Package 'RPushbullet'

July 21, 2019

Type Package

Title R Interface to the Pushbullet Messaging Service

Version 0.3.2

Date 2019-07-21

Author Dirk Eddelbuettel with contributions by Bill Evans, Mike Birdgeneau, Henrik Bengtsson, Seth Wenchel, Colin Gillespie and Chanyub Park

Maintainer Dirk Eddelbuettel <edd@debian.org>

Description An R interface to the Pushbullet messaging service which provides fast and efficient notifications (and file transfer) between computers, phones and tablets. An account has to be registered at the site <http://www.pushbullet.com> site to obtain a (free) API key.

Imports utils, stats, jsonlite, curl

SystemRequirements A user API key (which one can request from the website at <http://www.pushbullet.com>), and one or more devices to push messages to which may be any one of an (Android or iOS) phone, a (Chrome or Firefox, or Opera or Safari) browser or the (Windows or Mac) desktop application provided the corresponding Pushbullet 'app' has been installed on any one of these.

NeedsCompilation no

License GPL (>= 2)

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2019-07-21 13:40:02 UTC

R topics documented:

`RPushbullet-package`

R interface to the Pushbullet service

Description

The Pushbullet[1] service permits users to pass messenges between their computers, phones and other devices such as tablets. It offers immediacy which is perfect for alerting, and much more.

This package provides a programmatic interface from R.

Details

The Pushbullet API[2] offers a RESTful interface which requires an API key. A key can be obtained free of charge from Pushbullet[3]. Given such a key, and one or more registered devices, users can push messages to one or more device, or a given email address.

The main function is `pbPost` which can be used to send a message comprising a note (with free-form body and title), link (for sending a URL), or even a file. The message recipients is typically one (or several) of the devices known to the user (see the next section for details), it can also be an email address in which case Pushbullet[4] creates and sends an email to the given address.

Initialization

The authentication key, as well as the device id, nicknames for the devices and default device can all be declared in several ways.

One possibility is to use a file `.rpushbullet.json` in the `$HOME` directory. (Note that on Windows you may have to set the `$HOME` environment variable.) It uses the JSON format which uses a key:value pair notation; values may be arrays. A simple example follows.

```
{
    "key": "abc...YourKeyHereBetweenQuote....xyz",

    "devices": [
        "abc...SomeId.......xyz",
        "abc...SomeOtherId..xyz"
    ],

    "names": [
        "Phone",
```

[1] `http://www.pushbullet.com`
[2] `https://www.pushbullet.com/api`
[3] `https://www.pushbullet.com`
[4] `http://www.pushbullet.com`

```
        "Browser"
    ],

    "defaultdevice": "Phone"
}
```

The entire block is delimited by a pair of curly braces. Within the curly braces we have "key" and "devices" which are mandatory. Here "key" is expected to contain a single value; "devices" can be an array which is denoted by square brackets. Optionally a "names" single value or array can be used to assign nicknames to the devices. Lastly, a "defaultdevice" can be designated as well.

However, use of a configuration file is not mandatory. The arguments can also be supplied as global options (which could be done in the usual R startup files, see `Startup` for details) as well as via standard function arguments when calling the corresponding functions. When using global options, use the names `rpushbullet.key`, `rpushbullet.devices`, `rpushbullet.names`, and `rpushbullet.defaultdevice` corresponding to the entries in the JSON file shown above.

The `curl` binary is required, and is located at package initialization, along with the other load-time intializations described here. It is therefore strongly recommended to attach the package in the normal way via `library(RPushbullet)` rather than trying to access functions from the package namespace.

Author(s)

Dirk Eddelbuettel

References

See the Pushbullet documentation at the Pushbullet website[5].

See Also

The documentation for the main function `pbPost`, as well as the documentation for `pbGetDevices`.

`pbGetChannelInfo` *Details for a channel*

Description

Details for a channel

[5] `http://www.pushbullet.com`

Usage

```
pbGetChannelInfo(channel, no_recent_pushes = FALSE)

## Default S3 method:
pbGetChannelInfo(channel, no_recent_pushes = FALSE)

## S3 method for class 'pbChannelInfo'
print(x, ...)

## S3 method for class 'pbChannelInfo'
summary(object, ...)
```

Arguments

`channel`	The name of a Pushbullet channel as a string
`no_recent_pushes`	Should the returned returned object exclude recent pushs? `FALSE` (the default) will return up to 10 pushes. `TRUE` will exclude them
`x`	Default object for `print` method
`...`	Other optional arguments
`object`	Default object for `summary` method

Value

a list with infoabout a channel

Examples

```
xkcd <- pbGetChannelInfo("xkcd", TRUE)
summary(xkcd)
```

`pbGetDevices` *Get registered Pushbullet devices*

Description

Retrieve the list of devices registered for the given API key.

Usage

```
pbGetDevices(apikey = .getKey())

## Default S3 method:
pbGetDevices(apikey = .getKey())

## S3 method for class 'pbDevices'
```

```
print(x, ...)

## S3 method for class 'pbDevices'
summary(object, ...)
```

Arguments

apikey	The API key used to access the service. It can be supplied as an argument here, or via the file ~/.rpushbullet.json which is read at package initialization.
x	Default object for print method
...	Other optional arguments
object	Default object for summary method

Details

This function invokes the 'devices' functionality of the Pushbullet API; see https://docs.pushbullet.com/v2/devices for more details.

Value

The resulting JSON record is converted to a list and returned as a pbDevices object with appropriate print and summary methods.

Author(s)

Dirk Eddelbuettel

---

pbGetPosts *Get messages posted via Pushbullet*

Description

This function gets messages posted to Pushbullet.

Usage

```
pbGetPosts(apikey = .getKey(), limit = 10)
```

Arguments

apikey	The API key used to access the service. It can be supplied as an argument here, via the global option rpushbullet.key, or via the file ~/.rpushbullet.json which is read at package initialization (and, if found, also sets the global option). ~/.rpushbullet.json which is read at package initialization.
limit	Limit number of post. Default is 10.

Value

A data.frame result record is returned

Author(s)

Chanyub Park

Examples

```
## Not run:
pbGetPosts()

## End(Not run)
```

`pbGetUser` *Get info about a user*

Description

Get info about a user

Usage

```
pbGetUser(apikey = .getKey())

## Default S3 method:
pbGetUser(apikey = .getKey())

## S3 method for class 'pbUser'
print(x, ...)

## S3 method for class 'pbUser'
summary(object, ...)
```

Arguments

`apikey`	The API key used to access the service. It can be supplied as an argument here, or via the file `~/.rpushbullet.json` which is read at package initialization.
`x`	Default object for `print` method
`...`	Other optional arguments
`object`	Default object for `summary` method

Value

Invisibly returns info about a user

Examples

```
## Not run:
me <- pbGetUser()
summary(me)

## End(Not run)
```

`pbPost` *Post a message via Pushbullet*

Description

This function posts a message to Pushbullet. Different types of messages are supported: 'note', 'link', 'address', or 'file'.

Usage

```
pbPost(type = c("note", "link", "file"), title = "", body = "",
  url = "", filetype = "text/plain", recipients, email, channel, deviceind,
  apikey = .getKey(), devices = .getDevices(), verbose = FALSE,
  debug = FALSE)
```

Arguments

`type`	The type of post: one of 'note', 'link', or 'file'.
`title`	The title of the note being posted.
`body`	The body of the note or the (optional) body when the `type` is 'link'.
`url`	The URL of `type` is 'link', or the local path of a file to be sent if `type` is 'file'.
`filetype`	The MIME type for the file at `url` (if `type` is 'file') such as "text/plain" or "image/jpeg", defaults to "text/plain".
`recipients`	A character or numeric vector indicating the devices this post should go to. If missing, the default device is looked up from an optional setting, and if none has been set the push is sent to all devices.
`email`	An alternative way to specify a recipient is to specify an email address. If both `recipients` and `email` are present, `recipients` is used.
`channel`	A channel tag used to specify the name of the channel as the recipient. If either `recipients` or `email` are present, they will take precedence over `channel`.
`deviceind`	(Deprecated) The index (or a vector/list of indices) of the device(s) in the list of devices.
`apikey`	The API key used to access the service. It can be supplied as an argument here, via the global option `rpushbullet.key`, or via the file `~/.rpushbullet.json` which is read at package initialization (and, if found, also sets the global option).
`devices`	The device to which this post is pushed. It can be supplied as an argument here, or via the file `~/.rpushbullet.json` which is read at package initialization.

verbose	Boolean switch to add additional output
debug	Boolean switch to add even more debugging output

Details

This function invokes the 'pushes' functionality of the Pushbullet API; see https://docs.pushbullet.com/v2/pushes for more details.

When a 'note' is pushed, the recipient receives the title and body of the note. If a 'link' is pushed, the recipient's web browser is opened at the given URL. If an 'address' is pushed, the recipient's web browser is opened in map mode at the given address.

If 'recipients' argument is missing, the post is pushed to *all* devices in accordance with the API definition. If 'recipients' is text vector, it matched against the device names (from either the config file or a corresponding option). Lastly, if 'recipients' is a numeric vector, the post is pushed the corresponding elements in the devices vector.

In other words, the default of value of no specified recipients results in sending to all devices. If you want a particular subset of devices you have to specify it name or index. A default device can be set in the configuration file, or as a global option. If none is set, zero is used as a code to imply 'all' devices.

The earlier argument deviceind is now deprecated and will be removed in a later release.

Value

A JSON result record is return invisibly

Author(s)

Dirk Eddelbuettel

Examples

```
## Not run:
# A note
pbPost("note", "A Simple Test", "We think this should work.\nWe really do.")

# A URL -- should open browser
pbPost(type="link", title="Some title", body="Some URL",
       url="http://cran.r-project.org/package=RPushbullet")

# A file
pbPost(type="file", url=system.file("DESCRIPTION", package="RPushbullet"))

## End(Not run)
```

pbSetup *Create a JSON config file*

Description

Create a JSON config file

Usage

```
pbSetup(apikey, conffile, defdev)
```

Arguments

apikey	An *Access Token* provided by Pushbullet (see details). If not provided in the function call, the user will be prompted to enter one.
conffile	A string giving the path where the configuration file will be written. RPushbullet will automatically attempt load from the default location `~/.rpushbullet.json` (which can be changed via a `rpushbullet.dotfile`) entry in `options`).
defdev	An optional value for the default device; if missing (or `NA`) then an interactive prompt is used.

Details

This function writes a simple default configuration file based on a given apikey. It is intended to be run once to help new users setup RPushbullet. Running multiple times without overriding the `config_file` parameter will overwrite the default file. An *Access Token* may be obtained for free by logging into the Pushbullet website, going to `https://www.pushbullet.com/#settings`, and clicking on "Create Access Token".

Value

`NULL` is returned invisibly, but the function is called for its side effect of creating the configuration file.

Author(s)

Seth Wenchel and Dirk Eddelbuettel

Examples

```
## Not run:
# Interactive mode.  Just follow the prompts.
pbSetup()

## End(Not run)
```

`pbValidateConf` *Check if a configuration is valid*

Description

Check if a configuration is valid

Usage

```
pbValidateConf(conf = NULL)
```

Arguments

`conf`	Either a file path (like `~/.rpushbullet.json`) or a JSON string. If `NULL` (the default), the value of `getOption("rpushbullet.dotfile")` will be used.

Value

`TRUE` if both the api key and *all* devices are vaild. `FALSE` otherwise.

Examples

```
pbValidateConf('{"key":"a_fake_key","devices":["dev_iden1","dev_iden2"]}')
```