

Package: ovscout2 (via r-universe)

October 13, 2026

Title An R Shiny App for Scouting Volleyball Match Files

Version 1.2.1

Description Functions for scouting volleyball match files.

URL <https://github.com/openvolley/ovscout2>

BugReports <https://github.com/openvolley/ovscout2/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends datavolley (> 0.12.999)

Imports assertthat, curl, data.table, dplyr, digest, DT, fs, ggplot2, jpeg, htmltools, httpuv, htr, lubridate, methods, ovddata, ovideo (>= 0.17.2), perनावolley, qrcode, R6, R.utils, rappdirs, rintrojs, rlang, rmarkdown, servr (>= 0.15.1), shiny, shinythemes, shinyWidgets, stringr, sys, tidyr, uuid, volleyreport (>= 0.8.3)

Suggests av, covr, DBI, dbplyr, duckdb, fontawesome, knitr, rstudioapi, tcltk, testthat

Remotes openvolley/datavolley, openvolley/ovdata, openvolley/ovideo, openvolley/perनावolley, openvolley/volleyreport

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/pak/sysreqs

chromium cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libmagick++-dev gsfonts libavfilter-dev libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libpoppler-cpp-dev poppler-data libnode-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-09-12 08:57:19 UTC

RemoteUrl <https://github.com/openvolley/ovscout2>

RemoteRef HEAD

RemoteSha 8551c888a28ebc9b97de97e585ed59bd0bde4ccd

Contents

dv_create	2
dv_set_lineups	3
ov_app_styling	4
ov_default_attack_table	5
ov_default_click_shortcuts	5
ov_default_compound_table	6
ov_default_key_remapping	7
ov_default_scouting_table	7
ov_default_setter_calls_table	8
ov_default_winning_symbols	8
ov_install_lighttpd	9
ov_scouter	9
ov_scouting_options	12
ovscout2	14
Index	15

dv_create	Create a new datavolley match object
-----------	--------------------------------------

Description

Create a new datavolley 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 describing the home and visiting teams, with required columns team_id, team and optional columns coach, assistant, shirt_colour. The home team must be in the first row of this data frame
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_set_lineups

Enter the team lineups at the start of a set

Description

Enter the team lineups at the start of a set

Usage

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

Arguments

<code>x</code>	datavolley: a datavolley object
<code>set_number</code>	integer: set number, 1–3 for beach or 1–5 for indoor
<code>lineups</code>	list: two-element list with numeric vectors of player numbers. Each lineup is • 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
<code>setter_positions</code>	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
<code>setters</code>	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`

ov_app_styling	<i>Styling to apply to the app</i>
----------------	------------------------------------

Description

Colours can be either hex strings or names, but if using names they must be recognized both by R and as CSS colour names.

Usage

```
ov_app_styling()
```

Details

- `review_pane_width` is expressed as a percentage of the browser window width

Value

A named list of styling parameters

ov_default_attack_table
<i>Default attack combination code table</i>

Description

Default attack combination code table

Usage

ov_default_attack_table()

ov_simplified_attack_table()

Value

A tibble

ov_default_click_shortcuts
<i>Keyboard shortcuts for ov_scouter</i>

Description

ov_default_click_shortcuts apply when using scout_mode = "click", and ov_default_type_shortcuts apply when using scout_mode = "type"

Usage

ov_default_click_shortcuts()

ov_shortcuts(sc_type = "click", ...)

ov_default_type_shortcuts()

ov_default_playstable_shortcuts()

Arguments

sc_type	string: the shortcuts type, one of "click", "type", or "playstable"
...	: named values that are added to the default set of shortcuts. The name should match one of the existing entries, and its value should be the new shortcut key to use. If that shortcut key is already used elsewhere in the defaults, it will be removed from its default assignment

Details

Shortcuts should be defined in terms of the printable representation of the key (e.g. "a", "\$", "H", "Escape", "Enter"). See https://developer.mozilla.org/en-US/docs/Web/API/UI_Events/Keyboard_event_key_values for guidance. Shortcuts can optionally use modifier keys: "Ctrl-x" means pressing the control key and x simultaneously; similarly Alt-x, Meta-x, Shift-x. Be aware that some keys are hard-coded for specific functionality and might cause problems if you use them as shortcuts (e.g. "Enter", "Tab"), and some keys have browser-level or operating-system-level handling that cannot be overridden.

Value

A named list

the default shortcuts for click-scouting:

```
sc <- ov_default_click_shortcuts()
```

or equivalently:

```
sc <- ov_shortcuts("click")
```

use 'z' as a pause shortcut, along with the existing defaults:

```
sc <- ov_shortcuts("click", pause = "z")
```

ov_default_compound_table

Default compound skills table

Description

Default compound skills table

Usage

```
ov_default_compound_table()
```

Value

A tibble

`ov_default_key_remapping`*Default keyboard remapping for ov_scouter*

Description

These key remappings only apply to input into the scouting bar (when the scouter has been started with `scout_mode = "type"`).

Usage

```
ov_default_key_remapping(scout_mode = "click")
```

Arguments

<code>scout_mode</code>	string: either "click" for the guided point-and-click scouting interface, or "type" for the typing-based interface. Currently remapping has no effect with "click"
-------------------------	--

Value

A named list

`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 (where the setter has instructed the middle hitter to run)

Description

Default setter calls table (where the setter has instructed the middle hitter to run)

Usage

```
ov_default_setter_calls_table(data_type = "indoor")
```

Arguments

data_type string: "indoor", "beach". The returned table will be empty for "beach"

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_install_lighttpd	<i>Install lighttpd</i>
---------------------	-------------------------

Description

This is a helper function to install lighttpd. Currently it only works on Windows platforms. The lighttpd bundle will be downloaded from <http://lighttpd.dtech.hu/> and saved to your user appdata directory.

Usage

```
ov_install_lighttpd(force = FALSE)
```

Arguments

force	logical: force reinstallation if lighttpd already exists
-------	--

Value

the path to the installed executable

References

<http://lighttpd.dtech.hu/>

Examples

```
## Not run:  
  ov_install_lighttpd()  
  
## End(Not run)
```

ov_scouter	<i>Launch a Shiny app for scouting</i>
------------	--

Description

Launch a Shiny app for scouting

Usage

```
ov_scouter(
  dvw,
  video_file,
  court_ref,
  season_dir,
  auto_save_dir,
  scout_mode = "click",
  pause_on_type = 500,
  scoreboard = TRUE,
  ball_path = FALSE,
  playlist_display_option = "dv_codes",
  review_pane = TRUE,
  playback_rate = 1,
  scouting_options = ov_scouting_options(),
  app_styling = ov_app_styling(),
  click_shortcuts = ov_shortcuts("click"),
  type_shortcuts = ov_shortcuts("type"),
  playstable_shortcuts = ov_shortcuts("playstable"),
  key_remapping,
  scout_name = "",
  show_courtref = FALSE,
  dash = FALSE,
  host,
  launch_browser = TRUE,
  prompt_for_files = interactive(),
  ...
)
```

Arguments

<code>dvw</code>	string or datavolley: either the path to a dvw or ovs file or a datavolley object (e.g. as returned by dv_create()). Passing the file name (not the datavolley object) is required if any extra arguments are passed via <code>...</code> . <code>dvw</code> can also be an object as saved by <code>ov_scouter()</code> in ovs format. If <code>dvw</code> is "demo", the app will be started with a demonstration data set
<code>video_file</code>	string or FALSE: optionally, the path to the video file. If FALSE and <code>scout_mode</code> is "type", no video will be used (use this for e.g. live scouting). If <code>video_file</code> is not supplied or is NULL, the video file specified in the dvw file will be used. <code>video_file</code> can also be a URL (including a YouTube URL or video ID)
<code>court_ref</code>	data.frame or string: data.frame with the court reference (as returned by ovideo::ov_shiny_court_ref()) or the path to the rds file containing the output from this
<code>season_dir</code>	string: optional path to a directory with other dvw/ovs files from this season
<code>auto_save_dir</code>	string: optional path to a directory where the dvw will be saved automatically after each rally
<code>scout_mode</code>	string: either "click" for the guided point-and-click scouting interface, or "type" for the typing-based interface

pause_on_type	numeric: if greater than 0 and using scout_mode = "type", pause the video for this many milliseconds after each key press in the scouting bar
scoreboard	logical: if TRUE, show a scoreboard in the top-right of the video pane
ball_path	logical: if TRUE, show the ball path on the court inset diagram. Note that this will slow the app down slightly
playlist_display_option	string: what to show in the plays table? Either "dv_codes" (scouted codes) or "commentary" (a plain-language interpretation of the touches)
review_pane	logical: if TRUE, entry popups will be accompanied by a small video pane that shows a loop of the video of the action in question
playback_rate	numeric: starting playback rate of the video (1.0 is normal speed, higher is faster)
scouting_options	list: a named list with entries as per <code>ov_scouting_options()</code> . See Details, below
app_styling	list: named list of styling options, as returned by <code>ov_app_styling()</code>
click_shortcuts	list: named list of keyboard shortcuts, as returned by <code>ov_default_click_shortcuts()</code>
type_shortcuts	list: named list of keyboard shortcuts, as returned by <code>ov_default_type_shortcuts()</code>
playstable_shortcuts	list: named list of keyboard shortcuts that apply when in the plays table, as returned by <code>ov_default_playstable_shortcuts()</code>
key_remapping	list: a named list of key remappings, with entries as per <code>ov_default_key_remapping()</code> . If not provided, it will default to <code>ov_default_key_remapping(scout_mode)</code> (for whatever scout_mode is being used)
scout_name	string: the name of the scout (your name)
show_courtref	logical: if TRUE, show the court reference lines overlaid on the video
dash	logical: support live MPEG DASH streams? If not specified, will default to TRUE if video_file is a *.mpd stream. Note that DASH support is fragile at best. HLS streams are automatically supported and likely to be more reliable
host	string: the IP address of this machine. Only required if you intend to run the app on this machine but connect to it from a different machine (in which case use <code>ov_scouter(..., host = "www.xxx.yyy.zzz", launch_browser = FALSE)</code> on the other machine, where www.xxx.yyy.zzz is the IP address of this machine, i.e. the machine running the app)
launch_browser	logical: if TRUE, launch the app in the system's default web browser (passed to <code>shiny::runApp()</code> 's <code>launch.browser</code> parameter)
prompt_for_files	logical: if dvw was not specified, prompt the user to select the dvw file
...	: extra parameters passed to <code>datavolley::dv_read()</code> (if dvw is a provided as a string) and/or to the shiny server and UI functions

Details

A note on scouting_options

If a *.ovs file (i.e. a partially-scouted file, that was previously scouted using this app) has been provided in the dvw argument, then it will contain the scouting options used during the previous scouting session. Those options will be re-used EXCEPT if scouting_options are also provided here. Any scouting options provided here via the scouting_options argument will override options saved in the .ovs file. Thus, it is recommended that scouting_options not be provided here along with a .ovs file unless absolutely necessary. If necessary, only the specific, relevant elements of the scouting_options list should be provided. Note that *.dvw files do not contain saved options, only .ovs files that were scouted with this app.

Examples

```
## Not run:
  ov_scouter("demo")

## End(Not run)
```

ov_scouting_options	<i>Scouting options</i>
---------------------	-------------------------

Description

Scouting options

Usage

```
ov_scouting_options(
  end_convention = "actual",
  nbblockers = TRUE,
  default_nbblockers = NA,
  transition_sets = FALSE,
  set_quality = FALSE,
  attacks_by = "codes",
  zones_cones = "Z",
  team_system = "SHM3",
  setter_dump_code = "PP",
  second_ball_attack_code = "P2",
  overpass_attack_code = "PR",
  default_scouting_table = ov_default_scouting_table(),
  compound_table = ov_default_compound_table(),
  attack_table = ov_simplified_attack_table(),
  setter_calls = "none",
  setter_calls_table = ov_default_setter_calls_table()
)
```

Arguments

end_convention	string: either "actual" or "intended". Is the end coordinate of an attack or serve the actual end location (where the ball contacted the floor or out of bounds area), or the intended one. The actual might differ from the intended if there is a block touch or the ball hit the net. If "actual", and a block touch is recorded, then the end location of the attack will not be used for the dig location (the dig location will be missing)
nblockers	logical: scout the number of blockers on each attack?
default_nblockers	integer: if nblockers is TRUE, what number of blockers should we default to? If NA, no default. Having a default value might speed up the data entry process very slightly but it is not recommended because it's easy to forget to update it on each attack and then your data will include default-valued entries that you can't easily distinguish from genuine ones that you actually entered
transition_sets	logical: scout sets in transition? If FALSE, just the endpoint of each attack (i.e. the dig) and the subsequent counter-attack are scouted
set_quality	logical: are we assessing set quality? If FALSE sets will be classified as "error", "overpass set", or "positive" for all other sets
attacks_by	string: "codes" (X5, V5, etc) or "tempo" (high, medium, quick)
zones_cones	string: record attack directions as "Z"ones or "C"ones (ignored when scouting in 'click' mode, which always uses zones. But after click-scouting you can export your dvw with attack directions as cones, if you wish)
team_system	string: the assumed system that teams are using to assign e.g. passing and hitting responsibilities • "SHM3" - a setter-hitter-middle rotation, with 3 passers (the libero and two outside hitters)
setter_dump_code	string: the attack combination code for a setter dump
second_ball_attack_code	string: the attack combination code for a second-ball attack
overpass_attack_code	string: the attack combination code for an attack on an overpass
default_scouting_table	tibble: the table of scouting defaults (skill type and evaluation)
compound_table	tibble: the table of compound codes
attack_table	tibble: table of attack codes (X5, V5, etc) as returned by ov_default_attack_table() or ov_simplified_attack_table()
setter_calls	string: either "none", "reception" (setter calls in reception phase only), or "both" (setter calls in reception and transition)
setter_calls_table	tibble: table of setter calls (where the setter instructs the middle to run) as returned by ov_default_setter_calls_table()

Value

A named list

ovscout2	ovscout2
----------	-----------------

Description

Functions for scouting volleyball match files.

Author(s)

Maintainer: Ben Raymond <ben@untan.gl>

Authors:

- Adrien Ickowicz

Other contributors:

- Krystian Kowalski [contributor]
- Hugh Nguyen [contributor]
- Brady Williams [contributor]
- openvolley.org [originator]

See Also

Useful links:

- <https://github.com/openvolley/ovscout2>
- Report bugs at <https://github.com/openvolley/ovscout2/issues>

Index

`datavolley::dv_read()`, [11](#)
`dv_create`, [2](#)
`dv_create()`, [10](#)
`dv_set_lineups`, [3](#)

`ov_app_styling`, [4](#)
`ov_app_styling()`, [11](#)
`ov_default_attack_table`, [5](#)
`ov_default_attack_table()`, [3](#), [13](#)
`ov_default_click_shortcuts`, [5](#)
`ov_default_click_shortcuts()`, [11](#)
`ov_default_compound_table`, [6](#)
`ov_default_key_remapping`, [7](#)
`ov_default_key_remapping()`, [11](#)
`ov_default_playstable_shortcuts`
 (`ov_default_click_shortcuts`), [5](#)
`ov_default_playstable_shortcuts()`, [11](#)
`ov_default_scouting_table`, [7](#)
`ov_default_setter_calls_table`, [8](#)
`ov_default_setter_calls_table()`, [3](#), [13](#)
`ov_default_type_shortcuts`
 (`ov_default_click_shortcuts`), [5](#)
`ov_default_type_shortcuts()`, [11](#)
`ov_default_winning_symbols`, [8](#)
`ov_default_winning_symbols()`, [3](#)
`ov_install_lighttpd`, [9](#)
`ov_scouter`, [9](#)
`ov_scouting_options`, [12](#)
`ov_scouting_options()`, [11](#)
`ov_shortcuts`
 (`ov_default_click_shortcuts`), [5](#)
`ov_simplified_attack_table`
 (`ov_default_attack_table`), [5](#)
`ov_simplified_attack_table()`, [3](#), [13](#)
`ovideo::ov_shiny_court_ref()`, [10](#)
`ovscout2`, [14](#)
`ovscout2-package (ovscout2)`, [14](#)

`shiny::runApp()`, [11](#)