

# Package: sdmEvalToolUI (via r-universe)

October 1, 2026

**Title** SDM Evaluation Tool Shiny UI

**Version** 0.0.1

**Description** The goal of sdmEvalToolUI is to implement Shiny functionality according to the SDM Evaluation Tool requirements to evaluate materials.

**License** Apache License 2.0

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**Imports** bsicons, bslib, dplyr (>= 1.1.4), ggplot2 (>= 4.0.0), glue (>= 1.8.0), htmlwidgets (>= 1.6.4), jsonlite (>= 2.0.0), leaflet, leaflet.extras (>= 2.0.2), plotly (>= 4.11.0), purrr (>= 1.1.0), reactable, rlang, sdmEvalToolCore, sf (>= 1.0.21), shiny, shinycssloaders (>= 1.1.0), shinyjs (>= 2.1.0), shinymanager, stringr (>= 1.5.1), terra (>= 1.8.93), tidyr (>= 1.3.1), viridisLite (>= 0.4.2)

**Suggests** rmarkdown, shinytest2, testthat (>= 3.0.0), withr, waldo

**Remotes** LandSciTech/sdmEvaluationTool/sdmEvalToolCore

**Config/testthat/edition** 3

**Depends** R (>= 4.1)

**Config/roxygen2/version** 8.0.0

**RoxygenNote** 7.3.3

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libpng-dev libuv1-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

**Repository** <https://psolymos.r-universe.dev>

**Date/Publication** 2026-07-21 14:45:11 UTC

**RemoteUrl** <https://github.com/LandSciTech/sdmEvaluationTool>

**RemoteRef** HEAD

**RemoteSha** 32444f2dd7ef4af0025b8324a37b4f796d62fe6f

**RemoteSubdir** sdmEvalToolUI

## Contents

|                                               |    |
|-----------------------------------------------|----|
| add_control . . . . .                         | 5  |
| add_markers . . . . .                         | 6  |
| add_raster . . . . .                          | 6  |
| add_selected_markers . . . . .                | 7  |
| add_selected_subunits . . . . .               | 8  |
| add_subunits . . . . .                        | 9  |
| affirmative . . . . .                         | 10 |
| app_materials . . . . .                       | 10 |
| app_users . . . . .                           | 11 |
| base_map . . . . .                            | 11 |
| bslib_wrapper . . . . .                       | 12 |
| check_q_mismatch . . . . .                    | 13 |
| coords_to_poly . . . . .                      | 13 |
| db_connect_check . . . . .                    | 14 |
| deployment_subunits_map . . . . .             | 14 |
| deployment_subunits_prep . . . . .            | 15 |
| evals_answered . . . . .                      | 15 |
| evals_details . . . . .                       | 16 |
| evals_extract . . . . .                       | 17 |
| evals_list . . . . .                          | 17 |
| evals_table . . . . .                         | 18 |
| feature_ids . . . . .                         | 19 |
| fetch_questions . . . . .                     | 19 |
| fmt_species . . . . .                         | 20 |
| fmt_time . . . . .                            | 20 |
| get_material_settings . . . . .               | 21 |
| have_data . . . . .                           | 21 |
| identical_loose . . . . .                     | 22 |
| input_to_value . . . . .                      | 22 |
| lang . . . . .                                | 23 |
| map_has_group . . . . .                       | 23 |
| map_reactive_vals . . . . .                   | 24 |
| mod_comp_deployment_subunits_server . . . . . | 24 |
| mod_comp_deployment_subunits_ui . . . . .     | 25 |
| mod_comp_model_fit_server . . . . .           | 26 |
| mod_comp_model_fit_ui . . . . .               | 26 |
| mod_comp_model_summary_server . . . . .       | 27 |
| mod_comp_model_summary_ui . . . . .           | 27 |
| mod_comp_obs_chart_server . . . . .           | 28 |
| mod_comp_obs_chart_ui . . . . .               | 28 |
| mod_comp_observations_server . . . . .        | 29 |
| mod_comp_observations_ui . . . . .            | 30 |
| mod_comp_predictor_metadata_server . . . . .  | 30 |
| mod_comp_predictor_metadata_ui . . . . .      | 31 |
| mod_comp_predictor_raster_server . . . . .    | 31 |
| mod_comp_predictor_raster_ui . . . . .        | 32 |

|                                              |    |
|----------------------------------------------|----|
| mod_comp_spatial_prediction_server . . . . . | 33 |
| mod_comp_spatial_prediction_ui . . . . .     | 34 |
| mod_comp_template_server . . . . .           | 34 |
| mod_comp_template_spatial_server . . . . .   | 35 |
| mod_comp_template_spatial_ui . . . . .       | 36 |
| mod_comp_template_ui . . . . .               | 36 |
| mod_page_model_server . . . . .              | 37 |
| mod_page_model_ui . . . . .                  | 37 |
| mod_page_observations_server . . . . .       | 38 |
| mod_page_observations_ui . . . . .           | 38 |
| mod_page_overview_server . . . . .           | 39 |
| mod_page_overview_ui . . . . .               | 40 |
| mod_page_predictions_server . . . . .        | 40 |
| mod_page_predictions_ui . . . . .            | 41 |
| mod_page_predictors_server . . . . .         | 41 |
| mod_page_predictors_ui . . . . .             | 42 |
| mod_page_summary_server . . . . .            | 42 |
| mod_page_summary_ui . . . . .                | 43 |
| mod_page_template_server . . . . .           | 44 |
| mod_page_template_spatial_server . . . . .   | 44 |
| mod_page_template_spatial_ui . . . . .       | 45 |
| mod_page_template_ui . . . . .               | 45 |
| mod_page_test_server . . . . .               | 46 |
| mod_page_test_ui . . . . .                   | 46 |
| mod_utils_details_server . . . . .           | 47 |
| mod_utils_details_ui . . . . .               | 47 |
| mod_utils_evaluations_server . . . . .       | 48 |
| mod_utils_evaluations_ui . . . . .           | 49 |
| mod_utils_map_selections_server . . . . .    | 49 |
| mod_utils_map_selections_ui . . . . .        | 50 |
| mod_utils_map_sync_server . . . . .          | 51 |
| model_fit_prep . . . . .                     | 51 |
| model_fit_table . . . . .                    | 52 |
| model_summary_prep . . . . .                 | 53 |
| model_summary_table . . . . .                | 53 |
| named_ids . . . . .                          | 54 |
| obs_map . . . . .                            | 54 |
| obs_map_points . . . . .                     | 55 |
| obs_map_raster . . . . .                     | 56 |
| obs_prep . . . . .                           | 57 |
| obs_prep_points . . . . .                    | 58 |
| obs_prep_raster . . . . .                    | 58 |
| parent_vals . . . . .                        | 59 |
| predictor_metadata_prep . . . . .            | 60 |
| predictor_metadata_table . . . . .           | 60 |
| predictor_raster_layer . . . . .             | 61 |
| predictor_raster_map . . . . .               | 61 |
| predictor_raster_prep . . . . .              | 62 |

|                                         |    |
|-----------------------------------------|----|
| prep_deployments . . . . .              | 63 |
| prep_evaluations . . . . .              | 63 |
| prep_files . . . . .                    | 64 |
| prep_material_settings . . . . .        | 64 |
| prep_materials . . . . .                | 65 |
| prep_questions . . . . .                | 66 |
| response_to_json . . . . .              | 67 |
| save_evaluations . . . . .              | 68 |
| sdm_spinner . . . . .                   | 69 |
| sdm_theme . . . . .                     | 69 |
| sdm_tool . . . . .                      | 70 |
| sdm_tooltip . . . . .                   | 71 |
| set_options . . . . .                   | 71 |
| show_spatial . . . . .                  | 72 |
| skip_if_no_data . . . . .               | 73 |
| spatial_prediction_map . . . . .        | 73 |
| spatial_prediction_prep . . . . .       | 74 |
| template_prep . . . . .                 | 74 |
| template_spatial_map . . . . .          | 75 |
| template_spatial_prep . . . . .         | 76 |
| template_table . . . . .                | 76 |
| test_comp . . . . .                     | 77 |
| test_comp_deployment_subunits . . . . . | 78 |
| test_comp_model_fit . . . . .           | 78 |
| test_comp_model_summary . . . . .       | 79 |
| test_comp_obs_chart . . . . .           | 79 |
| test_comp_observations . . . . .        | 80 |
| test_comp_predictor_metadata . . . . .  | 80 |
| test_comp_predictor_raster . . . . .    | 81 |
| test_comp_spatial_prediction . . . . .  | 81 |
| test_comp_template . . . . .            | 82 |
| test_comp_template_spatial . . . . .    | 82 |
| test_evaluation_body . . . . .          | 83 |
| test_input_evals . . . . .              | 83 |
| test_page . . . . .                     | 84 |
| test_points . . . . .                   | 85 |
| test_questions . . . . .                | 85 |
| test_raster . . . . .                   | 86 |
| tt_material_settings . . . . .          | 86 |
| ui_questions . . . . .                  | 87 |
| ui_questions_update . . . . .           | 88 |
| validate_ids . . . . .                  | 88 |
| value_to_input . . . . .                | 89 |

---

|             |                                  |
|-------------|----------------------------------|
| add_control | <i>Add Layers control to map</i> |
|-------------|----------------------------------|

---

## Description

Add Layers control to map

## Usage

```
add_control(map, groups = character(0))
```

## Arguments

|        |                                                                                                                                     |
|--------|-------------------------------------------------------------------------------------------------------------------------------------|
| map    | Leaflet map                                                                                                                         |
| groups | Character vector. Layers or Group names to add to the control. Subunits are included by default. Only existing layers are included. |

## Value

Leaflet map with layer control buttons

## Examples

```
# No subunits
r <- template_spatial_prep()
base_map() |>
  add_raster(r, layer = "mean", palette = "viridis", name = "Mean") |>
  add_control(groups = "Mean")

# With subunits
if(have_data()) {
  r <- template_spatial_prep()
  s <- deployment_subunits_prep("deployment1")
  base_map() |>
    add_raster(r, layer = "mean", palette = "viridis", name = "Mean") |>
    add_subunits(s) |>
    add_control(groups = "Mean")
}
```

---

|             |                                         |
|-------------|-----------------------------------------|
| add_markers | <i>Add point markers to Leaflet map</i> |
|-------------|-----------------------------------------|

---

### Description

Add point markers to Leaflet map

### Usage

```
add_markers(
  map,
  data = leaflet::getMapData(map),
  colour_by = "layers",
  layer_by = "layers"
)
```

### Arguments

|           |                                                                                            |
|-----------|--------------------------------------------------------------------------------------------|
| map       | Leaflet map object.                                                                        |
| data      | Spatial data with markers to plot (defaults to data in the map object).                    |
| colour_by | Character. Column to use when colouring points.                                            |
| layer_by  | Character. Column to use to group points into layers (for <a href="#">add_control()</a> ). |

### Value

Leaflet map.

### Examples

```
base_map() |>
  add_markers(data = test_points()) |>
  add_control(groups = c("present", "absent")) # From 'layers' column
```

---

|            |                                                 |
|------------|-------------------------------------------------|
| add_raster | <i>Add raster as image layer to leaflet map</i> |
|------------|-------------------------------------------------|

---

### Description

Note that raster layers are attached to a new map pane with a defined z-index. This is so we can be sure the raster is above the map tiles (z-index 200, see `base_map()`), and yet below the subunits (default z-index of 400).

**Usage**

```
add_raster(
  map,
  raster,
  layer,
  name,
  palette,
  opacity = 0.8,
  add_legend = TRUE,
  min_0 = TRUE
)
```

**Arguments**

|            |                                                        |
|------------|--------------------------------------------------------|
| map        | Leaflet map                                            |
| raster     | Terra raster to add                                    |
| layer      | Layer in raster to add                                 |
| name       | Name of leaflet layer (for controls and ids)           |
| palette    | Character. Colour palette to use                       |
| opacity    | Numeric. Opacity of raster layer                       |
| add_legend | Logical. Whether or not to add a legend.               |
| min_0      | Logical. Whether to force the legend to start at zero. |

**Value**

Leaflet map

**Examples**

```
r <- template_spatial_prep()
base_map() |>
  add_raster(r, layer = "mean", palette = "viridis", name = "Mean")
```

---

add\_selected\_markers    *Add selected point markers to Leaflet map*

---

**Description**

Very similar to [add\\_markers\(\)](#) but colours according to type column which defines the selections and no option to add to controls.

**Usage**

```
add_selected_markers(map, data = leaflet::getMapData(map), colour_by = "type")
```

**Arguments**

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| map       | Leaflet map object.                                                     |
| data      | Spatial data with markers to plot (defaults to data in the map object). |
| colour_by | Character. Column to use when colouring points.                         |

**Value**

Leaflet map.

**Examples**

```
base_map() |>
  add_selected_markers(data = test_points())
```

---

add\_selected\_subunits *Add selected subunits to Leaflet map*

---

**Description**

Similar to [add\\_subunits\(\)](#) but colours subunits by fill colour and selection type.

**Usage**

```
add_selected_subunits(
  map,
  subunits = NULL,
  colour_by = "type",
  opacity = 0.8,
  fill_opacity = 0.2
)
```

**Arguments**

|                       |                                                                                     |
|-----------------------|-------------------------------------------------------------------------------------|
| map                   | Leaflet map object.                                                                 |
| subunits              | Spatial Data Frame. Subunits                                                        |
| colour_by             | Vector. Column to fill by (must be a factor with levels to get the correct legend). |
| opacity, fill_opacity | Numeric. Opacity.                                                                   |

**Examples**

```
subunits <- deployment_subunits_prep("deployment1") |>
  dplyr::mutate(
    type = c(rep("Very biased", 10), rep("Moderately biased", 10), rep(NA, 114)),
    type = factor(type)
  )

base_map() |>
  add_selected_subunits(subunits)
```

---

|              |                                    |
|--------------|------------------------------------|
| add_subunits | <i>Add subunits to Leaflet map</i> |
|--------------|------------------------------------|

---

**Description**

Add subunits to Leaflet map

**Usage**

```
add_subunits(map, subunits = NULL, colour_by = "subunit_id", opacity = 0.8)
```

**Arguments**

|           |                              |
|-----------|------------------------------|
| map       | Leaflet map object.          |
| subunits  | Spatial Data Frame. Subunits |
| colour_by | Vector. Column to fill by    |
| opacity   | Numeric. Opacity.            |

**Examples**

```
subunits <- deployment_subunits_prep("deployment1")

base_map() |>
  add_subunits(subunits)
```

---

|             |                                              |
|-------------|----------------------------------------------|
| affirmative | <i>Evaluation answers in the affirmative</i> |
|-------------|----------------------------------------------|

---

**Description**

Used to determine which parts should be displayed

**Usage**

```
affirmative(type = "standard")
```

**Arguments**

type                      Character. "standard" (currently for ordinal inputs), or "spatial"

**Value**

Character vector of affirmative answers

**Examples**

```
affirmative()  
affirmative("spatial")
```

---

|               |                                    |
|---------------|------------------------------------|
| app_materials | <i>Get all materials available</i> |
|---------------|------------------------------------|

---

**Description**

Get all materials available

**Usage**

```
app_materials()
```

**Value**

Data frame of all materials available

**Examples**

```
app_materials()
```

---

`app_users`*Get Users details*

---

**Description**

Get Users details

**Usage**

```
app_users()
```

**Value**

Data frame of user details

**Examples**

```
app_users()
```

---

`base_map`*Base Map with Provider Tiles*

---

**Description**

Create the base Leaflet map with standard tiles, selection draw toolbar and erase selection button. Also includes and custom Javascripts, here one for marking which layers are visible for use in selections filtering (see [mod\\_utils\\_map\\_selections\\_server\(\)](#)).

**Usage**

```
base_map(ns = identity)
```

**Arguments**

`ns` Namespace. Required for spatial selections.

**Examples**

```
base_map()
```

**Description**

- `sdm_card()` - Wrapper for `bslib::card()`, provides defaults for `class`, `full_screen` and `min_height`.
- `sdm_card_header()` - Wrapper for `bslib::card_header()`, provides default custom class (see `sdm_theme()`).
- `sdm_nav_panel()` - Wrapper for `bslib::nav_panel()`, provides default custom class (see `sdm_theme()`).
- `sdm_layout_sidebar()` - Wrapper for `bslib::layout_sidebar()`, provides defaults for `gap` and `border`.

**Usage**

```
sdm_card(..., class = "p-0", full_screen = TRUE, min_height = 400)
```

```
sdm_card_header(..., class = "bg-sdm")
```

```
sdm_nav_panel(title, ..., class = "sdm-tab-pane")
```

```
sdm_layout_sidebar(..., gap = 0, border = FALSE)
```

**Arguments**

|                          |                                                    |
|--------------------------|----------------------------------------------------|
| <code>...</code>         | See bslib function                                 |
| <code>class</code>       | Character vector. HTML classes to add to the card. |
| <code>full_screen</code> | Logical. Option to make card full screen.          |
| <code>min_height</code>  | Numeric. Minimum card height.                      |
| <code>title</code>       | Character. Nav panel title.                        |
| <code>gap</code>         | Numeric. Vertical gap between sidebar elements.    |
| <code>border</code>      | Logical. Whether or not to add a border.           |

**Value**

bslib function output

---

|                  |                                    |
|------------------|------------------------------------|
| check_q_mismatch | <i>Compare number of questions</i> |
|------------------|------------------------------------|

---

**Description**

Compares the number of questions from those evaluated (q2) to those calculated from the question data (q1). Questions from evaluated (q2) may be NA if evaluations haven't been finished. Will error if values don't match (ignoring NAs).

**Usage**

```
check_q_mismatch(q1, q2)
```

**Arguments**

|    |                                                                                                    |
|----|----------------------------------------------------------------------------------------------------|
| q1 | Vector of Integers. Number of questions calculated from Question data (total questions available). |
| q2 | Vector of Integers. Number of questions evaluated.                                                 |

**Value**

invisible or error if there is a mismatch.

**Examples**

```
check_q_mismatch(c(1, 1, 2, 10, 6), c(1, 1, 2, 10, 6))
check_q_mismatch(c(10, 7, 1), c(10, NA, NA))
# check_q_mismatch(c(10, 11, 1), c(10, 10, NA)) # Error
```

---

|                |                                       |
|----------------|---------------------------------------|
| coords_to_poly | <i>Convert coordinates to polygon</i> |
|----------------|---------------------------------------|

---

**Description**

For use when drawing selections.

**Usage**

```
coords_to_poly(geometry)
```

**Arguments**

|          |                          |
|----------|--------------------------|
| geometry | From drawing selections. |
|----------|--------------------------|

**Value**

Sf polygon

---

|                  |                              |
|------------------|------------------------------|
| db_connect_check | <i>db connect with check</i> |
|------------------|------------------------------|

---

**Description**

Put the checks here rather than sdmEvalToolCore to avoid Shiny dependency in core package.

**Usage**

```
db_connect_check(dbname = NULL, ...)
```

**Arguments**

|        |                                                     |
|--------|-----------------------------------------------------|
| dbname | DB name.                                            |
| ...    | Arguments passed to <code>DBI::dbConnect()</code> . |

**Value**

A database connection.

---

|                         |                                         |
|-------------------------|-----------------------------------------|
| deployment_subunits_map | <i>Create a Leaflet Map of Subunits</i> |
|-------------------------|-----------------------------------------|

---

**Description**

Create a Leaflet Map of Subunits

**Usage**

```
deployment_subunits_map(subunits, ns = identity)
```

**Arguments**

|          |                                          |
|----------|------------------------------------------|
| subunits | Spatial Data frame. Deployment subunits. |
| ns       | Namespace.                               |

**Value**

A leaflet map object

**Examples**

```
s <- deployment_subunits_prep("deployment1")
deployment_subunits_map(s)
```

---

|                          |                                    |
|--------------------------|------------------------------------|
| deployment_subunits_prep | <i>Prepare Deployment Subunits</i> |
|--------------------------|------------------------------------|

---

**Description**

Subunits are used by multiple mapping components for a deployment.

**Usage**

deployment\_subunits\_prep(deployment\_id)

**Arguments**

deployment\_id    Character. Deployment ID

**Value**

Subunits polygons

**Examples**

deployment\_subunits\_prep(deployment\_id = "deployment1")

---

|                |                                               |
|----------------|-----------------------------------------------|
| evals_answered | <i>Calculate number of answered questions</i> |
|----------------|-----------------------------------------------|

---

**Description**

Summarizes evaluation data to count total questions and completed questions.

**Usage**

evals\_answered(eval)

**Arguments**

eval                      Data frame. Evaluation data with response column

**Details**

This is used by the 'overview'/'summary'/'index' page.

**Value**

Data frame with columns: n\_q (total questions), n\_q\_complete (completed questions)

**Examples**

```
eval_data <- evals_extract(test_evaluation_body())
evals_answered(eval_data)

eval_data <- evals_extract(test_evaluation_body(component_id = "model_fit_a"))
evals_answered(eval_data)

eval_data <- evals_extract(test_evaluation_body(component_id = "model_fit_b"))
evals_answered(eval_data)
```

---

evals\_details

*Get Evaluation Details for User*


---

**Description**

Retrieves evaluation details for a specific user based on their role. For modelers, shows all evaluations across users for their deployments. For evaluators, shows only their own evaluations. Returns a data frame with deployment, model, species, component, and completion information.

**Usage**

```
evals_details(user_id, user_role)
```

**Arguments**

|           |                                                        |
|-----------|--------------------------------------------------------|
| user_id   | Character string. User identifier                      |
| user_role | Character string. User role ("modeler" or "evaluator") |

**Details**

This is used by the 'overview'/'summary'/'index' page.

**Value**

Data frame with columns:

- deployment\_model\_name
- evaluation\_create\_user\_name
- deployment\_name
- model\_name
- species\_display
- component\_name
- completed
- started
- n\_q\_display
- n\_q
- n\_q\_complete

**Examples**

```
evals_details("testuser", "evaluator")
evals_details("testuser", "modeler")
```

---

|               |                                      |
|---------------|--------------------------------------|
| evals_extract | <i>Extract evaluations from JSON</i> |
|---------------|--------------------------------------|

---

**Description**

Parses JSON-formatted evaluation body into a data frame. Opposite of [response\\_to\\_json\(\)](#).

**Usage**

```
evals_extract(json)
```

**Arguments**

|      |                                           |
|------|-------------------------------------------|
| json | Character. JSON-formatted evaluation data |
|------|-------------------------------------------|

**Value**

Data frame with evaluation information

**Examples**

```
evals_extract(test_evaluation_body())
```

---

|            |                                                  |
|------------|--------------------------------------------------|
| evals_list | <i>Convert questions to input list structure</i> |
|------------|--------------------------------------------------|

---

**Description**

For comparing questions to Shiny input answers, convert the questions to the same input list structure as returned in the Shiny app.

**Usage**

```
evals_list(questions)
```

**Arguments**

|           |                                                                                |
|-----------|--------------------------------------------------------------------------------|
| questions | Data frame. Data frame of questions; output of <code>prep_questions()</code> . |
|-----------|--------------------------------------------------------------------------------|

**Value**

List which imitates a Shiny input object

**Examples**

```

q <- prep_questions(
  "observations",
  "deployment1",
  "bam_v5_can71",
  "BBWA",
  "testuser"
)
evals_list(q)

q <- prep_questions(
  "model_summary",
  "deployment1",
  "bam_v5_can71",
  "BBWA",
  "testuser"
)
evals_list(q)

```

---

evals\_table

---

*Create a Reactable Overview table*


---

**Description**

Creates reactable table displaying evaluation progress for either modelers or evaluators. The table shows deployment-model combinations, species, components, and their completion status.

**Usage**

```
evals_table(tbl, user_role)
```

**Arguments**

|           |                                                     |
|-----------|-----------------------------------------------------|
| tbl       | Data frame. Evaluation details from evals_details() |
| user_role | Character. User role ("modeler", or "evaluator")    |

**Value**

A reactable table object

**Examples**

```
tbl <- evals_details("testuser", "evaluator")
evals_table(tbl, "evaluator")
```

---

|             |                            |
|-------------|----------------------------|
| feature_ids | <i>Extract feature ids</i> |
|-------------|----------------------------|

---

**Description**

Extract feature ids

**Usage**

```
feature_ids(features)
```

**Arguments**

features            The input for the draw all features (input\${map}\_draw\_all\_features).

**Value**

Numeric vector of leaflet ids

---

|                 |                                                 |
|-----------------|-------------------------------------------------|
| fetch_questions | <i>Low-level function to read in questions.</i> |
|-----------------|-------------------------------------------------|

---

**Description**

If deployment doesn't exist, uses `sdmEvalToolCore::default_questions()`.

**Usage**

```
fetch_questions(deployment_id, component_id)
```

**Arguments**

deployment\_id    Character. Deployment ID

component\_id    Character. Component ID

**Value**

Data frame of questions.

**Examples**

```
# Use default questions if deployment doesn't exist
fetch_questions("No deployment", "observations")

if(have_data()) {
  fetch_questions("deployment1", "observations")
}
```

---

|             |                                     |
|-------------|-------------------------------------|
| fmt_species | <i>Format species display names</i> |
|-------------|-------------------------------------|

---

**Description**

Format species display names

**Usage**

```
fmt_species(df)
```

**Arguments**

|    |                                                                                                           |
|----|-----------------------------------------------------------------------------------------------------------|
| df | Data frame to format. Must contain at least species name in French or English as well as scientific name. |
|----|-----------------------------------------------------------------------------------------------------------|

**Value**

Formatted species display names.

---

|          |                                      |
|----------|--------------------------------------|
| fmt_time | <i>Format time nicely for humans</i> |
|----------|--------------------------------------|

---

**Description**

Format time nicely for humans

**Usage**

```
fmt_time(time)
```

**Arguments**

|      |              |
|------|--------------|
| time | POSIXct time |
|------|--------------|

**Value**

Character. Nicely formatted date

**Examples**

```
fmt_time(Sys.time())  
fmt_time(as.POSIXct("2026-02-01 16:02"))
```

---

|                       |                              |
|-----------------------|------------------------------|
| get_material_settings | <i>Get Material Settings</i> |
|-----------------------|------------------------------|

---

**Description**

Get Material Settings

**Usage**

```
get_material_settings(x)
```

**Arguments**

x                      Object.

**Value**

The value of the material settings attribute

---

|           |                         |
|-----------|-------------------------|
| have_data | <i>Have local data?</i> |
|-----------|-------------------------|

---

**Description**

Checks whether the user has local data available for this tool. Used for examples (must be exported).

**Usage**

```
have_data()
```

**Value**

TRUE/FALSE

**Examples**

```
have_data()
```

---

|                 |                                  |
|-----------------|----------------------------------|
| identical_loose | <i>Loosely test if identical</i> |
|-----------------|----------------------------------|

---

**Description**

Used in `mod_utils_evaluations_server()` to loosely determine if evaluations are identical between saved and inputs (to determine if the unsaved notice should be used).

**Usage**

```
identical_loose(i1, i2)
```

**Arguments**

|    |           |
|----|-----------|
| i1 | Character |
| i2 | Character |

**Value**

Logical

---

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| input_to_value | <i>Helper function to format Shiny Input names as Question Values</i> |
|----------------|-----------------------------------------------------------------------|

---

**Description**

Helper function to format Shiny Input names as Question Values

**Usage**

```
input_to_value(i)
```

**Arguments**

|   |                                 |
|---|---------------------------------|
| i | Character vector of Input names |
|---|---------------------------------|

**Value**

Character string of Shiny Inputs names formatted as Question values

**Examples**

```
input_to_value("Strongly_Agree")
```

---

|      |                                |
|------|--------------------------------|
| lang | <i>Return the app language</i> |
|------|--------------------------------|

---

**Description**

Return the app language

**Usage**

```
lang()
```

**Value**

Character. Either "english" or "french"

**Examples**

```
lang()
```

---

|               |                                       |
|---------------|---------------------------------------|
| map_has_group | <i>Check if a group exists in map</i> |
|---------------|---------------------------------------|

---

**Description**

Note: this only works for Leaflet maps, NOT Leaflet Proxies. For example, this is why `mod_comp_predictor_raster` family of functions are a bit convoluted when it calls to adding layers and controls (because `add_controls()` only adds layers which are detected with this function).

**Usage**

```
map_has_group(map, group)
```

**Arguments**

|       |                                        |
|-------|----------------------------------------|
| map   | Leaflet Map (not Leaflet Proxy).       |
| group | Character Vector. Groups to check for. |

**Details**

This may be slightly fragile if the list structure of the leaflet objects change.

**Value**

Logical. True if detected, false otherwise.

---

|                   |                                              |
|-------------------|----------------------------------------------|
| map_reactive_vals | <i>Create reactive vals and keep updated</i> |
|-------------------|----------------------------------------------|

---

**Description**

Tracks interactive inputs from map selections. Depending on the origin, namespaced if necessary.

**Usage**

```
map_reactive_vals(
  input,
  map,
  x = c("draw_all_features", "draw_new_feature", "draw_stop", "marker_click",
        "shape_click"),
  js_x = c("clear_selection", "layers_visible")
)
```

**Arguments**

|       |                                                                                                            |
|-------|------------------------------------------------------------------------------------------------------------|
| input | Shiny input object.                                                                                        |
| map   | Character. Map name.                                                                                       |
| x     | Character vector. Names of interactive inputs which are namespaced to the map, so must be namespaced here. |
| js_x  | Character vector. Names of Javascript created interactive inputs which do not need to be namespaced.       |

**Details**

Used at the component-level for those using spatial selections.

**Value**

A reactiveValues() list, as well as having the side effect of creating `shiny::observe()` for each value in the list to be independently updated.

---

|                                     |                                             |
|-------------------------------------|---------------------------------------------|
| mod_comp_deployment_subunits_server | <i>Deployment Subunits component Server</i> |
|-------------------------------------|---------------------------------------------|

---

**Description**

Deployment Subunits component Server

Usage

```
mod_comp_deployment_subunits_server(  
  id = "comp_summary",  
  parent_id,  
  deployment_id,  
  model_id,  
  species_id,  
  spatial_selection,  
  spatial_ids,  
  map_views  
)
```

Arguments

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| id                | Module ID                                                                            |
| parent_id         | Parent tab ID (used to identify which tab is active)                                 |
| deployment_id     | Deployment ID                                                                        |
| model_id          | Model ID                                                                             |
| species_id        | Species ID                                                                           |
| spatial_selection | Spatial selection                                                                    |
| spatial_ids       | Spatial IDs                                                                          |
| map_views         | List. List of reactiveVals active_tab, set_by and view (list with zoom and lat/lon). |

Value

Module server function

---

|                                         |
|-----------------------------------------|
| mod_comp_deployment_subunits_ui         |
| <i>Deployment Subunits Component UI</i> |

---

Description

Deployment Subunits Component UI

Usage

```
mod_comp_deployment_subunits_ui(id = "comp_summary", header = "Comfort zones")
```

Arguments

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_deployment_subunits_ui()
```

---

|                                   |
|-----------------------------------|
| mod_comp_model_fit_server         |
| <i>Model Fit Component Server</i> |

---

**Description**

Model Fit Component Server

**Usage**

```
mod_comp_model_fit_server(id = "comp_model_fit", model_id, species_id)
```

**Arguments**

|            |            |
|------------|------------|
| id         | Module ID  |
| model_id   | Model ID   |
| species_id | Species ID |

**Value**

Module server function

---

|                       |                               |
|-----------------------|-------------------------------|
| mod_comp_model_fit_ui | <i>Model Fit Component UI</i> |
|-----------------------|-------------------------------|

---

**Description**

Model Fit Component UI

**Usage**

```
mod_comp_model_fit_ui(id = "comp_model_fit", header = "Model Fit")
```

**Arguments**

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_model_fit_ui()
```

---

```
mod_comp_model_summary_server
```

*Model Summary component Server*

---

**Description**

Model Summary component Server

**Usage**

```
mod_comp_model_summary_server(id = "comp_model_summary", model_id, species_id)
```

**Arguments**

|            |            |
|------------|------------|
| id         | Module ID  |
| model_id   | Model ID   |
| species_id | Species ID |

**Value**

Module server function

---

```
mod_comp_model_summary_ui
```

*Model Summary Component UI*

---

**Description**

Model Summary Component UI

**Usage**

```
mod_comp_model_summary_ui(id = "comp_model_summary", header = "Model Summary")
```

**Arguments**

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_model_summary_ui()
```

---

|                                            |
|--------------------------------------------|
| mod_comp_obs_chart_server                  |
| <i>Observations Chart Component Server</i> |

---

**Description**

Observations Chart Component Server

**Usage**

```
mod_comp_obs_chart_server(id = "comp_obs_chart", model_id, species_id)
```

**Arguments**

|            |            |
|------------|------------|
| id         | Module ID  |
| model_id   | Model ID   |
| species_id | Species ID |

**Value**

Module server function

---

|                       |                                        |
|-----------------------|----------------------------------------|
| mod_comp_obs_chart_ui | <i>Observations Chart Component UI</i> |
|-----------------------|----------------------------------------|

---

**Description**

Observations Chart Component UI

**Usage**

```
mod_comp_obs_chart_ui(id = "comp_obs_chart", header = "Observation Charts")
```

**Arguments**

|        |                            |
|--------|----------------------------|
| id     | Character. Shiny module ID |
| header | Header                     |

**Value**

Shiny UI

**Examples**

```
mod_comp_obs_chart_ui()
```

---

|                                      |
|--------------------------------------|
| mod_comp_observations_server         |
| <i>Observations component Server</i> |

---

**Description**

Observations component Server

**Usage**

```
mod_comp_observations_server(  
  id = "comp_observations",  
  parent_id,  
  deployment_id,  
  model_id,  
  species_id,  
  spatial_selection,  
  spatial_ids,  
  map_views  
)
```

**Arguments**

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| id                | Module ID                                                                            |
| parent_id         | Parent tab ID (used to identify which tab is active)                                 |
| deployment_id     | Deployment ID. Required for subunits.                                                |
| model_id          | Model ID                                                                             |
| species_id        | Species ID                                                                           |
| spatial_selection | Spatial selection                                                                    |
| spatial_ids       | Spatial IDs                                                                          |
| map_views         | List. List of reactiveVals active_tab, set_by and view (list with zoom and lat/lon). |

**Value**

Module server function

---

`mod_comp_observations_ui`*Observations Component UI*

---

**Description**

Observations Component UI

**Usage**

```
mod_comp_observations_ui(  
  id = "comp_observations",  
  header = "Map of Observations"  
)
```

**Arguments**

|                     |                            |
|---------------------|----------------------------|
| <code>id</code>     | Character. Shiny module ID |
| <code>header</code> | Header                     |

**Value**

Shiny UI

**Examples**

```
mod_comp_observations_ui()
```

---

`mod_comp_predictor_metadata_server`*Predictor Metadata component Server*

---

**Description**

Predictor Metadata component Server

**Usage**

```
mod_comp_predictor_metadata_server(id = "comp_predictor_metadata", model_id)
```

**Arguments**

|                       |           |
|-----------------------|-----------|
| <code>id</code>       | Module ID |
| <code>model_id</code> | Model ID  |

**Value**

Module server function

---

`mod_comp_predictor_metadata_ui`*Predictor Metadata Component UI*

---

**Description**

Predictor Metadata Component UI

**Usage**

```
mod_comp_predictor_metadata_ui(  
  id = "comp_predictor_metadata",  
  height = "40%",  
  header = "Predictor Metadata"  
)
```

**Arguments**

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| height | Height          |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_predictor_metadata_ui()
```

---

`mod_comp_predictor_raster_server`*Predictor Raster component Server*

---

**Description**

Predictor Raster component Server

**Usage**

```
mod_comp_predictor_raster_server(  
  id = "comp_predictor_raster",  
  parent_id,  
  deployment_id,  
  model_id,  
  map_views  
)
```

**Arguments**

|               |                                                                                      |
|---------------|--------------------------------------------------------------------------------------|
| id            | Module ID                                                                            |
| parent_id     | Parent tab ID (used to identify which tab is active)                                 |
| deployment_id | Deployment ID. Required for subunits.                                                |
| model_id      | Model ID                                                                             |
| map_views     | List. List of reactiveVals active_tab, set_by and view (list with zoom and lat/lon). |

**Value**

Module server function

---

|                              |                                      |
|------------------------------|--------------------------------------|
| mod_comp_predictor_raster_ui | <i>Predictor Raster Component UI</i> |
|------------------------------|--------------------------------------|

---

**Description**

Predictor Raster Component UI

**Usage**

```
mod_comp_predictor_raster_ui(
  id = "comp_predictor_raster",
  height = NULL,
  header = "Predictor Raster"
)
```

**Arguments**

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| height | Height          |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_predictor_raster_ui()
```

---

|                                            |
|--------------------------------------------|
| mod_comp_spatial_prediction_server         |
| <i>Spatial Prediction component Server</i> |

---

**Description**

Spatial Prediction component Server

**Usage**

```
mod_comp_spatial_prediction_server(  
  id = "comp_spatial_prediction",  
  parent_id,  
  deployment_id,  
  model_id,  
  species_id,  
  spatial_selection,  
  spatial_ids,  
  map_views  
)
```

**Arguments**

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| id                | Module ID                                                                            |
| parent_id         | Parent tab ID (used to identify which tab is active)                                 |
| deployment_id     | Deployment ID. Required for subunits.                                                |
| model_id          | Model ID                                                                             |
| species_id        | Species ID                                                                           |
| spatial_selection | Spatial selection. Required for spatial evaluations                                  |
| spatial_ids       | Spatial IDs. Required for spatial evaluations.                                       |
| map_views         | List. List of reactiveVals active_tab, set_by and view (list with zoom and lat/lon). |

**Value**

Module server function

---

mod\_comp\_spatial\_prediction\_ui  
*Spatial Prediction Component UI*

---

**Description**

Spatial Prediction Component UI

**Usage**

```
mod_comp_spatial_prediction_ui(  
  id = "comp_spatial_prediction",  
  header = "Predictions"  
)
```

**Arguments**

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_spatial_prediction_ui()
```

---

mod\_comp\_template\_server  
*Template component Server*

---

**Description**

Template component Server

**Usage**

```
mod_comp_template_server(id = "comp_template", model_id, species_id)
```

**Arguments**

|            |            |
|------------|------------|
| id         | Module ID  |
| model_id   | Model ID   |
| species_id | Species ID |

**Value**

Module server function

---

|                                          |
|------------------------------------------|
| mod_comp_template_spatial_server         |
| <i>Template Spatial component Server</i> |

---

**Description**

Template Spatial component Server

**Usage**

```
mod_comp_template_spatial_server(  
  id = "comp_template_spatial",  
  parent_id,  
  deployment_id,  
  model_id,  
  species_id,  
  spatial_selection,  
  spatial_ids,  
  map_views  
)
```

**Arguments**

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| id                | Module ID                                                                            |
| parent_id         | ID of the parent module, corresponds to the tab.                                     |
| deployment_id     | Deployment ID. Required for subunits.                                                |
| model_id          | Model ID                                                                             |
| species_id        | Species ID                                                                           |
| spatial_selection | Spatial selection. Required for spatial evaluations.                                 |
| spatial_ids       | Spatial IDs. Required for spatial evaluations.                                       |
| map_views         | List. List of reactiveVals active_tab, set_by and view (list with zoom and lat/lon). |

**Value**

Module server function

---

|                                      |
|--------------------------------------|
| mod_comp_template_spatial_ui         |
| <i>Template Spatial Component UI</i> |

---

**Description**

Template Spatial Component UI

**Usage**

```
mod_comp_template_spatial_ui(  
  id = "comp_template_spatial",  
  header = "Template Spatial"  
)
```

**Arguments**

|        |                            |
|--------|----------------------------|
| id     | Character. Shiny module ID |
| header | Header                     |

**Value**

Shiny UI

**Examples**

```
mod_comp_template_spatial_ui()
```

---

|                      |                              |
|----------------------|------------------------------|
| mod_comp_template_ui | <i>Template Component UI</i> |
|----------------------|------------------------------|

---

**Description**

Template Component UI

**Usage**

```
mod_comp_template_ui(id = "comp_template", header = "Template")
```

**Arguments**

|        |                 |
|--------|-----------------|
| id     | Shiny module ID |
| header | Header          |

**Value**

Shiny UI

**Examples**

```
mod_comp_template_ui()
```

---

mod\_page\_model\_server *Model Page Server*

---

**Description**

Model Page Server

**Usage**

```
mod_page_model_server(id = "model", ...)
```

**Arguments**

|     |                                                                                                                                                                                                                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID                                                                                                                                                                                                                                                                                         |
| ... | Additional arguments passed via <code>expand_dots</code> including <code>deployment_id</code> , <code>model_id</code> , <code>species_id</code> , <code>abandoned</code> , <code>unsaved</code> , <code>opts</code> list (see <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

mod\_page\_model\_ui *Model Page UI*

---

**Description**

Model Page UI

**Usage**

```
mod_page_model_ui(id, title, review_width = NULL)
```

**Arguments**

|              |                 |
|--------------|-----------------|
| id           | Shiny module ID |
| title        | Page title      |
| review_width | Review width    |

**Value**

Shiny UI

**Examples**

```
mod_page_model_ui("id", "title")
```

---

```
mod_page_observations_server
```

*Observations Page Server*

---

**Description**

Observations Page Server

**Usage**

```
mod_page_observations_server(id = "observations", ...)
```

**Arguments**

|                  |                                                                                                                                                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>id</code>  | Shiny module ID                                                                                                                                                                                                                                                                                                              |
| <code>...</code> | Additional arguments passed via <code>expand_dots</code> including <code>deployment_id</code> , <code>model_id</code> , <code>species_id</code> , <code>abandoned</code> , <code>unsaved</code> , <code>opts</code> , <code>map_views</code> . See <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

```
mod_page_observations_ui
```

*Observations Page UI*

---

**Description**

Observations Page UI

**Usage**

```
mod_page_observations_ui(id, title, review_width = NULL)
```

**Arguments**

|              |                                                                    |
|--------------|--------------------------------------------------------------------|
| id           | Shiny module ID                                                    |
| title        | Page title                                                         |
| review_width | Character. Width of the review sidebar in percentage of the screen |

**Value**

Shiny UI

**Examples**

```
mod_page_observations_ui("id", "title")
```

---

```
mod_page_overview_server
```

*Overview Page Server*

---

**Description**

Overview Page Server

**Usage**

```
mod_page_overview_server(id = "overview", ...)
```

**Arguments**

|     |                                                                                                                                                                                                                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID                                                                                                                                                                                                                                                                                         |
| ... | Additional arguments passed via <code>expand_dots</code> including <code>deployment_id</code> , <code>model_id</code> , <code>species_id</code> , <code>abandoned</code> , <code>unsaved</code> , <code>opts</code> list (see <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

|                      |                         |
|----------------------|-------------------------|
| mod_page_overview_ui | <i>Overview Page UI</i> |
|----------------------|-------------------------|

---

**Description**

Overview Page UI

**Usage**

```
mod_page_overview_ui(id, title)
```

**Arguments**

|       |                 |
|-------|-----------------|
| id    | Shiny module ID |
| title | Page title      |

**Value**

Shiny UI

**Examples**

```
mod_page_overview_ui("id", "title")
```

---

|                             |                                |
|-----------------------------|--------------------------------|
| mod_page_predictions_server | <i>Predictions Page Server</i> |
|-----------------------------|--------------------------------|

---

**Description**

Predictions Page Server

**Usage**

```
mod_page_predictions_server(id = "predictions", ...)
```

**Arguments**

|     |                                                                                                                                                                                                                                                                                                                              |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID                                                                                                                                                                                                                                                                                                              |
| ... | Additional arguments passed via <code>expand_dots</code> including <code>deployment_id</code> , <code>model_id</code> , <code>species_id</code> , <code>abandoned</code> , <code>unsaved</code> , <code>opts</code> , <code>map_views</code> . See <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

`mod_page_predictions_ui`*Predictions Page UI*

---

**Description**

Predictions Page UI

**Usage**

```
mod_page_predictions_ui(id, title, review_width = NULL)
```

**Arguments**

|                           |                 |
|---------------------------|-----------------|
| <code>id</code>           | Shiny module ID |
| <code>title</code>        | Page title      |
| <code>review_width</code> | Review width    |

**Value**

Shiny UI

**Examples**

```
mod_page_predictions_ui("id", "title")
```

---

`mod_page_predictors_server`*Predictors Page Server*

---

**Description**

Predictors Page Server

**Usage**

```
mod_page_predictors_server(id = "predictors", ...)
```

**Arguments**

|                  |                                                                                                                                                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>id</code>  | Shiny module ID                                                                                                                                                                                                                                                                                                              |
| <code>...</code> | Additional arguments passed via <code>expand_dots</code> including <code>deployment_id</code> , <code>model_id</code> , <code>species_id</code> , <code>abandoned</code> , <code>unsaved</code> , <code>opts</code> , <code>map_views</code> . See <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

|                           |
|---------------------------|
| mod_page_predictors_ui    |
| <i>Predictors Page UI</i> |

---

**Description**

Predictors Page UI

**Usage**

```
mod_page_predictors_ui(id, title, review_width = NULL)
```

**Arguments**

|              |                 |
|--------------|-----------------|
| id           | Shiny module ID |
| title        | Page title      |
| review_width | Review width    |

**Value**

Shiny UI

**Examples**

```
mod_page_predictors_ui("id", "title")
```

---

|                            |
|----------------------------|
| mod_page_summary_server    |
| <i>Summary Page Server</i> |

---

**Description**

Summary Page Server

**Usage**

```
mod_page_summary_server(id = "summary", ...)
```

**Arguments**

|     |                                                                                                                                                                                                               |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID                                                                                                                                                                                               |
| ... | Additional arguments passed via expand_dots including deployment_id, model_id, species_id, abandoned, unsaved, opts, map_views. See <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

|                     |                        |
|---------------------|------------------------|
| mod_page_summary_ui | <i>Summary Page UI</i> |
|---------------------|------------------------|

---

**Description**

Summary Page UI

**Usage**

```
mod_page_summary_ui(id, title, review_width = NULL)
```

**Arguments**

|              |                 |
|--------------|-----------------|
| id           | Shiny module ID |
| title        | Page title      |
| review_width | Review width    |

**Value**

Shiny UI

**Examples**

```
mod_page_summary_ui("id", "title")
```

---

mod\_page\_template\_server

*Template Page Server*


---

### Description

Template Page Server

### Usage

```
mod_page_template_server(id = "template", ...)
```

### Arguments

|     |                                                                                                                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID                                                                                                                                                                                         |
| ... | Additional arguments passed via expand_dots including deployment_id, model_id, species_id, abandoned, unsaved, opts list (see <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

### Value

Server function for Shiny module

---

mod\_page\_template\_spatial\_server

*Template Page Server*


---

### Description

Template Page Server

### Usage

```
mod_page_template_spatial_server(id = "template_spatial", ...)
```

### Arguments

|     |                                                                                                                                                                                                               |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID. NOTE for pages this should match a value in the tabs argument of <a href="#">sdm_tool()</a> .                                                                                                |
| ... | Additional arguments passed via expand_dots including deployment_id, model_id, species_id, abandoned, unsaved, opts, map_views. See <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

### Value

Server function for Shiny module

---

|                                 |
|---------------------------------|
| mod_page_template_spatial_ui    |
| <i>Template Spatial Page UI</i> |

---

**Description**

Template Spatial Page UI

**Usage**

```
mod_page_template_spatial_ui(id, title, review_width = NULL)
```

**Arguments**

|              |                 |
|--------------|-----------------|
| id           | Shiny module ID |
| title        | Page title      |
| review_width | Review width    |

**Value**

Shiny UI

**Examples**

```
mod_page_template_spatial_ui("id", "title")
```

---

|                      |                         |
|----------------------|-------------------------|
| mod_page_template_ui | <i>Template Page UI</i> |
|----------------------|-------------------------|

---

**Description**

Template Page UI

**Usage**

```
mod_page_template_ui(id, title, review_width = NULL)
```

**Arguments**

|              |                 |
|--------------|-----------------|
| id           | Shiny module ID |
| title        | Page title      |
| review_width | Review width    |

**Value**

Shiny UI

**Examples**

```
mod_page_template_ui("id", "title")
```

---

|                      |                         |
|----------------------|-------------------------|
| mod_page_test_server | <i>Test Page Server</i> |
|----------------------|-------------------------|

---

**Description**

Test Page Server

**Usage**

```
mod_page_test_server(id = "test", ...)
```

**Arguments**

|     |                                                                                                                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id  | Shiny module ID                                                                                                                                                                                         |
| ... | Additional arguments passed via expand_dots including deployment_id, model_id, species_id, abandoned, unsaved, opts list (see <a href="#">sdm_tool()</a> which programmatically calls all module pages. |

**Value**

Server function for Shiny module

---

|                  |                     |
|------------------|---------------------|
| mod_page_test_ui | <i>Test Page UI</i> |
|------------------|---------------------|

---

**Description**

Test Page UI

**Usage**

```
mod_page_test_ui(id, review_width = "50%")
```

**Arguments**

|              |                                                       |
|--------------|-------------------------------------------------------|
| id           | Shiny module ID                                       |
| review_width | Character. Width of the review sidebar in percentage. |

**Value**

Shiny UI

**Examples**

```
mod_page_test_ui("id")
```

---

`mod_utils_details_server`*Utility Module 'Details' Server*

---

**Description**

Used in `sdm_tool()` to create the Deployment Details left-hand sidebar.

**Usage**

```
mod_utils_details_server(id = "details", details)
```

**Arguments**

|                      |                                   |
|----------------------|-----------------------------------|
| <code>id</code>      | Shiny module ID                   |
| <code>details</code> | Data frame of details to display. |

**Value**

Server function for Shiny module

---

`mod_utils_details_ui` *Utility Module 'Details' UI*

---

**Description**

Used in `sdm_tool()` to create the Deployment Details left-hand sidebar.

**Usage**

```
mod_utils_details_ui(id = "details")
```

**Arguments**

|                 |                 |
|-----------------|-----------------|
| <code>id</code> | Shiny module ID |
|-----------------|-----------------|

**Value**

Shiny UI

**Examples**

```
mod_utils_details_ui()
```

---

`mod_utils_evaluations_server`*Utility Module 'Evaluations' Server*

---

## Description

Utility Module 'Evaluations' Server

## Usage

```
mod_utils_evaluations_server(  
  id = "evaluations",  
  component_id,  
  deployment_id,  
  model_id,  
  species_id,  
  spatial_ids = reactive(NULL),  
  opts,  
  abandoned,  
  unsaved  
)
```

## Arguments

|                            |                                                                                              |
|----------------------------|----------------------------------------------------------------------------------------------|
| <code>id</code>            | Shiny module ID                                                                              |
| <code>component_id</code>  | Character. Component ID                                                                      |
| <code>deployment_id</code> | Character. Deployment ID                                                                     |
| <code>model_id</code>      | Character. Model ID                                                                          |
| <code>species_id</code>    | Character. Species ID                                                                        |
| <code>spatial_ids</code>   | Reactive Character vector. Spatial Ids to use when populating spatial evaluation selectizes. |
| <code>opts</code>          | List. Options list of reactives, originates from <code>sdm_tool()</code> .                   |
| <code>abandoned</code>     | ReactiveVal indicating whether the evaluation has been abandoned.                            |
| <code>unsaved</code>       | ReactiveVal indicating whether any questions are unsaved.                                    |

## Value

Server function for Shiny module

---

`mod_utils_evaluations_ui`*Utility Module 'Evaluations' UI*

---

**Description**

Called by page-level modules to create the Evaluations UI in the right-hand side bar.

**Usage**

```
mod_utils_evaluations_ui(  
  id = "evaluations",  
  review_width = NULL,  
  level = "species"  
)
```

**Arguments**

|                           |                                                        |
|---------------------------|--------------------------------------------------------|
| <code>id</code>           | Shiny module ID                                        |
| <code>review_width</code> | Character. Starting width of the sidebar.              |
| <code>level</code>        | Character. Level of evaluations, "model" or "species". |

**Value**

Shiny UI

---

`mod_utils_map_selections_server`*Utility Module 'Map Selections' Server*

---

**Description**

Highlights spatial ids selected on the map, either through user selection or through the 'show' button from evaluations.

**Usage**

```
mod_utils_map_selections_server(  
  id,  
  data,  
  spatial_selection,  
  interactions,  
  parent_session = getDefaultReactiveDomain()  
)
```

**Arguments**

|                   |                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id                | Character. Shiny module ID                                                                                                                                              |
| data              | Spatial Data frame. Data from which selections will be made.                                                                                                            |
| spatial_selection | List of show_clicked reactiveVal which indicates which show button was last clicked and show_spatial_ids which contains a list of spatial ids by category to highlight. |
| interactions      | Interactions                                                                                                                                                            |
| parent_session    | Parent session                                                                                                                                                          |

**Value**

Shiny server

---

mod\_utils\_map\_selections\_ui

*Utility Module 'Map Selections' UI*

---

**Description**

Highlights spatial ids selected, either through user selection or through the 'show' button from evaluations. Called by component-level modules.

**Usage**

```
mod_utils_map_selections_ui(id)
```

**Arguments**

|    |                             |
|----|-----------------------------|
| id | Character. Shiny module ID. |
|----|-----------------------------|

**Value**

Shiny UI

---

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| mod_utils_map_sync_server | <i>Utility Module 'Map Synchronization' Server</i> |
|---------------------------|----------------------------------------------------|

---

**Description**

Synchronizes the map view port among tabs

**Usage**

```
mod_utils_map_sync_server(  
  id,  
  parent_id,  
  this_view,  
  map_views,  
  parent_session = getDefaultReactiveDomain()  
)
```

**Arguments**

- id                      Character. Shiny module ID
- parent\_id             Character. Parent tab ID (used to identify which tab is active)
- this\_view             Reactive. Reactive that returns the current map view (list with zoom and lat/lon)
- map\_views            List. List of reactiveVals active\_tab, set\_by and view (list with zoom and lat/lon).
- parent\_session       Shiny session. Parent session (used for leafletProxy)

**Value**

Shiny server

---

|                |                               |
|----------------|-------------------------------|
| model_fit_prep | <i>Prepare Model Fit Data</i> |
|----------------|-------------------------------|

---

**Description**

Prepare Model Fit Data

**Usage**

```
model_fit_prep(model_id, species_id)
```

**Arguments**

- model\_id            Character. Model ID
- species\_id        Character. Species ID

**Value**

Data frame

**Examples**

```
model_fit_prep(model_ = "bam_v5_can71", species_id = "BBW0")
```

---

|                 |                                   |
|-----------------|-----------------------------------|
| model_fit_table | Create a Table for Model Fit Data |
|-----------------|-----------------------------------|

---

**Description**

Create a Table for Model Fit Data

**Usage**

```
model_fit_table(model_fit)
```

**Arguments**

- model\_fit            Data frame with model fit information

**Value**

Reactable table

**Examples**

```
model_fit_prep(model_id = "bam_v5_can71", species_id = "BBW0") |>  
  model_fit_table()
```

---

|                    |                                   |
|--------------------|-----------------------------------|
| model_summary_prep | <i>Prepare Model Summary Data</i> |
|--------------------|-----------------------------------|

---

**Description**

Prepare Model Summary Data

**Usage**

```
model_summary_prep(model_id, species_id)
```

**Arguments**

|            |                       |
|------------|-----------------------|
| model_id   | Character. Model ID   |
| species_id | Character. Species ID |

**Value**

Data frame

**Examples**

```
model_summary_prep(model_id = "bam_v5_can71", species_id = "BBW0")
```

---

|                     |                                              |
|---------------------|----------------------------------------------|
| model_summary_table | <i>Create a Table for Model Summary Data</i> |
|---------------------|----------------------------------------------|

---

**Description**

Create a Table for Model Summary Data

**Usage**

```
model_summary_table(model_summary)
```

**Arguments**

|               |                                       |
|---------------|---------------------------------------|
| model_summary | Data frame. Model summary information |
|---------------|---------------------------------------|

**Value**

A reactable table object

**Examples**

```
model_summary_prep(model_id = "bam_v5_can71", species_id = "BBW0") |>  
  model_summary_table()
```

---

|           |                                                       |
|-----------|-------------------------------------------------------|
| named_ids | <i>Create named lists of ids for input selections</i> |
|-----------|-------------------------------------------------------|

---

### Description

Lists are `c(named = id)` to create a input selection with 'pretty' selections returning original ids.

### Usage

```
named_ids(df_db, id = "id", name = "name", match = NULL)
```

### Arguments

|                    |                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <code>df_db</code> | Data frame or database where ID values can be found.                                                                          |
| <code>id</code>    | Character. Pattern matching 'id' in the column name.                                                                          |
| <code>name</code>  | Character. Pattern matching 'name' in the column name.                                                                        |
| <code>match</code> | Character. ID/Name preface. Optional instead of id/name, useful if there are more than one pair of id/name in the data frame. |

### Value

Named character vector.

### Examples

```
df <- data.frame(
  species_id = c("X11", "Y22"),
  species_name = c("Pretty name", "Lovely name")
)

named_ids(df)
named_ids(df, match = "species")
```

---

|         |                                                 |
|---------|-------------------------------------------------|
| obs_map | <i>Create a Leaflet Map of Observation Data</i> |
|---------|-------------------------------------------------|

---

### Description

Can use points or rasters to map the data.

**Usage**

```
obs_map(
  obs,
  subunits = NULL,
  ns = identity,
  output_type = c("points", "raster"),
  ...
)
```

**Arguments**

|             |                                                                                                                                       |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|
| obs         | sf or Terra spatial data. Observations as points ( <a href="#">obs_prep()</a> ) or raster data ( <a href="#">obs_prep_raster()</a> ). |
| subunits    | Subunits                                                                                                                              |
| ns          | Namespace                                                                                                                             |
| output_type | Character. Points or raster.                                                                                                          |
| ...         | Other parameters.                                                                                                                     |

**Value**

A leaflet map object

**Examples**

```
o <- obs_prep(model_id = "bam_v5_can71", species_id = "BBWO")
s <- deployment_subunits_prep("deployment1")
r0 <- prep_materials(
  "spatial_prediction",
  model_id = "bam_v5_can71",
  species_id = "BBWO"
)
r <- obs_prep_raster(o, r0)

obs_map(o, s) # points
obs_map(r, s, output_type = "raster") # raster
```

---

|                |                                                       |
|----------------|-------------------------------------------------------|
| obs_map_points | <i>Create a Leaflet Map of Observation Point Data</i> |
|----------------|-------------------------------------------------------|

---

**Description**

Create a Leaflet Map of Observation Point Data

**Usage**

```
obs_map_points(obs, subunits = NULL, ns = identity, ...)
```

**Arguments**

|          |                             |
|----------|-----------------------------|
| obs      | sf data frame. Observations |
| subunits | Subunits                    |
| ns       | Namespace                   |
| ...      | Other parameters.           |

**Value**

A leaflet map object

**Examples**

```
p <- test_points()
obs_map_points(p)
```

---

|                |                                                        |
|----------------|--------------------------------------------------------|
| obs_map_raster | <i>Create a Leaflet Map of Observation Raster Data</i> |
|----------------|--------------------------------------------------------|

---

**Description**

Create a Leaflet Map of Observation Raster Data

**Usage**

```
obs_map_raster(obs, subunits = NULL, ns = identity, ...)
```

**Arguments**

|          |                             |
|----------|-----------------------------|
| obs      | sf data frame. Observations |
| subunits | Subunits                    |
| ns       | Namespace                   |
| ...      | Other parameters.           |

**Value**

A leaflet map object

**Examples**

```
o <- obs_prep(model_id = "bam_v5_can71", species_id = "BBWO")
r <- prep_materials(
  "spatial_prediction",
  model_id = "bam_v5_can71",
  species_id = "BBWO"
)
p <- obs_prep_raster(o, r)

obs_map_raster(p)
```

---

|          |                                 |
|----------|---------------------------------|
| obs_prep | <i>Prepare Observation Data</i> |
|----------|---------------------------------|

---

## Description

Loads and generally prepares the observation data. Wrapper around `obs_prep_points()` and `obs_prep_raster()`.

## Usage

```
obs_prep(model_id, species_id, output_type = c("points", "raster"), ...)
```

## Arguments

|                          |                                                                                                          |
|--------------------------|----------------------------------------------------------------------------------------------------------|
| <code>model_id</code>    | Character. Model ID                                                                                      |
| <code>species_id</code>  | Character. Species ID                                                                                    |
| <code>output_type</code> | Character. Points or raster.                                                                             |
| <code>...</code>         | Other parameters. Most importantly, <code>rast</code> argument if <code>output_type == "raster"</code> . |

## Value

Spatial data frame, sf if points, terra SpatRaster if raster.

## Examples

```
r0 <- prep_materials(
  "spatial_prediction",
  model_id = "bam_v5_can71",
  species_id = "BBW0"
)

# Points Prep
obs_prep(model_id = "bam_v5_can71", species_id = "BBW0")

# Raster Prep
obs_prep(model_id = "bam_v5_can71", species_id = "BBW0", "raster", rast = r0)
```

---

|                 |                                                   |
|-----------------|---------------------------------------------------|
| obs_prep_points | <i>Prepare points from Observation point data</i> |
|-----------------|---------------------------------------------------|

---

**Description**

Prepare points from Observation point data

**Usage**

```
obs_prep_points(obs, ...)
```

**Arguments**

|     |                       |
|-----|-----------------------|
| obs | Observation materials |
| ... | Unused.               |

**Value**

Sf spatial data frame

**Examples**

```
o <- prep_materials(
  "observations",
  model_id = "bam_v5_can71",
  species_id = "BBWO"
)
p <- obs_prep_points(o)
```

---

|                 |                                                   |
|-----------------|---------------------------------------------------|
| obs_prep_raster | <i>Prepare raster from Observation point data</i> |
|-----------------|---------------------------------------------------|

---

**Description**

Prepare raster from Observation point data

**Usage**

```
obs_prep_raster(obs, rast, scale = 10, ...)
```

**Arguments**

|       |                                                   |
|-------|---------------------------------------------------|
| obs   | Observation materials                             |
| rast  | Spatial predictions raster to get the extent etc. |
| scale | Raster scale.                                     |
| ...   | Unused.                                           |

**Value**

SpatRaster

**Examples**

```
o <- prep_materials(
  "observations",
  model_id = "bam_v5_can71",
  species_id = "BBWO"
)
r <- prep_materials(
  "spatial_prediction",
  model_id = "bam_v5_can71",
  species_id = "BBWO"
)
p <- obs_prep_raster(o, r)
```

---

parent\_vals

*Create vector of values to use downstream*

---

**Description**

Creates a vector of Spatial values to use downstream when referencing parent values in followup questions.

**Usage**

```
parent_vals(v)
```

**Arguments**

**v** Character vector of Questions values.

**Value**

Character vector of Question values formatted for Shiny Inputs, including only spatial values.

**Examples**

```
parent_vals(c("Slightly agree", "Very biased", "Moderately biased"))
```

---

|                                        |
|----------------------------------------|
| <code>predictor_metadata_prep</code>   |
| <i>Prepare Predictor Metadata Data</i> |

---

**Description**

Prepare Predictor Metadata Data

**Usage**

```
predictor_metadata_prep(model_id)
```

**Arguments**

`model_id`            Character. Model ID

**Value**

Data frame

**Examples**

```
predictor_metadata_prep("bam_v5_can71")
```

---

|                                              |
|----------------------------------------------|
| <code>predictor_metadata_table</code>        |
| <i>Create a Table for Predictor Metadata</i> |

---

**Description**

Create a Table for Predictor Metadata

**Usage**

```
predictor_metadata_table(predictor_metadata)
```

**Arguments**

`predictor_metadata`  
Data frame. Predictor metadata

**Value**

A reactable table object

**Examples**

```
predictor_metadata_prep("bam_v5_can71") |>
  predictor_metadata_table()
```

---

`predictor_raster_layer`

*Add a Prediction Raster Layer to an existing map*

---

**Description**

Add a Prediction Raster Layer to an existing map

**Usage**

```
predictor_raster_layer(map, raster = NULL, layers = NULL)
```

**Arguments**

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <code>map</code>    | Leaflet map.                                         |
| <code>raster</code> | SpatRaster. Raster from which to get layers to plot. |
| <code>layers</code> | Character vector. Layers in SpatRaster to plot.      |

**Value**

Leaflet map

**Examples**

```
r <- predictor_raster_prep("bam_v5_can71")
base_map() |>
  predictor_raster_layer(r, "mean")
```

---

`predictor_raster_map`    *Create a Leaflet Map of Predictor Raster Data*

---

**Description**

Create a Leaflet Map of Predictor Raster Data

**Usage**

```
predictor_raster_map(predictor_raster, layers = NULL, subunits = NULL)
```

**Arguments**

predictor\_raster      terra SpatRaster. Predictor information.

layers                Character vector. Layers in SpatRaster to plot.

subunits              Spatial Data frame. Deployment subunits.

**Value**

A leaflet map object

**Examples**

```
predictor_raster_prep("bam_v5_can71") |>
  predictor_raster_map(
    layers = "AHM_1km", # Must be a layer in prepared raster
    subunits = deployment_subunits_prep("deployment1")
  )
```

---

predictor\_raster\_prep    *Prepare Predictor Raster Data*

---

**Description**

Prepare Predictor Raster Data

**Usage**

```
predictor_raster_prep(model_id)
```

**Arguments**

model\_id              Character. Model ID

**Value**

terra Raster

**Examples**

```
predictor_raster_prep("bam_v5_can71")
```

---

|                  |                            |
|------------------|----------------------------|
| prep_deployments | <i>Prepare Deployments</i> |
|------------------|----------------------------|

---

**Description**

Prepare Deployments

**Usage**

```
prep_deployments(deployment_id, deployment_type)
```

**Arguments**

deployment\_id    Character. Deployment ID  
deployment\_type    Character. Type of data to load ("deployment\_questions" or "deployment\_subunits")

**Value**

Data frame of deployments

**Examples**

```
prep_deployments("deployment1", "deployment_questions")  
prep_deployments("deployment1", "deployment_subunits")  
prep_deployments("deployment2", "deployment_questions")
```

---

|                  |                                               |
|------------------|-----------------------------------------------|
| prep_evaluations | <i>Fetch and format submitted evaluations</i> |
|------------------|-----------------------------------------------|

---

**Description**

Fetch and format submitted evaluations

**Usage**

```
prep_evaluations(user_id, deployment_id = NULL)
```

**Arguments**

user\_id            Character. User ID.  
deployment\_id    Character. Deployment ID

**Value**

Data frame of evaluation details

Examples

```
prep_evaluations("testuser") # Returns 0 rows if no evaluations
```

---

|            |                          |
|------------|--------------------------|
| prep_files | <i>Generic file prep</i> |
|------------|--------------------------|

---

Description

Generic file prep

Usage

```
prep_files(path, name, ...)
```

Arguments

- path                    Character. File path to fetch.
- name                   Character. Name of data for messaging
- ...                    Named model, species, etc. values. See data from [sdmEvalToolCore::make\\_target\\_path\(\)](#).

Value

File contents depending on file type (see [sdmEvalToolCore::read\\_file\(\)](#))

---

|                        |                                  |
|------------------------|----------------------------------|
| prep_material_settings | <i>Prepare Material Settings</i> |
|------------------------|----------------------------------|

---

Description

Prepare Material Settings

Usage

```
prep_material_settings(component_id, model_id, species_id = NULL)
```

Arguments

- component\_id    Character. Component ID
- model\_id        Character. Model ID
- species\_id      Character. Species ID

**Value**

A list with material settings

**Examples**

```
prep_material_settings("observations", species_id = "BBW0", model_id = "bam_v5_can71")
prep_material_settings("predictor_metadata", model_id = "bam_v5_can71")
prep_material_settings("spatial_prediction", species_id = "BBW0", model_id = "bam_v5_can71")
prep_material_settings("model_summary", species_id = "BBW0", model_id = "bam_v5_can71")
prep_material_settings("model_fit", species_id = "BBW0", model_id = "bam_v5_can71")
prep_material_settings("model_metadata", model_id = "bam_v5_can71")
```

---

|                |                            |
|----------------|----------------------------|
| prep_materials | <i>Load material files</i> |
|----------------|----------------------------|

---

**Description**

Load material files

**Usage**

```
prep_materials(component_id, model_id, species_id = NULL)
```

**Arguments**

|              |                         |
|--------------|-------------------------|
| component_id | Character. Component ID |
| model_id     | Character. Model ID     |
| species_id   | Character. Species ID   |

**Value**

Loaded file as an R object, material settings returned as an attribute.

**Examples**

```
prep_materials("observations", species_id = "BBW0", model_id = "bam_v5_can71")
prep_materials("model_metadata", model_id = "bam_v5_can71")
prep_materials("predictor_metadata", model_id = "bam_v5_can71")
#prep_materials("predictor_raster", model_id = "bam_v5_can71")
prep_materials("spatial_prediction", species_id = "BBW0", model_id = "bam_v5_can71")
prep_materials("model_summary", species_id = "BBW0", model_id = "bam_v5_can71")
prep_materials("model_fit", species_id = "BBW0", model_id = "bam_v5_can71")

# Expect Errors
# prep_materials("observations", model_id = "", species_id = "")
# prep_materials("observations", model_id = "bam_v5_can71", species_id = "")
```

---

|                |                                     |
|----------------|-------------------------------------|
| prep_questions | <i>Prepare Evaluation Questions</i> |
|----------------|-------------------------------------|

---

### Description

Reads, combines and prepares questions for use in UIs. `model_id` and `species_id` required to create `material_id`.

### Usage

```
prep_questions(
  component_id,
  deployment_id,
  model_id,
  species_id,
  user_id = NULL
)
```

### Arguments

|                            |                          |
|----------------------------|--------------------------|
| <code>component_id</code>  | Character. Component ID  |
| <code>deployment_id</code> | Character. Deployment ID |
| <code>model_id</code>      | Character. Model ID      |
| <code>species_id</code>    | Character. Species ID    |
| <code>user_id</code>       | Character. User ID.      |

### Value

Data frame of questions

### Examples

```
# Return all questions
prep_questions("ALL", "deployment1", "bam_v5_can71", "BBW0")

# Return component specific questions
prep_questions("observations", "deployment1", "bam_v5_can71", "BBW0")
prep_questions("model_fit", "deployment1", "bam_v5_can71")
prep_questions("model_summary", "deployment1", "bam_v5_can71")

prep_questions(c("model_summary", "model_fit"), "deployment1", "bam_v5_can71")

prep_questions("predictor_raster", "deployment1", "bam_v5_can71")

# Return default questions
prep_questions("observations", "deployment_test", "bam_v5_can71", "BBW0")
```

```
# Follow up questions
prep_questions("model_fit", "deployment2", "bam_v5_can71", "BBW0")
prep_questions("observations", "deployment1", "bam_v5_can71", "BBW0")

# Add evaluations
#prep_questions("observations", "deployment1", "bam_v5_can71", "BBW0", "draper")
prep_questions("observations", "deployment1", "bam_v5_can71", "BBW0", "testuser")

# For the abandon review
prep_questions("app", "deployment1", "bam_v5_can71", "BBW0", "testuser")
```

---

|                  |                             |
|------------------|-----------------------------|
| response_to_json | Create JSON evaluation body |
|------------------|-----------------------------|

---

## Description

Takes component, questions data frame, and input list from Shiny app and converts responses (evaluations) to JSON body for saving to the data base.

## Usage

```
response_to_json(component_id, questions, input_list)
```

## Arguments

|              |                                                  |
|--------------|--------------------------------------------------|
| component_id | Character. Component ID.                         |
| questions    | Data frame. From <code>prep_questions()</code> . |
| input_list   | List of evaluation inputs.                       |

## Examples

```
q1 <- test_questions("model_fit", "deployment2", "bam_v5_can71", "BBW0")
q2 <- test_questions("model_summary", "deployment2", "bam_v5_can71", "BBW0")
i1 <- test_input_evals(q1)
i2 <- test_input_evals(q2)
response_to_json("model_fit", rbind(q1, q2), append(i1, i2))
```

---

|                  |                         |
|------------------|-------------------------|
| save_evaluations | <i>Save evaluations</i> |
|------------------|-------------------------|

---

## Description

Save evaluations

## Usage

```
save_evaluations(questions, input_list, user_id)
```

## Arguments

|            |                                                                                |
|------------|--------------------------------------------------------------------------------|
| questions  | Data frame. Data frame of questions; output of <code>prep_questions()</code> . |
| input_list | List. List of Shiny inputs; output of <code>reactiveValuesToList(input)</code> |
| user_id    | Character. User ID.                                                            |

## Examples

```
q <- prep_questions(
  "observations",
  "deployment_test",
  "bam_v5_can71",
  "BBW0",
  "testuser"
)
a <- test_input_evals(q) # Create example evaluations
save_evaluations(q, a, user_id = "testuser")

# Compare
# Fetch saved from database
q <- prep_questions(
  "observations",
  "deployment_test",
  "bam_v5_can71",
  "BBW0",
  "testuser"
)
q <- evals_list(q) # Convert to list to compare
waldo::compare(a, q)
```

---

|             |                                         |
|-------------|-----------------------------------------|
| sdm_spinner | Create spinner when loading UI elements |
|-------------|-----------------------------------------|

---

**Description**

Create spinner when loading UI elements

**Usage**

sdm\_spinner(ui\_element)

**Arguments**

ui\_element      Character. UI element name.

**Value**

UI element with spinner.

---

|           |                 |
|-----------|-----------------|
| sdm_theme | Theme for bslib |
|-----------|-----------------|

---

**Description**

Theme for bslib

**Usage**

sdm\_theme()

**Value**

bslib theme

**Examples**

sdm\_theme()

---

`sdm_tool`*Launch the SDM Evaluation Tool Shiny Application*

---

## Description

Launch the SDM Evaluation Tool Shiny Application

## Usage

```
sdm_tool(  
  lang = "english",  
  options = list(host = "0.0.0.0", port = 7405),  
  tabs = c("overview", "predictions", "observations", "model", "predictors", "summary"),  
  user = NULL,  
  user_db = NULL,  
  ...  
)
```

## Arguments

|                      |                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>lang</code>    | Character. Language of app; either english or french.                                                                  |
| <code>options</code> | List. Shiny app options (passed to options in <a href="#">shiny::shinyApp()</a> ).                                     |
| <code>tabs</code>    | Character. List the tabs/pages that the UI should have.                                                                |
| <code>user</code>    | Character. User id. Placeholder for now, used for testing.                                                             |
| <code>user_db</code> | A data frame with credentials data or path to SQLite database created with <a href="#">shinymanager::create_db()</a> . |
| <code>...</code>     | Other arguments passed to <a href="#">shiny::shinyApp()</a> .                                                          |

## Value

A Shiny app object. Launched in browser/viewer if interactive.

## Examples

```
sdm_tool()
```

---

|             |                          |
|-------------|--------------------------|
| sdm_tooltip | <i>Generate tool tip</i> |
|-------------|--------------------------|

---

**Description**

No tooltip if text NULL.

**Usage**

```
sdm_tooltip(  
  tooltip_text,  
  trigger = bsicons::bs_icon("info-circle", title = "Details")  
)
```

**Arguments**

|              |                                                                            |
|--------------|----------------------------------------------------------------------------|
| tooltip_text | Character. Text to make into the tooltip contents.                         |
| trigger      | Code. Generally HTML tag lists to attach tooltip to. Defaults to (i) icon. |

**Value**

bslib tooltip

**Examples**

```
sdm_tooltip("testing")  
sdm_tooltip(  
  "testing",  
  shiny::tagList("Hi", bsicons::bs_icon("info-circle", title = "Details"))  
)
```

---

|             |                                |
|-------------|--------------------------------|
| set_options | <i>Set sdmEvalTool options</i> |
|-------------|--------------------------------|

---

**Description**

May not be necessary in the long run...

**Usage**

```
set_options(...)
```

**Arguments**

|     |                              |
|-----|------------------------------|
| ... | Named list of options to set |
|-----|------------------------------|

**Value**

list of options currently set

---

|              |                                             |
|--------------|---------------------------------------------|
| show_spatial | <i>Get a list of selected spatial areas</i> |
|--------------|---------------------------------------------|

---

**Description**

Returns a list of selected spatial areas by question value. Used to show selections on the map when "Show identified area" button associated with spatial questions is clicked.

**Usage**

```
show_spatial(show_clicked, questions, input)
```

**Arguments**

|              |                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------|
| show_clicked | Button input for "Show identified area" for a specific spatial question. Only contains buttons just clicked. |
| questions    | Data frame of original questions.                                                                            |
| input        | Input.                                                                                                       |

**Value**

Named list of Question values and spatial subunits for each.

**Examples**

```
# Example from test_page_summary()
c <- "deployment_subunits"
d <- "deployment1"
m <- "bam_v5_can71"
s <- "BBW0"
o <- 2
v <- "Confident"
sp <- paste(d, m, s, c, o, 0, sep = "_")

input <- list(c(4,5,9))
names(input) <- paste0(sp, "-", v)
clicked <- paste0(sp, "-show")
questions <- test_questions(c, d, m, s)
show_spatial(clicked, questions, input)

# Example from test_page_predictions()
c <- "spatial_prediction"
d <- "deployment1"
m <- "bam_v5_can71"
s <- "BBW0"
```

```

o <- 1
v <- c("Severe_over", "Moderate_over", "Accurate", "Moderate_under", "Severe_under", "Unknown")
sp <- paste(d, m, s, c, o, 0, sep = "_")

input <- list(c(4,5,9), NULL, NULL, c(3,5,10), NULL, NULL)
names(input) <- paste0(sp, "-", v)
clicked <- paste0(sp, "-show")
questions <- test_questions(c, d, m, s)
show_spatial(clicked, questions, input)

```

---

|                 |                                                   |
|-----------------|---------------------------------------------------|
| skip_if_no_data | <i>Skip testthat tests if no data is detected</i> |
|-----------------|---------------------------------------------------|

---

**Description**

Skip testthat tests if no data is detected

**Usage**

```
skip_if_no_data()
```

**Value**

Logical

---

|                        |                                                        |
|------------------------|--------------------------------------------------------|
| spatial_prediction_map | <i>Create a Leaflet Map of Spatial Prediction Data</i> |
|------------------------|--------------------------------------------------------|

---

**Description**

Create a Leaflet Map of Spatial Prediction Data

**Usage**

```
spatial_prediction_map(spatial_prediction, subunits = NULL, ns = identity)
```

**Arguments**

|                    |                                          |
|--------------------|------------------------------------------|
| spatial_prediction | terra Raster. Spatial predictions        |
| subunits           | Spatial Data frame. Deployment subunits. |
| ns                 | Namespace.                               |

**Value**

A leaflet map object

Examples

```
p <- spatial_prediction_prep(model_id = "bam_v5_can71", species_id = "BBW0")
s <- deployment_subunits_prep("deployment1")
spatial_prediction_map(p)
spatial_prediction_map(p, s)
```

---

|                         |                                        |
|-------------------------|----------------------------------------|
| spatial_prediction_prep | <i>Prepare Spatial Prediction Data</i> |
|-------------------------|----------------------------------------|

---

Description

Prepare Spatial Prediction Data

Usage

```
spatial_prediction_prep(model_id, species_id)
```

Arguments

|            |                     |
|------------|---------------------|
| model_id   | Character. Model ID |
| species_id | Character. Model ID |

Value

terra Raster

Examples

```
spatial_prediction_prep(model_id = "bam_v5_can71", species_id = "BBW0")
```

---

|               |                              |
|---------------|------------------------------|
| template_prep | <i>Prepare Template Data</i> |
|---------------|------------------------------|

---

Description

Prepare Template Data

Usage

```
template_prep(model_id, species_id)
```

**Arguments**

|            |                       |
|------------|-----------------------|
| model_id   | Character. Model ID   |
| species_id | Character. Species ID |

**Value**

Data frame

**Examples**

```
template_prep(model_ = "bam_v5_can71", species_id = "BBWO")
```

---

|                      |                                                      |
|----------------------|------------------------------------------------------|
| template_spatial_map | <i>Create a Leaflet Map of Template Spatial Data</i> |
|----------------------|------------------------------------------------------|

---

**Description**

Create a Leaflet Map of Template Spatial Data

**Usage**

```
template_spatial_map(template_spatial, subunits = NULL, ns = identity)
```

**Arguments**

|                  |                                          |
|------------------|------------------------------------------|
| template_spatial | terra Raster. Template spatial data      |
| subunits         | Spatial Data frame. Deployment subunits. |
| ns               | Namespace.                               |

**Value**

A leaflet map object

**Examples**

```
p <- template_spatial_prep(model_id = "bam_v5_can71", species_id = "BBWO")
s <- deployment_subunits_prep("deployment1")
template_spatial_map(p)
template_spatial_map(p, s)
```

---

|                       |                                      |
|-----------------------|--------------------------------------|
| template_spatial_prep | <i>Prepare Template Spatial Data</i> |
|-----------------------|--------------------------------------|

---

**Description**

Prepare Template Spatial Data

**Usage**

```
template_spatial_prep(model_id, species_id)
```

**Arguments**

|            |                     |
|------------|---------------------|
| model_id   | Character. Model ID |
| species_id | Character. Model ID |

**Value**

terra Raster

**Examples**

```
template_spatial_prep(model_id = "bam_v5_can71", species_id = "BBW0")
```

---

|                |                                         |
|----------------|-----------------------------------------|
| template_table | <i>Create a Table for Template Data</i> |
|----------------|-----------------------------------------|

---

**Description**

Create a Table for Template Data

**Usage**

```
template_table(template)
```

**Arguments**

|          |                                      |
|----------|--------------------------------------|
| template | Data frame with template information |
|----------|--------------------------------------|

**Value**

Reactable table

**Examples**

```
template_prep(model_id = "bam_v5_can71", species_id = "BBW0") |>
  template_table()
```

---

|           |                                       |
|-----------|---------------------------------------|
| test_comp | Create Test App for Component Modules |
|-----------|---------------------------------------|

---

**Description**

Not all arguments are used by all modules.

**Usage**

```
test_comp(  
  module,  
  use = c("parent_id", "deployment_id", "model_id", "species_id", "spatial_ids",  
    "spatial_selection", "map_views"),  
  deployment_id = "deployment1",  
  model_id = "bam_v5_can71",  
  species_id = "BBW0",  
  spatial_ids = NULL,  
  spatial_selection = list(show_clicked = NULL, show_spatial_ids = NULL),  
  map_views = list(active_tab = NULL, set_by = NULL, view = NULL)  
)
```

**Arguments**

|                   |                                                                                |
|-------------------|--------------------------------------------------------------------------------|
| module            | Character. Module Name.                                                        |
| use               | Character vector. Objects to include                                           |
| deployment_id     | Character. Example Deployment ID.                                              |
| model_id          | Character. Example Model ID.                                                   |
| species_id        | Character. Example Species ID.                                                 |
| spatial_ids       | Character. Optional starting spatial ids.                                      |
| spatial_selection | List. Optional starting spatial selection (show_clicked and show_spatial_ids). |
| map_views         | List. Optional set map views for map synchronization.                          |

**Value**

Shiny App

---

`test_comp_deployment_subunits`*Test the Deployment Subunits Component*

---

**Description**

Test the Deployment Subunits Component

**Usage**

```
test_comp_deployment_subunits(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_deployment_subunits()
```

---

`test_comp_model_fit`*Test the Model Fit Component*

---

**Description**

Test the Model Fit Component

**Usage**

```
test_comp_model_fit(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_model_fit()
```

---

`test_comp_model_summary`*Test the Model Summary Component*

---

**Description**

Test the Model Summary Component

**Usage**

```
test_comp_model_summary(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_model_summary()
```

---

`test_comp_obs_chart`*Test the Observations Chart Component*

---

**Description**

Test the Observations Chart Component

**Usage**

```
test_comp_obs_chart(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_obs_chart()
```

---

`test_comp_observations`*Test the Observations Component*

---

**Description**

Test the Observations Component

**Usage**

```
test_comp_observations(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_observations()
```

---

`test_comp_predictor_metadata`*Test the Predictor Metadata Component*

---

**Description**

Test the Predictor Metadata Component

**Usage**

```
test_comp_predictor_metadata(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_predictor_metadata()
```

---

`test_comp_predictor_raster`*Test the Predictor Raster Component*

---

**Description**

Test the Predictor Raster Component

**Usage**

```
test_comp_predictor_raster(...)
```

**Arguments**

... Other arguments.

**Value**

A Shiny app object

**Examples**

```
test_comp_predictor_raster()
```

---

`test_comp_spatial_prediction`*Test the Spatial Prediction Component*

---

**Description**

Test the Spatial Prediction Component

**Usage**

```
test_comp_spatial_prediction(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_spatial_prediction()
```

---

|                    |                                    |
|--------------------|------------------------------------|
| test_comp_template | <i>Test the Template Component</i> |
|--------------------|------------------------------------|

---

**Description**

Test the Template Component

**Usage**

```
test_comp_template(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_template()
```

---

|                            |                                            |
|----------------------------|--------------------------------------------|
| test_comp_template_spatial | <i>Test the Template Spatial Component</i> |
|----------------------------|--------------------------------------------|

---

**Description**

Test the Template Spatial Component

**Usage**

```
test_comp_template_spatial(...)
```

**Arguments**

... Arguments passed to other functions.

**Value**

A Shiny app object

**Examples**

```
test_comp_template_spatial()
```

---

test\_evaluation\_body     *Create dummy json evaluation body*


---

### Description

This can be used for testing loading/extracting/counting/etc. questions. These are created by running the App, saving data, and then extracting the JSON bodies. They can be updated the same way. Note that there are different bodies to test different kinds of behaviour.

### Usage

```
test_evaluation_body(component_id = "observations")
```

### Arguments

component\_id     Character. Id of the test evaluation body to use. Currently "observations", "model\_fit\_a", or "model\_fit\_b"

### Value

Character. JSON string for hypothetical evaluation

### Examples

```
test_evaluation_body()

## Not run:
# After created through app with 'testuser' and `deployment2` retrieved with:
db_read_evaluations(db_connect_check(), "deployment2", "testuser") |>
  dplyr::filter(component_id == "model_fit") |>
  dplyr::slice(1) |>
  dplyr::pull(evaluation_body)

## End(Not run)
```

---

test\_input\_evals     *Create dummy input values for evaluations*


---

### Description

Create dummy input values for evaluations

### Usage

```
test_input_evals(questions)
```

**Arguments**

questions            Data frame. Output of .prep\_questions()

**Value**

List of dummy input values. Mimics and input object from the Shiny app.

**Examples**

```
q <- test_questions()
test_input_evals(q)
```

---

|           |                                  |
|-----------|----------------------------------|
| test_page | Create Test App for Page Modules |
|-----------|----------------------------------|

---

**Description**

Not all arguments are used by all modules.

**Usage**

```
test_page(
  module,
  deployment_id = "deployment1",
  model_id = "bam_v5_can71",
  species_id = "BBW0",
  user_id = "testuser",
  user_role = "evaluator",
  user_admin = FALSE,
  map_views = list(active_tab = NULL, set_by = NULL, view = NULL)
)
```

**Arguments**

module            Character. Module Name.  
deployment\_id    Character. Example Deployment ID.  
model\_id          Character. Example Model ID.  
species\_id       Character. Example Species ID.  
user\_id           Character. Example User ID.  
user\_role        Character. Example User Role.  
user\_admin       Logical. Example User Admin status.  
map\_views        List. Optional set map views for map synchronization.

**Value**

Shiny App

---

|             |                            |
|-------------|----------------------------|
| test_points | Create dummy sf point data |
|-------------|----------------------------|

---

**Description**

Create dummy sf point data

**Usage**

```
test_points()
```

**Value**

sf data frame

**Examples**

```
test_points()
```

---

|                |                                    |
|----------------|------------------------------------|
| test_questions | Create dummy questions for testing |
|----------------|------------------------------------|

---

**Description**

Create dummy questions for testing

**Usage**

```
test_questions(  
  component = "observations",  
  deployment_id = "deployment_test",  
  model_id = "bam_v5_can71",  
  species_id = "BBW0",  
  types = c("spatial", "ordinal", "simple_text"),  
  user_id = "testuser"  
)
```

**Arguments**

|               |                 |
|---------------|-----------------|
| component     | Component.      |
| deployment_id | Deployment ID.  |
| model_id      | Model ID.       |
| species_id    | Species ID.     |
| types         | Question types. |
| user_id       | User id.        |

**Examples**

```
test_questions()
```

---

|             |                                    |
|-------------|------------------------------------|
| test_raster | Create dummy terra SpatRaster data |
|-------------|------------------------------------|

---

**Description**

Create dummy terra SpatRaster data

**Usage**

```
test_raster()
```

**Value**

SpatRaster

**Examples**

```
test_raster()
```

---

|                      |                                       |
|----------------------|---------------------------------------|
| tt_material_settings | Create tooltip from material settings |
|----------------------|---------------------------------------|

---

**Description**

Create tooltip from material settings

**Usage**

```
tt_material_settings(material)
```

**Arguments**

|          |                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------|
| material | R Object depending on the type of material loaded. May have attributes such as 'material_settings'. |
|----------|-----------------------------------------------------------------------------------------------------|

**Value**

bslib tooltip output from [sdm\\_tooltip\(\)](#).

---

`ui_questions`*Create dynamic question inputs*

---

### Description

Create dynamic question inputs in the server using the prepared questions data frame.

### Usage

```
ui_questions(questions, spatial_ids = NULL, width = NULL)
```

### Arguments

|                          |                                                                                                                                                                                                                                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>questions</code>   | Data frame prepared by <code>prep_questions()</code> for a specific <code>user_id</code> .                                                                                                                                                                                                                |
| <code>spatial_ids</code> | Character vector. [NOT CURRENTLY USED, see <a href="#">ui_questions_update()</a> ]. Only applies if using 'client-side selectize'. All possible Spatial ID options for spatial inputs choices. Generally passed in as <code>ReactiveVal</code> (see <a href="#">mod_page_template_spatial_server()</a> ). |
| <code>width</code>       | Numeric. Optional width of Shiny UI input.                                                                                                                                                                                                                                                                |

### Value

Shiny `tagList` of UI elements

### Examples

```
q <- prep_questions(
  "observations",
  "deployment1",
  "bam_v5_can71",
  "BBW0",
  user_id = "testuser"
)
ui_questions(q)

# More than one component
q <- prep_questions(
  c("model_fit", "model_summary"),
  "deployment2",
  "bam_v5_can71",
  "BBW0",
  user_id = "testuser"
)
u <- ui_questions(q)
```

---

|                     |                                             |
|---------------------|---------------------------------------------|
| ui_questions_update | <i>Update existing Spatial Shiny inputs</i> |
|---------------------|---------------------------------------------|

---

### Description

Because there may be many possible selections for Spatial inputs, we use Server-side processing to speed things up. This means that we initialize the Spatial input, and then we have to update it with the spatial input options (i.e. the list of spatial ids in the drop down menu). This function performs that update when the ids are ready (see the use of `ui_questions_update()` in `mod_utils_evaluations_server()`).

### Usage

```
ui_questions_update(questions, spatial_ids = NULL)
```

### Arguments

|             |                                                                                                                                                                                                                                                                                                     |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| questions   | Data frame prepared by <code>prep_questions()</code> for a specific user_id.                                                                                                                                                                                                                        |
| spatial_ids | Character vector. [NOT CURRENTLY USED, see <code>ui_questions_update()</code> ]. Only applies if using 'client-side selectize'. All possible Spatial ID options for spatial inputs choices. Generally passed in as <code>ReactiveVal</code> (see <code>mod_page_template_spatial_server()</code> ). |

### Value

Nothing. Performs Javascript update (see `shiny::updateSelectizeInput()`).

---

|              |                           |
|--------------|---------------------------|
| validate_ids | <i>Validate input ids</i> |
|--------------|---------------------------|

---

### Description

Validate input ids

### Usage

```
validate_ids(...)
```

### Arguments

|     |                                                                                                                                        |
|-----|----------------------------------------------------------------------------------------------------------------------------------------|
| ... | Named arguments containing id values to validate (e.g., <code>deployment_id</code> , <code>model_id</code> , <code>species_id</code> ) |
|-----|----------------------------------------------------------------------------------------------------------------------------------------|

### Value

NULL (invisibly), or throws validation error if any id is missing

---

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| value_to_input | <i>Helper function to format Question values as Shiny Input names</i> |
|----------------|-----------------------------------------------------------------------|

---

**Description**

Helper function to format Question values as Shiny Input names

**Usage**

```
value_to_input(v)
```

**Arguments**

`v` Character vector of Question values.

**Value**

Character string of values formatted for Shiny Inputs

**Examples**

```
value_to_input("Strongly Agree")
```

# Index

add\_control, 5  
add\_control(), 6  
add\_markers, 6  
add\_markers(), 7  
add\_raster, 6  
add\_selected\_markers, 7  
add\_selected\_subunits, 8  
add\_subunits, 9  
add\_subunits(), 8  
affirmative, 10  
app\_materials, 10  
app\_users, 11  
  
base\_map, 11  
bslib::card(), 12  
bslib::card\_header(), 12  
bslib::layout\_sidebar(), 12  
bslib::nav\_panel(), 12  
bslib\_wrapper, 12  
  
check\_q\_mismatch, 13  
coords\_to\_poly, 13  
  
db\_connect\_check, 14  
DBI::dbConnect(), 14  
deployment\_subunits\_map, 14  
deployment\_subunits\_prep, 15  
  
evals\_answered, 15  
evals\_details, 16  
evals\_extract, 17  
evals\_list, 17  
evals\_table, 18  
  
feature\_ids, 19  
fetch\_questions, 19  
fmt\_species, 20  
fmt\_time, 20  
  
get\_material\_settings, 21  
  
have\_data, 21  
  
identical\_loose, 22  
input\_to\_value, 22  
  
lang, 23  
  
map\_has\_group, 23  
map\_reactive\_vals, 24  
mod\_comp\_deployment\_subunits\_server, 24  
mod\_comp\_deployment\_subunits\_ui, 25  
mod\_comp\_model\_fit\_server, 26  
mod\_comp\_model\_fit\_ui, 26  
mod\_comp\_model\_summary\_server, 27  
mod\_comp\_model\_summary\_ui, 27  
mod\_comp\_obs\_chart\_server, 28  
mod\_comp\_obs\_chart\_ui, 28  
mod\_comp\_observations\_server, 29  
mod\_comp\_observations\_ui, 30  
mod\_comp\_predictor\_metadata\_server, 30  
mod\_comp\_predictor\_metadata\_ui, 31  
mod\_comp\_predictor\_raster\_server, 31  
mod\_comp\_predictor\_raster\_ui, 32  
mod\_comp\_spatial\_prediction\_server, 33  
mod\_comp\_spatial\_prediction\_ui, 34  
mod\_comp\_template\_server, 34  
mod\_comp\_template\_spatial\_server, 35  
mod\_comp\_template\_spatial\_ui, 36  
mod\_comp\_template\_ui, 36  
mod\_page\_model\_server, 37  
mod\_page\_model\_ui, 37  
mod\_page\_observations\_server, 38  
mod\_page\_observations\_ui, 38  
mod\_page\_overview\_server, 39  
mod\_page\_overview\_ui, 40  
mod\_page\_predictions\_server, 40  
mod\_page\_predictions\_ui, 41  
mod\_page\_predictors\_server, 41  
mod\_page\_predictors\_ui, 42

mod\_page\_summary\_server, 42  
 mod\_page\_summary\_ui, 43  
 mod\_page\_template\_server, 44  
 mod\_page\_template\_spatial\_server, 44  
 mod\_page\_template\_spatial\_server(), 87, 88  
 mod\_page\_template\_spatial\_ui, 45  
 mod\_page\_template\_ui, 45  
 mod\_page\_test\_server, 46  
 mod\_page\_test\_ui, 46  
 mod\_utils\_details\_server, 47  
 mod\_utils\_details\_ui, 47  
 mod\_utils\_evaluations\_server, 48  
 mod\_utils\_evaluations\_server(), 22, 88  
 mod\_utils\_evaluations\_ui, 49  
 mod\_utils\_map\_selections\_server, 49  
 mod\_utils\_map\_selections\_server(), 11  
 mod\_utils\_map\_selections\_ui, 50  
 mod\_utils\_map\_sync\_server, 51  
 model\_fit\_prep, 51  
 model\_fit\_table, 52  
 model\_summary\_prep, 53  
 model\_summary\_table, 53  
  
 named\_ids, 54  
  
 obs\_map, 54  
 obs\_map\_points, 55  
 obs\_map\_raster, 56  
 obs\_prep, 57  
 obs\_prep(), 55  
 obs\_prep\_points, 58  
 obs\_prep\_points(), 57  
 obs\_prep\_raster, 58  
 obs\_prep\_raster(), 55, 57  
  
 parent\_vals, 59  
 predictor\_metadata\_prep, 60  
 predictor\_metadata\_table, 60  
 predictor\_raster\_layer, 61  
 predictor\_raster\_map, 61  
 predictor\_raster\_prep, 62  
 prep\_deployments, 63  
 prep\_evaluations, 63  
 prep\_files, 64  
 prep\_material\_settings, 64  
 prep\_materials, 65  
 prep\_questions, 66  
 prep\_questions(), 67  
  
 response\_to\_json, 67  
 response\_to\_json(), 17  
  
 save\_evaluations, 68  
 sdm\_card(bslib\_wrapper), 12  
 sdm\_card\_header(bslib\_wrapper), 12  
 sdm\_layout\_sidebar(bslib\_wrapper), 12  
 sdm\_nav\_panel(bslib\_wrapper), 12  
 sdm\_spinner, 69  
 sdm\_theme, 69  
 sdm\_theme(), 12  
 sdm\_tool, 70  
 sdm\_tool(), 37–41, 43, 44, 46–48  
 sdm\_tooltip, 71  
 sdm\_tooltip(), 86  
 sdmEvalToolCore::default\_questions(), 19  
 sdmEvalToolCore::make\_target\_path(), 64  
 sdmEvalToolCore::read\_file(), 64  
 set\_options, 71  
 shiny::observe(), 24  
 shiny::shinyApp(), 70  
 shiny::updateSelectizeInput(), 88  
 shinymanager::create\_db(), 70  
 show\_spatial, 72  
 skip\_if\_no\_data, 73  
 spatial\_prediction\_map, 73  
 spatial\_prediction\_prep, 74  
  
 template\_prep, 74  
 template\_spatial\_map, 75  
 template\_spatial\_prep, 76  
 template\_table, 76  
 test\_comp, 77  
 test\_comp\_deployment\_subunits, 78  
 test\_comp\_model\_fit, 78  
 test\_comp\_model\_summary, 79  
 test\_comp\_obs\_chart, 79  
 test\_comp\_observations, 80  
 test\_comp\_predictor\_metadata, 80  
 test\_comp\_predictor\_raster, 81  
 test\_comp\_spatial\_prediction, 81  
 test\_comp\_template, 82  
 test\_comp\_template\_spatial, 82  
 test\_evaluation\_body, 83  
 test\_input\_evals, 83  
 test\_page, 84  
 test\_points, 85

test\_questions, [85](#)  
test\_raster, [86](#)  
tt\_material\_settings, [86](#)  
  
ui\_questions, [87](#)  
ui\_questions\_update, [88](#)  
ui\_questions\_update(), [87](#), [88](#)  
  
validate\_ids, [88](#)  
value\_to\_input, [89](#)