Package: huecontroller (via r-universe)

September 18, 2024

Title Control Philips Hue lights using the R programming language

Version 0.0.0.9005

Description Control Philips Hue lights using the R programming language

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports usethis, magrittr, purrr, httr, jsonlite, shinyWidgets, shinyjs, shiny

URL https://giocomai.github.io/huecontroller

Repository https://giocomai.r-universe.dev

RemoteUrl https://github.com/giocomai/huecontroller

RemoteRef HEAD

RemoteSha 5ae53ed2e74200b6c0c9731c248efcf6ec08d736

Contents

`hue_convert_to_hue_sat` *Convert hex colour to*

Description

Convert hex colour to

Usage

```
hue_convert_to_hue_sat(colour)
```

Arguments

`colour`	A colour name as listed by `colours()` or a hexadecimal colour string.

Value

A named list with two elements: `hue` and `sat`

Examples

```
hue_convert_to_hue_sat("red")
hue_convert_to_hue_sat("#E414FF")
if (interactive()) {
  hue_convert_to_hue_sat(colourpicker::colourPicker(numCols = 1))
}
```

hue_delete_group *Delete group*

Description

Delete group

Usage

```
hue_delete_group(id)
```

Arguments

id	If numeric, numeric id of group. If character, name of group You can check id and names with `hue_get_groups_names()`

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_delete_light *Delete light*

Description

Delete light

Usage

```
hue_delete_light(id)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_lights_names()`

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_get_groups *Get all details about groups*

Description

Get all details about groups

Usage

```
hue_get_groups()
```

Value

A list

hue_get_groups_names *Get names of available groups*

Description

Get names of available groups

Usage

```
hue_get_groups_names()
```

hue_get_group_lights *Get state of given group*

Description

Get state of given group

Usage

```
hue_get_group_lights(id)
```

Arguments

id	If numeric, numeric id of group. If character, name of group. You can check id and names with `hue_get_groups_names()`

Value

An integer vector with id of lights included in a group.

hue_get_group_state *Get state of given group*

Description

Get state of given group

Usage

```
hue_get_group_state(id)
```

Arguments

id	If numeric, numeric id of group. If character, name of group. You can check id and names with `hue_get_groups_names()`

Value

A list with details on the state and attribute of the given group

hue_get_lights *Get details about all lights*

Description

Get details about all lights

Usage

```
hue_get_lights()
```

Value

A list with details about all lights

hue_get_lights_names *Get names of all available lights*

Description

Get names of all available lights

Usage

```
hue_get_lights_names()
```

hue_get_light_state *Get light state*

Description

Get light state

Usage

```
hue_get_light_state(id)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_lights_names()`

hue_mod_group_card_server *Server function for shiny app*

Description

Server function for shiny app

Usage

```
hue_mod_group_card_server(id)
```

Arguments

id	Module id

hue_mod_group_card_ui *UI function for shiny app*

Description

UI function for shiny app

Usage

```
hue_mod_group_card_ui(
  id,
  group,
  onoff,
  brightness,
  temperature,
  min_temperature,
  max_temperature
)
```

Arguments

id	Module id
group	Group id

hue_mod_light_card_server *Server function for shiny app*

Description

Server function for shiny app

Usage

```
hue_mod_light_card_server(id)
```

Arguments

id	Module id

hue_mod_light_card_ui *UI function for shiny app*

Description

UI function for shiny app

Usage

```
hue_mod_light_card_ui(
  id,
  light,
  onoff,
  brightness,
  temperature,
  min_temperature,
  max_temperature
)
```

Arguments

id	Module id
light	Light id

hue_output_group_id *Check if given id is valid*

Description

Check if given id is valid

Usage

```
hue_output_group_id(id)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_lights_names()`

Value

Always return an integer corresponding to a given light.

hue_output_id *Check if given id is valid*

Description

Check if given id is valid

Usage

```
hue_output_id(id)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_lights_names()`

Value

Always return an integer corresponding to a given light.

hue_settings *Set IP and username to interact with the api*

Description

Set IP and username to interact with the api

Usage

```
hue_settings(ip = NULL, username = NULL)
```

Arguments

ip	A character vector of length one. The IP of your hue bridge, e.g. "192.168.0.1"
username	A username generated to interact with the API. For details, https://developers.meethue.com/develop/get-started-2/

hue_set_group_brightness *Set brightness of group*

Description

Set brightness of group

Usage

```
hue_set_group_brightness(id, brightness, by = 10)
```

Arguments

id	If numeric, numeric id of group. If character, name of group. You can check id and names with `hue_get_groups_names()`
brightness	A numeric value, or one of either "brighter" (or "+") or "darker" (or "-"). The highest value is 254, the lowest 1.
by	Numeric, defaults to 10. Ignored if brightness is numeric. Otherwise determines the size of the increment/decrement.

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_set_group_colour *Set colour of a group of lights with relevant capabilities*

Description

N.B. Colour accuracy may not be perfect. See the official documentation for details, and consider using custom parameters with `hue_set_group_state()` if this function does not achieve what you expect. https://developers.meethue.com/develop/application-design-guidance/color-conversion-formulas-rgb-to-xy-and-back/

Usage

```
hue_set_group_colour(id, colour, transition_time = 0.4)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_groups_names()`
colour	A colour name as listed by `colours()` or a hexadecimal colour string.
transition_time	Duration of the transition in seconds. Defaults to 0.4 seconds.

Examples

```
if (interactive()) {
  hue_set_group_colour(id = 11, colour = "green")
}
```

`hue_set_group_state` *Change state of Hue groups*

Description

Change state of Hue groups

Usage

```
hue_set_group_state(id, params)
```

Arguments

`id`	If numeric, numeric id of group. If character, name of group. You can check id and names with `hue_get_groups_names()`
`params`	A named list. For a full list of available parameters, check https://developers.meethue.com/develop/hue-api/groupds-api/

`hue_set_group_temperature`
Set temperature of group

Description

Set temperature of group

Usage

```
hue_set_group_temperature(id, temperature, by = 10)
```

Arguments

`id`	If numeric, numeric id of group. If character, name of group. You can check id and names with `hue_get_groups_names()`
`temperature`	A numeric value, or one of either "warmer" (or "+") or "colder" (or "-"). The warmest value available is usually 500, the coldest 153.
`by`	Numeric, defaults to 10. Ignored if temperature is numeric. Otherwise determines the size of the increment/decrement.

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_set_light_brightness *Set brightness of light*

Description

Set brightness of light

Usage

```
hue_set_light_brightness(id, brightness, by = 10)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with hue_get_lights_names()
brightness	A numeric value, or one of either "brighter" (or "+") or "darker" (or "-"). The highest value is 254, the lowest 1.
by	Numeric, defaults to 10. Ignored if brightness is numeric. Otherwise determines the size of the increment/decrement.

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_set_light_colour *Set colour of a Hue light with relevant capabilities*

Description

N.B. Colour accuracy may not be perfect. See the official documentation for details, and consider using custom parameters with hue_set_light_state() if this function does not achieve what you expect. https://developers.meethue.com/develop/application-design-guidance/color-conversion-formulas-rgb-to-xy-and-back/

Usage

```
hue_set_light_colour(id, colour, transition_time = 0.4)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with hue_get_lights_names()
colour	A colour name as listed by colours() or a hexadecimal colour string.
transition_time	Duration of the transition in seconds. Defaults to 0.4 seconds.

Examples

```
if (interactive()) {
  hue_set_light_colour(id = 11, colour = "blue")
}
```

`hue_set_light_state` *Change state of Hue lights*

Description

Change state of Hue lights

Usage

```
hue_set_light_state(id, params)
```

Arguments

`id`	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_lights_names()`
`params`	A named list. For a full list of available parameters, check https://developers.meethue.com/develop/hue-api/lights-api/

`hue_set_light_temperature`
Set temperature of light

Description

Set temperature of light

Usage

```
hue_set_light_temperature(id, temperature, by = 10)
```

Arguments

`id`	If numeric, numeric id of light. If character, name of light. You can check id and names with `hue_get_lights_names()`
`temperature`	A numeric value, or one of either "warmer" (or "+") or "colder" (or "-"). The warmest value available is usually 500, the coldest 153.
`by`	Numeric, defaults to 10. Ignored if temperature is numeric. Otherwise determines the size of the increment/decrement.

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_shiny_controller *Interactive interface to control lights*

Description

options Options to be passed to the shiny app. See `?shiny::shinyApp()` for details.

Usage

```
hue_shiny_controller(options = list())
```

hue_turn_group_off *Turn off group*

Description

Turn off group

Usage

```
hue_turn_group_off(id)
```

Arguments

`id`	If numeric, numeric id of group. If character, name of group. You can check id and names with `hue_get_groups_names()`

Value

Silently returns http response from the bridge, but mostly used for its side effects.

hue_turn_group_on *Turn on group*

Description

Turn on group

Usage

```
hue_turn_group_on(id)
```

Arguments

id	If numeric, numeric id of group. If character, name of group. You can check id and names with hue_get_groups_names()

Value

Silently returns http response from the bridge, but mostly used for its side effects.

---

hue_turn_light_off *Turn off light*

Description

Turn off light

Usage

```
hue_turn_light_off(id)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with hue_get_lights_names()

Value

Silently returns http response from the bridge, but mostly used for its side effects.

---

hue_turn_light_on *Turn on light*

Description

Turn on light

Usage

```
hue_turn_light_on(id)
```

Arguments

id	If numeric, numeric id of light. If character, name of light. You can check id and names with hue_get_lights_names()

Value

Silently returns http response from the bridge, but mostly used for its side effects.

Index