

Package ‘pushoverr’

July 2, 2014

Type Package

Title Send push notifications using Pushover

Version 0.1.3

Date 2014-06-18

Maintainer Brian Connelly <bdc@bconnelly.net>

Description A package for sending push notifications to iOS and Android devices using Pushover

License BSD_2_clause + file LICENSE

URL <https://github.com/briandconnelly/pushoverr>

BugReports <https://github.com/briandconnelly/pushoverr/issues>

ByteCompile no

NeedsCompilation no

Depends R (>= 3.0.0),

Imports httr, methods

Author Brian Connelly [aut, cre]

Repository CRAN

Date/Publication 2014-06-19 01:10:01

R topics documented:

check_receipt	2
content_value,PushoverResponse-method	3
get_devices	4
get_pushover_sounds	5
headers,PushoverResponse-method	6
http_status_code,PushoverResponse-method	6
is.success,PushoverResponse-method	7

is.valid_device	8
is.valid_receipt	8
is.valid_token	9
pushover	10
PushoverMessage	11
PushoverMessage-class	12
pushoverr	13
PushoverResponse	14
PushoverResponse-class	15
pushover_priorities	16
receipt,PushoverResponse-method	16
request,PushoverResponse-method	17
send,PushoverMessage-method	18
send_pushovermessage	18
set_pushover_app	19
show,PushoverMessage-method	21
show,PushoverResponse-method	21
status,PushoverResponse-method	22
validate_key	22
validate_PushoverMessage	23
validate_PushoverResponse	24

Index 25

check_receipt	<i>Determine whether or not an emergency message has been acknowledged</i>
---------------	--

Description

check_receipt issues a query to Pushover to determine whether or not an emergency-priority message has been acknowledged and when (if applicable). If a callback URL was specified with the message, it will also report whether or not that callback URL was POSTed to and when.

is.acknowledged checks to see whether or not the given emergency message receipt has been acknowledged or not.

Usage

```
check_receipt(receipt, ...)
```

```
is.acknowledged(receipt, info = TRUE, ...)
```

Arguments

receipt	A message receipt (e.g., 'KAWXTswy4cekx6vZbHBKbCKk1c1fdf')
...	An application token can be specified with token
info	is.acknowledged will print out additional information about who acknowledged the message, when, and if and when the callback URL

Details

is.acknowledged returns whether a message was acknowledged (TRUE) or not (FALSE) as well as some information about when the message was received and by whom.

Value

A [PushoverResponse](#) object containing the response from the server

is.acknowledged returns a boolean indicating whether the message has been acknowledged (TRUE) or not (FALSE)

Note

Message receipts can be acquired using [receipt](#) on the [PushoverResponse](#) object storing the response to an emergency message.

The token argument is necessary, however it does not need to be given if the application token have been set with [set_pushover_app](#).

Examples

```
## Not run:
response <- check_receipt(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
                          receipt='KAWXTswy4cekx6vZbHBKbCCKk1c1fdf')

if(is.acknowledged(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
                   receipt='KAWXTswy4cekx6vZbHBKbCCKk1c1fdf'))
{
  cat('Message has been read.\n')
}

## End(Not run)
```

content_value,PushoverResponse-method

Get a value from a API query response

Description

Pushover API calls return JSON data containing parameter-value data related to the query. content_value extracts the value for a given parameter from a [PushoverResponse-class](#) object.

Usage

```
## S4 method for signature 'PushoverResponse'
content_value(object, param)
```

Arguments

object	A PushoverResponse-class object
param	The name of a parameter in the response

Value

A string containing the value associated with the given parameter.

Note

All values are returned as strings. Numeric, date, and other types should be coerced with [as.numeric](#), etc.

Examples

```
## Not run:
# Get the acknowledged status of an emergency message
message <- PushoverMessage(message='EMERGENCY!', priority=2,
                           token='KzGDORePK8gMac0QOYAMyEEuzJnyUi ',
                           user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG')
response <- send(message)
acknowledged <- as.numeric(content_value(response, 'acknowledged'))

## End(Not run)
```

get_devices

Get a list of the user's registered devices

Description

get_devices queries Pushover to receive a list of the devices given by the user (specified by their user key)

is.device determines whether or not a given device name is registered to the given user (specified by their user key)

Usage

```
get_devices(...)

is.device(device, ...)
```

Arguments

...	An app token and/or user key can be specified with the token and user arguments, respectively
device	A device name (e.g., 'phone')

Value

get_devices returns a vector of device names

is.device returns a boolean value indicating if the device is registered (TRUE) or not (FALSE)

Note

The token and user arguments are necessary, however they do not need to be given if they have been set with [set_pushover_user](#) and [set_pushover_app](#), respectively.

The token and user arguments are necessary, however they do not need to be given if they have been set with [set_pushover_user](#) and [set_pushover_app](#), respectively.

Examples

```
## Not run:
available_devs <- get_devices(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
                             user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG')

## End(Not run)
## Not run:
is.device(device='phone',
          token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
          user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG')

## End(Not run)
```

get_pushover_sounds *Get a list of available message sounds*

Description

get_pushover_sounds queries Pushover to receive a list of the sounds that may be played on a user's device when a message is received. Sounds are specified when the message is sent.

Usage

```
get_pushover_sounds(...)
```

Arguments

... An application token can be specified with token

Value

A list of available sounds and their descriptions

Examples

```
## Not run:
sounds <- get_pushover_sounds()
for (s in names(sounds))
{
  cat(paste(s, ":", sounds[s], "\n"))
}

## End(Not run)
```

headers,PushoverResponse-method

Get the HTTP headers returned by a Pushover API query

Description

Aside from standard HTTP parameters, headers in Pushover API responses also contain information about the application, including the app's limit on the number of messages sent per month ('x-limit-app-limit'), how many messages remain ('x-limit-app-remaining'), and when the message counter will be reset ('x-limit-app-reset').

Usage

```
## S4 method for signature 'PushoverResponse'
headers(object)
```

Arguments

object A [PushoverResponse-class](#) object

Value

A list of values from the HTTP header

http_status_code,PushoverResponse-method

Return the HTTP status code returned by a Pushover API query

Description

Get the HTTP status code returned from a Pushover API query. Successful queries return 200, 4xx indicates query errors (see the errors list in the content slot of a [PushoverResponse-class](#) object), and 500 indicates a connection problem.

Usage

```
## S4 method for signature 'PushoverResponse'  
http_status_code(object)
```

Arguments

object A [PushoverResponse-class](#) object

Value

The numeric HTTP status code used in response to a Pushover query

is.success,PushoverResponse-method

Return whether a PushoverResponse indicated success (TRUE) or not (FALSE)

Description

Return whether a PushoverResponse indicated success (TRUE) or not (FALSE)

Usage

```
## S4 method for signature 'PushoverResponse'  
is.success(object)
```

Arguments

object A [PushoverResponse-class](#) object

Value

Whether the query was a success (TRUE) or not (FALSE)

See Also

[status](#)

is.valid_device *Determine whether or not a given device name is valid*

Description

is.valid_device determines whether or not a given device name is valid or not according to Pushover's specifications. It does not determine whether or not the given device is registered to a user.

Usage

```
is.valid_device(device)
```

Arguments

device A device name (e.g., 'phone')

Value

A boolean value indicating if the device name is valid (TRUE) or not (FALSE)

Examples

```
## Not run:  
is.valid_device(device='phone')  
  
## End(Not run)
```

is.valid_receipt *Determine whether or not a given message receipt is valid*

Description

is.valid_receipt determines whether or not a given message receipt is valid or not according to Pushover's specifications. It does not determine whether or not the given receipt actually exists.

Usage

```
is.valid_receipt(receipt)
```

Arguments

receipt A message receipt (e.g., 'KAWXTswy4cekx6vZbHBKbCKk1c1fdf')

Value

A boolean value indicating if the message receipt is valid (TRUE) or not (FALSE)

Note

To acquire a message receipt, send an emergency priority message

See Also

[check_receipt](#)

Examples

```
## Not run:
is.valid_receipt(token='KAWXTswy4cekx6vZbHBKbCKk1c1fdf')

## End(Not run)
```

is.valid_token	<i>Determine whether or not a given API token is valid</i>
----------------	--

Description

is.valid_token determines whether or not a given application token is valid or not according to Pushover's specifications. It does not determine whether or not the given token is associated with an application.

Usage

```
is.valid_token(token)
```

Arguments

token A application token (e.g., 'KzGDORePK8gMaC0QOYAMyEEuzJnyUi')

Value

A boolean value indicating if the application token is valid (TRUE) or not (FALSE)

Note

To acquire an application token, register your token at <https://pushover.net/apps>

Examples

```
## Not run:
is.valid_token(token='KzGDORePK8gMaC0QOYAMyEEuzJnyU')

## End(Not run)
```

pushover *Send a message using Pushover*

Description

pushover sends a message (push notification) to a user or group. Messages can be given different priorities, play different sounds, or require acknowledgments. A unique request token is returned. The `pushover_normal`, `pushover_silent`, `pushover_quiet`, `pushover_high`, and `pushover_emergency` functions send messages with those priorities.

Usage

```
pushover(message, ...)
```

Arguments

message	The message to be sent (max. 512 characters)
...	Any additional message parameters (see PushoverMessage-class)

Value

A list containing a Pushover request token and a receipt token for emergency priority messages. When used outside of an assignment, these return values will not be displayed for non-emergency messages.

Note

Pushover user/group keys and application tokens are required for a message. They can either be specified as arguments or be set earlier with [set_pushover_user](#) and [set_pushover_app](#), respectively.

Examples

```
## Not run:
# Send a pushover message
pushover(message='Hello World!', token='KzGDORePK8gMaC0QOYAMyEEuzJnyUi',
          user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG')

# User keys and app tokens can be set ahead of time
set_pushover_user('KAWXTswy4cekx6vZbHBKbCKk1c1fdf')
set_pushover_app('KzGDORePK8gMaC0QOYAMyEEuzJnyUi')
pushover(message='so much less work!')

# Send a message with high priority and a title
pushover_high(message='The sky is falling', title='Alert')

# Send an emergency message. Emergency messages will be re-sent until they
# are acknowledged (in this case, every 60 seconds)
pushover_emergency(message='TAXES ARE DUE AT MIDNIGHT!', retry=60)
```



```
# If the app token and user key have already been set, messages can be
# created with just the message argument
set_pushover_user('KAWXTswy4cekx6vZbHBKbCKk1c1fdf')
set_pushover_app('KzGDORePK8gMaC0QOYAMyEEuzJnyUi')
m3 <- PushoverMessage(message='so much less work!')

## End(Not run)
```

PushoverMessage-class *The PushoverMessage class*

Description

PushoverMessage objects represent a Pushover message and implements all of the features available in Pushover's API (<https://pushover.net/api>). PushoverMessage objects are used to build queries that are sent to Pushover's servers.

Slots

message The message to be sent (max. 512 characters)

token The application token

user The user or group key to send the message to

device The device to send the notification to (optional)

title The title of the message (optional)

url A URL to be included in the message (optional, max. 512 characters)

url_title A title for the given url (optional, max. 100 characters)

priority The message's priority. One of: -2 (silent), -1 (quiet), 0 (normal, default), 1 (high), 2 (emergency). Quiet messages do not play a sound. Emergency messages require acknowledgement.

timestamp The time to associate with the message (default: now, format: UNIX time)

sound The sound to be played when the message is received (see [get_pushover_sounds](#))

callback A callback URL. For emergency priority, a POST request will be sent to this URL when the message is acknowledged (see <https://pushover.net/api#receipt>)

retry The number of seconds between re-sending of an unacknowledged emergency message (default: 60, min: 30)

expire The number of seconds until an unacknowledged emergency message will stop being resent (default: 3600, max: 86400).

Note

PushoverMessage objects are created with the [PushoverMessage](#) constructor (see Examples below).

See Also[PushoverMessage](#)**Examples**

```
## Not run:
library(pushoverr)

# Create a PushoverMessage
m1 <- PushoverMessage(message='Hi there',
                       token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
                       user='KAWXTswy4cekx6vZbHBKbCKk1c1fdf')

## End(Not run)
```

pushoverr	<i>pushoverr: Send push notifications to iOS and Android devices using Pushover</i>
-----------	---

Description

pushoverr is package for sending push notifications to iOS and Android devices using [Pushover](#). Messages can be quickly sent using the [pushover](#), [pushover_silent](#), [pushover_quiet](#), [pushover_normal](#), [pushover_high](#), and [pushover_emergency](#) functions. Behind the scenes, messages are created using [PushoverMessage](#) objects, and responses are stored in [PushoverResponse](#) objects.

Details

For more information, browse the package index by following the link at the bottom of this help page. For release information, run `news(package=='pushoverr')`.

Disclaimer

This package and its author are not affiliated with [Superblock, LLC](#), creators of Pushover.

Author(s)

Brian Connelly

See Also

GitHub Repository: <https://github.com/briandconnelly/pushoverr>

Bug Reports and Feature Requests: <https://github.com/briandconnelly/pushoverr/issues>

Pushover: <https://pushover.net>

Pushover API: <https://pushover.net/api>

PushoverResponse *Create a PushoverResponse to store a Pushover server response*

Description

The PushoverResponse function is a constructor that creates [PushoverResponse-class](#) objects. These objects store information from responses to Pushover queries

Usage

```
PushoverResponse(status = NA_integer_, request = NA_character_,
  status_code = NA_integer_, headers = NA, content = NA)
```

Arguments

status	The status response from Pushover 1=good, 0=problem
request	The unique identifier associated with the message
status_code	The HTTP status code returned
headers	A list containing the headers in the HTTP response
content	A list containing the content from the response. This will vary depending on the type of query sent to the server.

See Also

[PushoverResponse-class](#)

Examples

```
## Not run:
library(pushoverrr)

# Create PushoverResponse object
response <- POST(url="https://api.pushover.net:443/1/messages.json",
  body=params)
response <- PushoverResponse(status=1,
  request='8345bfe5fbd7d346028f2863de77c8c4',
  status_code=200, cheaders=resp$headers,
  content=content(rsp))

## End(Not run)
```

PushoverResponse-class

The PushoverResponse class

Description

PushoverResponse objects store information from responses to Pushover queries

Slots

status The status response from Pushover 1=good, 0=problem

request The unique identifier associated with the message

status_code The HTTP status code returned

headers A list containing the headers in the HTTP response

content A list containing the content from the response. This will vary depending on the type of query sent to the server.

Note

PushoverResponse objects can be created with [new](#) or with the [PushoverResponse](#) constructor (see Examples below).

See Also

[PushoverResponse](#)

Examples

```
## Not run:
library(pushoverr)

# Create PushoverResponse object based on a POST to Pushover
response <- POST(url="https://api.pushover.net:443/1/messages.json",
  body=params)
response <- new("PushoverResponse", status=1,
  request='8345bfe5fbd7d346028f2863de77c8c4',
  status_code=200,
  headers=resp$headers,
  content=content(rsp))

# Create PushoverResponse object using constructor
response <- POST(url="https://api.pushover.net:443/1/messages.json",
  body=params)
response <- PushoverResponse(status=1,
  request='8345bfe5fbd7d346028f2863de77c8c4',
  status_code=200, cheaders=resp$headers,
  content=content(rsp))

## End(Not run)
```

pushover_priorities *List of the priorities available and their descriptions*

Description

List of the priorities available and their descriptions

Usage

pushover_priorities

Format

List of 5
 \$ -2: chr "silent"
 \$ -1: chr "quiet"
 \$ 0 : chr "normal"
 \$ 1 : chr "high"
 \$ 2 : chr "emergency"

receipt,PushoverResponse-method
Get the receipt from a Pushover server response (M)

Description

Receipts are unique tokens returned by Pushover servers in response to emergency priority messages being sent. Receipts can be used to query whether or not the message has been acknowledged, when, and by whom. This can be done with **pushoverr** using either [check_receipt](#) or [is.acknowledged](#).

Usage

```
## S4 method for signature 'PushoverResponse'
receipt(object)
```

Arguments

object A [PushoverResponse-class](#) object containing a response from Pushover following a request

Value

A string containing a unique receipt token

See Also

[check_receipt](#), [is.acknowledged](#) for checking receipt status

Examples

```
## Not run:
# Send an emergency message to a group, and see if it's been acknowledged
msg <- PushoverMessage(message='Get back to work',
                        token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
                        user='gznej3rKEVAvPUxu9vvNnqpmZpokzF')
response <- send(msg)
r <- receipt(response)

if(is.acknowledged(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi',
                  receipt=r))
{
  cat('Message has been read.\n')
}

## End(Not run)
```

request,PushoverResponse-method

Return the Pushover response request token

Description

Responses to Pushover queries include a unique request token. This method returns the request token for the given PushoverResponse object.

Usage

```
## S4 method for signature 'PushoverResponse'
request(object)
```

Arguments

object A [PushoverResponse-class](#) object

Value

The string request token included as response to a Pushover query

send, PushoverMessage-method
Send a Pushover message

Description

send sends the given Pushover message (represented by a [PushoverMessage-class](#)) object

Usage

```
## S4 method for signature 'PushoverMessage'
send(object)
```

Arguments

object A [PushoverMessage-class](#) object

Value

A [PushoverResponse](#) object containing information about the server's response.

Examples

```
## Not run:
library(pushoverrr)

message <- PushoverMessage(message='Hello World',
                             token='KzGD0RePK8gMac0Q0YAMyEEuzJnyUi',
                             user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG')
response <- send(message)

## End(Not run)
```

send_pushovermessage *Send a PushoverMessage object*

Description

send_PushoverMessage sends a given PushoverMessage object to Pushover. This function is not directly called, but is used when [send](#) is called with a PushoverMessage object.

Usage

```
send_pushovermessage(object)
```

Arguments

object A [PushoverMessage-class](#) object

Value

A [PushoverResponse](#) object containing information about the server's response.

set_pushover_app	<i>Store Pushover app and user information and use for all subsequent queries</i>
------------------	---

Description

set_pushover_app allows an application token to be stored that will be used for all subsequent calls that require a token (e.g., [pushover](#)).

pushover_app.isset determines whether or not a Pushover app has been set

unset_pushover_app removes a stored Pushover app token, which means that a token will have to be provided to all subsequent API calls.

get_pushover_app returns the token associated with the current Pushover app

set_pushover_user allows a user key to be stored that will be used for all subsequent calls that require a user key (e.g., [pushover](#)).

pushover_user.isset indicates whether or not a Pushover user/group is currently set

unset_pushover_user removes a stored Pushover user/group, which means that a key will have to be provided to all subsequent API calls.

get_pushover_user gets the key associated with the current Pushover user/group

Usage

```
set_pushover_app(token, user = NA)
```

```
pushover_app.isset()
```

```
unset_pushover_app()
```

```
get_pushover_app()
```

```
set_pushover_user(user)
```

```
pushover_user.isset()
```

```
unset_pushover_user()
```

```
get_pushover_user()
```

Arguments

token	The application token
user	If a user/group key is given, that information will also be saved for subsequent queries. This is equivalent to calling <code>set_pushover_user</code> .

Value

`pushover_app.isset` returns a boolean indicating whether the app token has been set (TRUE) or not (FALSE).

`get_pushover_app` returns a string representing the token associated with the current app (if one is set)

`pushover_user.isset` returns boolean indicating whether the user/group is set (TRUE) or not (FALSE).

`get_pushover_user` returns a string containing a Pushover user/group key

Note

This behavior can temporarily be overridden by providing an app token or user key as argument to the function in question.

The Pushover API calls don't differentiate between users and groups (and therefore neither does `pushover`). This function will remove any user or group key that is saved.

Examples

```
## Not run:
# Set the Pushover user account to use
set_pushover_app('KzGD0RePK8gMaC0Q0YAMyEEuzJnyUi')

# Determine whether the Pushover user account has been set
if(pushover_app.isset())
{
  cat(paste('The Pushover app token has been set to', get_pushover_app()))
}

## End(Not run)
## Not run:
# Set the Pushover user account to use
set_pushover_user('uQiRzpo4DXghDmr9QzzfQu27cmVRsG')

# Determine whether the Pushover user account has been set
if(pushover_user.isset())
{
  cat(paste('Pushover user has been set to', get_pushover_user()))
}
```

When both the app token and user account have been set, we can send a message without providing either of them:

```
pushover('this took so much less typing!')
pushover_high('and again!')
```

```
## End(Not run)
## Not run:
# Forget about the current Pushover user
unset_pushover_user()

## End(Not run)
```

show,PushoverMessage-method

Print information about a PushoverMessage object

Description

show_PushoverMessage prints information about a given PushoverMessage object. This function is automatically called when a PushoverMessage object is referenced alone, when returned from a function and not stored, or when [show](#) is called with a PushoverMessage object.

Usage

```
## S4 method for signature 'PushoverMessage'
show(object)
```

Arguments

object A [PushoverMessage-class](#) object

show,PushoverResponse-method

Print information about a PushoverResponse object

Description

show prints information about a given [PushoverResponse-class](#) object. This function is automatically called when a PushoverResponse object is referenced alone, when returned from a function and not stored, or when [show](#) is called with a [PushoverResponse-class](#) object.

Usage

```
## S4 method for signature 'PushoverResponse'
show(object)
```

Arguments

object A [PushoverResponse-class](#) object

status, PushoverResponse-method

Return the Pushover response status

Description

Pushover API calls return a status code, which indicates the success of a query (1=success, 0=problem). The status method extracts the numeric status code from a [PushoverResponse-class](#) object.

Usage

```
## S4 method for signature 'PushoverResponse'
status(object)
```

Arguments

object A [PushoverResponse-class](#) object

Value

The numeric status value stored in the object (1=success, 0=problem)

See Also

[is.success](#)

validate_key

Determine whether or not a given user/group key is valid

Description

validate_key issues a query to Pushover to determine whether or not a given user (or group) key is valid. If a device is specified, the query will also see if the given device is registered to that user.

is.valid_key returns a boolean value indicating whether the user/device is valid (TRUE) or not (FALSE).

Usage

```
validate_key(user, device = NA_character_, ...)
```

```
is.valid_key(user, device = NA, ...)
```

Arguments

user	A user or group key (e.g., 'uQiRzpo4DXghDmr9QzzfQu27cmVRsG')
device	A device name (e.g., 'phone')
...	Any additional parameters, such as an application token set_pushover_app .

Value

validate_key returns a [PushoverResponse](#) object containing the response from the server
 is.valid_key returns a boolean value indicating whether the user/device is valid (TRUE) or not (FALSE).

Note

To acquire a user key, create an account at <https://pushover.net>

Examples

```
## Not run:
response <- validate_key(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyU',
                        user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG')
response_dev <- validate_key(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyU',
                            user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG',
                            device='phone')

if(is.valid_key(token='KzGD0RePK8gMaC0Q0YAMyEEuzJnyU',
               user='uQiRzpo4DXghDmr9QzzfQu27cmVRsG',
               device='phone'))
{
  cat('I can send to this device!')
}

## End(Not run)
```

validate_PushoverMessage

Validate a given PushoverMessage object

Description

validate_PushoverMessage determines whether or not a given PushoverMessage object has valid slot values. It does not determine whether or not the given application token or user/group keys are registered with Pushover. This second step can be done with [validate_key](#) or [is.valid_key](#). This function is automatically called when a PushoverMessage object is created or [validObject](#) is called with an existing PushoverMessage object.

Usage

```
validate_PushoverMessage(object)
```

Arguments

object A [PushoverMessage](#) object

Value

A boolean value indicating if the PushoverMessage object is valid (TRUE) or not (FALSE)

Note

To acquire a user key, register an account at <https://pushover.net>

To acquire an application token, register your token at <https://pushover.net/apps>

validate_PushoverResponse

Validate a given PushoverResponse object

Description

validate_PushoverResponse determines whether or not a given PushoverResponse object has valid slot values. This function is automatically called when a new PushoverResponse object is created or [validObject](#) is called with an existing object.

Usage

```
validate_PushoverResponse(object)
```

Arguments

object A [PushoverResponse](#) object

Value

A boolean value indicating if the PushoverResponse object is valid (TRUE) or not (FALSE)

Index

- *Topic **datasets**
 - pushover_priorities, 16
- as.numeric, 4
- check_receipt, 2, 9, 16, 17
- content_value
 - (content_value, PushoverResponse-method), 3
- content_value, PushoverResponse-method, 3
- GenPushoverMessage
 - (PushoverMessage-class), 12
- GenPushoverResponse
 - (PushoverResponse-class), 15
- get_devices, 4
- get_pushover_app (set_pushover_app), 19
- get_pushover_sounds, 5, 12
- get_pushover_user (set_pushover_app), 19
- headers
 - (headers, PushoverResponse-method), 6
- headers, PushoverResponse-method, 6
- http_status_code
 - (http_status_code, PushoverResponse-method), 6
- http_status_code, PushoverResponse-method, 6
- is.acknowledged, 16, 17
- is.acknowledged (check_receipt), 2
- is.device (get_devices), 4
- is.success, 22
- is.success
 - (is.success, PushoverResponse-method), 7
- is.success, PushoverResponse-method, 7
- is.valid_device, 8
- is.valid_key, 23
- is.valid_key (validate_key), 22
- is.valid_receipt, 8
- is.valid_token, 9
- new, 15
- pushover, 10, 13, 19
- pushover_app.isset (set_pushover_app), 19
- pushover_emergency, 13
- pushover_emergency (pushover), 10
- pushover_high, 13
- pushover_high (pushover), 10
- pushover_normal, 13
- pushover_normal (pushover), 10
- pushover_priorities, 16
- pushover_quiet, 13
- pushover_quiet (pushover), 10
- pushover_silent, 13
- pushover_silent (pushover), 10
- pushover_user.isset (set_pushover_app), 19
- PushoverMessage, 11, 12, 13, 24
- PushoverMessage-class, 12
- pushoverr, 13
- pushoverr-package (pushoverr), 13
- PushoverResponse, 3, 13, 14, 15, 18, 19, 23, 24
- PushoverResponse-class, 15
- receipt, 3
- receipt
 - (receipt, PushoverResponse-method), 16
- receipt, PushoverResponse-method, 16
- request
 - (request, PushoverResponse-method), 17
- request, PushoverResponse-method, 17
- send, 18

send (send,PushoverMessage-method), [18](#)
send,PushoverMessage-method, [18](#)
send_pushovermessage, [18](#)
set_pushover_app, [3](#), [5](#), [10](#), [11](#), [19](#), [23](#)
set_pushover_user, [5](#), [10](#), [11](#), [20](#)
set_pushover_user (set_pushover_app), [19](#)
show, [21](#)
show,PushoverMessage-method, [21](#)
show,PushoverResponse-method, [21](#)
status, [7](#)
status
 (status,PushoverResponse-method),
 [22](#)
status,PushoverResponse-method, [22](#)

unset_pushover_app (set_pushover_app),
 [19](#)
unset_pushover_user (set_pushover_app),
 [19](#)

validate_key, [22](#), [23](#)
validate_PushoverMessage, [23](#)
validate_PushoverResponse, [24](#)
validObject, [23](#), [24](#)