Package ‘vembedr’

October 12, 2022

Title Embed Video in HTML
Version 0.1.5
Description A set of functions for generating HTML to embed hosted video in your R Markdown documents or Shiny applications.
Depends R (>= 3.3.0)
License MIT + file LICENSE
Encoding UTF-8
Imports htmltools, httr, stringr, magrittr, glue, assertthat, utils, lifecycle
RoxygenNote 7.1.2
URL https://github.com/ijlyttle/vembedr
BugReports https://github.com/ijlyttle/vembedr/issues
Suggests testthat, knitr, rmarkdown, covr, here, fs, conflicted, usethis
VignetteBuilder knitr
NeedsCompilation no
Author Ian Lyttle [aut, cre] (<https://orcid.org/0000-0001-9962-4849>), Schneider Electric [cph]
Maintainer Ian Lyttle <ian.lyttle@se.com>
Repository CRAN
Date/Publication 2021-12-11 19:30:02 UTC

R topics documented:

vembedr-package .......................................................... 2
embed ................................................................. 2
embed_url ............................................................. 4
embed_user2016 ...................................................... 5
get_service ............................................................ 6
rickroll ............................................................... 7
The vembedr package lets you embed video into your HTML pages for these services:

Details

- YouTube
- Vimeo
- Box
- Microsoft Stream

It provides two categories of functions:

- **embed** functions, to specify a video to embed: e.g. `embed_youtube()`, `embed_url()`
- **use** functions, to modify the embedding: e.g. `use_start_time()`, `use_rounded()`

You can use the pipe (|> or %>%) to chain *embed function-calls* with *use function-calls.*
Usage

```r
embed_box(
  id,
  custom_domain = getOption("vembedr.box_custom_domain"),
  width = NULL,
  height = 300,
  ratio = c("16by9", "4by3"),
  frameborder = 0,
  allowfullscreen = TRUE
)
```

```r
eMBED_MSS(long = TRUE, id, width = NULL, height = 300, ratio = c("16by9", "4by3"), query = NULL)
```

```r
eMBED_VIMEO(id, width = NULL, height = 300, ratio = c("16by9", "4by3"), frameborder = 0, allowfullscreen = TRUE, query = NULL, fragment = NULL)
```

```r
eMBED_YOUTUBE(id, width = NULL, height = 300, ratio = c("16by9", "4by3"), frameborder = 0, allowfullscreen = TRUE, query = NULL)
```

Arguments

- `id` character, identifier provided by the service
- `custom_domain` character, (used by Box) name of Box-instance to use. It can be useful to use `getOption("vembedr.box_custom_domain")` if you are using a corporate instance of Box. If NULL, it will use the standard Box instance.
- `width` numeric, width of iframe (px)
Details

These services allow you to customize a lot of things by specifying an optional query string. The specification for the query string will differ according to the service being used:

- **YouTube** https://developers.google.com/youtube/player_parameters
- **Vimeo** https://developer.vimeo.com/player/embedding
- **Box** https://developer.box.com/guides/embed/box-embed/#programmatically
- **Microsoft Stream** https://docs.microsoft.com/en-us/stream/portal-embed-video

Value

Object with S3 class vembedr_embed.

See Also

use_start_time()

Examples

```r
eMBED_YOUTUBE("dQw4w9WgXcQ")
eMBED_VIMEO("45196609")
eMBED_BOX("m5do45hvz32iv2aors3urf5pgkxxazx")
eMBED_MSSTREAM("ae21b0ac-4a2b-41f4-b3fc-f1720dd20f48")
```

Description

You can use this function to embed video using only the URL and you do not need any customization beyond the start-time. It works for all the services supported by the embed() family of functions.

Usage

```r
embed_url(url)
```
Arguments

url character, URL of web-page for video

Details

This function calls suggest_embed() then parses and evaluates the code.

Value

Object with S3 class vembedr_embed.

See Also

suggest_embed()

Examples

embed_url("https://youtu.be/1-vcErQPofQ?t=28s")

Description

[Deprecated]

These functions are deprecated: links to Microsoft Channel 9 no longer work.

Usage

embed_user2016(
    id,
    width = NULL,
    height = 300,
    ratio = c("16by9", "4by3"),
    frameborder = 0,
    allowfullscreen = TRUE
)

embed_user2017(
    id,
    width = NULL,
    height = 300,
    ratio = c("16by9", "4by3"),
    frameborder = 0,
    allowfullscreen = TRUE
)
get_service

Determine service based on URL

Usage
get_service(url)

Arguments
url character, URL of web-page for video

Value
character identifying the video service

Examples
get_service("https://youtu.be/1-vcErOPofQ?t=28s")
Description
If you want to experiment with the arguments to `embed()`, such as query, but do not have a particular video in mind, this function may be useful to you.

Usage
rickroll_vimeo(...)

rickroll_youtube(...)

Arguments
... arguments (other than id) passed on to `embed()`

Details
Please note that the YouTube video seems no longer embeddable.

Value
Object with S3 class vembedr_embed.

Examples
rickroll_vimeo()
rickroll_youtube()

Description
Use this function to specify the horizontal alignment of the iframe within the enclosing div.

Usage
use_align(embed, align = c("left", "right", "center", "justified"))

Arguments
embed vembedr_embed object, created using an `embed()` function
align character, indicates type of alignment
Value

Object with S3 class vembedr_embed.

---

use_bs_responsive  Make size responsive

Description

If your HTML page includes Twitter Bootstrap 3, you can use this function to make the size of the iframe responsive within the enclosing div.

Usage

use_bs_responsive(embed)

Arguments

embed  vembedr_embed object, created using an embed() function

Value

Object with S3 class vembedr_embed.

---

use_rounded  Make rounded corners

Description

You can use this function to make rounded corners for the enclosing <div>.

Usage

use_rounded(embed, radius = NULL)

Arguments

embed  vembedr_embed object, created using an embed() function
radius  numeric or character, css property for the border-radius for the <iframe/>. Numeric values will be interpreted as number of pixels.

Value

Object with S3 class vembedr_embed.
**Description**

This function provides you a consistent way to specify the start time, regardless of the service. Please note that Box does not provide a means to specify the start time.

**Usage**

```r
## S3 method for class 'vembedr_embed_box'
use_start_time(embed, ...)  

## S3 method for class 'vembedr_embed_channel9'
use_start_time(embed, start_time, is_paused = TRUE, ...)

## S3 method for class 'vembedr_embed_msstream'
use_start_time(embed, start_time, ...)

## S3 method for class 'vembedr_embed_vimeo'
use_start_time(embed, start_time, ...)

## S3 method for class 'vembedr_embed_youtube'
use_start_time(embed, start_time, ...)

use_start_time(embed, ...)

## Default S3 method:
use_start_time(embed, ...)
```

**Arguments**

- `embed` `vembedr_embed` object, created using an `embed()` function
- `...` generic arguments to pass through
- `start_time` numeric (seconds), or character (e.g. "3m15s")
- `is_paused` logical, for "Channel 9" specifies if the video should be paused at this time

**Details**

The `start_time` argument can take a variety of formats; these inputs all evaluate to the same value:

- "0h1m0s","0h01m00s","0h1m"
- "1m0s","1m"
- "60s",60

Please note that for Vimeo, you can specify a start time, but you can not specify that the video be paused at this time. In other words, it is like "autoplay" is set to TRUE, and you cannot unset it.
Value

Object with S3 class `vembedr_embed`.

Examples

```r
rickroll_youtube() %>%
  use_start_time("3m32s")
```

---

### Description

Knowledge of these classes is not needed for day-to-day use. Rather, it is a bookkeeping device used to make it clearer to a developer how to add a new service.

### Details

We use S3 classes to distinguish an embed object, and to denote which service it uses. Objects of these classes are created by `embed_url()` and each service’s embed function.

- **vembedr_embed**
  - base class for all services
  - HTML `<div>`
  - contains the embed code

There is an additional class attached according to the service:

- `vembedr_embed_youtube`
- `vembedr_embed_youtube_short`
- `vembedr_embed_vimeo`
- `vembedr_embed_channel9`
- `vembedr_embed_box`
- `vembedr_embed_msstream`

To support parsing, there is an internal S3 class attached to the URL being processed. It is named according to the service:

- `vembedr_url_youtube`
- `vembedr_url_youtube_short`
- `vembedr_url_vimeo`
- `vembedr_url_channel9`
- `vembedr_url_box`
- `vembedr_url_msstream`
Index

* embed
  embed, 2
  embed(), 4, 6–9
  embed_box (embed), 2
  embed_channel19 (embed_user2016), 5
  embed_msstream (embed), 2
  embed_url, 4
  embed_url(), 2, 10
  embed_user2016, 5
  embed_user2017 (embed_user2016), 5
  embed_vimeo (embed), 2
  embed_youtube (embed), 2
  embed_youtube(), 2

get_service, 6

rickroll, 7
  rickroll_channel19 (embed_user2016), 5
  rickroll_vimeo (rickroll), 7
  rickroll_youtube (rickroll), 7

suggest_embed(), 5

use_align, 7
use_bs_responsive, 8
use_rounded, 8
use_rounded(), 2
use_start_time
  (use_start_time.vembedr_embed_box), 9
use_start_time(), 2, 4
use_start_time.vembedr_embed_box, 9

vembedr-package, 2
vembedr-s3-classes, 10