

Package: ovscout (via r-universe)

August 26, 2024

Title An R Shiny App for Editing, Syncing, and Scouting Volleyball Match Files

Version 0.5.0

Description Functions for video synchronisation, scout file editing, and more from volleyball match files.

URL <https://github.com/openvolley/ovscout>

BugReports <https://github.com/openvolley/ovscout/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends datavalley (> 0.12.999)

Imports assertthat, base64enc, data.table, dplyr, digest, DT, fs, ggplot2, htmltools, httr, lubridate, ovideo (>= 0.0.5), rintrojs, servr (>= 0.15.1), shiny, shinyTime, stringr, sys, tidyr, tidyselect, uuid

Suggests covr, tcltk, testthat

Remotes openvolley/datavalley, openvolley/ovideo

RoxygenNote 7.2.1

Repository <https://openvolley.r-universe.dev>

RemoteUrl <https://github.com/openvolley/ovscout>

RemoteRef HEAD

RemoteSha 2aa95a8d39b6aa65a1560c0ef2a44ea134960ac2

Contents

dv_create	2
dv_scout_from_console	3
dv_set_lineups	4

ovscout	5
ov_default_attack_table	5
ov_default_compound_table	5
ov_default_scouting_table	6
ov_default_setter_calls_table	6
ov_default_winning_symbols	6
ov_shiny_video_sync	7
rotations	7
vwModalDialog	8

Index 10

dv_create	<i>Create a new datavalley match object</i>
-----------	---

Description

Create a new datavalley match object

Usage

```
dv_create(
  match,
  more,
  teams,
  players_h,
  players_v,
  video_file,
  attacks = ov_simplified_attack_table(),
  setter_calls = ov_default_setter_calls_table(),
  winning_symbols = ov_default_winning_symbols(),
  zones_or_cones = "Z",
  regulation = "indoor rally point",
  comments
)
```

Arguments

match	list or single-row data.frame: (optional) with components date (defaults to current date), time (defaults to current time), season, league, phase, home_away, day_number, match_number, regulation, zones_or_cones. zones_or_cones can also be provided directly
more	list or single-row data.frame: (optional) with components referees, spectators, receipts, city, arena, scout
teams	data.frame: a 2-row data frame, with required columns team_id, team and optional columns coach, assistant, shirt_colour

players_h, players_v	data.frame: with required columns number, firstname, lastname, and optional columns player_id, role (character vector with "outside", "opposite", "middle", "libero", "setter"), nickname, special_role (character vector with "L", "C", or NA), foreign (logical, defaults to FALSE)
video_file	string: (optional) path to video file
attacks	data.frame: as returned by <code>ov_simplified_attack_table()</code> or <code>ov_default_attack_table()</code>
setter_calls	data.frame: as returned by <code>ov_default_setter_calls_table()</code>
winning_symbols	data.frame: as returned by <code>ov_default_winning_symbols()</code>
zones_or_cones	string: "Z" or "C". Will be ignored if zones_or_cones is provided in the match parameter
regulation	string: "indoor rally point", "beach rally point", or "indoor sideout". Will be ignored if regulation is provided in the match parameter
comments	character: optional vector of length up to 5, of comments

Value

A datavolley object

Examples

```
x <- dv_create(teams = data.frame(team_id = c("TM1", "TM2"), team = c("Team 1", "Team 2")),
  comments = "Test file",
  players_h = data.frame(firstname = toupper(letters[1:7]), lastname = "Player",
    number = 1:7),
  players_v = data.frame(firstname = letters[10:15], lastname = "VisPlayer",
    number = 10:15))

## enter the team lineups for set 1
x <- dv_set_lineups(x, set_number = 1, lineups = list(6:1, 15:10), setter_positions = c(2, 1))
```

dv_scout_from_console *Enter scout codes from the console*

Description

Probably only useful for testing.

Usage

```
dv_scout_from_console(
  x,
  prompt = "SCOUT> ",
  compound_table = ov_default_compound_table(),
  default_scouting_table = ov_default_scouting_table()
)
```

Arguments

x	datavolley: a datavolley object as returned by <code>dv_create()</code>
prompt	string: the prompt to show
compound_table	tibble: the table of default compound codes
default_scouting_table	tibble: the table of scouting defaults (skill type and evaluation)

Value

A modified version of x, with rows added to the plays2 component

dv_set_lineups	<i>Enter the team lineups at the start of a set</i>
----------------	---

Description

Enter the team lineups at the start of a set

Usage

```
dv_set_lineups(x, set_number, lineups, setter_positions, setters)
```

Arguments

x	datavolley: a datavolley object
set_number	integer: set number, 1–3 for beach or 1–5 for indoor
lineups	list: two-element list with numeric vectors of player numbers. Each lineup is <ul style="list-style-type: none"> • for indoor, of length 6, 7, or 8 (first 6 are player jersey numbers in positions 1–6, elements 7 and 8 are optionally the libero jersey numbers) • for beach, of length 2
setter_positions	integer: two-element integer vector giving the position on court of the two setters. At least one of <code>setter_positions</code> or <code>setters</code> must be provided for indoor. Ignored for beach
setters	integer: two-element integer vector giving the jersey numbers of the two setters. At least one of <code>setter_positions</code> or <code>setters</code> must be provided for indoor. Ignored for beach

Value

A modified version of x

ovscout	ovscout
---------	----------------

Description

Functions for video synchronisation, scout file editing, and more from volleyball match files.

ov_default_attack_table

Default attack combination code table

Description

Default attack combination code table

Usage

ov_default_attack_table()

ov_simplified_attack_table()

Value

A tibble

ov_default_compound_table

Default compound skills table

Description

Default compound skills table

Usage

ov_default_compound_table()

Value

A tibble

ov_default_scouting_table

Default scouting (type and evaluation for each skill) table

Description

Default scouting (type and evaluation for each skill) table

Usage

ov_default_scouting_table()

Value

A tibble

ov_default_setter_calls_table

Default setter calls table

Description

Default setter calls table

Usage

ov_default_setter_calls_table()

Value

A tibble

ov_default_winning_symbols

Default winning symbols table

Description

Default winning symbols table

Usage

ov_default_winning_symbols()

Value

A tibble

ov_shiny_video_sync *Launch a Shiny app for video synchronisation and scout editing*

Description

Launch a Shiny app for video synchronisation and scout editing

Usage

```
ov_shiny_video_sync(
  dwv,
  video_file = NULL,
  launch_browser = TRUE,
  prompt_for_files = interactive(),
  ...
)
```

Arguments

dwv	string or datavalley: either the path to a dwv file (which will be read by read_dv) or a datavalley object as returned by that function. Passing the file name (not the datavalley object) is required if any extra arguments are passed via ...
video_file	string: optionally, the path to the video file. If not supplied (or NULL) the video file specified in the dwv file will be used. Provide video_file = NA to run the app without a video file
launch_browser	logical: if TRUE, launch the app in the system's default web browser (passed to runApp 's launch.browser parameter)
prompt_for_files	logical: if dwv was not specified, prompt the user to select the dwv file
...	: extra parameters passed to read_dv (if dwv is a provided as a string) and/or to the shiny server and UI functions

See Also

[read_dv](#)

rotations *Get team rotations point by point*

Description

Get team rotations point by point

Usage

```
rotations(x, team, start_point_id, set_number, new_rotation, new_rotation_id)
```

Arguments

`x` datavolley or data.frame: a datavolley object as returned by `read_dv`, or the plays component of that object

`team` string: team name

`start_point_id` vector: vector of pointids for which to return the rotation

`set_number` value: Set number

`new_rotation` vector: vector of player number, positioned from 1 to 6

`new_rotation_id` character: as for `new_rotation`, but providing player IDs instead of jersey numbers

Value

list of 2 data.frames. Current data.frame, without changes, and updated data.frame, with new player rotation.

See Also

[read_dv](#)

Examples

```
## Not run:
x <- read_dv(dv_example_file(), insert_technical_timeouts=FALSE)
new_x <- rotations(x, start_point_id = 25, new_rotation = c(9,6,15,4,12,7))

## End(Not run)
```

vwModalDialog

Variable width modal dialog

Description

Variable width modal dialog

Usage

```
vwModalDialog(..., width = 90)
```

Arguments

`...` : as for [shiny::modalDialog\(\)](#)

`width` numeric: percentage of viewport width

Value

As for `shiny::modalDialog()`

Examples

```
## Not run:  
  showModal(vwModalDialog(title = "Wide dialog", "blah", width = 90))  
  
## End(Not run)
```

Index

`dv_create`, 2
`dv_create()`, 4
`dv_scout_from_console`, 3
`dv_set_lineups`, 4

`ov_default_attack_table`, 5
`ov_default_attack_table()`, 3
`ov_default_compound_table`, 5
`ov_default_scouting_table`, 6
`ov_default_setter_calls_table`, 6
`ov_default_setter_calls_table()`, 3
`ov_default_winning_symbols`, 6
`ov_default_winning_symbols()`, 3
`ov_shiny_video_sync`, 7
`ov_simplified_attack_table`
 (`ov_default_attack_table`), 5
`ov_simplified_attack_table()`, 3
`ovscout`, 5

`read_dv`, 7, 8
`rotations`, 7
`runApp`, 7

`shiny::modalDialog()`, 8, 9
`vwModalDialog`, 8