Package ‘asciicast’

December 13, 2019

Title Create 'Ascii' Screen Casts from R Scripts

Version 1.0.0

Description Record 'asciicast' screen casts from R scripts. Convert them to animated SVG images, to be used in 'README' files, or blog posts. Includes 'asciinema-player' as an 'HTML' widget, and an 'asciicast' 'knitr' engine, to embed 'ascii' screen casts in 'Rmarkdown' documents.

License MIT + file LICENSE

Imports curl, jsonlite, processx (>= 3.4.0), tibble, utils, uuid, V8

Suggests cli, htmlwidgets, knitr, rmarkdown

Encoding UTF-8

LazyData true

RoxygenNote 7.0.1.9000

VignetteBuilder asciicast

NeedsCompilation no

Author Gábor Csárdi [aut, cre],
Romain Francois [aut],
Mario Nebl [aut] (https://github.com/marionebl/svg-term author),
Marcin Kulik [aut] (https://github.com/asciinema/asciinema-player author)

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Repository CRAN

Date/Publication 2019-12-13 13:00:02 UTC

R topics documented:

`asciicast-package` *asciicast parameters*

Description

You can set asciicast parameters in the header of the recorded R script. The header is in DCF format (see `read.dcf()`), but all lines are prefixed with #' comments.

Details

The DCF header may specify arbitrary parameters. We list here the parameters that are interpreted by the asciicast functions.

Recording parameters:

- `allow_errors`: Whether to cast errors properly. If this is set to `TRUE`, then asciicast overwrites the `"error"` option. Only change this if you know what you are doing.
- `cols`: Width of the terminal, in number of characters.
- `empty_wait`: How long to wait for empty lines in the script file, in seconds.
- `end_wait`: Delay at the very end, in seconds.
- `env`: Environment variables to include in the case JSON file. Defaults to `list(TERM = "xterm-256color",SHELL = "/bin/zsh")`.
- `idle_time_limit`: Time limit for the cast not printing anything, in seconds. By default there is no limit.
- `record_env`: Environment variables to set for the R subprocess.
- `rows`: Height of the terminal, in number of characters.
- `start_wait`: Delay at the beginning, in seconds.
- `timeout`: Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops. Update this for slow running code, that produces no output as it runs.
- `timestamp`: Time stamp of the recording, defaults to `Sys.time()`, this is included in the cast JSON file.
- `title`: Title of the cast, this is included in the cast JSON file.
- `typing_speed`: Average typing speed, per keypress, in seconds.

Asciinema player parameters:

- author: Author, displayed in the titlebar in fullscreen mode.
- author_img_url: URL of the author's image, displayed in the titlebar in fullscreen mode.
- author_url: URL of the author's homepage/profile. Author name (author above) is linked to this URL.
- autoplay: Whether to start playing the cast automatically.
- cols: Width of the terminal, in number of characters.
- font_size: Size of terminal font. Possible values: small, medium, big, any css font-size value (e.g. 15px).
- idle_time_limit: Time limit for the cast not printing anything, in seconds. By default there is no limit.
- loop: Whether to loop the playback.
- poster_frame: Which frame to use (in seconds) as the preview picture.
- poster_text: Text to use as the preview picture. Defaults to the title.
- rows: Height of the terminal, in number of characters.
- speed: Whether to play slower or faster. 1 is normal speed.
- start_at: Where to start the playback from, in seconds.
- theme: Theme to use, currently it has to be a string, one of "asciinema", "tango", "solarized-dark", "solarized-light", "monokai". The first one is the default.
- title: Title of the cast.

Parameters for SVG files:

- at: Timestamp of single frame to render, in seconds.
- cols: Width of the terminal, in number of characters.
- cursor: Enable cursor rendering.
- end_at: Upper range of timeline to render in seconds.
- padding: Distance between text and image bounds.
- padding_x: Distance between text and image bounds on x axis.
- padding_y: Distance between text and image bounds on y axis.
- rows: Height of the terminal, in number of characters.
- start_at: Where to start the playback from, in seconds.
- window: Render with window decorations.

See Also

Other asciicast functions: asciicast_start_process(), read_cast(), record(), write_json()

asciicast_knitr_options *Default R options to set in the background R process for knits*

Description

You can pass these options to `init_knitr_engine()`, after possibly overriding some of them.

Usage

```
asciicast_knitr_options()
```

Value

List of options.

See Also

Other asciicast in Rmd: `init_knitr_engine()`

Examples

```
asciicast_knitr_options()
```

asciicast_start_process *Start an asciicast background process*

Description

This is for expert use, if you want to run multiple recordings in the same process.

Usage

```
asciicast_start_process(
  timeout = 10,
  allow_errors = TRUE,
  startup = NULL,
  record_env = NULL,
  echo = TRUE
)
```

Arguments

- `timeout` Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops. Update this for slow running code, that produces no output as it runs.
- `allow_errors` Whether to cast errors properly. If this is set to `TRUE`, then asciicast overwrites the `"error"` option. Only change this if you know what you are doing.
- `startup` Quoted language object to run in the subprocess before starting the recording.
- `record_env` Environment variables to set for the R subprocess.
- `echo` Whether to echo the input to the terminal. If `FALSE`, then only the output is shown.

Value

The R process, a processx::process object.

See Also

Other asciicast functions: `asciicast-package`, `read_cast()`, `record()`, `write_json()`

Examples

```
# Use the same R process to record multiple casts
process <- asciicast_start_process()
script1 <- "a <- runif(10)\n"
script2 <- "a\n"
cast1 <- record(textConnection(script1), process = process)
cast2 <- record(textConnection(script2), process = process)
cast1
cast2
```

`asciinema_player` *asciinema player HTML widget*

Description

You can use this widget in Rmd files or Shiny applications, the same way as other HTML widgets.

Usage

```
asciinema_player(
  cast,
  start_at = 0,
  rows = NULL,
  cols = NULL,
  autoplay = NULL,
  loop = NULL,
  speed = NULL,
  title = NULL,
  author = NULL,
  author_url = NULL,
  author_img_url = NULL,
  poster_text = NULL,
  poster_frame = NULL,
  font_size = NULL,
  theme = NULL,
  idle_time_limit = NULL,
  html_height = NULL,
  html_width = NULL,
  element_id = NULL
)
```

Arguments

`cast` `asciicast` object.

`start_at` Where to start the playback from, in seconds.

`rows` Number of rows, defaults to the number of rows in the recording, or 24 if not specified in the cast.

`cols` Number of columns, defaults to the number columns in the recording, or 80 if not specified in the cast.

`autoplay` Whether to start playing the cast automatically.

`loop` Whether to loop the playback.

`speed` Whether to play slower or faster. 1 is normal speed.

`title` If specified, it overrides the title in the recording.

`author` Author, displayed in the titlebar in fullscreen mode.

`author_url` URL of the author's homepage/profile. Author name (author above) is linked to this URL.

`author_img_url` URL of the author's image, displayed in the titlebar in fullscreen mode.

`poster_text` if not `NULL`, used as the text of the poster (preview).

`poster_frame` Text to use as the preview picture. Defaults to the title.

`font_size` Size of terminal font. Possible values: small, medium, big, any css `font-size` value (e.g. 15px).

`theme` Theme.

`idle_time_limit`
Time limit for the cast not printing anything, in seconds. By default there is no limit.

`html_height` HTML height of the widget.

`html_width` HTML width of the widget.

`element_id` HTML id of the widget's element. If `NULL`, then the id is generated randomly.

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
asciinema_player(cast)
```

`clear_screen` *Merge multiple ASCII casts into one*

Description

The new cast will inherit its options (screen size, etc.) from the first cast in the argument list. The options of the rest of the casts are ignored.

Usage

```
clear_screen()

pause(secs)

merge_casts(...)
```

Arguments

`secs`	Number of seconds to wait.
`...`	Ascii casts to merge or merge commands. Merge commands provide a way to insert pause, clear the screen, etc., between casts.

Details

`pause()` inserts a pause of the specified seconds between the casts.

`clear_screen()` clears the screen between two casts.

Value

An `asciicast` object.

Examples

```
# merge two casts, with a pause, and clear screen between them
cast1 <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
cast2 <- read_cast(system.file("examples", "dplyr.cast", package = "asciicast"))
cast <- merge_casts(cast1, pause(3), clear_screen(), cast2)
play(cast)
```

`default_theme` *The default asciicast theme*

Description

Currently only used for `write_svg()`

Usage

```
default_theme()
```

Value

A named list.

See Also

Other SVG functions: `play()`, `write_svg()`

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
svg_file <- tempfile(fileext = ".svg")
mytheme <- modifyList(default_theme(), list(cursor = c(255, 0, 0)))
write_svg(cast, svg_file, theme = mytheme)
```

`init_knitr_engine` *Initialize the asciicast knitr engine*

Description

Call this function in your Rmd file, to enable creating asciinema casts from code chunks.

Usage

```
init_knitr_engine(
  echo = FALSE,
  same_process = TRUE,
  timeout = 10,
  allow_errors = TRUE,
  startup = NULL,
  record_env = NULL,
  echo_input = TRUE,
  options = asciicast_knitr_options()
)
```

Arguments

`echo` Whether to print the code of asciicast chunks.

`same_process` Whether to run all asciicast chunks *in the same* R process. To restart this R process, call `init_knitr_engine()` again.

`timeout` Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops. Update this for slow running code, that produces no output as it runs.

`allow_errors` Whether to cast errors properly. If this is set to `TRUE`, then asciicast overwrites the `"error"` option. Only change this if you know what you are doing.

`startup` Quoted language object to run in the subprocess before starting the recording.

`record_env` Environment variables to set for the R subprocess.

`echo_input` Whether to echo the input in the asciicast recording.

`options` R options to set (via `base::options()`, in the background R process that performs the recording. See `asciicast_knitr_options()` for the defaults.

Examples

Call this function from an Rmd chunk and then you can use the asciicast knitr engine:

~~~
```{r echo = FALSE, results = "hide"}
asciicast::init_knitr_engine()
```

```{asciicast, cache = TRUE}`
#' Rows: 10
# This is an asciicast example
loadedNamespaces()
```
~~~

See Also

Other asciicast in Rmd: `asciicast_knitr_options()`

`play` *Play asciinema cast as an SVG image in the default browser*

Description

Uses `write_svg()` to create an SVG image for a cast, in a temporary file, and then previews a minimal HTML file with the SVG image, in the default browser.

Usage

```
play(cast, ...)
```

Arguments

cast asciicast object

... Additional arguments are passed to write_svg().

Value

The path of the temporary SVG file, invisibly.

See Also

Other SVG functions: default_theme(), write_svg()

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
play(cast)
```

read_cast *Import an asciicast from an asciicast JSON file*

Description

Import an asciicast from an asciicast JSON file

Usage

```
read_cast(json)
```

Arguments

json Path to JSON asciicast file, version 2: https://github.com/asciinema/asciinema/blob/master/doc/asciicast-v2.md. If a numeric id, then it is taken as a public https://asciinema.org recording id, that is downloaded. It can also be a URL of private https://asciinema.org link.

Value

asciicast object.

See Also

Other asciicast functions: asciicast-package, asciicast_start_process(), record(), write_json()

Examples

```
c1 <- read_cast("https://asciinema.org/a/uHQwIVpiZvu0Ioio8KYx6Uwlj.cast?dl=1")
play(c1)

c2 <- read_cast(258660)
play(c2)


c3 <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
play(c3)
```

`record` *Record an asciinema screencast*

Description

Record an asciinema screencast

Usage

```
record(
  script,
  typing_speed = NULL,
  empty_wait = NULL,
  cols = NULL,
  rows = NULL,
  title = NULL,
  timestamp = NULL,
  env = NULL,
  idle_time_limit = NULL,
  allow_errors = TRUE,
  timeout = NULL,
  start_wait = NULL,
  end_wait = NULL,
  record_env = NULL,
  startup = NULL,
  echo = TRUE,
  process = NULL
)
```

Arguments

`script` Path of an R script to record. It can also be a readable R connection or URL, as it is passed to `base::readLines()`.

`typing_speed` Average typing speed, per keypress, in seconds.

empty_wait How long to wait for empty lines in the script file, in seconds.

cols Width of the terminal, in number of characters.

rows Height of the terminal, in number of characters. If it the string "auto", then it will be determined automatically, by including all output on the screen.

title Title of the cast, this is included in the cast JSON file.

timestamp Time stamp of the recording, defaults to Sys.time(), this is included in the cast JSON file.

env Environment variables to include in the case JSON file. Defaults to list(TERM = "xterm-256color",SHELL = "/bin/zsh").

idle_time_limit
Time limit for the cast not printing anything, in seconds. By default there is no limit.

allow_errors Whether to cast errors properly. If this is set to TRUE, then asciicast overwrites the "error" option. Only change this if you know what you are doing.

timeout Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops. Update this for slow running code, that produces no output as it runs.

start_wait Delay at the beginning, in seconds.

end_wait Delay at the very end, in seconds.

record_env Environment variables to set for the R subprocess.

startup Quoted language object to run in the subprocess before starting the recording.

echo Whether to echo the input to the terminal. If FALSE, then only the output is shown.

process A processx subprocess to run the cast in. By default a new subprocess is started. You can reuse a process by calling asciicast_start_process() first, and supplying the returned process here.

Value

An asciicast object, write this to file with write_json().

See Also

Other asciicast functions: asciicast-package, asciicast_start_process(), read_cast(), write_json()

Examples

```
script <- system.file("examples", "hello.R", package = "asciicast")
cast <- record(script)
play(cast)
```

write_json *Write an ascii cast to file*

Description

The file uses the asciinema file format, version 2: https://github.com/asciinema/asciinema/blob/master/doc/asciicast-v2.md.

Usage

```
write_json(cast, path)
```

Arguments

cast	asciicast object.
path	Path to write to.

See Also

Other asciicast functions: asciicast-package, asciicast_start_process(), read_cast(), record()

Examples

```
script <- system.file("examples", "hello.R", package = "asciicast")
cast <- record(script)
json <- tempfile(fileext = ".json")
write_json(cast, json)
```

write_svg *Create animated SVG from an asciicast*

Description

Create animated SVG from an asciicast

Usage

```
write_svg(
  cast,
  path,
  window = NULL,
  start_at = NULL,
  end_at = NULL,
  at = NULL,
  cursor = NULL,
  rows = NULL,
  cols = NULL,
  padding = NULL,
  padding_x = NULL,
  padding_y = NULL,
  omit_last_line = NULL,
  theme = NULL
)
```

Arguments

cast asciicast object.

path Path to SVG file to create.

window Render with window decorations.

start_at Lower range of timeline to render in seconds.

end_at Upper range of timeline to render in seconds.

at Timestamp of single frame to render, in seconds. Alternatively it can be "end", to take a snapshot at the end of the cast, after all output is done.

cursor Enable cursor rendering.

rows Height in lines.

cols Width in columns.

padding Distance between text and image bounds.

padding_x Distance between text and image bounds on x axis.

padding_y Distance between text and image bounds on y axis.

omit_last_line Whether to omit the last line of the cast. This often just the prompt, and sometimes it is not worth showing.

theme A named list to override the default theme (see default_theme()).

See Also

Other SVG functions: default_theme(), play()

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
svg_file <- tempfile(fileext = ".svg")
write_svg(cast, svg_file)
```

Index