

# Package ‘rvkstat’

December 11, 2019

**Type** Package

**Title** Interface to API 'vk.com'

**Version** 2.6.3

**Date** 2019-12-11

**Author** Alexey Seleznev

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

**Description** Load data from vk.com api about your communiti users and views, ads performance, post on user wall and etc. For more detail see <[https://vk.com/dev/first\\_guide](https://vk.com/dev/first_guide)>.

**License** GPL-2

**Depends** R (>= 3.5.0)

**Imports** RCurl, jsonlite, httr, tidyr

**Language** ru

**Encoding** UTF-8

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

**URL** <http://selesnow.github.io/rvkstat>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-12-11 10:50:02 UTC

## R topics documented:

rvkstat-package . . . . .	2
vkAuth . . . . .	4
vkGetAdAccounts . . . . .	5
vkGetAdAgeStats . . . . .	6
vkGetAdBudget . . . . .	8
vkGetAdCampaigns . . . . .	9
vkGetAdCategories . . . . .	10
vkGetAdCityStats . . . . .	11

vkGetAdClients . . . . .	13
vkGetAdGenderAgeStats . . . . .	14
vkGetAdGenderStats . . . . .	16
vkGetAdPostsReach . . . . .	18
vkGetAds . . . . .	20
vkGetAdsLayout . . . . .	22
vkGetAdStatistics . . . . .	23
vkGetDbCities . . . . .	26
vkGetDbCountries . . . . .	27
vkGetDbRegions . . . . .	28
vkGetGroupStat . . . . .	29
vkGetGroupStatAge . . . . .	30
vkGetGroupStatCity . . . . .	31
vkGetGroupStatCountries . . . . .	32
vkGetGroupStatGender . . . . .	33
vkGetGroupStatGenderAge . . . . .	34
vkGetGroupStatPostReach . . . . .	35
vkGetGroupToken . . . . .	37
vkGetToken . . . . .	38
vkGetUserDialogs . . . . .	38
vkGetUserFriends . . . . .	40
vkGetUserGroups . . . . .	42
vkGetUserWall . . . . .	44
<b>Index</b>	<b>46</b>

---

rvkstat-package	<i>Interface to API 'vk.com'</i>
-----------------	----------------------------------

---

## Description

Load data from vk.com api about your communiti users and views, ads performance, post on user wall and etc. For more detail see <[https://vk.com/dev/first\\_guide](https://vk.com/dev/first_guide)>.

## Details

The DESCRIPTION file:

```

Package:      rvkstat
Type:        Package
Title:       Interface to API 'vk.com'
Version:     2.6.3
Date:       2019-12-11
Author:     Alexey Seleznev
Maintainer: Alexey Seleznev <selesnow@gmail.com>
Description: Load data from vk.com api about your communiti users and views, ads performance, post on user wall and etc.
License:    GPL-2
Depends:    R (>= 3.5.0)

```

Imports: RCurl, jsonlite, httr, tidyr  
 Language: ru  
 Encoding: UTF-8  
 BugReports: <https://github.com/selesnow/rvkstat/issues>  
 URL: <http://selesnow.github.io/rvkstat>

#### Index of help topics:

rvkstat-package	Interface to API 'vk.com'
vkAuth	Authorization in 'Vkontakte'
vkGetAdAccounts	List of Advertising Account
vkGetAdAgeStats	Statistics on Audience Coverage by Age.
vkGetAdBudget	Budget of Advertising Cabinet 'Vkontakte'.
vkGetAdCampaigns	Campaign list
vkGetAdCategories	List of Possible Subjects of Advertisements
vkGetAdCityStats	Statistics on Audience Coverage in The Context of Cities.
vkGetAdClients	Client list.
vkGetAdGenderAgeStats	Statistics on Audience Coverage by Age and Gender.
vkGetAdGenderStats	Statistics on Audience Coverage by Gender.
vkGetAdPostsReach	Statistics on The Coverage of Advertising Entries 'Vkontakte'.
vkGetAdStatistics	Statistics of advertising in 'VKontakte'.
vkGetAds	List of Ads From 'Vkontakte'.
vkGetAdsLayout	Appearance of Advertisements.
vkGetDbCities	City Dictionary.
vkGetDbCountries	Country Dictionary.
vkGetDbRegions	Regions list.
vkGetGroupStat	Statistics on visiting the community in 'Vkontakte'.
vkGetGroupStatAge	Community Statistics by User Age Bucket
vkGetGroupStatCity	Statistics of the Vkontakte community visitors by cities.
vkGetGroupStatCountries	Statistics of The Vkontakte Community Visitors By Country.
vkGetGroupStatGender	Loading data by gender of the community visitors
vkGetGroupStatGenderAge	Statistics on Age Groups And Gender of The 'VKontakte' Community.
vkGetGroupStatPostReach	Statistics of Posts on The Community Wall.
vkGetGroupToken	Get the Access Key of Community 'Vkontakte'.
vkGetToken	Get vkontakte developer token
vkGetUserDialogs	Returns The List of Conversations of The

vkGetUserFriends	Current User or Community. Frend List in 'Vkontakte'
vkGetUserGroups	List of 'Vkontakte' communities.
vkGetUserWall	Get Posts From Userwall.

**Author(s)**

Alexey Seleznev

Maintainer: Alexey Seleznev <selesnow@gmail.com>

---

vkAuth	<i>Authorization in 'Vkontakte'</i>
--------	-------------------------------------

---

**Description**

Allows you to get a token to access the API using the scheme Authorization Code Flow - [https://vk.com/dev/authcode\\_flow\\_us](https://vk.com/dev/authcode_flow_us)

**Usage**

```
vkAuth(app_id = 5344605,
       app_secret = "CAyGdHhdgEBEW8fJCmzS",
       apiVersion = "5.68")
```

**Arguments**

app_id	Application ID, you can find it on the application settings page.
app_secret	Secure application key, you can find it on the application settings page.
apiVersion	API version.

**Value**

The list of credentials to work with the API Vkontakte

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## Auth in VK
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## End(Not run)
```

---

vkGetAdAccounts	<i>List of Advertising Account</i>
-----------------	------------------------------------

---

**Description**

Returns a list of available advertising spaces.

**Usage**

```
vkGetAdAccounts(access_token = NULL, api_version = NULL)
```

**Arguments**

access_token	API access tokens obtained using vkAuth or vkGetToken functions
api_version	Vkontakte API version.

**Value**

Date frame with a list of advertising offices and their parameters with the following values:

account_id	ID of the advertising account.
account_type	Type of advertising office.
account_status	The status of the advertising office. (1 - active, 0 - inactive)
access_role	User privileges in the advertising office.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:  
## Auth in VK  
my_tok <- vkAuth(app_id = 1,  
                 app_secret = "H2Pk8htyFD8024mZaPHm")  
  
my_vk_acc <- vkGetAdAccounts(my_tok$access_token)  
  
## End(Not run)
```

---

vkGetAdAgeStats      *Statistics on Audience Coverage by Age.*

---

### Description

Returns statistics on the reach of advertising campaigns and ads by age.

### Usage

```
vkGetAdAgeStats(account_id = NULL, ids_type = "campaign", ids = NULL,
                period = "day", date_from = Sys.Date() - 30,
                date_to = Sys.Date(), api_version = NULL,
                access_token = NULL)
```

### Arguments

account_id	The ID of the advertising account, a list of all available advertising offices can be obtained using the function vkGetAdAccounts.
ids_type	The type of objects requested that are listed in the ids parameter. Valid values "ad" - are ads, "campaigns" - are campaigns.
ids	Vector id of the requested ads, campaigns, clients or account, depending on what is specified in the ids_type parameter.
period	The way data is grouped by date is day-by-day statistics, month is month-by-month statistics, overall is all-time statistics. Time limits are specified in the date_from and date_to arguments.
date_from	The start date of the displayed statistics in the format YYYY-MM-DD.
date_to	End date of the displayed statistics in the format YYYY-MM-DD.
access_token	API access tokens obtained using vkAuth or vkGetToken functions.
api_version	Vkontakte API version for send request.

### Value

Date frame with community statistics with the following values:

id	id of the object from the ids argument.
day	(if the period is equal to day) day in the format YYYY-MM-DD.
month	(if period is equal to month) - day in the format YYYY-MM.
day_from	(if period is overall) the start date of the reporting period.
day_to	(if period is overall) end date of the reporting period.
type	object type from the ids_type parameter.
age	age.
impressions_rate	part of the audience that viewed the ad, from 0 to 1.
clicks_rate	(the part of the audience that clicked on the ad, from 0 to 1.



```
## End(Not run)
```

---

vkGetAdBudget	<i>Budget of Advertising Cabinet 'Vkontakte'.</i>
---------------	---

---

### Description

Returns the current balance of the advertising account Vkontakte.

### Usage

```
vkGetAdBudget(account_id = NULL, access_token = NULL,  
              api_version = NULL)
```

### Arguments

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version.

### Author(s)

Alexey Seleznev

### Examples

```
## Not run:  
## auth  
my_tok <- vkAuth(app_id = 1,  
                app_secret = "H2Pk8htyFD8024mZaPHm")  
  
# Receive the rest of means from an advertising office  
vk_budget <- vkGetAdBudget(account_id = 1,  
                          access_token = my_tok$access_token)  
  
## End(Not run)
```

---

vkGetAdCampaigns	<i>Campaign list</i>
------------------	----------------------

---

**Description**

Returns a list of available advertising campaigns from the advertising office Vkontakte.

**Usage**

```
vkGetAdCampaigns(account_id = NULL, client_id = NULL, include_deleted =
                 TRUE, campaign_ids = "null", api_version = NULL,
                 access_token = NULL)
```

**Arguments**

account_id	The ID of the advertising account, a list of all available advertising offices can be obtained using the function vkGetAdAccounts.
client_id	Client identifier from which advertisements are requested. Available and required for advertising agencies.
include_deleted	Boolean TRUE or FALSE, flag, specifying the need to display archived ads.
campaign_ids	Campaign ids.
access_token	API access token, obtained using the functions vkAuth or vkGetToken.
api_version	Vkontakte API version for send request.

**Value**

Date frame with a list of advertising offices and their parameters with the following values:

id	campaign ID.
type	campaign type.
name	campaign name.
status	campaign status.
day_limit	daily campaign limit in rubles.
all_limit	total campaign limit in rubles.
start_time	campaign start time.
stop_time	campaign stop time.
create_time	campaign creation time.
update_time	the last time the campaign was modified.

**Author(s)**

Alexey Seleznev

## Examples

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 app_secret = "H2Pk8htyFD8024mZaPHm")

## get campaing list
my_vk_camp <- vkGetAdCampaigns(account_id = 1,
                               access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetAdCategories      *List of Possible Subjects of Advertisements*

---

## Description

Returns a table with all possible topics and their subsections of the topics of advertisements from 'Vkontakte'

## Usage

```
vkGetAdCategories(access_token = NULL, api_version = NULL)
```

## Arguments

access\_token      API access tokens obtained using vkAuth or vkGetToken functions  
api\_version      Vkontakte API version for send request.

## Value

Date frame with the following values:

id                      subject identifier.  
name                    subject name.  
subcategories\_id  
                        identifier of the subsection.  
ubcategories\_name  
                        subsection name.

## Author(s)

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 client_secret = "H2Pk8htyFD8024mZaPHm")

# list of subjects
vk_ad_categories <- vkGetAdCategories(access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetAdCityStats	<i>Statistics on Audience Coverage in The Context of Cities.</i>
------------------	--

---

**Description**

Returns statistics on audience reach by advertising campaign and ads by city.

**Usage**

```
vkGetAdCityStats(account_id = NULL, ids_type = "campaign", ids = NULL,
                 period = "day", date_from = Sys.Date() - 30, date_to =
                 Sys.Date(), api_version = NULL, access_token = NULL)
```

**Arguments**

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
ids_type	The type of objects requested, which are listed in the ids parameter. Valid ad values are ads, campaigns are campaigns.
ids	Vector id of the requested ads, campaigns, clients or account, depending on what is specified in the ids_type parameter.
period	The method of grouping data by date. day - statistics by days, month - statistics by months, overall - statistics for all time. Temporary restrictions are given by the arguments date_from and date_to.
date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version.



```

                                date_to = "2017-09-10",
                                access_token = my_tok$access_token)

# 3.Receive a static report on audience coverage in the context
# of cities by advertising campaigns without a temporary breakdown
## 3.1. We get a list of advertising campaigns
camp <- vkGetAdCampaigns(account_id = 1, access_token = my_tok$access_token)
##3.2. Get statistics on advertising campaigns
vk_ad_city_stat_total <- vkGetAdCityStats(account_id = 1,
                                           ids_type = "campaign",
                                           ids = camp$id ,
                                           period = "overall",
                                           date_from = "2010-01-01",
                                           date_to = "2017-09-10",
                                           access_token = my_tok$access_token)

## End(Not run)

```

---

vkGetAdClients	<i>Client list.</i>
----------------	---------------------

---

### Description

Returns a list of advertising agency clients.

### Usage

```
vkGetAdClients(account_id = NULL, api_version = NULL,
               access_token = NULL)
```

### Arguments

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version.

### Value

Date frame with community statistics with the following values:

id	client id.
name	customer name.
day_limit	client's daily limit in rubles.
all_limit	total customer limit in rubles.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 app_secret = "H2Pk8htyFD8024mZaPHm")

## We get a list of clients from the agent account
vk_clients <- vkGetAdClients(account_id = 1,
                             access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetAdGenderAgeStats *Statistics on Audience Coverage by Age and Gender.*

---

**Description**

Returns statistics on the reach of advertising campaigns and ads by age and sex.

**Usage**

```
vkGetAdGenderAgeStats(account_id = NULL, ids_type = "campaign", ids = NULL,
                      period = "day", date_from = Sys.Date() - 30, date_to =
                      Sys.Date(), api_version = NULL, access_token = NULL)
```

**Arguments**

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
ids_type	The type of objects requested, which are listed in the ids parameter. Valid ad values are ads, campaigns are campaigns.
ids	Vector id of the requested ads, campaigns, clients or account, depending on what is specified in the ids_type parameter.
period	The method of grouping data by date. day - statistics by days, month - statistics by months, overall - statistics for all time. Temporary restrictions are given by the arguments date_from and date_to.
date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
access_token	API access token, obtained using the functions vkAuth or vkGetToken.
api_version	Vkontakte API version.



```

period = "month",
date_from = "2010-01-01",
date_to = "2017-09-10",
access_token = my_tok$access_token)

# 3. Get a static message on reaching the audience
# for advertising campaigns without a temporary breakdown
## 3.1. We get a list of advertising campaigns
camp <- vkGetAdCampaigns(account_id = 1, access_token = my_tok$access_token)
## 3.2. We get statistics on advertising campaigns
vk_ad_gender_age_stat_total <- vkGetAdGenderAgeStats(account_id = 1,
                                                    ids_type = "campaign",
                                                    ids = camp$id ,
                                                    period = "overall",
                                                    date_from = "2010-01-01",
                                                    date_to = "2017-09-10",
                                                    access_token = my_tok$access_token)

## End(Not run)

```

---

vkGetAdGenderStats      *Statistics on Audience Coverage by Gender.*

---

### Description

Returns audience coverage statistics for advertising campaigns and ads by gender.

### Usage

```

vkGetAdGenderStats(account_id = NULL, ids_type = "campaign", ids = NULL,
                    period = "day", date_from = Sys.Date() - 30, date_to =
                    Sys.Date(), api_version = NULL, access_token = NULL)

```

### Arguments

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
ids_type	The type of objects requested, which are listed in the ids parameter. Valid values: ad - are ads, campaigns - are campaigns.
ids	Vector id of the requested ads, campaigns, clients or account, depending on what is specified in the ids_type parameter.
period	The method of grouping data by date. day - statistics by days, month - statistics by months, overall - statistics for all time. Temporary restrictions are given by the arguments date_from and date_to.
date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.

date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version.

**Value**

Date frame with community statistics with the following values:

1. id id of the object from the ids argument.
2. day (if the period is equal to day) day in the format YYYY-MM-DD.
3. month (if period is equal to month) - day in the format YYYY-MM.
4. day\_from (if period is overall) the start date of the reporting period.
5. day\_to (if period is overall) end date of the reporting period.
6. type object type from the ids\_type parameter.
7. gender gender, f - female, m - male.
8. impressions\_rate part of the audience that viewed the ad, from 0 to 1.
9. clicks\_rate (the part of the audience that clicked on the ad, from 0 to 1.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

# 1. Get a stat message on audience coverage by sex,
# advertising campaigns in the context of days
## 1.1. We get a list of advertising campaigns
camp <- vkGetAdCampaigns(account_id = 1, access_token = my_tok$access_token)
## 1.2. We get statistics on advertising campaigns
vk_ad_gender_stat_day <- vkGetAdGenderStats(account_id = 1,
                                             ids_type = "campaign",
                                             ids = camp$id,
                                             period = "day",
                                             date_from = "2010-01-01",
                                             date_to = "2017-09-10",
                                             access_token = my_tok$access_token)

# 2. Receive a stat message on the ads in the context of months
## 2.1. We get a list of advertising campaigns
ads <- vkGetAds(account_id = account_id, access_token = my_tok$access_token)
## 2.2. Get ad statistics
vk_ad_gender_stat_month <- vkGetAdGenderStats(account_id = 1,
```

```

                                ids_type = "ad",
                                ids = ads$id ,
                                period = "month",
                                date_from = "2010-01-01",
                                date_to = "2017-09-10",
                                access_token = my_tok$access_token)

# 3. Get a stat message on reaching the audience for
# advertising campaigns without a temporary breakdown
## 3.1. We get a list of advertising campaigns
camp <- vkGetAdCampaigns(account_id = 1, access_token = my_tok$access_token)
## 3.2. We get statistics on advertising campaigns
vk_ad_gender_stat_total <- vkGetAdGenderStats(account_id = 1,
                                                ids_type = "campaign",
                                                ids = camp$id ,
                                                period = "overall",
                                                date_from = "2010-01-01",
                                                date_to = "2017-09-10",
                                                access_token = my_tok$access_token)

## End(Not run)

```

---

vkGetAdPostsReach      *Statistics on The Coverage of Advertising Entries 'Vkontakte'.*

---

### Description

Returns detailed statistics on the reach of advertisements from ads and campaigns to promote community entries.

### Usage

```
vkGetAdPostsReach(account_id = NULL, ids_type = "campaign", ids = NULL,
                  api_version = NULL, access_token = NULL)
```

### Arguments

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
ids_type	The type of objects requested, which are listed in the ids parameter. Valid ad values are ads, campaigns are campaigns.
ids	Vector of id of the requested ads, campaigns, clients or account, depending on what is specified in the ids_type parameter.
access_token	API access tokens obtained using vkAuth or vkGetToken functions
api_version	Vkontakte API version.



```
## Get statistics on advertising campaigns
post_reach <- vkGetAdPostsReach(account_id = 1600134264,
                                ids_type = "campaign",
                                ids = camp$id,
                                access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetAds

*List of Ads From 'Vkontakte'.*


---

### Description

Provide to get a list of advertisements from the advertising office 'Vkontakte'.

### Usage

```
vkGetAds(account_id = NULL, client_id = NULL, include_deleted =
          TRUE, campaign_ids = "null", ad_ids = "null",
          status_names = TRUE, api_version = NULL, access_token
          = NULL)
```

### Arguments

account_id	The ID of the advertising account, a list of all available advertising offices can be obtained using the function vkGetAdAccounts.
client_id	Client identifier from which advertisements are requested. Available and required for advertising agencies.
include_deleted	Boolean TRUE or FALSE, flag, specifying the need to display archived ads.
campaign_ids	Numeric vector, filter by advertising campaigns.
ad_ids	Numeric vector, ad filter.
access_token	API access token, obtained using the functions vkAuth or vkGetToken.
api_version	Vkontakte API version.
status_names	If TRUE you get status names, in FALSE status Ids.

### Details

If campaign\_ids and ad\_ids filters are enabled at the same time, the following ads will be displayed: 1. If the ad id is specified in ad\_ids, then it is displayed. 2. If the ad belongs to an advertising campaign whose id is specified in campaign\_ids, then it is displayed. 3. If there is an ad with the id specified in ad\_ids and it belongs to a campaign whose id is specified in campaign\_ids, then this ad campaign is ignored, i.e. for her, rule 2 does not work.

No more than 2000 ads are output from no more than 2000 campaigns.

**Value**

Date frame with the following values:

id	ad id.
campaign_id	campaign id.
name	name of the announcement.
status	ad status.
ad_format	ad format.
approved	ad moderation status.
all_limit	total ad limit in rubles. 0 - no limit is set.
create_time	date and time of ad creation.
9. update_time	the date and time of the last ad change.
10. age_restriction	label about ad age limit.
11. category1_id	ID of the subject or subsection of the subject of the announcement, a list of topics can be obtained using the function vkGetAdCategories.
12. category2_id	Subject ID or sub topic of the ad. Additional topics, a list of topics can be obtained using the function vkGetAdCategories.
13. cost_type	type of payment.
14. cpc	price per switch in kopecks.
15. cpm	price per 1000 impressions in kopecks.
16. impressions_limit	limit the number of times this ad is shown per user.
17. impressions_limited	a sign that the number of ad impressions per user is limited.
18. ad_platform	advertising platforms where the ad will be displayed. (if ad_format is 1, 0 - VKontakte and partner sites, 1 - only VKontakte, (if ad_format is 9) all - all sites, desktop - full version of the site, mobile - mobile site and applications.)
19. video	1 - the ad is a video ad.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,app_secret = "H2Pk8htyFD8024mZaPHm")
# ads list
my_vk_ads <- vkGetAds(account_id = 11111111,
                      access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetAdsLayout	<i>Appearance of Advertisements.</i>
----------------	--------------------------------------

---

### Description

Allows you to get a list of advertisements and a description of their appearance from the advertising office 'Vkontakte'.

### Usage

```
vkGetAdsLayout(account_id = NULL, client_id = NULL, include_deleted =
    TRUE, campaign_ids = "null", ad_ids = "null",
    api_version = NULL, status_names = TRUE, access_token
    = NULL)
```

### Arguments

account_id	Advertising account ID, a list of all available advertising accounts can be obtained using the function vkGetAdAccounts.
client_id	Client identifier from which advertisements are requested. Available and required for advertising agencies.
include_deleted	Boolean TRUE or FALSE, flag, specifying whether to display archived ads.
campaign_ids	Numeric vector, filter by advertising campaigns.
ad_ids	Numeric vector, ad filter.
status_names	Numeric vector, filter by advertising campaigns.
api_version	Vkontakte API version.
access_token	API access token, obtained using the functions vkAuth or vkGetToken

### Details

If campaign\_ids and ad\_ids filters are enabled at the same time, the following ads will be displayed: 1. If the ad id is specified in ad\_ids, then it is displayed. 2. If the ad belongs to an advertising campaign whose id is specified in campaign\_ids, then it is displayed. 3. If there is an ad with the id specified in ad\_ids and it belongs to a campaign whose id is specified in campaign\_ids, then this ad campaign is ignored, i.e. for her, rule 2 does not work.

No more than 2000 ads are output from no more than 2000 campaigns.

The function is based on the API method ads.getAdsLayout, you can find out all the details of its work at <https://vk.com/dev/ads.getAdsLayout>.

**Value**

Data frame with next values:

1. id            ad id.
2. campaign\_id   campaign id.
3. ad\_format     ad format.
4. cost\_type     type of payment.
5. video         1 - the ad is a video ad.
6. title         ad title.
7. description   description of the advertisement.
8. link\_url      link to the advertised object.
9. link\_domain   domain of the advertised site..
10. preview\_link  
                  link by clicking on which you can view the advertisement as it looks on the site.  
                  The link is available for 30 minutes after the vkGetAdsLayout function.
11. image\_src    ad image url.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 app_secret = "H2Pk8htyFD8024mZaPHm")
# get Ads Layouts
my_vk_ads <- vkGetAdsLayout(account_id = 11111111,
                             access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetAdStatistics      *Statistics of advertising in 'VKontakte'.*

---

**Description**

returns performance statistics for advertisements, campaigns, customers, or the entire account.

**Usage**

```
vkGetAdStatistics(account_id = NULL, ids_type = "office", ids = NULL,
                  period = "overall", date_from = Sys.Date() - 30,
                  date_to = Sys.Date(), api_version = NULL, access_token
                  = NULL)
```

**Arguments**

account_id	The ID of the advertising account, a list of all available advertising offices can be obtained using the function vkGetAdAccounts.
ids_type	The type of objects requested that are listed in the ids parameter. Valid ad values are ads, campaign campaigns, client clients, office office.
ids	Vector id of the requested ads, campaigns, clients or account, depending on what is specified in the ids_type parameter.
period	The method of grouping data by date. day - statistics by days, month - statistics by months, overall - statistics for all time. Temporary restrictions are given by the arguments date_from and date_to.
date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version.

**Value**

Date frame with advertising statistics with the following values:

1. id id of the object from the ids argument.
2. day (if the period is equal to day) day in the format YYYY-MM-DD.
3. month (if period is equal to month) - day in the format YYYY-MM.
4. day\_from (if period is overall) the start date of the reporting period.
5. day\_to (if period is overall) end date of the reporting period.
6. type object type from the ids\_type parameter.
7. spent money spent.
8. impressions number of views.
9. clicks number of unique visitors.
10. reach (if ids\_type is ad or campaign and period is equal to day or month) - coverage.
11. video\_views (if ids\_type is ad) - video views (for video ads).
12. video\_views\_half (if ids\_type is ad) - half the video views (for video ads)
13. video\_views\_full (if ids\_type is ad) - views of the whole video (for video ads).
14. video\_clicks\_site (if ids\_type is equal to ad) - transitions to the advertiser's website from video ads (for video ads).
15. join\_rate (if ids\_type is ad or campaign) - group membership, event, subscriptions to a public page or application settings (only if the ad contains a direct link to the corresponding VK page).
16. lead\_form\_sends ount of sended lead forms from your ads.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 app_secret = "H2Pk8htyFD8024mZaPHm")

# 1. Get a static message on advertising
# campaigns in the context of days
## 1.1. We get a list of advertising campaigns
camp <- vkGetAdCampaigns(account_id = 1, access_token = my_tok$access_token)
## 1.2. We get statistics on advertising campaigns
vk_stat_by_campaign <- vkGetAdStatistics(account_id = 1,
                                         ids_type = "campaign",
                                         ids = camp$id ,
                                         period = "day",
                                         date_from = "2010-01-01",
                                         date_to = "2017-09-10",
                                         access_token = my_tok$access_token)

# 2. Receive a static message on the ads in the context of months
## 2.1. We get a list of advertising campaigns
ads <- vkGetAds(account_id = account_id, access_token = my_tok$access_token)
## 2.2. Get ad statistics
vk_stat_by_ads <- vkGetAdStatistics(account_id = 1,
                                   ids_type = "ad",
                                   ids = ads$id ,
                                   period = "month",
                                   date_from = "2010-01-01",
                                   date_to = "2017-09-10",
                                   access_token = my_tok$access_token)

# 3. Receive the general static of the advertising cabinet, in this example
# it is assumed that the id of the advertising cabinet is 1
vk_stat_by_account <- vkGetAdStatistics(account_id = 1,
                                       ids_type = "office",
                                       ids = 1,
                                       period = "overall",
                                       date_from = "2010-01-01",
                                       date_to = "2017-09-10",
                                       access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetDbCities	<i>City Dictionary.</i>
---------------	-------------------------

---

**Description**

Returns a list of cities and countries.

**Usage**

```
vkGetDbCities(country_id = TRUE, region_id = NULL, q = NULL,
              need_all = TRUE, api_version = NULL, access_token =
              NULL)
```

**Arguments**

country_id	country identifier obtained using the vkGetDbCountries function.
region_id	identifier of the region whose city you want to receive. (this parameter is optional).
q	search query string. For example, "Saint". The maximum string length is 15 characters.
need_all	Boolean TRUE or FALSE, flag — return a list of all countries.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version.

**Details**

the q parameter is not specified, then a list of the largest cities in the given country will be returned. If the q parameter is specified, a list of cities that are relevant to the search query will be returned.

**Value**

Date frame with community statistics with the following values:

cid	country id.
title	country name.
important	mark key cities for the current user.
area	area.
region	region.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## Get a list of cities for Ukraine, because Ukraine id = 2
vk_cities <- vkGetDbCities(country_id = 2,
                           access_token = my_tok$access_token)

## Get the list of cities for the search query "St."
vk_cities_search <- vkGetDbCities(country_id = 1,
                                  q = "St.",
                                  access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetDbCountries	<i>Country Dictionary.</i>
------------------	----------------------------

---

**Description**

Get list of countries.

**Usage**

```
vkGetDbCountries(need_all = TRUE, code = NULL, api_version = NULL,
                 access_token = NULL)
```

**Arguments**

need_all	Boolean TRUE or FALSE, flag — return a list of all countries.
code	Text vector, two-letter codes of countries in ISO 3166-1 alpha-2, for which you need to give information. Example c("RU", "UA", "BY"), a list of all codes can be viewed at the link - <a href="https://vk.com/dev/country_codes">https://vk.com/dev/country_codes</a> .
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version for send rquest.

**Details**

If the need\_all and code parameters are not specified, then a short list of countries located closest to the current user's country is returned. If the need\_all parameter is specified, a list of all countries will be returned. If the code parameter is specified, only countries with the listed ISO 3166-1 alpha-2 codes will be returned. A list of all country codes can be obtained at the link - [https://vk.com/dev/country\\_codes](https://vk.com/dev/country_codes)

**Value**

Date frame with community statistics with the following values:

1. cid            country id.
2. title         country name.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## get dictionary
vk_countries <- vkGetDbCountries(need_all = T,
                                code = c("RU", "UA", "BY"),
                                access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetDbRegions	<i>Regions list.</i>
----------------	----------------------

---

**Description**

Get regions dictionary.

**Usage**

```
vkGetDbRegions(country_id = NULL, q = NULL, api_version = NULL,
               access_token = NULL)
```

**Arguments**

- |              |  |
|--------------|--|
| country_id   | country identifier obtained using the vkGetDbCountries function.                   |
| q            | search query string. For example, Len. The maximum string length is 15 characters. |
| access_token | API access token, obtained using the functions vkAuth or vkGetToken.               |
| api_version  | Vkontakte API version for send rquest.   |

**Details**

If the q parameter is not specified, then a list of all regions in the given country will be returned. If the q parameter is specified, a list of regions that are relevant to the search query will be returned.

**Value**

Data frame contains community statistics with the following values:

1. region\_id     region identifier.
2. title         name of the region.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 app_secret = "H2Pk8htyFD8024mZaPHm")

## Get a list of regions for Ukraine, because Ukraine id = 2
vk_regions <- vkGetDbRegions(country_id = 2,
                             access_token = my_tok$access_token)

##get a list of cities of Ukraine on request "Ode"
vk_regions_search <- vkGetDbRegions(country_id = 2,
                                    q = "Ode",
                                    access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetGroupStat                    *Statistics on visiting the community in 'Vkontakte'.*

---

**Description**

Load data from the 'Vkontakte' communities by the number of views, subscriptions, replies and community visitors by date.

**Usage**

```
vkGetGroupStat(date_from = Sys.Date(),
               date_to = Sys.Date(),
               group_id = NULL, access_token = NULL)
```

**Arguments**

date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
group_id	Community ID.
access_token	API access token, obtained using the functions vkAuth or vkGetToken



**Arguments**

date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
group_id	Community ID.
access_token	API access token, obtained using the functions vkAuth or vkGetToken

**Value**

Date frame with statistics on community visitors with the following values:

1. Date the date for which statistics are given.
2. Visitors number of unique visitors.
3. AgeGroup age interval.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,app_secret = "H2Pk8htyFD8024mZaPHm")

## Get statistics on age groups of visitors
communityDataAge <- vkGetGroupStatAge(date_from = "2016-01-01",
                                       date_to = "2016-04-01",
                                       group_id = 1,
                                       access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetGroupStatCity      *Statistics of the Vkontakte community visitors by cities.*

---

**Description**

The function vkGetGroupStatCity allows you to get data about the city of community visitors by date.

**Usage**

```
vkGetGroupStatCity(date_from = Sys.Date(),
                   date_to = Sys.Date(),
                   group_id = NULL,
                   access_token = NULL)
```

**Arguments**

date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
group_id	Community ID.
access_token	API access token, obtained using the functions vkAuth or vkGetToken

**Value**

Date frame with statistics on community visitors with the following values:

Date	date for which statistics is given.
Visitors	number of unique visitors.
CityID	city identifier or "other" for the "other cities" section.
CityName	city name.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,app_secret = "H2Pk8htyFD8024mZaPHm")

## load statistic
communityDataCity <- vkGetGroupStatCity(
  date_from = "2016-01-01",
  date_to = "2016-04-01",
  group_id = 1,
  access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetGroupStatCountries

*Statistics of The Vkontakte Community Visitors By Country.*

---

**Description**

Allows you to get data about the city of community visitors by date.

**Usage**

```
vkGetGroupStatCountries(date_from = Sys.Date(),
  date_to = Sys.Date(),
  group_id = NULL,
  access_token = NULL)
```

**Arguments**

date_from	The start date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
group_id	Community ID.
access_token	API access token, obtained using the functions vkAuth or vkGetToken

**Value**

Date frame with statistics on community visitors with the following values:

Date	date for which statistics is given.
Visitors	number of unique visitors.
CountryID	country identifier
CountryCode	two-letter country code (for example, "RU").
CountryName	country name.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## load data
communityDataCountry <- vkGetGroupStatCountries(
  date_from = "2016-01-01",
  date_to = "2016-04-01",
  group_id = 1,
  access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetGroupStatGender *Loading data by gender of the community visitors*

---

**Description**

Allows you to download data on the gender of community visitors by date.

**Usage**

```
vkGetGroupStatGender(date_from = Sys.Date(),
  date_to = Sys.Date(),
  group_id = NULL,
  access_token = NULL)
```

**Arguments**

date_from	The start date of the displayed statistics in the format YYYY-MM-DD.
date_to	End date of the displayed statistics in the format YYYY-MM-DD.
group_id	Community ID.
access_token	API access token, obtained using the functions vkAuth or vkGetToken

**Value**

Date frame with statistics on community visitors with the following values:

1. Date the date for which statistics are given.
2. Visitors number of unique visitors.
3. Gender gender ("m" is male, "f" is female).

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## load data about users by gender
communityDataGender <- vkGetGroupStatGender(
  date_from = "2016-01-01",
  date_to = "2016-04-01",
  group_id = 1,
  access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetGroupStatGenderAge

*Statistics on Age Groups And Gender of The 'VKontakte' Community.*

---

**Description**

Provide to load to download statistics on the sex and age group of community visitors by date.

**Usage**

```
vkGetGroupStatGenderAge(date_from = Sys.Date(),
  date_to = Sys.Date(),
  group_id = NULL,
  access_token = NULL)
```

**Arguments**

date_from	The starting date of the displayed statistics in the format YYYY-MM-DD.
date_to	The end date of the displayed statistics in the format YYYY-MM-DD.
group_id	Community ID.
access_token	API access token, obtained using the functions vkAuth or vkGetToken

**Value**

Date frame with statistics on community visitors with the following values:

1. Date the date for which statistics are given.
2. Visitors number of unique visitors.
3. Gender gender ("m" is male, "f" is female).
4. AgeGroup age interval.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,app_secret = "H2Pk8htyFD8024mZaPHm")

## Loading data by users by gender and age group
communityDataGenderAge <- vkGetGroupStatGenderAge(
  date_from = "2016-01-01",
  date_to = "2016-04-01",
  group_id = 1,
  access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetGroupStatPostReach

*Statistics of Posts on The Community Wall.*

---

**Description**

Returns statistics for writing to the wall.

**Usage**

```
vkGetGroupStatPostReach(owner_id = NULL,
  post_ids = NULL,
  api_version = NULL,
  access_token = NULL)
```

**Arguments**

owner_id	community ID - the owner of the entry. Indicated with a minus sign.
post_ids	vector containing record identifiers, please note - statistics data are available only for the 300 latest (most recent) posts on the community wall.
api_version	An optional argument, the API version number to which the request will be executed.
access_token	API access tokens obtained using vkAuth or vkGetToken functions.

**Details**

Statistics records are currently available only to communities with the number of participants from 5000 and above, as well as official communities.

Please note - statistics data are available only for the latest 300 (most recent) posts on the community wall. Common errors may occur during execution. Their description is on a separate page - <https://vk.com/dev/errors>.

**Value**

Date frame with the following values:

1. reach\_viral viral coverage (if recording was promoted with targeted advertising)
2. reach\_ads advertising reach (if the record was promoted using targeted advertising).
3. reach\_subscribers reach subscribers.
4. reach\_total total coverage.
5. links follow the link.
6. to\_group transitions to the community.
7. join\_group entry into the community.
8. report number of complaints about the record.
9. hide number of entries hiding.
10. unsubscribe number of unsubscribe members.
11. post\_id post id.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1,
                 app_secret = "H2Pk8htyFD8024mZaPHm")

## Get the list of community messages
```

```

my_vk_wall <- vkGetUserWall(user_id = -111111,
                           access_token = my_tok$access_token)

## Get sub-stats for the latest 300 community entries
post_reach <- vkGetGroupStatPostReach(owner_id = -2222222,
                                       post_ids = my_vk_wall$id,
                                       access_token = my_tok$access_token)

## End(Not run)

```

---

vkGetGroupToken      *Get the Access Key of Community 'Vkontakte'.*

---

### Description

This key allows you to work with the API on behalf of a group, meeting or public page. For example, with its help you can respond to community subscribers to messages received in its address. With the community access key, you can call those methods that have a special mark in the general list (<https://vk.com/dev/methods>). The vkGetGroupToken function gets the community access key using the Implicit flow scheme ([https://vk.com/dev/implicit\\_flow\\_group](https://vk.com/dev/implicit_flow_group)).

### Usage

```
vkGetGroupToken(app_id = 5344605, group_ids = NULL)
```

### Arguments

app_id	The ID of your Vkontakte application, which is located in the "Application ID" field in the application settings.
group_ids	A vector containing the ID of the groups to which you want to get an access code.

### Value

Object containing a group access token

### Author(s)

Alexey Seleznev

### Examples

```

## Not run:
## Get vk api token, where 1 is the application ID,
## and 11,22,33 is the group ID.
myToken <- vkGetGroupToken(appId = 1,
                          group_ids = c(11,22,33))

## End(Not run)

```

---

vkGetToken	<i>Get vkontakte developer token</i>
------------	--------------------------------------

---

**Description**

vkGetToken open web link in browser and generate developer token, which you need for work with vk.com api

**Usage**

```
vkGetToken(app_id = 5344605)
```

**Arguments**

app\_id            Your app id from filed in app page vkontakte.

**Value**

R object with vk.com token

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## Get vk api token
myToken <- vkGetToken(app_id = "1")

## End(Not run)
```

---

vkGetUserDialogs	<i>Returns The List of Conversations of The Current User or Community.</i>
------------------	--

---

**Description**

Return a list of dialogs, or rather the last message from each dialog, in order to get a list of message dialogs, you must pass the community access key received by using the vkGetGroupToken function to the access\_token argument, in order to get a list of user dialogs access obtained with vkGetToken

**Usage**

```
vkGetUserDialogs(offset = NULL, count = NULL,
                 start_message_id = NULL,
                 preview_length = 0, unread = 0,
                 important = 0, unanswered = 0,
                 api_version = NULL, access_token = NULL)
```

**Arguments**

offset	Integer, offset required to select a specific subset of dialogs.
count	A positive number, the default is 20, the maximum value is 200, the number of conversations you need to receive.
start_message_id	A positive number, a message identifier starting from which you want to return a list of conversations. See below for details.
preview_length	A positive number, the number of characters to cut the message. Specify 0 if you do not want to cut the message. (by default, messages are not cropped). Please note that the text is cut off by words, the exact number of characters may not match the specified value.
unread	1 - return only dialogs with unread incoming messages. Default: 0.
important	1 - return dialogs marked “Important” (for community posts).
unanswered	1 - return dialogs marked “Unanswered” (for community messages).
api_version	API access token, obtained using vkAuth or vkGetToken
access_token	API access token, obtained using the functions vkAuth or vkGetToken

**Details**

If the `start_message_id` parameter was passed, the position of the dialog in the list will be found, the last message identifier of which is equal to `start_message_id` (or the closest one to it). Starting from this position, `count` dialogs will be returned. The offset in this case will be counted from this position (it may be negative).

API method which is used in the function: `messages.getDialogs` (<https://vk.com/dev/messages.getDialogs>)

The `messages.getDialogs` method returns a collection of objects describing private messages. A list of all possible fields for describing personal messages can be found at the link - <https://vk.com/dev/objects/message>.

**Value**

Date frame with the following values:

1. `id` message identifier (not returned for forwarded messages).
2. `date` date the message was sent.
3. `out` message type (0 - received, 1 - sent, not returned for forwarded messages).
4. `user_id` identifier of the user in the dialogue with which the message is located.
5. `read_state` status of the message (0 - not read, 1 - read, not returned for forwarding messages).
6. `title` the title of the message or conversation.
7. `body` message text.
8. `random_id` identifier used when sending a message. Returned for outgoing messages only

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
# Get access key for the community
group_token <- vkGetGroupToken(app_id = 111111,
                               group_ids = 222222)

# Get dialogs list
message <- vkGetUserDialogs(access_token = group_token)

## End(Not run)
```

---

vkGetUserFriends      *Frend List in 'Vkontakte'*

---

**Description**

Returns a list of user friends identifiers and extended information about the user's friends.

**Usage**

```
vkGetUserFriends(user_id = NULL, access_token = NULL, api_version =
                 NULL, name_case = "nom")
```

**Arguments**

user_id	user ID for which you want to get a list of friends. If the parameter is not specified, it is considered that it is equal to the identifier of the current user.
name_case	case for declining the user's first and last name. Possible values: nominative - nom, genitive - gen, dative - dat, accusative - acc, ablative - ins, prepositional - abl. Default nom.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version for send rquest.

**Value**

Date frame with community statistics with the following values:

1. user_id	user id.
2. first_name	name.
3. last_name	last name
4. gender	sex.
5. nickname	nickname of the user.
6. domain	short page address. Returns a string containing the short address of the page (for example, andrew). If not assigned, it returns "id" + user_id, for example, id35828305.

7. `bdate` date of birth. Returns in the format D.M.YYYY or D.M (if the year of birth is hidden). If the date of birth is hidden entirely, the field is absent in the answer.
8. `city` city id, city name can be obtained using the function `vkGetDbCities`.
9. `country` country id, country name can be obtained using the function `vkGetDbCountries`.
10. `photo_50` url of a square photo of a user having a width of 50 pixels. If the user does not have a photo, he returns `http://vk.com/images/camera_c.gif`.
11. `photo_100` url of a square photo of a user having a width of 100 pixels. If the user doesn't have a photo, he returns `http://vk.com/images/camera_b.gif`.
12. `photo_200_orig` url of a user photo that has a width of 200 pixels. If the user doesn't have a photo, he returns `http://vk.com/images/camera_a.gif`.
13. `has_mobile` information about whether the user's mobile phone number is known. Return values: 1 - known, 0 - unknown.
14. `online` information about whether the user is currently online.
15. `can_post` information about whether the current user can post on the wall. Possible values: 1 - can, 0 - can not.
16. `can_see_all_posts` information about whether the current user can see someone else's posts on the wall. Possible values: 1 - can, 0 - can not.
17. `can_write_private_message` information about whether the current user can send a private message. Possible values: 1 - can, 0 - can not.
18. `status` user status. Returns a string containing the text of the status located in the profile under the user name.
19. `last_seen_time` date and time of last visit.
20. `last_seen_platform` the type of platform through which the last entry was made. Possible values: 1 - mobile version, 2 - application for iPhone, 3 - application for iPad, 4 - application for Android, 5 - application for Windows Phone, 6 - application for Windows 10, 7 - full version of the site.
21. `university` university id.
22. `university_name` university name.
23. `faculty` faculty id.
24. `faculty_name` faculty name.
25. `graduation` graduation year.
26. `education_form` learning form
27. `education_status` status (for example, "Graduate (Specialist)")
28. `relation` marital status of the user. Possible values: 1 - not married / not married, 2 - have a friend / have a girlfriend, 3 - engaged / engaged, 4 - married / married, 5 - everything is difficult, 6 - in active search, 7 - in love / in love, 8 - in a civil marriage, 0 - not specified.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## user list
my_vk_friends <- vkGetUserFriends(user_id = 7437995,
                                  access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetUserGroups	<i>List of 'Vkontakte' communities.</i>
-----------------	---

---

**Description**

Returns the list of communities for the specified user.

**Usage**

```
vkGetUserGroups(user_id = NULL, filter = NULL, api_version = NULL,
                access_token = NULL)
```

**Arguments**

user_id	user ID for which you want to get a list of friends. If the parameter is not specified, it is considered that it is equal to the identifier of the current user.
filter	A list of community filters to be returned, separated by commas. The values are admin, editor, moder, groups, publics, events. By default, all user communities are returned. If the admin filter is specified, the communities in which the user is an administrator, editor - administrator or editor, moder - administrator, editor or moderator will be returned.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version for send request.

**Value**

Date frame with community statistics with the following values:

gid	community id.
name	community name.
screen_name	short address, for example, apiclub.

is_closed	is the community closed. Possible values: 0 - open, 1 - closed, 2 - private
type	community type: group - group, page - public page, event - event.
description	community description text.
wiki_page	name of the main wiki page.
members_count	number of members.
start_date	date of foundation.
can_post	information about whether the current user can post on the community wall. Possible values: 1 - can, 0 - can not.
can_see_all_posts	information about whether it is allowed to see other people's posts on the group wall. Possible values: 1 - can, 0 - can not.
activity	public page status bar. For groups, a string value is returned, whether the group is open or not, and for events the start date.
status	community status.
fixed_post	pinned post ID.
verified	information about whether the community is verified. Possible values: 1 - can, 0 - can not.
site	the address of the site from the "web site" field in the community description.
can_create_topic	information on whether the current user can create a new discussion in the group. Possible values: 1 - can, 0 - can not.
photo	URL of the main photo with a size of 50x50px.
photo_medium	URL of the main photo with a size of 100x100px.
photo_big	The URL of the main photo in the maximum size.

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:
## auth
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")

## get groups
my_vk_community <- vkGetUserGroups(user_id = 7437995,
                                   access_token = my_tok$access_token)

## End(Not run)
```

---

vkGetUserWall	<i>Get Posts From Userwall.</i>
---------------	---------------------------------

---

**Description**

Return posts list from userwall 'Vkontakte'.

**Usage**

```
vkGetUserWall(user_id = NULL, domain = NULL, filter = NULL,
              api_version = NULL, access_token = NULL)
```

**Arguments**

user_id	user ID for which you want to get a list of friends. If the parameter is not specified, it is considered that it is equal to the identifier of the current user.
domain	short address of user or community.
filter	determines what types of posts on the wall should be received. Possible values: suggests — suggested entries on the community wall, postponed — deferred entries, owner — owner wall entries, others — not wall owner entries, all — all entries on the wall (owner + others), Default: all.
access_token	API access token, obtained using the functions vkAuth or vkGetToken
api_version	Vkontakte API version for send request.

**Value**

Date frame with the following values:

1. id record ID.
2. from\_id identifier of the post author.
3. to\_id identifier of the owner of the wall on which the entry is placed.
4. date post publication time
5. post\_type record type, can be one of the following values: post, copy, reply, postpone, suggest.
6. text text entry.
7. can\_delete information about whether the current user can delete a post (1 — maybe, 0 — cannot).
8. comments\_count number of comments.
9. likes\_count the number of users who liked the post.
10. reposts\_count number of users who copied the record.
11. attachment\_type type of media attachment of the record (photos, links, etc.)

**Author(s)**

Alexey Seleznev

**Examples**

```
## Not run:  
## auth  
my_tok <- vkAuth(app_id = 1, app_secret = "H2Pk8htyFD8024mZaPHm")  
  
## get posts  
my_vk_wall <- vkGetUserWall(user_id = 7437995,  
                             access_token = my_tok$access_token)  
  
## End(Not run)
```

# Index

## \*Topic **package**

rvkstat-package, [2](#)

rvkstat (rvkstat-package), [2](#)

rvkstat-package, [2](#)

vkAuth, [4](#)

vkGetAdAccounts, [5](#)

vkGetAdAgeStats, [6](#)

vkGetAdBudget, [8](#)

vkGetAdCampaigns, [9](#)

vkGetAdCategories, [10](#)

vkGetAdCityStats, [11](#)

vkGetAdClients, [13](#)

vkGetAdGenderAgeStats, [14](#)

vkGetAdGenderStats, [16](#)

vkGetAdPostsReach, [18](#)

vkGetAds, [20](#)

vkGetAdsLayout, [22](#)

vkGetAdStatistics, [23](#)

vkGetDbCities, [26](#)

vkGetDbCountries, [27](#)

vkGetDbRegions, [28](#)

vkGetGroupStat, [29](#)

vkGetGroupStatAge, [30](#)

vkGetGroupStatCity, [31](#)

vkGetGroupStatCountries, [32](#)

vkGetGroupStatGender, [33](#)

vkGetGroupStatGenderAge, [34](#)

vkGetGroupStatPostReach, [35](#)

vkGetGroupToken, [37](#)

vkGetToken, [38](#)

vkGetUserDialogs, [38](#)

vkGetUserFriends, [40](#)

vkGetUserGroups, [42](#)

vkGetUserWall, [44](#)