <- myTarGetStats(date_from = Sys.Date() - 7,
                              date_to   = Sys.Date(),
                              object_type = "users",
                              metrics = "all",
                              login = "agency_login")

## End(Not run)
```

---

myTarRefreshToken	<i>Refresh Access Token</i>
-------------------	-----------------------------

---

**Description**

Refresh Access Token

**Usage**

```
myTarRefreshToken(old_auth = NULL, client_id = NULL, client_secret = NULL)
```

**Arguments**

old_auth	Your old auth object. See <a href="#">myTarAuth</a>
client_id	Your client ID
client_secret	Your client secret

**Value**

No return value, called for side effects

**Examples**

```
## Not run:  
auth <- myTarRefreshToken(old_auth = myTargetAuth,  
                          client_id = "xxxx",  
                          client_secret = "xxxx.")  
  
## End(Not run)
```

---

myTarSetLogin	<i>Set MyTarget login</i>
---------------	---------------------------

---

**Description**

Set MyTarget login

**Usage**

```
myTarSetLogin(login)
```

**Arguments**

login	Your login, or client name in MyTarget account
-------	--

**Value**

No return value, called for side effects

---

myTarSetTokenPath	<i>Set path to auth cache</i>
-------------------	-------------------------------

---

**Description**

Set path to auth cache

**Usage**

```
myTarSetTokenPath(token_path)
```

**Arguments**

token_path	Path to directory where you save credential data
------------	--

**Value**

No return value, called for side effects

# Index

## \* package

rmytarget-package, 2

myTarAuth, 4, 6–8, 10, 11, 14, 19

myTarGetAdList, 6

myTarGetCampaignList, 7

myTarGetClientList, 8

myTarGetFastStats, 9

myTarGetGoallList, 10

myTarGetGoalsStats, 10

myTarGetInAppStats, 11

myTarGetOfflineConversionsStats, 12

myTarGetStats, 13

myTarRefreshToken, 19

myTarSetLogin, 19

myTarSetTokenPath, 20

rmytarget (rmytarget-package), 2

rmytarget-package, 2