

Package ‘sensibo.sky’

October 14, 2022

Title Access to 'Sensibo Sky' API V2 for Air Conditioners Remote Control

Version 1.0.0

Description Provides an interface to the 'Sensibo Sky' API which allows to remotely control non-smart air conditioning units.

See <<https://sensibo.com>> for more informations.

URL <https://github.com/theclue/sensibo.sky>

BugReports <https://github.com/theclue/sensibo.sky/issues>

Depends R (>= 3.0)

License MIT + file LICENSE

LazyData true

NeedsCompilation no

Imports httr, jsonlite, glue

Suggests testthat

RoxygenNote 6.1.1

Author Gabriele Baldassarre [aut, cre]

Maintainer Gabriele Baldassarre <gabriele@gabrielebaldassarre.com>

Repository CRAN

Date/Publication 2019-06-24 13:10:03 UTC

R topics documented:

sensibo.sky-package	2
sensibo.pod.historical	2
sensibo.pod.info	3
sensibo.pod.probe	3
sensibo.pod.set	4
sensibo.pod.smartmode	5
sensibo.pod.smartmode.set	6
sensibo.pod.state	7
sensibo.pod.states	8
sensibo.pods	9

Index**10**

sensibo.sky-package *Access to Sensibo Sky API V2 for Smart A/C Remote Control*

Description

Access to Sensibo Sky API V2 for Smart A/C Remote Control

sensibo.pod.historical

Probe the historical measurements sent by a given air conditioner (pod) for up to 7 days in the past.

Description

Probe the historical measurements sent by a given air conditioner (pod) for up to 7 days in the past.

Usage

```
sensibo.pod.historical(pod, days = 1, key = getOption("sensibo.key"))
```

Arguments

pod	(character) Pod unique id.
days	(integer) The number of days of measurements to get (max 7), including current day.
key	(character) API key from https://home.sensibo.com/me/api .

Value

A data frame with with time, temperature and humidity values.

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Get a week of data values from the first pod
pod.data <- sensibo.pod.historical(pods.id[1], days = 7)

## End(Not run)
```

sensibo.pod.info	<i>Gets additional info of a specific air conditioner (pod) of the authenticated user.</i>
------------------	--

Description

Gets additional info of a specific air conditioner (pod) of the authenticated user.

Usage

```
sensibo.pod.info(pod, key = getOption("sensibo.key"))
```

Arguments

pod	(character) Pod unique id.
key	(character) API key from https://home.sensibo.com/me/api .

Value

A list of the available details for the given pod or an empty list if the pod doesn't exist.

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Getting the details for the first pod
pod.details <- sensibo.pod.info(pods.id[1])

## End(Not run)
```

sensibo.pod.probe	<i>Probe the last measurements sent by a given air conditioner (pod). In a typical configuration, Sensibo Sky Remote send fresh data to the server every 90 seconds.</i>
-------------------	--

Description

Probe the last measurements sent by a given air conditioner (pod). In a typical configuration, Sensibo Sky Remote send fresh data to the server every 90 seconds.

Usage

```
sensibo.pod.probe(pod, key = getOption("sensibo.key"))
```

Arguments

```
pod          (character) Pod unique id.
key          (character) API key from https://home.sensibo.com/me/api.
```

Value

A list with time, temperature and humidity values. A 'seconds since last send' field is also provided.

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Get the most current data probed by the first pod
pod.fresh.data <- sensibo.pod.probe(pods.id[1])

## End(Not run)
```

sensibo.pod.set	<i>Set the air conditioner (pod) state. NULL properties will be left unchanged. Please bear in mind that valid values might depend of the A/C model.</i>
-----------------	--

Description

Set the air conditioner (pod) state. NULL properties will be left unchanged. Please bear in mind that valid values might depend of the A/C model.

Usage

```
sensibo.pod.set(pod, on = NULL, mode = NULL, fan = NULL,
  unit = NULL, temperature = NULL, swing = NULL,
  key = getOption("sensibo.key"))
```

Arguments

pod	(character) Pod unique id.
on	(logical) Set to TRUE to turn the pod on, FALSE to turn off instead.
mode	(character) Set the mode from 'cool', 'hot', 'dry' and 'fan'
fan	(character) Set the fan speed from 'low', 'medium', 'high' and 'auto'.
unit	(character) Set the temperature unit: 'C' for Celsius, 'F' for Fahrenheit.
temperature	(numeric) Set the target temperature
swing	(character) Set the swing mode from 'stopped' and 'rangeFull'
key	(character) API key from https://home.sensibo.com/me/api .

Value

A list with the updated state and the result of the command (Success/Failure).

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Update the status of the first pod
pod.newstate <- sensibo.pod.set(pods.id[1], on = TRUE, mode = "cool", temperature = 26)

## End(Not run)
```

`sensibo.pod.smartmode` *Get the climate react settings for a given air conditioner (pod).*

Description

Get the climate react settings for a given air conditioner (pod).

Usage

```
sensibo.pod.smartmode(pod, key = getOption("sensibo.key"))
```

Arguments

pod	(character) Pod unique id.
key	(character) API key from https://home.sensibo.com/me/api .

Value

A list with the Climate React feature settings.

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Get the climate react settings for the first pod
pod.smartsettings <- sensibo.pod.smartmode(pods.id[1])

## End(Not run)
```

sensibo.pod.smartmode.set

Set the air conditioner (pod) Climate Reactmode. Set to TRUE to enable and FALSE to disable.

Description

Set the air conditioner (pod) Climate Reactmode. Set to TRUE to enable and FALSE to disable.

Usage

```
sensibo.pod.smartmode.set(pod, enable = NULL,
  key = getOption("sensibo.key"))
```

Arguments

pod	(character) Pod unique id. Required: TRUE
enable	(logical) Set to TRUE to turn on the climate react, FALSE to turn off instead.
key	(character) API key from https://home.sensibo.com/me/api.

Value

A list with the updated status of Climate React settings change.

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Enable the Climate React mode for the first pod
sensibo.pod.smartmode.set(pods.id[1], TRUE)

## End(Not run)
```

sensibo.pod.state *Get info from a specific state of a given air conditioner (pod).*

Description

Get info from a specific state of a given air conditioner (pod).

Usage

```
sensibo.pod.state(pod, state, key = getOption("sensibo.key"))
```

Arguments

pod	(character) Pod unique id.
state	(character) State id to be retrieved.
key	(character) API key from https://home.sensibo.com/me/api.

Value

A list with the requested state details.

Examples

```
## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Getting the current state of the first pod
```

```

pod.current <- sensibo.pod.states(pods.id[1], n = 1)

## Get more details of the given state (if available)
pod.state.details <- sensibo.pod.state(pods.id[1], pod.current[1])

## End(Not run)

```

`sensibo.pod.states` *Get the specified air conditioner (pod) current and previous states.*

Description

Get the specified air conditioner (pod) current and previous states.

Usage

```
sensibo.pod.states(pod, n = 10, key = getOption("sensibo.key"))
```

Arguments

<code>pod</code>	(character) Pod unique id.
<code>n</code>	(integer) The number of states to get (max 20); n=1 will get only the current state.
<code>key</code>	(character) API key from https://home.sensibo.com/me/api .

Value

A list with the requested states, starting from the current one.

Examples

```

## Not run:
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api
# and added to a 'sensibo.sky' global option.
#
# options("sensibo.key" = <Your Sensibo API Key>)

## Getting the list of pods available to the user
pods.id <- sensibo.pods()

## Getting the current state of the first pod
## Even if we're asking for the current state only, a list is returned anyway
pod.current <- sensibo.pod.states(pods.id[1], n = 1)

## End(Not run)

```

sensibo.pods	<i>Gets a list of the air conditioners (pods) IDs of the authenticated user.</i>
--------------	--

Description

Gets a list of the air conditioners (pods) IDs of the authenticated user.

Usage

```
sensibo.pods(key = getOption("sensibo.key"))
```

Arguments

key (character) API key from <https://home.sensibo.com/me/api>.

Value

A character vectors with available pods unique IDs.

Examples

```
## Not run:  
# Assuming that a valid Sensibo Sky API Key was created on https://home.sensibo.com/me/api  
# and added to a 'sensibo.sky' global option.  
#  
# options("sensibo.key" = <Your Sensibo API Key>)  
  
## Getting the list of pods available to the user  
pods.id <- sensibo.pods()  
  
## End(Not run)
```

Index

`sensibo.pod.historical`, [2](#)
`sensibo.pod.info`, [3](#)
`sensibo.pod.probe`, [3](#)
`sensibo.pod.set`, [4](#)
`sensibo.pod.smartmode`, [5](#)
`sensibo.pod.smartmode.set`, [6](#)
`sensibo.pod.state`, [7](#)
`sensibo.pod.states`, [8](#)
`sensibo.pods`, [9](#)
`sensibo.sky` (`sensibo.sky-package`), [2](#)
`sensibo.sky-package`, [2](#)