

Package ‘psychTestR’

September 11, 2018

Type Package

Title Framework for delivering psychological tests via Shiny

Version 2.0.0

Author Peter M. C. Harrison

Maintainer Peter M. C. Harrison <pmc.harrison@gmail.com>

Description Framework for delivering psychological tests via Shiny.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports assertthat (>= 0.2.0),

shiny (>= 1.1.0),

catR (>= 3.16),

htmltools (>= 0.3.6),

magrittr (>= 1.5),

methods,

parallel,

plyr (>= 1.8.4),

RPushbullet (>= 0.3.1),

R.utils (>= 2.6.0),

shinyjs (>= 1.0),

shinyBS (>= 0.61),

shinythemes (>= 1.1.1),

tidyr (>= 0.8.1),

utils,

R6 (>= 2.2.2),

markdown (>= 0.8),

rstudioapi (>= 0.7)

RoxygenNote 6.1.0

Suggests testthat (>= 2.0.0)

Byte-Compile yes

R topics documented:

| | |
|---------------------------------|---|
| answer | 3 |
| answer<- | 3 |
| assert_global_is_null | 4 |

| | |
|--|----|
| audio_NAFC_page | 4 |
| basic_demographics_default_intro | 5 |
| begin_module | 5 |
| check_dirs | 5 |
| close_test | 6 |
| code_block | 6 |
| demo | 6 |
| demo.get_url_params | 7 |
| demo.set_url_params | 7 |
| demo_options | 7 |
| display_error | 8 |
| dropdown_page | 8 |
| elt_save_results_to_disk | 9 |
| end_module | 9 |
| error | 9 |
| error<- | 10 |
| final_page | 10 |
| finish_test_and_give_code | 10 |
| get_basic_demographics | 11 |
| get_global | 11 |
| get_local | 12 |
| get_p_id | 12 |
| get_results | 13 |
| get_session_info | 13 |
| get_supplementary_results_dir | 14 |
| get_url_params | 14 |
| i18n | 15 |
| i18n_dict | 15 |
| is.test_element | 16 |
| is.timeline | 16 |
| is_test_closed | 17 |
| loop_while | 17 |
| make_test | 18 |
| make_ui_NAFC | 18 |
| NAFC_page | 19 |
| new_results_section | 20 |
| new_timeline | 20 |
| one_button_page | 21 |
| open_test | 21 |
| page | 22 |
| pt_options | 22 |
| p_id | 23 |
| p_id<- | 23 |
| reactive_page | 24 |
| register_next_results_section | 24 |
| results | 25 |
| results<- | 25 |
| save_result | 25 |
| save_results_to_disk | 26 |
| set_global | 26 |
| set_local | 27 |
| set_url_param | 27 |

answer

3

skip_n_pages

27

test_options

28

test_permissions

29

text_input_page

30

timeline

30

trigger_button

31

video_NAFC_page

31

volume_calibration_page

32

Index

33

| | |
|--------|---------------|
| answer | <i>Answer</i> |
|--------|---------------|

Description

Accesses the answer slot of the participant’s state object. The answer slot contains the participant’s last response.

Usage

answer(state)

Arguments

state Participant’s state object.

| | |
|----------|---------------|
| answer<- | <i>Answer</i> |
|----------|---------------|

Description

Updates the answer slot of the participant’s state object. The answer slot contains the participant’s last response.

Usage

answer(state) <- value

Arguments

state Participant’s state object.
value New value for answer.

`assert_global_is_null` *Assert that a global variable is NULL*

Description

Throws an error if a certain global variable is not NULL.

Usage

```
assert_global_is_null(key, state)
```

Arguments

| | |
|--------------------|---|
| <code>key</code> | Key of the global variable that should be NULL. |
| <code>state</code> | The participant's state object. |

`audio_NAFC_page` *Make NAFC audio page*

Description

Creates an n-alternative forced-choice page with an audio prompt.

Usage

```
audio_NAFC_page(label, prompt, choices, url, labels = NULL,
  type = tools::file_ext(url), save_answer = TRUE,
  on_complete = NULL, arrange_choices_vertically = length(choices) >
  2L, wait = TRUE, loop = FALSE, admin_ui = NULL)
```

Arguments

| | |
|--------------------------|--|
| <code>label</code> | Label for the current page (character scalar). |
| <code>prompt</code> | Prompt to be displayed above the response choices. Can be either a scalar character (e.g. "What is 2 + 2?") or an object of class "shiny.tag", e.g. <code>shiny::tags\$p("What is 2 + 2?")</code> . |
| <code>choices</code> | Character vector of choices for the participant. If unnamed, then these values will be used both for button IDs and for button labels. If named, then values will be used for button IDs and names will be used for button labels. |
| <code>url</code> | URL to the audio. Can be an absolute URL (e.g. "http://mysite.com/audio.mp3") or a URL relative to the /www directory (e.g. "audio.mp3"). |
| <code>labels</code> | Optional vector of labels for the NAFC choices. If not NULL, will overwrite the names of choices. This vector of labels can either be a character vector or a list of Shiny tag objects, e.g. as created by <code>shiny::HTML()</code> . |
| <code>type</code> | Audio type (e.g. 'mp3'). Defaults to the provided file extension. |
| <code>save_answer</code> | See page . |
| <code>on_complete</code> | See page . |

| | |
|----------------------------|--|
| arrange_choices_vertically | Whether to arrange the response buttons vertically (the default) as opposed to horizontally. |
| wait | Whether to wait for the audio to finish before displaying the response buttons. |
| loop | Whether the audio should loop. |
| admin_ui | See page . |

basic_demographics_default_intro

Default demographics introduction

Description

Default introduction to the demographics section.

Usage

```
basic_demographics_default_intro()
```

begin_module

Begin module

Description

Returns a code block that begins a psychTestR module. Modules have their own set of local variables. They also have identifying labels that are stored alongside result entries created during the module.

Usage

```
begin_module(label)
```

Arguments

| | |
|-------|----------------------------------|
| label | Module label (scalar character). |
|-------|----------------------------------|

check_dirs

Check directories

Description

Checks that all directories as specified in the test options are writeable.

Usage

```
check_dirs(opt)
```

Arguments

| | |
|-----|--|
| opt | Options list as created by <code>test_options()</code> . |
|-----|--|

| | |
|------------|-------------------|
| close_test | <i>Close test</i> |
|------------|-------------------|

Description

Closes the test.

Usage

```
close_test(opt)
```

Arguments

| | |
|-----|---|
| opt | Test options list as created by test_options(). |
|-----|---|

| | |
|------------|-------------------|
| code_block | <i>Code block</i> |
|------------|-------------------|

Description

Creates a code block.

Usage

```
code_block(fun)
```

Arguments

| | |
|-----|--|
| fun | Function to execute within code block. This function's argument list must include It may also include any of the following named arguments: state, the participant's state object; opt, which is the test's option list as created by test_options(); input, the current page's Shiny input object; output, the current page's Shiny output object; session, the current Shiny session object; elts, the timeline (i.e. list of test elements). |
|-----|--|

| | |
|------|-------------|
| demo | <i>Demo</i> |
|------|-------------|

Description

Determines whether the current session is a demo session.

Usage

```
demo(state)
```

Arguments

| | |
|-------|---------------------------------|
| state | The participant's state object. |
|-------|---------------------------------|

Value

TRUE if the current session is a demo session, FALSE otherwise.

| | |
|---------------------|-------------------------------------|
| demo.get_url_params | <i>Demo: Getting URL parameters</i> |
|---------------------|-------------------------------------|

Description

Demonstrates how psychTestR can get URL parameters.

Usage

```
demo.get_url_params()
```

| | |
|---------------------|-------------------------------------|
| demo.set_url_params | <i>Demo: Getting URL parameters</i> |
|---------------------|-------------------------------------|

Description

Demonstrates how psychTestR can set URL parameters.

Usage

```
demo.set_url_params()
```

| | |
|--------------|---------------------|
| demo_options | <i>Demo options</i> |
|--------------|---------------------|

Description

Test options list for demo purposes.

Usage

```
demo_options(...)
```

Arguments

... Arguments to be passed to [test_options\(\)](#).

| | |
|---------------|--------------------------------------|
| display_error | <i>Display a popup error message</i> |
|---------------|--------------------------------------|

Description

Displays a popup error message and terminates the testing session.

Usage

```
display_error(...)
```

Arguments

... Arguments to be pasted together (without separator) to form the error message.

| | |
|---------------|--------------------------------|
| dropdown_page | <i>Make dropdown list page</i> |
|---------------|--------------------------------|

Description

Creates a page where the response is to be selected from a dropdown list.

Usage

```
dropdown_page(label, prompt, choices, alternative_choice = FALSE,
  alternative_text = "Other (please state)", save_answer = TRUE,
  validate = dropdown_page.validate(alternative_choice,
  alternative_text), on_complete = NULL, next_button_text = "Next",
  max_width_pixels = 200, admin_ui = NULL)
```

Arguments

| | |
|--------------------|--|
| label | Page label (scalar character). |
| prompt | Prompt to be displayed above the response choices. Can be either a scalar character (e.g. "What is 2 + 2?") or an object of class "shiny.tag", e.g. shiny::tags\$p("What is 2 + 2?") |
| choices | Character vector of choices for the participant. If names are provided, then these names will be used for display, whereas the values will be stored in the results. |
| alternative_choice | Whether or not to give the participant the option of providing a free-text response instead of selecting one of the dropdown options. |
| alternative_text | Prompt for the free-text box (only relevant if alternative_choice is set to TRUE). |
| save_answer | See page . |
| validate | See page . |
| on_complete | See page . |
| next_button_text | Text to display on the next-page button (character scalar). |

| | |
|------------------|--|
| max_width_pixels | Maximum width of the response UI, in pixels. |
| admin_ui | See page . |

elt_save_results_to_disk
Save results to disk (test element version)

Description

Returns a test element that saves the current participant's results to disk.

Usage

```
elt_save_results_to_disk(complete)
```

Arguments

| | |
|----------|---|
| complete | Whether or not the participant has now completed the test (scalar Boolean). |
|----------|---|

| | |
|------------|-------------------|
| end_module | <i>End module</i> |
|------------|-------------------|

Description

Returns a code block that ends a psychTestR module. Modules have their own set of local variables. They also have identifying labels that are stored alongside result entries created during the module.

Usage

```
end_module()
```

| | |
|-------|--------------|
| error | <i>Error</i> |
|-------|--------------|

Description

Accesses the error slot of the participant's state object. This slot takes values of NULL or a character scalar. When not NULL, the server will display its value as an error message to the participant.

Usage

```
error(state)
```

Arguments

| | |
|-------|-----------------------------|
| state | Participant's state object. |
|-------|-----------------------------|

| | |
|---------|--------------|
| error<- | <i>Error</i> |
|---------|--------------|

Description

Sets the error slot of the participant's state object. This slot takes values of NULL or a character scalar. When not NULL, the server will display its value as an error message to the participant.

Usage

```
error(state) <- value
```

Arguments

| | |
|-------|-------------------------------|
| state | Participant's state object. |
| value | New value for the error slot. |

| | |
|------------|-----------------------|
| final_page | <i>New final page</i> |
|------------|-----------------------|

Description

Creates a page that concludes the test.

Usage

```
final_page(body, admin_ui = NULL)
```

Arguments

| | |
|----------|---|
| body | Page body. Can be either a scalar character (e.g. "Welcome to the test!") or an object of class "shiny.tag", e.g. shiny::tags\$p("Welcome to the test!"). |
| admin_ui | See page . |

| | |
|---------------------------|----------------------------------|
| finish_test_and_give_code | <i>Finish test and give code</i> |
|---------------------------|----------------------------------|

Description

psychTestR logic for finishing the current test and giving the participant a randomly generated code that can then be used for organising payment (e.g. for Amazon Mechanical Turk).

Usage

```
finish_test_and_give_code(researcher_email)
```

Arguments

researcher_email

The researcher's email address (given in case the participant has any further questions).

get_basic_demographics

Get basic demographics

Description

Gets basic demographic information from the participant.

Usage

```
get_basic_demographics(intro = basic_demographics_default_intro(),  
  gender = TRUE, age = TRUE, occupation = TRUE, education = TRUE)
```

Arguments

| | |
|------------|---|
| intro | Introductory page to the demographics section. If NULL, no introduction is given. |
| gender | Ask the participant's gender? (Boolean) |
| age | Ask the participant's age? (Boolean) |
| occupation | Ask the participant's occupation? (Boolean) |
| education | Ask the participant's education? (Boolean) |

get_global

Get global variable

Description

Gets a global variable from the participant's testing session.

Usage

```
get_global(key, state)
```

Arguments

| | |
|-------|--|
| key | The variable's key (character scalar). |
| state | The participant's state object. |

Value

The global variable with key equal to key.

get_local

Get local variable

Description

Gets a local variable from the participant's testing session.

Usage

```
get_local(key, state)
```

Arguments

| | |
|-------|--|
| key | The variable's key (character scalar). |
| state | The participant's state object. |

Value

The local variable with key equal to key.

get_p_id

Get participant ID

Description

A psychTestR page that gets the participant to enter their ID.

Usage

```
get_p_id(prompt = "Please enter your participant ID.",
  placeholder = "e.g. 10492817", button_text = "Next",
  width = "300px", validate = "auto", admin_ui = NULL)
```

Arguments

| | |
|-------------|---|
| prompt | Prompt to display (character scalar or Shiny tag object) |
| placeholder | Placeholder text for the text box (character scalar). |
| button_text | Text for the submit button (character scalar). |
| width | Width of the text box (character scalar, should be valid HTML). |
| validate | See page . |
| admin_ui | See page . |

Note

Participant IDs must be between 1 and 100 characters long, and solely comprise alphanumeric characters and underscores.

`get_results`*Get results*

Description

Gets the participants current results, optionally along with session information.

Usage

```
get_results(state, complete, add_session_info = FALSE)
```

Arguments

| | |
|-------------------------------|---|
| <code>state</code> | The participant's state object. |
| <code>complete</code> | Whether the participant has now completed the test (Boolean). |
| <code>add_session_info</code> | Whether to add session information (Boolean). |

Value

A results object.

`get_session_info`*Get session info*

Description

Gets information about the current participant's session, including things like participant ID, time started, and number of restarts.

Usage

```
get_session_info(state, complete)
```

Arguments

| | |
|-----------------------|--|
| <code>state</code> | The participant's state object. |
| <code>complete</code> | Whether or not the session is now considered complete (this only affects the returned session info). |

Value

List containing session information.

| | |
|-------------------------------|--|
| get_supplementary_results_dir | <i>Supplementary results directory</i> |
|-------------------------------|--|

Description

Returns the location of the supplementary results directory. The supplementary results directory provides a place to store results in addition to the results saved by the psychTestR function `save_results_to_disk`. The researcher has full control over what to put in the supplementary results directory.

Usage

```
get_supplementary_results_dir(opt)
```

Arguments

| | |
|-----|--|
| opt | Options list as created by <code>test_options()</code> . |
|-----|--|

Value

Character scalar identifying the (relative) path to the supplementary results directory.

Note

The supplementary results directory is typically stored within the main results directory. It will be deleted if the researcher presses the 'Delete all results' button in the admin panel.

| | |
|----------------|---------------------------|
| get_url_params | <i>Get URL parameters</i> |
|----------------|---------------------------|

Description

Gets the current URL parameters as stored in state.

Usage

```
get_url_params(state)
```

Arguments

| | |
|-------|---------------------------------|
| state | The participant's state object. |
|-------|---------------------------------|

| | |
|------|------------------|
| i18n | <i>Translate</i> |
|------|------------------|

Description

Translates text into a language determined by the current testing session. Translation is carried out using a dictionary defined at test creation (see [i18n_dict](#)). The function can typically only be used within a call to [new_timeline](#), which defines the dictionary in which to look up i18n. Calling i18n outside this context will typically result in an error.

Usage

```
i18n(x, html = TRUE, sub = character())
```

Arguments

| | |
|------|---|
| x | Character scalar identifying the term to be translated. |
| html | Boolean; whether or not the translation output should be parsed as HTML or as plain text (the default is TRUE, corresponding to HTML parsing). |
| sub | A named character vector or list defining substitutions to make within the translation output. The text of the translation output may contain passages such as <code>{{var1}}</code> ; if <code>sub = c(var1 = "Hello")</code> , then <code>{{var1}}</code> will be replaced with the text Hello. |

Value

The translated output (typically a character scalar or a `shiny::HTML()` output).

| | |
|-----------|-------------------|
| i18n_dict | <i>Dictionary</i> |
|-----------|-------------------|

Description

Dictionaries allow psychTestR tests to support multiple languages during test administration. These dictionaries (class name: `i18n_dict`) define mappings between string keys (e.g. `prompt1`) and (potentially) multiple outputs, each corresponding to different languages (e.g. English, German). The dictionary is passed as an argument to [new_timeline](#), and provides the context for the evaluation of calls to [i18n](#), responsible for translating individual terms.

Usage

```
i18n_dict
```

Format

An object of class `R6ClassGenerator` of length 24.

Methods

`i18n_dict$new()` defines and returns a new dictionary which can be saved as an object. Its first argument, `x`, should be a data frame defining the dictionary. Each row of this data frame should correspond to a term to be translated. It should have a column entitled 'key', which defines the identifying keys for each term. All other columns should provide translations into different languages, with the language being identified by the column name, according to ISO 639-2 conventions. The second argument, `markdown`, is a scalar Boolean that determines whether the data frame is formatted in Markdown or not (Markdown allows for efficient and readable text markup, e.g. italic and bold). Text is parsed according to standard Markdown conventions, with one addition: two successive backslashes (written `"\"` in a text file, or `"\\"` in R) are interpreted as a new paragraph.

`x$as.data.frame`, where `x` is a dictionary object, converts the dictionary back to a data frame representation.

`x$translate(key, language, allow_missing)` uses the dictionary `x` to translate `key` into `language`. If `allow_missing` is TRUE, then NULL is returned when no translation is found, otherwise an error is thrown.

| | |
|------------------------------|----------------------------------|
| <code>is.test_element</code> | <i>Is object a test element?</i> |
|------------------------------|----------------------------------|

Description

Checks whether an object is a test element.

Usage

```
is.test_element(x)
```

Arguments

| | |
|----------------|------------------|
| <code>x</code> | Object to check. |
|----------------|------------------|

| | |
|--------------------------|---------------------------------|
| <code>is.timeline</code> | <i>Is it a timeline object?</i> |
|--------------------------|---------------------------------|

Description

Checks whether an object is a timeline object.

Usage

```
is.timeline(x)
```

Arguments

| | |
|----------------|-----------------|
| <code>x</code> | Object to check |
|----------------|-----------------|

Value

TRUE if object is a timeline object, FALSE otherwise.

| | |
|----------------|------------------------|
| is_test_closed | <i>Is test closed?</i> |
|----------------|------------------------|

Description

Checks whether the test is closed.

Usage

```
is_test_closed(opt)
```

Arguments

opt Test options list as created by test_options().

Value

TRUE if the test is closed, FALSE otherwise.

| | |
|------------|-------------|
| loop_while | <i>Loop</i> |
|------------|-------------|

Description

Repeatedly show a series of test elements to the participant while a condition is satisfied. The elements are shown at least once; at the end of each pass, the test function is checked.

Usage

```
loop_while(test, logic)
```

Arguments

| | |
|-------|--|
| test | Test function to execute. Argument list should include ...; further permitted arguments are: state, the participant's state object; input, the Shiny input object for the current page; output, the Shiny output object for the current page; session, the Shiny session object; opt, test options as created by test_options(). |
| logic | List of psychTestR test elements. |

| | |
|-----------|-----------------------------|
| make_test | <i>Make psychTestR test</i> |
|-----------|-----------------------------|

Description

The top-level function that defines a psychTestR test. This should be the final line of your app script, typically entitled app.R.

Usage

```
make_test(elts, opt = demo_options(), custom_admin_panel = NULL)
```

Arguments

`elts` List of test elements defining the test's timeline.

`opt` Options list as created by `test_options()`.

`custom_admin_panel`

This argument supports the construction of a custom admin panel. The argument should be a function that runs during the Shiny server function. The argument list of this function should include `...`; it may also include `state`, `input`, `output`, and `session`, all with their traditional meanings from Shiny or psychTestR. This function should include an expression of the form `output$custom_admin_panel <- shiny::renderUI(...)` where the code inside `renderUI()` defines the UI of the admin panel using standard Shiny UI vocabulary. Further documentation for this option should be forthcoming.

| | |
|--------------|--------------------------|
| make_ui_NAFC | <i>Make NAFC buttons</i> |
|--------------|--------------------------|

Description

Creates HTML code for n-alternative forced-choice response options.

Usage

```
make_ui_NAFC(choices, labels = NULL, hide = FALSE,
  arrange_vertically = length(choices) > 2L, id = "response_ui")
```

Arguments

`choices` Character vector of choices for the participant. If unnamed, then these values will be used both for button IDs and for button labels. If named, then values will be used for button IDs and names will be used for button labels.

`labels` Optional vector of labels for the NAFC choices. If not NULL, will overwrite the names of choices. This vector of labels can either be a character vector or a list of Shiny tag objects, e.g. as created by `shiny::HTML()`.

`hide` Whether the response buttons should be hidden (possibly to be shown later).

| | |
|--------------------|--|
| arrange_vertically | Whether to arrange the response buttons vertically (the default) as opposed to horizontally. |
| id | HTML ID for the div containing the response buttons. |

| | |
|-----------|----------------------|
| NAFC_page | <i>New NAFC page</i> |
|-----------|----------------------|

Description

Creates an n-alternative forced-choice page.

Usage

```
NAFC_page(label, prompt, choices, labels = NULL, save_answer = TRUE,
  arrange_vertically = length(choices) > 2L, hide_response_ui = FALSE,
  response_ui_id = "response_ui", on_complete = NULL,
  admin_ui = NULL)
```

Arguments

| | |
|--------------------|--|
| label | Label for the current page (character scalar). |
| prompt | Prompt to be displayed above the response choices. Can be either a scalar character (e.g. "What is 2 + 2?") or an object of class "shiny.tag", e.g. <code>shiny::tags\$p("What is 2 + 2?")</code> . |
| choices | Character vector of choices for the participant. If unnamed, then these values will be used both for button IDs and for button labels. If named, then values will be used for button IDs and names will be used for button labels. |
| labels | Optional vector of labels for the NAFC choices. If not NULL, will overwrite the names of choices. This vector of labels can either be a character vector or a list of Shiny tag objects, e.g. as created by <code>shiny::HTML()</code> . |
| save_answer | See page . |
| arrange_vertically | Whether to arrange the response buttons vertically (the default) as opposed to horizontally. |
| hide_response_ui | Whether to begin with the response interface hidden (it can be subsequently made visible through Javascript, using the element ID as set in <code>response_ui_id</code> . See <code>audio_NAFC_page</code> for an example.). |
| response_ui_id | HTML ID for the response user interface. |
| on_complete | See page . |
| admin_ui | See page . |

| | |
|---------------------|----------------------------|
| new_results_section | <i>New results section</i> |
|---------------------|----------------------------|

Description

Returns a test element that initialises a new results section.

Usage

```
new_results_section(label)
```

Arguments

| | |
|-------|--|
| label | Label to give the new results section. |
|-------|--|

| | |
|--------------|---------------------|
| new_timeline | <i>New timeline</i> |
|--------------|---------------------|

Description

Creates a new timeline. Timelines allow tests to support multiple languages.

Usage

```
new_timeline(x = stop("x not supplied"), dict = NULL,
             default_lang = "EN")
```

Arguments

| | |
|--------------|---|
| x | Expression defining a series of test elements. This expression will be evaluated once for every language provided in the dictionary argument (dict). A call to <code>i18n(key)</code> , where key is a term defined in dict, will be translated to its definitions in each respective language. |
| dict | Dictionary object as created by <code>i18n_dict\$new</code> . |
| default_lang | If no dictionary is supplied, then <code>new_timeline()</code> assumes that the current language is <code>default_lang</code> . |

Note

Debugging is difficult within `new_timeline()` because of its underlying macro definition. When building a test, we recommend defining small timelines first and subsequently combining them. This helps to narrow down the source of any errors.

| | |
|-----------------|----------------------------|
| one_button_page | <i>New one-button page</i> |
|-----------------|----------------------------|

Description

Creates a page with a prompt and one button which, when clicked, advances to the next page. This is typically used for giving the participant information about the test.

Usage

```
one_button_page(body, admin_ui = NULL, button_text = "Next",  
  on_complete = NULL)
```

Arguments

| | |
|-------------|---|
| body | Page body. Can be either a scalar character (e.g. "Welcome to the test!") or an object of class "shiny.tag", e.g. shiny::tags\$p("Welcome to the test!"). |
| admin_ui | See page . |
| button_text | Text to display on the button. Should be a scalar character vector. |
| on_complete | See page . |

| | |
|-----------|------------------|
| open_test | <i>Open test</i> |
|-----------|------------------|

Description

Opens the test.

Usage

```
open_test(opt)
```

Arguments

| | |
|-----|---|
| opt | Test options list as created by <code>test_options()</code> . |
|-----|---|

| | |
|------|-----------------|
| page | <i>New page</i> |
|------|-----------------|

Description

This is the most general way to create a psychTestR page.

Usage

```
page(ui, admin_ui = NULL, label = NULL, final = FALSE,
      get_answer = NULL, save_answer = FALSE, validate = NULL,
      on_complete = NULL)
```

Arguments

| | |
|-------------|---|
| ui | Page UI. Can be either a scalar character (e.g. "Welcome to the test!") or an object of class "shiny.tag", e.g. shiny::tags\$p("Welcome to the test!"). |
| admin_ui | Optional UI component for the admin panel. |
| label | Page label (character scalar). |
| final | Whether or not the page is the final page in the test. |
| get_answer | Optional function for extracting the participant's answer from the current page. The argument list should include ..., input, and optionally state, the participant's state object. The function should extract the answer from the Shiny input object and return it as its output. |
| save_answer | Whether or not to save the answer. |
| validate | Optional validation function. The argument list should include ..., and any of: answer, the participant's most recent answer; state, the participant's state object; input, the current page's Shiny input object; opt, the test's option list as created by test_options(); session, the current Shiny session object. It should return TRUE for a successful validation; for an unsuccessful validation, it should return either FALSE or a character scalar error message. If validation fails then the page will be refreshed, usually to give the user a chance to revise their input. |
| on_complete | Optional function to execute on leaving the page (after successful validation). The argument list should include ..., and any of: state, the participant's state object; answer, the participant's most recent answer; input, the current page's Shiny input object; session, the current Shiny session object; opt, the test's option list as created by test_options(). |

| | |
|------------|---------------------------------|
| pt_options | <i>Alias for test_options()</i> |
|------------|---------------------------------|

Description

Alias for test_options() (deprecated).

Usage

```
pt_options(...)
```

Arguments

... Arguments to pass to `test_options()`.

| | |
|------|---------------------------|
| p_id | <i>Get participant ID</i> |
|------|---------------------------|

Description

Returns the participant's ID.

Usage

```
p_id(state)
```

Arguments

state The participant's state object.

| | |
|--------|----------------------------|
| p_id<- | <i>Sets participant ID</i> |
|--------|----------------------------|

Description

Sets the participant's ID.

Usage

```
p_id(state) <- value
```

Arguments

state The participant's state object.
value The value to set the ID to.

 reactive_page

Reactive page

Description

Creates a reactive page. A reactive page is defined by a function that is called once the participant reaches a given location in the timeline. This function must be idempotent, i.e. calling it multiple times should have the same effect as calling it once).

Usage

```
reactive_page(fun)
```

Arguments

| | |
|-----|--|
| fun | Function that returns a page argument. This function's argument list must include It should also include one or more named arguments that provide information for generating the new page: potential arguments include state, corresponding to the participant's state object, answer, which is the most recent participant response, and opt, which is the test's option list as created by test_options. The function should always return an object of class page. |
|-----|--|

 register_next_results_section

Register next results section

Description

Register next results section

Usage

```
register_next_results_section(x, label)
```

Arguments

| | |
|-------|--|
| x | Where should the new section be registered? Methods exist for results objects and for state objects. |
| label | Label for the new section. |

| | |
|---------|-----------------------|
| results | <i>Access results</i> |
|---------|-----------------------|

Description

Accesses the results slot in state.

Usage

```
results(state)
```

Arguments

| | |
|-------|---------------------------------|
| state | The participant's state object. |
|-------|---------------------------------|

| | |
|-----------|--------------------|
| results<- | <i>Set results</i> |
|-----------|--------------------|

Description

Sets the results slot in state.

Usage

```
results(state) <- value
```

Arguments

| | |
|-------|-----------------------------------|
| state | The participant's state object. |
| value | Value to set the results slot to. |

| | | | | |
|-------------|------------|---------------------------------------|----|-------------|
| save_result | #' @export | get_next_results_section | <- | function(x) |
| | | UseMethod("get_next_results_section") | #' | @ex- |
| | port | get_next_results_section.state | <- | function(x) |
| | | get_next_results_section(results(x)) | | |

Description

```
get_next_results_section.results <- function(x) y <- attr(x, "new_section") z <- names(results) if
(!is.null(y)) return(y) if (length(z) > 0) return(z[length(z)]) stop() else names(results)
```

Save result

Usage

```
save_result(place, label, value)
```

Arguments

| | |
|-------|---|
| place | Place to save the results. Methods exist for results objects and for state objects. |
| label | Label for the new datum. |
| value | Value of the new datum. |

Details

Saves a new datum into the participant's results store.

| | |
|----------------------|-----------------------------|
| save_results_to_disk | <i>Save results to disk</i> |
|----------------------|-----------------------------|

Description

Saves the current participant's results to disk. This function can be called e.g. within a code block.

Usage

```
save_results_to_disk(complete, state, opt, ...)
```

Arguments

| | |
|----------|---|
| complete | Whether or not the participant has now completed the test (scalar Boolean). |
| state | The participant's state object. |
| opt | Test options as created by test_options(). |
| ... | Further arguments are allowed but ignored (for back-compatibility). |

| | |
|------------|----------------------------|
| set_global | <i>Set global variable</i> |
|------------|----------------------------|

Description

Sets a global variable for the participant's testing session.

Usage

```
set_global(key, value, state)
```

Arguments

| | |
|-------|--|
| key | The variable's key (character scalar). |
| value | The value to set the variable to. |
| state | The participant's state object. |

| | |
|-----------|---------------------------|
| set_local | <i>Set local variable</i> |
|-----------|---------------------------|

Description

Sets a local variable for the participant's testing session.

Usage

```
set_local(key, value, state)
```

Arguments

| | |
|-------|--|
| key | The variable's key (character scalar). |
| value | The value to set the variable to. |
| state | The participant's state object. |

| | |
|---------------|----------------------------|
| set_url_param | <i>Set a URL parameter</i> |
|---------------|----------------------------|

Description

Sets a URL parameter.

Usage

```
set_url_param(key, value, session, state)
```

Arguments

| | |
|---------|-----------------------------------|
| key | URL parameter key. |
| value | URL parameter value. |
| session | The participant's session object. |
| state | The participant's state object. |

| | |
|--------------|-------------------|
| skip_n_pages | <i>Skip pages</i> |
|--------------|-------------------|

Description

Skips a certain number of pages, typically in response to a participant action.

Usage

```
skip_n_pages(state, n)
```

Arguments

| | |
|-------|--|
| state | Participant's state object. |
| n | Number of pages to skip (negative numbers skip backwards). |

| | |
|--------------|---------------------|
| test_options | <i>Test options</i> |
|--------------|---------------------|

Description

Defines the options for running a given test.

Usage

```
test_options(title, admin_password, researcher_email = NULL,
  max_num_participants = NULL, demo = FALSE, languages = "EN",
  log_error = TRUE, show_full_error_msg = TRUE, notify_error = FALSE,
  notify_new_participant = FALSE, pushbullet_email = NULL,
  pushbullet_apikey = NULL, max_participants_msg = NULL,
  server_closed_msg = NULL, problems_info = "default",
  theme = shinythemes::shinytheme("readable"), auto_p_id = TRUE,
  enable_resume_session = TRUE, allow_any_p_id_url = FALSE,
  force_p_id_from_url = FALSE, enable_admin_panel = TRUE,
  output_dir = "output", session_timeout_min = 7 * 24 * 60,
  clean_sessions_interval_min = 15, logo = NULL, logo_width = NULL,
  logo_height = NULL)
```

Arguments

| | |
|------------------------|--|
| title | The test's title. |
| admin_password | Password to access the admin panel. |
| researcher_email | Researcher's email; used in participant help message. |
| max_num_participants | Maximum number of participant completes before the test automatically closes (excludes pilot participants as indicated through the admin panel). |
| demo | Whether the test is to be run in demo mode. |
| languages | Character vector of languages that may be selected via the URL parameter 'language'. If no language is provided by the URL parameter, defaults to the first language in this vector. Languages should be encoded according to ISO 639-2 conventions. |
| log_error | Whether to log errors (this feature is under development). |
| show_full_error_msg | Whether to show full error messages. |
| notify_error | Whether to send a Pushbullet message when the test experiences an error (this feature is under development). |
| notify_new_participant | Whether to send a Pushbullet message when a new participant completes the test. |
| pushbullet_email | Email to send Pushbullet notifications to. |
| pushbullet_apikey | Pushbullet API key (see https://www.pushbullet.com/). |

| | |
|-----------------------------|---|
| max_participants_msg | Message to display when the participant quota is reached (NULL gives default). |
| server_closed_msg | Message to display when the server is closed via the admin panel (NULL gives default). |
| problems_info | Message to display at the bottom of the screen with advice about what to do if a problem occurs. Defaults to a standard message including the researcher's email (if provided). |
| theme | Shiny theme: see e.g. the shinythemes package. |
| auto_p_id | Whether or not to automatically generate an ID for each participant. |
| enable_resume_session | Whether to allow participants to resume sessions by refreshing the page. |
| allow_any_p_id_url | Whether to allow new participants to give themselves an arbitrary participant ID by passing it as a URL parameter. |
| force_p_id_from_url | Whether to force new participants to provide their participant ID as a URL parameter. |
| enable_admin_panel | Whether to enable the admin panel. |
| output_dir | String identifying psychTestR's output directory. |
| session_timeout_min | Minimum time until a participant's session times out (minutes). |
| clean_sessions_interval_min | How often should psychTestR check for expired participant sessions (minutes). |
| logo | Path to a logo to display in the header (optional). |
| logo_width | Logo width, e.g. "100px". |
| logo_height | Logo height, e.g. "50px". |

| | |
|------------------|-------------------------------|
| test_permissions | <i>Test write permissions</i> |
|------------------|-------------------------------|

Description

Test the ability to write to a given directory.

Usage

```
test_permissions(dir)
```

Arguments

| | |
|-----|--------------------|
| dir | Directory to test. |
|-----|--------------------|

| | |
|-----------------|------------------------|
| text_input_page | <i>Text input page</i> |
|-----------------|------------------------|

Description

Creates a page where the participant puts their answer in a text box.

Usage

```
text_input_page(label, prompt, one_line = TRUE, save_answer = TRUE,
  placeholder = NULL, button_text = "Next", width = "300px",
  height = "100px", validate = NULL, on_complete = NULL,
  admin_ui = NULL)
```

Arguments

| | |
|-------------|--|
| label | Label for the current page (character scalar). |
| prompt | Prompt to display (character scalar or Shiny tag object). |
| one_line | Whether the answer box only has one line of text. |
| save_answer | See page . |
| placeholder | Placeholder text for the text box (character scalar). |
| button_text | Text for the submit button (character scalar). |
| width | Width of the text box (character scalar, should be valid HTML). |
| height | Height of the text box (character scalar, should be valid HTML). |
| validate | See page . |
| on_complete | See page . |
| admin_ui | See page . |

| | |
|----------|-----------------|
| timeline | <i>Timeline</i> |
|----------|-----------------|

Description

Timelines are series of psychTestR test elements that chain together to form a test. They support internationalisation, defining parallel series of test elements for the available languages.

Usage

```
timeline
```

Format

An object of class R6ClassGenerator of length 24.

Creation

Timelines are created using [new_timeline](#).

Manipulation

Timelines can be combined with other timelines and with test elements using [c](#).

Usage

Timelines are ultimately passed to [make_test](#).

Other methods

`x$get(language, i)` returns a list of test elements corresponding to language as extracted from the timeline x, or, if i is not NULL, the ith such test element.

`x$drop_languages(drop)` removes support for a set of languages from timeline x, where drop is the character vector of languages to remove.

| | |
|----------------|-----------------------|
| trigger_button | <i>Trigger button</i> |
|----------------|-----------------------|

Description

A version of `shiny::actionButton` that triggers the next `psychTestR` page.

Usage

```
trigger_button(inputId, label, icon = NULL, width = NULL)
```

Arguments

| | |
|---------|----------------------------------|
| inputId | See actionButton |
| label | See actionButton |
| icon | See actionButton |
| width | See actionButton |

| | |
|-----------------|-----------------------------|
| video_NAFC_page | <i>Make NAFC video page</i> |
|-----------------|-----------------------------|

Description

Creates an n-alternative forced-focused choice page with a video prompt.

Usage

```
video_NAFC_page(label, prompt, choices, url, labels = NULL,
  type = tools::file_ext(url), save_answer = TRUE,
  on_complete = NULL, video_width = "100%",
  arrange_choices_vertically = length(choices) > 2L, wait = TRUE,
  loop = FALSE, admin_ui = NULL)
```

Arguments

| | |
|----------------------------|--|
| label | Label for the current page (character scalar). |
| prompt | Prompt to be displayed above the response choices. Can be either a scalar character (e.g. "What is 2 + 2?") or an object of class "shiny.tag", e.g. <code>shiny::tags\$p("What is 2 + 2?")</code> . |
| choices | Character vector of choices for the participant. If unnamed, then these values will be used both for button IDs and for button labels. If named, then values will be used for button IDs and names will be used for button labels. |
| url | URL to the video. Can be an absolute URL (e.g. "http://mysite.com/video.mp4") or a URL relative to the /www directory (e.g. "video.mp4"). |
| labels | Optional vector of labels for the NAFC choices. If not NULL, will overwrite the names of choices. This vector of labels can either be a character vector or a list of Shiny tag objects, e.g. as created by <code>shiny::HTML()</code> . |
| type | Video type (e.g. 'mp4'). Defaults to the provided file extension. |
| save_answer | See page . |
| on_complete | See page . |
| video_width | Video width, as passed to HTML (e.g. '50px'). |
| arrange_choices_vertically | Whether to arrange the response buttons vertically (the default) as opposed to horizontally. |
| wait | Whether to wait for the video to finish before displaying the response buttons. |
| loop | Whether the video should loop. |
| admin_ui | See page . |

volume_calibration_page

Make volume calibration page

Description

Creates a page for the participant to calibrate their volume, using example audio, and the volume controls on their computer.

Usage

```
volume_calibration_page(url, type = tools::file_ext(url),
  prompt = NULL, button_text = "Next", on_complete = NULL,
  admin_ui = NULL)
```

Arguments

| | |
|-------------|---|
| url | URL to the audio. Can be an absolute URL (e.g. "http://mysite.com/audio.mp3") or a URL relative to the /www directory (e.g. "audio.mp3"). |
| type | Audio type (e.g. 'mp3'). Defaults to the provided file extension. |
| prompt | Prompt to be displayed. If left NULL, a sensible English prompt is provided. |
| button_text | Button text (scalar character). |
| on_complete | See page . |
| admin_ui | See page . |

Index

*Topic **datasets**

- i18n_dict, [15](#)
- timeline, [30](#)

- actionButton, [31](#)
- answer, [3](#)
- answer<-, [3](#)
- assert_global_is_null, [4](#)
- audio_NAFC_page, [4](#)

- basic_demographics_default_intro, [5](#)
- begin_module, [5](#)

- c, [31](#)
- check_dirs, [5](#)
- close_test, [6](#)
- code_block, [6](#)

- demo, [6](#)
- demo.get_url_params, [7](#)
- demo.set_url_params, [7](#)
- demo_options, [7](#)
- display_error, [8](#)
- dropdown_page, [8](#)

- elt_save_results_to_disk, [9](#)
- end_module, [9](#)
- error, [9](#)
- error<-, [10](#)

- final_page, [10](#)
- finish_test_and_give_code, [10](#)

- get_basic_demographics, [11](#)
- get_global, [11](#)
- get_local, [12](#)
- get_p_id, [12](#)
- get_results, [13](#)
- get_session_info, [13](#)
- get_supplementary_results_dir, [14](#)
- get_url_params, [14](#)

- i18n, [15](#), [15](#), [20](#)
- i18n_dict, [15](#), [15](#), [20](#)
- is.test_element, [16](#)
- is.timeline, [16](#)
- is_test_closed, [17](#)

- loop_while, [17](#)

- make_test, [18](#), [31](#)
- make_ui_NAFC, [18](#)

- NAFC_page, [19](#)
- new_results_section, [20](#)
- new_timeline, [15](#), [20](#), [30](#)

- one_button_page, [21](#)
- open_test, [21](#)

- p_id, [23](#)
- p_id<-, [23](#)
- page, [4](#), [5](#), [8–10](#), [12](#), [19](#), [21](#), [22](#), [30](#), [32](#)
- pt_options, [22](#)

- reactive_page, [24](#)
- register_next_results_section, [24](#)
- results, [25](#)
- results<-, [25](#)

- save_result, [25](#)
- save_results_to_disk, [26](#)
- set_global, [26](#)
- set_local, [27](#)
- set_url_param, [27](#)
- skip_n_pages, [27](#)

- test_options, [7](#), [23](#), [28](#)
- test_permissions, [29](#)
- text_input_page, [30](#)
- timeline, [30](#)
- trigger_button, [31](#)

- video_NAFC_page, [31](#)
- volume_calibration_page, [32](#)