Package ‘ruODK’

August 10, 2020

Type Package

Title An R Client for the ODK Central API

Version 0.9.1

Description Utilities to access and tidy up data from ODK
Central's API. ODK Central is OpenDataKit's clearinghouse for
digitally captured data <https://docs.opendatakit.org/central-intro/>.
ODK Central's API is documented at
<https://odkcentral.docs.apiary.io/>.

License GPL-3

URL https://ropensci.github.io/ruODK/,
https://github.com/ropensci/ruODK

BugReports https://github.com/ropensci/ruODK/issues

Depends R (>= 3.4)

Imports clisymbols (>= 1.2.0),
crayon (>= 1.3.4),
dplyr (>= 0.8.5),
fs (>= 1.4.1),
glue (>= 1.4.0),
httr (>= 1.4.1),
janitor (>= 2.0.1),
lifecycle (>= 0.1.0),
lubridate (>= 1.7.8),
magrittr (>= 1.5),
purrr (>= 0.3.4),
readr (>= 1.3.1),
rlang (>= 0.4.5),
stringr (>= 1.4.0),
tibble (>= 2.1.3),
tidyr (>= 1.0.3),
tidyselect (>= 1.0.0),
xml2 (>= 1.2.2)

Suggests covr (>= 3.4.0),
DT (>= 0.9),
ggplot2 (>= 3.2.1),
knitr (>= 1.26),
leaflet (>= 2.0.3),
listviewer (>= 3.0.0),
mapview (>= 2.7.8),
rmarkdown (>= 1.17),
roxygen2 (>= 7.1.0),
sf (>= 0.9-5),
testthat (>= 2.3.2),
usethis (>= 1.6.0),
vcr (>= 0.5.4),
webshot (>= 0.5.2)

VignetteBuilder knitr

RdMacros lifecycle

Encoding UTF-8

Language en_AU

LazyData true

RoxygenNote 7.1.1

X-schema.org-applicationCategory Data Access

X-schema.org-keywords database, open-data, opendatakit, odk, api, data, dataset

R topics documented:

`attachment_get` *Download attachments and return the local path.*

Description

Stable

Usage

```
attachment_get(
  sid,
  fn,
  local_dir = "media",
  separate = FALSE,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  verbose = get_ru_verbose()
)
```

Arguments

`sid`	One or many ODK submission UUIDs, an MD5 hash.
`fn`	One or many ODK form attachment filenames, e.g. "1558330537199.jpg".
`local_dir`	The local folder to save the downloaded files to, default: "media".
`separate`	(logical) Whether to separate locally downloaded files into a subfolder named after the submission uuid within ‘local_dir‘, default: FALSE. The defaults mirror the behaviour of `submission_export`, which keeps all attachment files together in a folder ‘media‘. Enable this option if downloaded files collide on idential names. This can happen if two data collection devices by chance generate the same filename for two respective media files, e.g. ‘DCIM0001.jpg‘.
`pid`	The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.
`fid`	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.
`url`	The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

un — The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw — The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

verbose — Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

This function is the workhorse for handle_ru_attachments. This function is vectorised and can handle either one or many records. Parameters submission_uuid and attachment_filename accept single or exactly the same number of multiple values. The other parameters are automatically repeated.

The media attachments are downloaded into a folder given by ‘local_dir‘:

workdir/media/filename1.jpg

workdir/media/filename2.jpg

workdir/media/filename3.jpg

Value

The relative file path for the downloaded attachment(s)

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/'-form-attachments/downloading-a-form-attachment

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/attachments/downloading-an-attachment

Other utilities: attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
# Step 1: Setup ruODK with OData Service URL (has url, pid, fid)
ruODK::ru_setup(svc = "...")
a_local_dir <- here::here()

# Step 2: Get unparsed submissions
fresh_raw <- odata_submission_get(parse = FALSE)

# Step 3: Get attachment field "my_photo"
fresh_parsed <- fresh_raw %>%
  odata_submission_rectangle() %>%
  dplyr::mutate(
    my_photo = attachment_get(id,
      my_photo,
      local_dir = a_local_dir,
      verbose = TRUE
    )
    # Repeat for all other attachment fields
  )

## End(Not run)
```

attachment_link *Prefix attachment columns from CSV export with a local attachment file path.*

Description

Stable

Usage

```
attachment_link(data_tbl, form_schema, att_path = "media")
```

Arguments

data_tbl	The downloaded submissions from submission_export read into a ‘tibble‘ by readr::read_csv.
form_schema	The ‘form_schema‘ for the submissions. E.g. the output of ‘ruODK::form_schema()‘.
att_path	A local path, default: "media" (as per .csv.zip export). Selected columns of the dataframe (containing attchment filenames) are prefixed with ‘att_path‘, thus turning them into relative paths.

Value

The dataframe with attachment columns modified to contain relative paths to the downloaded attachment files.

See Also

Other restful-api: attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
t <- tempdir()
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

# Predict filenames (with knowledge of form)
fid <- get_default_fid()
fid_csv <- fs::path(t, glue::glue("{fid}.csv"))
fid_csv_tae <- fs::path(t, glue::glue("{fid}-taxon_encounter.csv"))
fs <- form_schema()

# Download the zip file
se <- ruODK::submission_export(
  local_dir = t,
  overwrite = FALSE,
  verbose = TRUE
)

# Unpack the zip file
f <- unzip(se, exdir = t)
fs::dir_ls(t)

# Prepend attachments with media/ to turn into relative file paths
data_quadrat <- fid_csv %>%
  readr::read_csv(na = c("", "NA", "na")) %>%
  janitor::clean_names() %>%
  handle_ru_datetimes(fs) %>%
  attachment_link(fs)

## End(Not run)
```

attachment_list *List all attachments for a list of submission instances.*

Description

List all attachments for a list of submission instances.

Usage

```
attachment_list(
  iid,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`iid` A list of submission instance IDs, e.g. from `submission_list$instance_id`.

`pid` The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default pid through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.

`fid` The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Value

A tibble containing some high-level details of the submission attachments. One row per submission attachment, columns are submission attributes:

* name: The attachment filename, e.g. 12345.jpg * exists: Whether the attachment for that submission exists on the server.

See Also

`https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/attachments/listing-expected-submission-attachments`

`https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/’-form-attachments/listing-expected-form-attachments`

Other restful-api: `attachment_link()`, `audit_get()`, `form_detail()`, `form_list()`, `form_schema()`, `form_xml()`, `project_create()`, `project_detail()`, `project_list()`, `submission_detail()`, `submission_export()`, `submission_get()`, `submission_list()`

Examples

```
## Not run:
# Step 1: Setup ruODK with OData Service URL (has url, pid, fid)
```

```
ruODK::ru_setup(svc = "...")

# Step 2: List all submissions of form
sl <- submission_list()

# Step 3a: Get attachment list for first submission
al <- get_one_submission_attachment_list(sl$instance_id[[1]])

# Ste 3b: Get all attachments for all submissions
all <- attachment_list(sl$instance_id)

## End(Not run)
```

---

audit_get *Get server audit log entries.*

Description

Stable

Usage

```
audit_get(
  action = NULL,
  start = NULL,
  end = NULL,
  limit = NULL,
  offset = NULL,
  url = Sys.getenv("ODKC_URL"),
  un = Sys.getenv("ODKC_UN"),
  pw = Sys.getenv("ODKC_PW")
)
```

Arguments

`action` string. The action to filter the logs, e.g. "user.create". See `https://odkcentral.docs.apiary.io/#reference/system-endpoints/server-audit-logs/` for the full list of available actions.

`start` string. The ISO8601 timestamp of the earliest log entry to return. E.g. ‘2000-01-01z‘ or ‘2000-12-31T23:59.999z‘, ‘2000-01-01T12:12:12+08‘ or ‘2000-01-01+08‘.

`end` string. The ISO8601 timestamp of the last log entry to return.

`limit` integer. The max number of log entries to return.

`offset` integer. The number of log entries to skip.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

un The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Details

Parameters to filter the audit logs: ‘action=form.create&start=2000-01-01z&end=2000-12-31T23

Value

A tibble containing server audit logs. One row per audited action, columns are submission attributes:

* actor_id: integer. The ID of the actor, if any, that initiated the action. * action: string. The action that was taken. * actee_id: uuid, string. The ID of the permissioning object against which the action was taken. * details: list. Additional details about the action that vary according to the type of action. * logged_at: dttm. Time of action on server.

See Also

https://odkcentral.docs.apiary.io/#reference/system-endpoints/server-audit-logs/getting-audit-log-entries

Other restful-api: attachment_link(), attachment_list(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

logs <- audit_get()

# With search parameters
logs <- audit_get(
  action = "project.update",
  start = "2019-08-01Z",
  end = "2019-08-31Z",
  limit = 100,
  offset = 0
)
```

```
# With partial search parameters
logs <- audit_get(
  limit = 100,
  offset = 0
)

logs %>% knitr::kable(.)

# audit_get returns a tibble
class(logs)
# > c("tbl_df", "tbl", "data.frame")

# Audit details
names(logs)
# > "actor_id" "action" "actee_id" "details" "logged_at"

## End(Not run)
```

drop_null_coords *Drop any NULL coordinates from a GeoJSON geometry.*

Description

This helper patches a bug/feature in ODK Central (versions 0.7-0.9), where geotrace / geoshape GeoJSON contains a last coordinate pair with NULL lat/lon (no alt/acc), and WKT ends in ‘, undefined NaN‘.

Usage

```
drop_null_coords(x)
```

Arguments

x	A GeoJSON geometry parsed as nested list. E.g. ‘geo_gj$path_location_path_gps‘.

Details

While split_geotrace and split_geoshape modify the WKT inline, it is more maintainable to separate the GeoJSON cleaner into this function.

This helper drops the last element of a GeoJSON coordinate list if it is ‘list(NULL, NULL)‘.

Value

The nested list minus the last element (if NULL).

See Also

Other utilities: attachment_get(), attachment_url(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
# A snapshot of geo data with trailing empty coordinates.
data("geo_gj88")

len_coords <- length(geo_gj88$path_location_path_gps[[1]]$coordinates)

length(geo_gj88$path_location_path_gps[[1]]$coordinates[[len_coords]]) %>%
  testthat::expect_equal(2)

geo_gj88$path_location_path_gps[[1]]$coordinates[[len_coords]][[1]] %>%
  testthat::expect_null()

geo_gj88$path_location_path_gps[[1]]$coordinates[[len_coords]][[2]] %>%
  testthat::expect_null()

# The last coordinate pair is a list(NULL, NULL).
# Invalid coordinates like these are a choking hazard for geospatial
# packages. We should remove them before we can convert ODK data into native
# spatial formats, such as sf.
str(geo_gj88$path_location_path_gps[[1]]$coordinates[[len_coords]])

geo_gj_repaired <- geo_gj88 %>%
  dplyr::mutate(
    path_location_path_gps = path_location_path_gps %>%
      purrr::map(drop_null_coords)
  )

len_coords_repaired <- length(
  geo_gj_repaired$path_location_path_gps[[1]]$coordinates
)
testthat::expect_equal(len_coords_repaired + 1, len_coords)
```

---

form_detail *Show details for one form.*

Description

Stable

Usage

```
form_detail(
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`pid` The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default pid through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.

`fid` The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Value

A tibble with one row and all form metadata as columns.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/'-individual-form

Other restful-api: `attachment_link()`, `attachment_list()`, `audit_get()`, `form_list()`, `form_schema()`, `form_xml()`, `project_create()`, `project_detail()`, `project_list()`, `submission_detail()`, `submission_export()`, `submission_get()`, `submission_list()`

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

# With explicit credentials, see tests
fl <- form_list()

# The first form in the test project
f <- form_detail(fid = fl$fid[[1]])

# form_detail returns exactly one row
nrow(f)
# > 1

# form_detail returns all form metadata as columns: name, xmlFormId, etc.
names(f)

# > "name" "fid" "version" "state" "submissions" "created_at"
# > "created_by_id" "created_by" "updated_at" "last_submission" "hash"

## End(Not run)
```

`form_list` *List all forms.*

Description

Stable

Usage

```
form_list(
  pid = get_default_pid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`pid` The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Value

A tibble with one row per form and all form metadata as columns.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/forms

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_schema(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

# With default pid
fl <- form_list()

# With explicit pid
fl <- form_list(pid = 1)

class(fl)
# > c("tbl_df", "tbl", "data.frame")

## End(Not run)
```

form_schema *Show the schema of one form.*

Description

Stable

Usage

```
form_schema(
  flatten = FALSE,
  odata = FALSE,
  parse = TRUE,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  odkc_version = get_default_odkc_version(),
  verbose = get_ru_verbose()
)
```

Arguments

`flatten` Whether to flatten the resulting list of lists (TRUE) or not (FALSE, default). Only applies to ODK Central version < 0.8.

`odata` Whether to sanitise the field names to match the way they will be outputted for OData. While the original field names as given in the XForms definition may be used as-is for CSV output, OData has some restrictions related to the domain-qualified identifier syntax it uses. Only applies to ODK Central version < 0.8. Default: FALSE.

`parse` Whether to parse the form schema into a tibble of form field type and name. This uses `form_schema_parse` internally. If used together with ‘flatten=TRUE‘, `form_schema` will raise a warning and return the unparsed, flattened form schema. Only applies to ODK Central version < 0.8. Default: TRUE.

`pid` The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.

`fid` The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default `un` through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

`odkc_version` The ODK Central version as decimal number (major.minor). ‘ruODK‘ uses this parameter to adjust for breaking changes in ODK Central. Default: `get_default_odkc_version` or 0.8 if unset. Set default `get_default_odkc_version` through `ru_setup(odkc_version=0.8)`. See `vignette("Setup",package = "ruODK")`.

`verbose` Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

ODK Central has introduced a new API endpoint in version 0.8 which returns a parsed and flattened list of fields. This replaces the nested form schema which is challenging to parse.

While users of newest ODK Central versions (> 0.8) can ignore the legacy support for ODK Central's earlier form schema API, users of ODK Central version < 0.8 can set an environment variable ODKC_VERSION to their ODKC's version in format <major>.<minor> e.g. 0.7. This variable caters for future breaking changes.

Either way, form_schema will always return a tibble with columns name, type, path and ruodk_name.

Value

A tibble or nested list (v0.7) containing the form definition. At the lowest nesting level, each form field consists of a list of two nodes, ‘name‘ (the underlying field name) and ‘type‘ (the XForms field type, as in "string", "select1", "geopoint", "binary" and so on). These fields are nested in lists of tuples ‘name‘ (the XForms screen name), ‘children‘ (the fields as described above), ‘type‘ ("structure" for non- repeating screens, "repeat" for repeating screens). A list with ‘name‘ "meta" may precede the structure, if several metadata fields are captured (e.g. "instanceId", form start datetimes etc.). In all cases for ODK Central 0.8, and with default parameters (parse=TRUE) for ODK Central 0.7, form_schema returns a tibble with the columns:

- name The field name as given in the form schema.
- type The field type, e.g. "string", "select1", etc.
- path The XForms path of the field,
- ruodk_name The predicted field name as generated by odata_submission_get, prefixed by the path.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/'-individual-form/getting-form-schema-fields

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/'-individual-form/retrieving-form-schema-json

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

# With explicit pid and fid
fs_defaults <- form_schema(pid = 1, fid = "build_xformsId")
```

```
# With current ODK Central (v0.8)
fs <- form_schema()

# With defaults, ODK Central v0.7
fs_nested <- form_schema(
  flatten = FALSE,
  odata = FALSE,
  parse = FALSE,
  odkc_version = 0.7
)
listviewer::jsonedit(fs_nested)

fs_flattened <- form_schema(
  flatten = TRUE,
  odata = FALSE,
  parse = FALSE,
  odkc_version = 0.7
)
listviewer::jsonedit(fs_flattened)

# form_schema returns a nested list. There's nothing to change about that.
class(fs_nested)
# > "list"

class(fs_flattened)
# > "list"

# This assumes knowledge of that exact form being tested.
# First node: type "structure" (a field group) named "meta".
fs_nested[[1]]$type
# > "structure"

fs_nested[[1]]$name
# > "meta"

# The first node contains children, which means it's an XForms field group.
names(fs_nested[[1]])
# > "name" "children" "type"

# Next node: a "meta" field of type "string" capturing the  "instanceId".
# First child node of "meta": type "string", name "instanceId".
fs_nested[[1]]$children[[1]]$type
# > "string"
fs_nested[[1]]$children[[1]]$name
# > "instanceID"

# In the flattened version, the field's and it's ancestors' names are the
# components of "path".
fs_flattened[[1]]$path
# > "meta". "instanceId"

fs_flattened[[1]]$type
# > "string"
# Last node: a "meta" field capturing the datetime of form completion
fs_flattened[[length(fs_flattened)]]$type
# > "dateTime"
fs_nested[[length(fs_nested)]]$type
# > "dateTime"

# Parsed into a tibble of form field type/name:
# Useful to inform further parsing of submission data (attachments, dates)
fs <- form_schema(parse = TRUE, odkc_version = 0.7)
fs <- form_schema(odkc_version = 0.8)

# Attachments: used by handle_ru_attachments
fs %>% dplyr::filter(type == "binary")

# dateTime: used by handle_ru_datetimes
fs %>% dplyr::filter(type == "dateTime")

# Point location: used by handle_ru_geopoints
fs %>% dplyr::filter(type == "geopoint")

## End(Not run)
```

form_schema_parse *Parse a form_schema into a tibble of fields with name, type, and path.*

Description

Stable

Usage

```
form_schema_parse(fs, path = "Submissions", verbose = get_ru_verbose())
```

Arguments

fs The output of form_schema as nested list

path The base path for form fields. Default: "Submissions". form_schema_parse recursively steps into deeper nesting levels, which are reflected as separate OData tables. The returned value in ‘path‘ reflects the XForms group name, which translates to separate screens in ODK Collect. Non-repeating form groups will be flattened out into the main Submissions table. Repeating groups are available as separate OData tables.

verbose Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

This function is used by form_schema for older versions of ODK Central (pre 0.8). These return the form schema as XML, requiring the quite involved code of form_schema_parse, while newer ODK Central versions return JSON, which is parsed directly in form_schema.

The ‘form_schema‘ returned from ODK Central versions < 0.8 is a nested list of lists containing the form definition. The form definition consists of fields (with a type and name), and form groups, which are rendered as separate ODK Collect screens. Form groups in turn can also contain form fields.

form_schema_parse recursively unpacks the form and extracts the name and type of each field. This information then informs handle_ru_attachments, handle_ru_datetimes, handle_ru_geopoints, handle_ru_geotraces, and handle_ru_geoshapes.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
# Option 1: in two steps, ODKC Version 0.7
fs <- form_schema(flatten = FALSE, parse = FALSE, odkc_version = 0.7)
fsp <- form_schema_parse(fs)

# Option 2: in one go
fsp <- form_schema(parse = TRUE)

fsp

## End(Not run)
```

---

form_xml *Show the XML representation of one form as list.*

Description

Stable

Usage

```
form_xml(
  parse = TRUE,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

parse — Whether to parse the XML into a nested list, default: TRUE

pid — The numeric ID of the project, e.g.: 2. Default: get_default_pid. Set default pid through ru_setup(pid="..."). See vignette("Setup",package = "ruODK").

fid — The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: get_default_fid. Set default fid through ru_setup(fid="..."). See vignette("Setup",package = "ruODK").

url — The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").

un — The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw — The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Value

The form XML as a nested list.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/'-individual-form/retrieving-form-xml

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)
```

```
# With explicit pid and fid
fxml_defaults <- form_xml(1, "build_xformsId")

# With defaults
fxml <- form_xml()
listviewer::jsonedit(fxml)

# form_xml returns a nested list
class(fxml)
# > "list"

## End(Not run)
```

`fq_attachments` *A tibble of submission attachments.*

Description

Stable

Usage

```
fq_attachments
```

Format

A tibble of submission attachments.

Source

The output of `attachment_list` run on submissions of the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

fq_data *Parsed submission data for an ODK Central form.*

Description

Stable

Usage

```
fq_data
```

Format

The output of `odata_submission_get` for a set of example data. A tidy tibble referencing the attachments included in the vignettes and documentation at a relative path ‘attachments/media/<filename>.<ext>‘.

Details

The parsed OData response for the submissions of an ODK Central form. This form represents a Flora Quadrat, which is a ca 50 by 50 m quadrat of a uniform plant community.

The XML and .odkbuild versions for this form are available as ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ and ‘system.file("extdata", "FloraQuadrat04.odkbuild", package = "ruODK")‘, respectively.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ and `odata_submission_get`.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt` `fq_data_strata` *Parsed submission data for a subgroup of an ODK Central form.*

Description

Stable

Usage

```
fq_data_strata
```

Format

The output of `odata_submission_get` for a set of example data. A tidy tibble referencing the attachments included in the vignettes and documentation at a relative path ‘attachments/media/<filename>.<ext>‘.

Details

The parsed OData response for the subgroup of an ODK Central form.

This subgroup represents vegetation strata as per the NVIS classification. A vegetation stratum is a layer of plants with the same height, and dominated by one or few plant taxa. Plant communities can be made of up to five strata, with two to three being most common.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ and `odata_submission_get`.

See Also

Other included: `fq_attachments`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

fq_data_taxa *Parsed submission data for a subgroup of an ODK Central form.*

Description

Stable

Usage

```
fq_data_taxa
```

Format

The output of `odata_submission_get` for a set of example data. A tidy tibble referencing the attachments included in the vignettes and documentation at a relative path ‘attachments/media/<filename>.<ext>‘.

Details

The parsed OData response for a subgroup of an ODK Central form.

This subgroup represents an individual plant taxon which is encountered by the enumerators. Typically, one voucher specimen is taken for each distinct encountered plant taxon. A field name is allocated by the enumerators, which can be the proper canonical name (if known) or any other moniker. The voucher specimens are later determined by taxonomic experts, who then provide the real, terminal taxonomic name for a given voucher specimen.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ and `odata_submission_get`.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_form_detail` *A tibble of form metadata.*

Description

Stable

Usage

```
fq_form_detail
```

Format

A tibble of form metadata.

Source

The output of `form_detail` run on submissions of the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_form_list` *A tibble of forms.*

Description

Stable

Usage

```
fq_form_list
```

Format

A tibble of forms

Source

The output of `form_list`. run on the project.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw, geo_wkt

fq_form_schema *JSON form schema for an ODK Central form.*

Description

Stable

Stable

Usage

```
fq_form_schema

fq_form_schema
```

Format

The output of ‘ruODK::form_schema()‘, a tibble with columns "type", "name" and "path" and one row per form field.

A tibble of form fields and field types.

Details

The parsed form schema of an ODK Central form.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

This data is used to build vignettes offline and without the need for credentials to an ODK Central server. The test suite ensures that the "canned" data is identical to the "live" data.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ and ‘ruODK::form_schema()‘.

The output of form_schema run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_form_xml` *A nested list of a form definition.*

Description

Stable

Usage

```
fq_form_xml
```

Format

A nested list of a form definition.

Source

The output of `form_xml` run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

fq_meta *OData metadata document for an ODK Central form.*

Description

Stable

Usage

```
fq_meta
```

Format

A list of lists

Details

The OData response for the metadata of an ODK Central form.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw, geo_wkt

fq_project_detail *A tibble of project metadata.*

Description

Stable

Usage

```
fq_project_detail
```

Format

A tibble of project metadata.

Source

The output of `project_detail` run on the project containing the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_project_list` *A tibble of project metadata.*

Description

Stable

Usage

```
fq_project_list
```

Format

A tibble of project metadata.

Source

The output of `project_list` run on all projects on the configured ODK Central server.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

fq_raw *OData submission data for an ODK Central form.*

Description

Stable

Usage

```
fq_raw
```

Format

A list of lists

Details

The OData response for the submissions of an ODK Central form. This form represents a Flora Quadrat, which is a ca 50 by 50 m quadrat of a uniform plant community.

The XML and .odkbuild versions for this form are available as ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ and ‘system.file("extdata", "FloraQuadrat04.odkbuild", package = "ruODK")‘, respectively.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

fq_raw_strata *OData submission data for a subgroup of an ODK Central form.*

Description

Stable

Usage

```
fq_raw_strata
```

Format

A list of lists

Details

The OData response for the subgroup of an ODK Central form.

This subgroup represents vegetation strata as per the NVIS classification. A vegetation stratum is a layer of plants with the same height, and dominated by one or few plant taxa. Plant communities can be made of up to five strata, with two to three being most common.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_raw_taxa` *OData submission data for a subgroup of an ODK Central form.*

Description

Stable

Usage

```
fq_raw_taxa
```

Format

A list of lists

Details

The OData response for a subgroup of an ODK Central form.

This subgroup represents an individual plant taxon which is encountered by the enumerators. Typically, one voucher specimen is taken for each distinct encountered plant taxon. A field name is allocated by the enumerators, which can be the proper canonical name (if known) or any other moniker. The voucher specimens are later determined by taxonomic experts, who then provide the real, terminal taxonomic name for a given voucher specimen.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

See ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_submissions` *A nested list of submission data.*

Description

Stable

Usage

```
fq_submissions
```

Format

A nested list of submission data.

Source

The output of `submission_get` run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘ using submission instance IDs from `submission_list`.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_submission_list` *A tibble of submission metadata.*

Description

Stable

Usage

```
fq_submission_list
```

Format

A tibble of submission metadata.

Source

The output of `submission_list` run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw, geo_wkt

---

fq_svc *OData service document for an ODK Central form.*

Description

Stable

Usage

```
fq_svc
```

Format

A tibble with one row per submission data endpoint.

Details

The OData response for the metadata of an ODK Central form.

This data is kept up to date with the data used in vignettes and package tests. The data is comprised of test records with nonsensical data. The forms used to capture this data are development versions of real-world forms.

Source

OData service document for ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw, geo_wkt

fq_zip_data *A tibble of the main data table of records from a test form.*

Description

Stable

Usage

```
fq_zip_data
```

Format

A tibble of main records from a test form.

Source

submission_export run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw, geo_wkt

fq_zip_strata *A tibble of a repeated sub-group of records from a test form.*

Description

Stable

Usage

```
fq_zip_strata
```

Format

A tibble of repeated sub-group of records from a test form.

Source

submission_export run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fq_zip_taxa` *A tibble of a repeated sub-group of records from a test form.*

Description

Stable

Usage

```
fq_zip_taxa
```

Format

A tibble of repeated sub-group of records from a test form.

Source

`submission_export` run on the test form ‘system.file("extdata", "FloraQuadrat04.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fs_v7` *The parsed XML form_schema of a form from ODK Central v0.6.*

Description

Stable

Usage

```
fs_v7
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 12 rows and 3 columns.

Source

`form_schema_parse(fs_v7_raw)`

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

`fs_v7_raw` *The unparsed XML form_schema of a form from ODK Central v0.6 as nested list.*

Description

Stable

Usage

```
fs_v7_raw
```

Format

An object of class `list` of length 6.

Source

`form_schema(odkc_version = 0.7,parse = FALSE)`

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

geo_fs *The form_schema of a form containing geofields in GeoJSON.*

Description

Stable

Usage

```
geo_fs
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 19 rows and 4 columns.

Source

`form_schema` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

geo_gj *The parsed submissions of a form containing geofields in GeoJSON.*

Description

Stable

Usage

```
geo_gj
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 51 columns.

Source

`odata_submission_get(wkt=FALSE,parse=TRUE)` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_wkt88, geo_wkt_raw, geo_wkt

geo_gj88 *The parsed submissions of a form containing geofields in GeoJSON with trailing empty coordinates present.*

Description

Stable

Usage

```
geo_gj88
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 51 columns.

Details

This issue was fixed in #88. ODK Central versions 0.7 - 0.9 export geotraces and geoshapes with trailing empty coordinates. ruODK has a patch to drop trailing empty coordinates. This dataset is used to test the patch in ruODK.

Source

`odata_submission_get(wkt=FALSE,parse=TRUE)` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw, geo_wkt

geo_gj_raw *The unparsed submissions of a form containing geofields in GeoJSON.*

Description

Stable

Usage

```
geo_gj_raw
```

Format

An object of class `list` of length 2.

Source

`odata_submission_get(wkt=FALSE,parse=FALSE)` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj`, `geo_wkt88`, `geo_wkt_raw`, `geo_wkt`

geo_wkt *The parsed submissions of a form containing geofields in WKT.*

Description

Stable

Usage

```
geo_wkt
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 48 columns.

Source

`odata_submission_get(wkt=TRUE,parse=TRUE)` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt88, geo_wkt_raw

geo_wkt88 *The parsed submissions of a form containing geofields in WKT with trailing empty coordinates present.*

Description

Stable

Usage

```
geo_wkt88
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 48 columns.

Details

This issue was fixed in #88. ODK Central versions 0.7 - 0.9 export geotraces and geoshapes with trailing empty coordinates. ruODK has a patch to drop trailing empty coordinates. This dataset is used to test the patch in ruODK.

Source

`odata_submission_get(wkt=TRUE,parse=TRUE)` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: fq_attachments, fq_data_strata, fq_data_taxa, fq_data, fq_form_detail, fq_form_list, fq_form_schema, fq_form_xml, fq_meta, fq_project_detail, fq_project_list, fq_raw_strata, fq_raw_taxa, fq_raw, fq_submission_list, fq_submissions, fq_svc, fq_zip_data, fq_zip_strata, fq_zip_taxa, fs_v7_raw, fs_v7, geo_fs, geo_gj88, geo_gj_raw, geo_gj, geo_wkt_raw, geo_wkt

geo_wkt_raw *The unparsed submissions of a form containing geofields in WKT.*

Description

Stable

Usage

```
geo_wkt_raw
```

Format

An object of class `list` of length 2.

Source

`odata_submission_get(wkt=TRUE,parse=FALSE)` run on the test form ‘system.file("extdata", "Locations.xml", package = "ruODK")‘.

See Also

Other included: `fq_attachments`, `fq_data_strata`, `fq_data_taxa`, `fq_data`, `fq_form_detail`, `fq_form_list`, `fq_form_schema`, `fq_form_xml`, `fq_meta`, `fq_project_detail`, `fq_project_list`, `fq_raw_strata`, `fq_raw_taxa`, `fq_raw`, `fq_submission_list`, `fq_submissions`, `fq_svc`, `fq_zip_data`, `fq_zip_strata`, `fq_zip_taxa`, `fs_v7_raw`, `fs_v7`, `geo_fs`, `geo_gj88`, `geo_gj_raw`, `geo_gj`, `geo_wkt88`, `geo_wkt`

get_one_attachment *Download one media attachment.*

Description

Stable

Usage

```
get_one_attachment(
  pth,
  fn,
  src,
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  verbose = get_ru_verbose()
)
```

Arguments

pth A local file path to save the attachment to.

fn The attachment filename, as per ODK form submission. If NA, no file will be downloaded, but NA will be returned.

src The attachment's download URL, generated by attachment_url.

url The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").

un The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

verbose Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

This is a helper function used by attachment_get. This function is not vectorised, but mapped by attachment_get to a tibble of input parameters.

Value

The relative local path to the downloaded attachment or NA.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
# Step 1: Setup ruODK with OData Service URL (has url, pid, fid)
ruODK::ru_setup(svc = "...")

# Step 2: Construct attachment_url
att_url <- ruODK:::attachment_url(
  "uuid:d3bcefea-32a8-4dbc-80ca-4ecb0678e2b0",
  "filename.jpg"
)

# Step 3: Get one attachment
local_fn <- get_one_attachment("media/filename.jpg", "filename.jpg", att_url)
```

```
# In real life: done in bulk behind the scenes during odata_submission_get()

## End(Not run)
```

get_one_submission *Download one submission.*

Description

This function is the workhorse for the vectorised function submission_get, which gets all submissions for a list of submission IDs.

Usage

```
get_one_submission(
  iid,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`iid` The ‘instance_id‘, a UUID, as returned by `submission_list`.

`pid` The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default pid through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.

`fid` The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Details

Note this function returns a nested list containing any repeating subgroups. As the presence and length of repeating subgroups is non-deterministic and entirely depends on the completeness of the submission data, we cannot rectangle them any further here. Rectangling requires knowledge of the form schema and the completeness of submission data.

Stable

Value

A nested list of submission data.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/submissions/retrieving-submission-xml

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

# With explicit credentials, see tests
sl <- submission_list()

sub <- get_one_submission(sl$instance_id[[1]])
listviewer::jsonedit(sub)

# The details for one submission depend on the form fields
length(sub)
# > 11

# The items are the field names. Repeated groups have the same name.
names(sub)
# > "meta"                      "encounter_start_datetime" "reporter"
# > "device_id"                 "location"                 "habitat"
# > "vegetation_structure"      "perimeter"                "taxon_encounter"
# > "taxon_encounter"          "encounter_end_datetime"

## End(Not run)
```

get_one_submission_attachment_list

List all attachments of one submission.

Description

Stable

Usage

```
get_one_submission_attachment_list(
  iid,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

iid	The ‘instance_id‘, a UUID, as returned by submission_list.
pid	The numeric ID of the project, e.g.: 2. Default: get_default_pid. Set default pid through ru_setup(pid="..."). See vignette("Setup",package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: get_default_fid. Set default fid through ru_setup(fid="..."). See vignette("Setup",package = "ruODK").
url	The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").
un	The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").
pw	The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Details

When a Submission is created, either over the OpenRosa or the REST interface, its XML data is analysed to determine which file attachments it references: these may be photos or video taken as part of the survey, or an audit/timing log, among other things. Each reference is an expected attachment, and these expectations are recorded permanently alongside the Submission. With this subresource, you can list the expected attachments, see whether the server actually has a copy or not, and download, upload, re-upload, or clear binary data for any particular attachment.

You can retrieve the list of expected Submission attachments at this route, along with a boolean flag indicating whether the server actually has a copy of the expected file or not. If the server has a file, you can then append its filename to the request URL to download only that file.

Value

A tibble containing some high-level details of the submission attachments. One row per submission attachment, columns are submission attributes:

* name: The attachment filename, e.g. 12345.jpg * exists: Whether the attachment for that submission exists on the server.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/attachments/listing-expected-submission-attachments

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/'-form-attachments/listing-expected-form-attachments

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

sl <- submission_list()

al <- get_one_submission_attachment_list(sl$instance_id[[1]])
al %>% knitr::kable(.)
```

```
# attachment_list returns a tibble
class(al)
# > c("tbl_df", "tbl", "data.frame")

# Submission attributes are the tibble's columns
names(al)
# > "name" "exists"

## End(Not run)
```

handle_ru_attachments *Download and link submission attachments according to a form schema.*

Description

Stable

Usage

```
handle_ru_attachments(
  data,
  form_schema,
  local_dir = "media",
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  verbose = get_ru_verbose()
)
```

Arguments

`data`	Submissions rectangled into a tibble. E.g. the output of "‘ ruODK::odata_submission_get(parse = FALSE) ruODK::odata_submission_rectangle() "‘
`form_schema`	The ‘form_schema‘ for the submissions. E.g. the output of ‘ruODK::form_schema()‘.
`local_dir`	The local folder to save the downloaded files to, default: "media".
`pid`	The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.
`fid`	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.
`url`	The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

un The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

verbose Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

For a given tibble of submissions, download and link attachments for all columns which are marked in the form schema as type "binary".

Value

The submissions tibble with all attachments downloaded and linked to a ‘local_dir‘.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
library(magrittr)
data("fq_raw")
data("fq_form_schema")
t <- tempdir()
fs::dir_ls(t) %>% fs::file_delete()
fq_with_att <- fq_raw %>%
  ruODK::odata_submission_rectangle() %>%
  ruODK::handle_ru_attachments(
    form_schema = fq_form_schema,
    local_dir = t,
    pid = ruODK::get_test_pid(),
    fid = ruODK::get_test_fid(),
    url = ruODK::get_test_url(),
    un = ruODK::get_test_un(),
    pw = ruODK::get_test_pw(),
    verbose <- ruODK::get_ru_verbose()
  )
# There should be files in local_dir
testthat::expect_true(fs::dir_ls(t) %>% length() > 0)

## End(Not run)
```

`handle_ru_datetimes` *Parse datetimes of submission data according to a form schema.*

Description

Stable

Usage

```
handle_ru_datetimes(
  data,
  form_schema,
  orders = c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd"),
  tz = get_default_tz(),
  verbose = get_ru_verbose()
)
```

Arguments

`data` Submissions rectangled into a tibble. E.g. the output of “‘ ruODK::odata_submission_get(parse = FALSE) ruODK::odata_submission_rectangle() “‘

`form_schema` The ‘form_schema‘ for the submissions. E.g. the output of ‘ruODK::form_schema()‘.

`orders` (vector of character) Orders of datetime elements for lubridate. Default: `c("YmdHMS","YmdHMSz","Ymd HMS","Ymd HMSz","Ymd","ymd")`.

`tz` A timezone to convert dates and times to. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s timezone can be set globally or per function.

`verbose` Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

For a given tibble of submissions, parse all columns which are marked in the form schema as type "date" or "dateTime" using a set of lubridate orders and a given timezone.

Value

The submissions tibble with all date/dateTime columns mutated as lubridate datetimes.

See Also

Other utilities: `attachment_get()`, `attachment_url()`, `drop_null_coords()`, `form_schema_parse()`, `get_one_attachment()`, `get_one_submission_attachment_list()`, `get_one_submission()`, `handle_ru_attachments()`, `handle_ru_geopoints()`, `handle_ru_geoshapes()`, `handle_ru_geotraces()`, `isodt_to_local()`, `odata_submission_rectangle()`, `predict_ruodk_name()`, `prepend_uuid()`, `ru_msg_abort()`, `ru_msg_info()`, `ru_msg_noop()`, `ru_msg_success()`, `ru_msg_warn()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

Examples

```
## Not run:
library(magrittr)
data("fq_raw")
data("fq_form_schema")

fq_with_dates <- fq_raw %>%
  ruODK::odata_submission_rectangle() %>%
  ruODK::handle_ru_datetimes(form_schema = fq_form_schema)

dplyr::glimpse(fq_with_dates)

## End(Not run)
```

---

handle_ru_geopoints *Split all geopoints of a submission tibble into their components.*

Description

Stable

Usage

```
handle_ru_geopoints(data, form_schema, wkt = FALSE, verbose = get_ru_verbose())
```

Arguments

data	Submissions rectangled into a tibble. E.g. the output of ``‘ ruODK::odata_submission_get(parse = FALSE) ruODK::odata_submission_rectangle() ``‘
form_schema	The ‘form_schema‘ for the submissions. E.g. the output of ‘ruODK::form_schema()‘.
wkt	Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.
verbose	Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘'s verbosity can be set globally or per function.

Details

For a given tibble of submissions, find all columns which are listed in the form schema as type `geopoint`, and extract their components. Extracted components are longitude (X), latitude (Y), altitude (Z, where given), and accuracy (M, where given).

The original column is retained to allow parsing into other spatially enabled formats.

Value

The submissions tibble with all geopoints retained in their original format, plus columns of their coordinate components as provided by `split_geopoint`.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
library(magrittr)
data("gep_fs")
data("geo_gj_raw")
data("geo_wkt_raw")

# GeoJSON
geo_gj_parsed <- geo_gj_raw %>%
  ruODK::odata_submission_rectangle(form_schema = geo_fs) %>%
  ruODK::handle_ru_geopoints(form_schema = geo_fs, wkt = FALSE)

dplyr::glimpse(geo_gj_parsed)

# WKT
geo_wkt_parsed <- geo_wkt_raw %>%
  ruODK::odata_submission_rectangle(form_schema = geo_fs) %>%
  ruODK::handle_ru_geopoints(form_schema = geo_fs, wkt = TRUE)

dplyr::glimpse(geo_wkt_parsed)
```

handle_ru_geoshapes *Split all geoshapes of a submission tibble into their components.*

Description

Stable

Usage

```
handle_ru_geoshapes(
  data,
  form_schema,
  wkt = FALSE,
  odkc_version = get_default_odkc_version(),
  verbose = get_ru_verbose()
)
```

Arguments

`data` Submissions rectangled into a tibble. E.g. the output of “‘ ruODK::odata_submission_get(parse = FALSE) ruODK::odata_submission_rectangle(form_schema = ...) “‘

`form_schema` The ‘form_schema‘ for the submissions. E.g. the output of ‘ruODK::form_schema()‘.

`wkt` Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.

`odkc_version` The ODK Central version as decimal number (major.minor). ‘ruODK‘ uses this parameter to adjust for breaking changes in ODK Central. Default: `get_default_odkc_version` or 0.8 if unset. Set default `get_default_odkc_version` through `ru_setup(odkc_version=0.8)`. See `vignette("Setup",package = "ruODK")`.

`verbose` Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

For a given tibble of submissions, find all columns which are listed in the form schema as type geoshape, and extract their components. Extracted components are longitude (X), latitude (Y), altitude (Z, where given), and accuracy (M, where given) of the first point of the geoshape.

The original column is retained to allow parsing into other spatially enabled formats.

Value

The submissions tibble with all geoshapes retained in their original format, plus columns of their first point’s coordinate components as provided by `split_geoshape`.

See Also

Other utilities: `attachment_get()`, `attachment_url()`, `drop_null_coords()`, `form_schema_parse()`, `get_one_attachment()`, `get_one_submission_attachment_list()`, `get_one_submission()`, `handle_ru_attachments()`, `handle_ru_datetimes()`, `handle_ru_geopoints()`, `handle_ru_geotraces()`, `isodt_to_local()`, `odata_submission_rectangle()`, `predict_ruodk_name()`, `prepend_uuid()`, `ru_msg_abort()`, `ru_msg_info()`, `ru_msg_noop()`, `ru_msg_success()`, `ru_msg_warn()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

Examples

```
## Not run:
library(magrittr)
data("gep_fs")
data("geo_wkt_raw")
data("geo_gj_raw")

# GeoJSON
geo_gj_parsed <- geo_gj_raw %>%
  ruODK::odata_submission_rectangle(form_schema = geo_fs) %>%
  ruODK::handle_ru_geoshapes(form_schema = geo_fs, wkt = FALSE)

dplyr::glimpse(geo_gj_parsed)
```

```
# WKT
geo_wkt_parsed <- geo_wkt_raw %>%
  ruODK::odata_submission_rectangle(form_schema = geo_fs) %>%
  ruODK::handle_ru_geoshapes(form_schema = geo_fs, wkt = TRUE)

dplyr::glimpse(geo_wkt_parsed)

## End(Not run)
```

---

handle_ru_geotraces *Split all geotraces of a submission tibble into their components.*

Description

Stable

Usage

```
handle_ru_geotraces(
  data,
  form_schema,
  wkt = FALSE,
  odkc_version = get_default_odkc_version(),
  verbose = get_ru_verbose()
)
```

Arguments

data	Submissions rectangled into a tibble. E.g. the output of "' ruODK::odata_submission_get(parse = FALSE) ruODK::odata_submission_rectangle(form_schema = ...) "'
form_schema	The 'form_schema' for the submissions. E.g. the output of 'ruODK::form_schema()'.
wkt	Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.
odkc_version	The ODK Central version as decimal number (major.minor). 'ruODK' uses this parameter to adjust for breaking changes in ODK Central. Default: `get_default_odkc_version` or 0.8 if unset. Set default `get_default_odkc_version` through `ru_setup(odkc_version=0.8)`. See `vignette("Setup",package = "ruODK")`.
verbose	Whether to display debug messages or not. Read 'vignette("setup", package = "ruODK")' to learn how 'ruODK''s verbosity can be set globally or per function.

Details

For a given tibble of submissions, find all columns which are listed in the form schema as type `geotrace`, and extract their components. Extracted components are longitude (X), latitude (Y), altitude (Z, where given), and accuracy (M, where given) of the first point of the geotrace.

The original column is retained to allow parsing into other spatially enabled formats.

Value

The submissions tibble with all geotraces retained in their original format, plus columns of their first point's coordinate components as provided by split_geotrace.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
library(magrittr)
data("gep_fs")
data("geo_wkt_raw")
data("geo_gj_raw")

# GeoJSON
geo_gj_parsed <- geo_gj_raw %>%
  ruODK::odata_submission_rectangle(form_schema = geo_fs) %>%
  ruODK::handle_ru_geotraces(form_schema = geo_fs, wkt = FALSE)

dplyr::glimpse(geo_gj_parsed)

# WKT
geo_wkt_parsed <- geo_wkt_raw %>%
  ruODK::odata_submission_rectangle(form_schema = geo_fs) %>%
  ruODK::handle_ru_geotraces(form_schema = geo_fs, wkt = TRUE)

dplyr::glimpse(geo_wkt_parsed)

## End(Not run)
```

odata_metadata_get *Retrieve metadata from an OData URL ending in .svc as list of lists.*

Description

Stable

Usage

```
odata_metadata_get(
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`pid`	The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.
`fid`	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.
`url`	The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.
`un`	The ODK Central username (an email address). Default: `get_default_un`. Set default `un` through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.
`pw`	The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Value

A nested list containing Edmx (dataset schema definition) and .attrs (Version).

See Also

`https://odkcentral.docs.apiary.io/#reference/odata-endpoints/odata-form-service/metadata-document`

Other odata-api: `odata_service_get()`, `odata_submission_get()`

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

meta <- odata_metadata_get()
listviewer::jsonedit(meta)

## End(Not run)
```

odata_service_get *Retrieve service metadata from an OData URL ending in .svc as tibble.*

Description

Stable

Usage

```
odata_service_get(
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

pid	The numeric ID of the project, e.g.: 2. Default: get_default_pid. Set default pid through ru_setup(pid="..."). See vignette("Setup",package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: get_default_fid. Set default fid through ru_setup(fid="..."). See vignette("Setup",package = "ruODK").
url	The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").
un	The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").
pw	The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Value

A tibble with one row per submission data endpoint. Columns: name, kind, url.

See Also

https://odkcentral.docs.apiary.io/#reference/odata-endpoints/odata-form-service/service-document

Other odata-api: odata_metadata_get(), odata_submission_get()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

svc <- odata_service_get()
svc

## End(Not run)
```

---

odata_submission_get *Retrieve and rectangle form submissions, parse dates, geopoints, download and link attachments.*

Description

Stable

Usage

```
odata_submission_get(
  table = "Submissions",
  skip = NULL,
  top = NULL,
  count = FALSE,
  wkt = FALSE,
  parse = TRUE,
  download = TRUE,
  orders = c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd"),
  local_dir = "media",
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  odkc_version = get_default_odkc_version(),
  tz = get_default_tz(),
  verbose = get_ru_verbose()
)
```

Arguments

`table` The submission EntityType, or in plain words, the table name. Default: `Submissions` (the main table). Change to `Submissions.GROUP_NAME` for repeating form groups. The group name can be found through `odata_service_get`.

`skip` The number of rows to be omitted from the results. Example: 10, default: `NA` (none skipped).

`top` The number of rows to return. Example: 100, default: `NA` (all returned).

`count` If TRUE, an `@odata.count` property will be returned in the response from ODK Central. Default: `FALSE`.

`wkt` If TRUE, geospatial data will be returned as WKT (Well Known Text) strings. Default: `FALSE`, returns GeoJSON structures. Note that accuracy is only returned through GeoJSON.

`parse` Whether to parse submission data based on form schema. Dates and datetimes will be parsed into local time. Attachments will be downloaded, and the field updated to the local file path. Point locations will be split into components; GeoJSON (`wkt=FALSE`) will be split into latitude, longitude, altitude and accuracy (with anonymous field names), while WKT will be split into longitude, latitude,and altitude (missing accuracy) prefixed by the original field name. See details for the handling of geotraces and geoshapes. Default: TRUE.

`download` Whether to download attachments to `local_dir` or not. If in the future ODK Central supports hot-linking attachments, this parameter will replace attachment file names with their fully qualified attachment URL. Default: TRUE.

`orders` (vector of character) Orders of datetime elements for lubridate. Default: `c("YmdHMS","YmdHMSz","Ymd HMS","Ymd HMSz","Ymd","ymd")`.

`local_dir` The local folder to save the downloaded files to, default: `"media"`.

`pid` The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.

`fid` The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

`odkc_version` The ODK Central version as decimal number (major.minor). ‘ruODK‘ uses this parameter to adjust for breaking changes in ODK Central. Default: `get_default_odkc_version` or 0.8 if unset. Set default `get_default_odkc_version` through `ru_setup(odkc_version=0.8)`. See `vignette("Setup",package = "ruODK")`.

`tz`	A timezone to convert dates and times to. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s timezone can be set globally or per function.
`verbose`	Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

`odata_submission_get` downloads submissions from (default) the main form group (submission table) including any non-repeating form groups, or from any other table as specified by parameter ‘table‘.

With parameter `parse=TRUE` (default), submission data is parsed into a tibble. Any fields of type `dateTime` or `date` are parsed into dates, with an optional parameter `tz` to specify the local timezone.

A parameter `local_dir` (default: `media`) specifies a local directory for downloaded attachment files. Already existing, previously downloaded attachments will be retained.

With parameter ‘wkt=TRUE‘, spatial fields will be returned as WKT, rather than GeoJSON. In addition, fields of type ‘geopoint‘ will be split into latitude, longitude, and altitude, prefixed with the original field name. E.g. a field ‘start_location‘ of type ‘geopoint‘ will be split into ‘start_location_latitude‘, ‘start_location_longitude‘, and ‘start_location_altitude‘. The field name prefix will allow multiple fields of type ‘geopoint‘ to be split into their components without naming conflicts.

Geotraces (lines) and gepshapes (polygons) will be retained in their original format, plus columns of their first point’s coordinate components as provided by `split_geotrace` and `split_geoshape`, respectively.

The only remaining manual step is to optionally join any sub-tables to the master table.

The parameter `verbose` enables diagnostic messages along the download and parsing process.

With parameter `parse=FALSE`, submission data is presented as nested list, which is the R equivalent of the JSON structure returned from the API. From there, `odata_submission_rectangle` can rectangle the data into a tibble, and subsequent lines of `handle_ru_datetimes`, `handle_ru_attachments`, `handle_ru_geopoints`, `handle_ru_geotraces`, and `handle_ru_geoshapes` parse dates, download and link file attachments, and extract coordinates from geofields. `ruODK` offers this manual and explicit pathway as an option to investigate and narrow down unexpected or unwanted behaviour.

Value

A list of lists.

- `value` contains the submissions as list of lists.
- `@odata.context` is the URL of the metadata.
- `@odata.count` is the total number of rows in the table.

See Also

`https://odkcentral.docs.apiary.io/#reference/odata-endpoints/odata-form-service`

`https://odkcentral.docs.apiary.io/#reference/odata-endpoints/odata-form-service/data-document`

Other odata-api: `odata_metadata_get()`, `odata_service_get()`

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

form_tables <- ruODK::odata_service_get()
data <- odata_submission_get() # default: main data table
data <- odata_submission_get(table = form_tables$url[1]) # same, explicitly
data_sub1 <- odata_submission_get(table = form_tables$url[2]) # sub-table 1
data_sub2 <- odata_submission_get(table = form_tables$url[3]) # sub-table 2

# Skip one row, return the next 1 rows (top), include total row count
data <- odata_submission_get(
  table = form_tables$url[1],
  skip = 1,
  top = 1,
  count = TRUE
)

## End(Not run)
```

---

odata_submission_rectangle

Rectangle the output of `odata_submission_get(parse=FALSE)` *into a tidy tibble and unnest all levels.*

Description

Stable

Usage

```
odata_submission_rectangle(
  data,
  names_repair = "universal",
  names_sep = "_",
  form_schema = NULL,
  verbose = get_ru_verbose()
)
```

Arguments

`data` A nested list of lists as given by `odata_submission_get`.

`names_repair` The argument `names_repair` for `tidyr::unnest_wider`, default: "universal".

`names_sep` The argument `names_sep` for `tidyr::unnest_wider`, default: "_". Un-nested variables inside a list column will be prefixed by the list column name, separated by `names_sep`. This avoids unsightly repaired names such as `latitude...1`.

`form_schema` An optional form_schema, like the output of `form_schema`. If a form schema is supplied, location fields will not be unnested. While WKT location fields contain plain text and will never be unnested, GeoJSON location fields would cause errors during unnesting.

`verbose` Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Value

The submissions as un-nested tibble

See Also

Other utilities: `attachment_get()`, `attachment_url()`, `drop_null_coords()`, `form_schema_parse()`, `get_one_attachment()`, `get_one_submission_attachment_list()`, `get_one_submission()`, `handle_ru_attachments()`, `handle_ru_datetimes()`, `handle_ru_geopoints()`, `handle_ru_geoshapes()`, `handle_ru_geotraces()`, `isodt_to_local()`, `predict_ruodk_name()`, `prepend_uuid()`, `ru_msg_abort()`, `ru_msg_info()`, `ru_msg_noop()`, `ru_msg_success()`, `ru_msg_warn()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

Examples

```
## Not run:
# Using canned data
data_parsed <- odata_submission_rectangle(fq_raw, verbose = TRUE)
# Field "device_id" is known part of fq_raw
testthat::expect_equal(
  data_parsed$device_id[[1]],
  fq_raw$value[[1]]$device_id
)

# fq_raw has two submissions
testthat::expect_equal(length(fq_raw$value), nrow(data_parsed))

## End(Not run)
```

odata_svc_parse *Retrieve URL, project ID, and form ID from an ODK Central OData service URL*

Description

Stable

Usage

```
odata_svc_parse(svc)
```

Arguments

svc	(character) The OData service URL of a form as provided by the ODK Central form submissions tab. Example: "https://sandbox.central.getodk.org/v1/projects/14/forms/build_Flora-Quadrat-0-2_1558575936.svc"

Value

A named list with three components (all of type character):

- `url` The ODK Central base URL.
- `pid` The project ID.
- `fid` The form ID.

See Also

Other ru_settings: `ru_settings()`, `ru_setup()`, `yell_if_error()`, `yell_if_missing()`

project_create *Create a new project.*

Description

Experimental

Usage

```
project_create(
  name,
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`name` The desired name of the project. Can contain whitespace.

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Value

A tibble with one row per project and all project metadata as columns as per ODK Central API docs.

See Also

`https://odkcentral.docs.apiary.io/#reference/project-management/projects/creating-a-project`

Other restful-api: `attachment_link()`, `attachment_list()`, `audit_get()`, `form_detail()`, `form_list()`, `form_schema()`, `form_xml()`, `project_detail()`, `project_list()`, `submission_detail()`, `submission_export()`, `submission_get()`, `submission_list()`

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

p <- project_create("Test Project")
knitr::kable(p)

# project_create returns a tibble
class(p)
# > "tbl_df" "tbl" "data.frame"

# columns are project metadata
names(p)
# > "id" "name" "archived"

## End(Not run)
```

project_detail *List all details of one project.*

Description

While the API endpoint will return all details for one project, `project_detail` will fail with incorrect or missing authentication.

Usage

```
project_detail(
  pid = get_default_pid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

`pid`	The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.
`url`	The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.
`un`	The ODK Central username (an email address). Default: `get_default_un`. Set default `un` through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.
`pw`	The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Details

Stable

Value

A tibble with exactly one row for the project and all project metadata as columns as per ODK Central API docs. Column names are renamed from ODK's ‘camelCase‘ to ‘snake_case‘. Values differ to values returned by ODK Central API:

* archived: FALSE (if NULL) else TRUE * dates: NA if NULL

See Also

https://odkcentral.docs.apiary.io/#reference/project-management/projects/getting-project-details

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_list(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

pd <- project_detail()

pd %>%
  dplyr::select(-"verbs") %>%
  knitr::kable(.)

## End(Not run)
```

project_list *List all projects.*

Description

While the API endpoint will return all projects, project_list will fail with incorrect or missing authentication.

Usage

```
project_list(
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

url The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").

un The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Details

Stable

Value

A tibble with one row per project and all project metadata as columns as per ODK Central API docs.

See Also

https://odkcentral.docs.apiary.io/#reference/project-management/projects/listing-projects

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_detail(), submission_detail(), submission_export(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

pl <- project_list()
knitr::kable(pl)

# project_list returns a tibble
class(pl)
# > "tbl_df" "tbl" "data.frame"

# columns are project metadata
names(pl)
# > "id" "name" "forms" "app_users" "created_at" "updated_at"
# > "last_submission" "archived"

## End(Not run)
```

`ru_msg_abort` *rlang::abort() with a red error message with a cross symbol.*

Description

Stable

Usage

```
ru_msg_abort(message)
```

Arguments

`message`	<chr> A message to print

See Also

Other utilities: `attachment_get()`, `attachment_url()`, `drop_null_coords()`, `form_schema_parse()`, `get_one_attachment()`, `get_one_submission_attachment_list()`, `get_one_submission()`, `handle_ru_attachments()`, `handle_ru_datetimes()`, `handle_ru_geopoints()`, `handle_ru_geoshapes()`, `handle_ru_geotraces()`, `isodt_to_local()`, `odata_submission_rectangle()`, `predict_ruodk_name()`, `prepend_uuid()`, `ru_msg_info()`, `ru_msg_noop()`, `ru_msg_success()`, `ru_msg_warn()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

Examples

```
## Not run:
ru_msg_abort("This is an error, abort.")

## End(Not run)
```

`ru_msg_info` *Print a blue info message with an info symbol.*

Description

Stable

Usage

```
ru_msg_info(message)
```

Arguments

`message`	<chr> A message to print

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
ru_msg_info("This is an info message.")
```

---

ru_msg_noop — *Print a green noop message with a filled circle symbol.*

Description

Stable

Usage

```
ru_msg_noop(message)
```

Arguments

message	<chr> A message to print

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
ru_msg_noop("This is a noop message.")
```

ru_msg_success *Print a green success message with a tick symbol.*

Description

Stable

Usage

```
ru_msg_success(message)
```

Arguments

message	<chr> A message to print

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_warn(), split_geopoint(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
ru_msg_success("This is a success message.")
```

ru_msg_warn *rlang::warn() with a yellow warning message with a warning symbol.*

Description

Stable

Usage

```
ru_msg_warn(message)
```

Arguments

message	<chr> A message to print

See Also

Other utilities: `attachment_get()`, `attachment_url()`, `drop_null_coords()`, `form_schema_parse()`, `get_one_attachment()`, `get_one_submission_attachment_list()`, `get_one_submission()`, `handle_ru_attachments()`, `handle_ru_datetimes()`, `handle_ru_geopoints()`, `handle_ru_geoshapes()`, `handle_ru_geotraces()`, `isodt_to_local()`, `odata_submission_rectangle()`, `predict_ruodk_name()`, `prepend_uuid()`, `ru_msg_abort()`, `ru_msg_info()`, `ru_msg_noop()`, `ru_msg_success()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

Examples

```
## Not run:
ru_msg_warn("This is a warning.")

## End(Not run)
```

`ru_settings` *Get or set* `ruODK` *settings.*

Description

Stable

Usage

```
ru_settings()

get_default_pid()

get_default_fid()

get_default_url()

get_default_un()

get_default_pw()

get_default_tz()

get_test_url()

get_test_un()

get_test_pw()

get_test_pid()

get_test_fid()
```

```
get_test_fid_zip()

get_test_fid_att()

get_test_fid_gap()

get_test_fid_wkt()

get_ru_verbose()

get_default_odkc_version()

get_test_odkc_version()
```

Value

`ru_settings` prints your default ODK Central project ID, form ID, url, username, and password, corresponding optional test server as well as verbosity settings. `ru_setup` sets your production and test settings, while `get_(default/test)_*` get each of those respective settings.

See Also

`ru_setup`, `get_default_pid`, `get_default_fid`, `get_default_url`, `get_default_un`, `get_default_pw`, `get_default_tz`, `get_default_odkc_version`, `get_test_pid`, `get_test_fid`, `get_test_fid_zip`, `get_test_fid_att`, `get_test_fid_gap`, `get_test_fid_wkt`, `get_test_url`, `get_test_un`, `get_test_pw`, `get_test_odkc_version`, `get_ru_verbose`.

Other ru_settings: `odata_svc_parse()`, `ru_setup()`, `yell_if_error()`, `yell_if_missing()`

Examples

```
ru_settings()
```

---

ru_setup *Configure default `ruODK` settings.*

Description

Settings are returned invisibly and additionally printed depending on `get_ru_verbose`.

Usage

```
ru_setup(
  svc = NULL,
  pid = NULL,
  fid = NULL,
  url = NULL,
  un = NULL,
  pw = NULL,
  tz = NULL,
  odkc_version = NULL,
  test_svc = NULL,
  test_pid = NULL,
  test_fid = NULL,
  test_fid_zip = NULL,
  test_fid_att = NULL,
  test_fid_gap = NULL,
  test_fid_wkt = NULL,
  test_url = NULL,
  test_un = NULL,
  test_pw = NULL,
  test_odkc_version = NULL,
  verbose = NULL
)
```

Arguments

svc
: (optional, character) The OData service URL of a form. This parameter will set `pid`, `fid`, and `url`. It is sufficient to supply `svc`, `un`, and `pw`.

pid
: (optional, character) The ID of an existing project on `url`. This will override the project ID from `svc`. A numeric value for `pid` will be converted to character.

fid
: (optional, character) The alphanumeric ID of an existing form in `pid`. This will override the form ID from `svc`.

url
: An ODK Central URL, e.g. "https://sandbox.central.getodk.org". This will override the ODK Central base URL from `svc`.

un
: An ODK Central username which is the email of a "web user" in the specified ODK Central instance `url` (optional, character).

pw
: The password for user `un` (optional, character).

tz
: Global default time zone. ‘ruODK‘’s time zone is determined in order of precedence:

- Function parameter: e.g. `odata_submission_get(tz = "Australia/Perth")`
- ‘ruODK‘ setting: `ru_setup(tz = "Australia/Perth")`
- Environment variable ‘RU_TIMEZONE‘ (e.g. set in ‘.Renviron‘)
- UTC (GMT+00)

odkc_version
: The ODK Central version as major/minor version, e.g. 0.8.

test_svc
: (optional, character) The OData service URL of a test form. This parameter will set `test_pid`, `test_fid`, and `test_url`. It is sufficient to supply `test_svc`, `test_un`, and `test_pw` to configure testing.

test_pid
: (optional, character) The numeric ID of an existing project on `test_url`. This will override the project ID from `test_svc`. A numeric value for `test_pid` will be converted to character.

`test_fid` (optional, character) The alphanumeric ID of an existing form in `test_pid`. This will override the form ID from `test_svc`. This form is used as default form in all tests, examples, vignettes, data, and Rmd templates.

`test_fid_zip` (optional, character) The alphanumeric ID of an existing form in `test_pid`. This will override the form ID from `test_svc`. Provide the form ID of a form with few submissions and without attachments. This form is used to test the repeated download of all form submissions.

`test_fid_att` (optional, character) The alphanumeric ID of an existing form in `test_pid`. This will override the form ID from `test_svc`. Provide the form ID of a form with few submissions and few attachments. This form is used to test downloading and linking attachments.

`test_fid_gap` (optional, character) The alphanumeric ID of an existing form in `test_pid`. This will override the form ID from `test_svc`. Provide the form ID of a form with gaps in the first submission. This form is used to test parsing incomplete submissions.

`test_fid_wkt` (optional, character) The alphanumeric ID of an existing form in `test_pid`. This will override the form ID from `test_svc`. Provide the form ID of a form with geopoints, geotraces, and geoshapes.

`test_url` (optional, character) A valid ODK Central URL for testing. This will override the ODK Central base URL from `svc`.

`test_un` (optional, character) A valid ODK Central username (email) privileged to view the test project(s) at `test_url`.

`test_pw` (optional, character) The valid ODK Central password for `test_un`.

`test_odkc_version`

The ODK Central test server's version as major/minor version, e.g. 0.8.

`verbose` Global default for ‘ruODK‘ verbosity. ‘ruODK‘ verbosity is determined in order of precedence:

- Function parameter: e.g. `odata_submission_get(verbose = TRUE)`
- ‘ruODK‘ setting: `ru_setup(verbose = TRUE)`
- Environment variable ‘RU_VERBOSE‘ (e.g. set in ‘.Renviron‘)
- ‘FALSE‘.

Details

Stable

`ru_setup` sets ODK Central connection details. `ruODK`'s functions default to use the default project ID, form ID, URL, username, and password unless specified explicitly.

Any parameters not specified will remain unchanged. It is therefore possible to set up username and password initially with `ru_setup(un="XXX",pw="XXX")`, and switch between forms with `ru_setup(svc="XXX")`, supplying the form's OData service URL. ODK Central conveniently provides the OData service URL in the form submission tab, which in turn contains base URL, project ID, and form ID.

`ruODK`'s automated tests require a valid ODK Central URL, and a privileged username and password of a "web user" on that ODK Central instance, as well as an existing project and form.

See Also

Other ru_settings: `odata_svc_parse()`, `ru_settings()`, `yell_if_error()`, `yell_if_missing()`

Examples

```
# `ruODK` users only need default settings to their ODK Central:
ru_setup(url = "https://my-odkc.com", un = "me@email.com", pw = "...")

# `ruODK` contributors and maintainers need specific ODK Central
# instances to run tests and build vignettes, see contributing guide:
ru_setup(
  url = "https://odkcentral.dbca.wa.gov.au",
  un = "me@email.com",
  pw = "...",
  test_url = "https://sandbox.central.getodk.org",
  test_un = "me@email.com",
  test_pw = "...",
  test_pid = 14,
  test_fid = "build_Flora-Quadrat-0-2_1558575936",
  test_fid_zip = "build_Spotlighting-0-6_1558333698",
  test_fid_att = "build_Flora-Quadrat-0-1_1558330379",
  test_fid_gap = "build_Turtle-Track-or-Nest-1-0_1569907666",
  test_fid_wkt = "build_Locations_1589344221",
  verbose = TRUE
)
```

`split_geopoint` *Annotate a dataframe containing a geopoint column with lon, lat, alt.*

Description

Stable

Usage

```
split_geopoint(data, colname, wkt = FALSE)
```

Arguments

`data` (dataframe) A dataframe with a geopoint column.

`colname` (chr) The name of the geopoint column. This column will be retained.

`wkt` Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.

Details

This function is used by `handle_ru_geopoints` on all `geopoint` fields as per `form_schema`.

Value

The given dataframe with the WKT POINT column <colname>, plus three new columns, <colname>_longitude, <colname>_latitude, <colname>_altitude. The three new columns are prefixed with the original colname to avoid naming conflicts with any other geopoint columns.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geoshape(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
df_wkt <- tibble::tibble(
  stuff = c("asd", "sdf", "sdf"),
  loc = c(
    "POINT (115.99 -32.12 20.01)",
    "POINT (116.12 -33.34 15.23)",
    "POINT (114.01 -31.56 23.56)"
  )
)
df_wkt_split <- df %>% split_geopoint("loc", wkt = TRUE)
testthat::expect_equal(
  names(df_wkt_split),
  c("stuff", "loc", "loc_longitude", "loc_latitude", "loc_altitude")
)

# With package data
data("gep_fs")
data("geo_wkt_raw")
data("geo_gj_raw")

# Find variable names of geopoints
geo_fields <- geo_fs %>%
  dplyr::filter(type == "geopoint") %>%
  magrittr::extract2("ruodk_name")
geo_fields[1] # First geotrace in data: point_location_point_gps

# Rectangle but don't parse submission data (GeoJSON and WKT)
geo_gj_rt <- geo_gj_raw %>%
  odata_submission_rectangle(form_schema = geo_fs)
geo_wkt_rt <- geo_wkt_raw %>%
  odata_submission_rectangle(form_schema = geo_fs)

# Data with first geopoint split
gj_first_gt <- split_geopoint(geo_gj_rt, geo_fields[1], wkt = FALSE)
gj_first_gt$point_location_point_gps_longitude
```

```
wkt_first_gt <- split_geopoint(geo_wkt_rt, geo_fields[1], wkt = TRUE)
wkt_first_gt$point_location_point_gps_longitude

## End(Not run)
```

---

split_geoshape *Annotate a dataframe containing a geoshape column with lon, lat, alt of the geotrace's first point.*

Description

Stable

Usage

```
split_geoshape(data, colname, wkt = FALSE, odkc_version = odkc_version)
```

Arguments

data	(dataframe) A dataframe with a geoshape column.
colname	(chr) The name of the geoshape column. This column will be retained.
wkt	Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.
odkc_version	The ODK Central version as decimal number (major.minor). 'ruODK' uses this parameter to adjust for breaking changes in ODK Central. Default: get_default_odkc_version or 0.8 if unset. Set default get_default_odkc_version through ru_setup(odkc_version=0.8). See vignette("Setup",package = "ruODK").

Details

This function is used by handle_ru_geopoints on all geopoint fields as per form_schema.

Value

The given dataframe with the geoshape column <colname>, plus three new columns, <colname>_longitude, <colname>_latitude, <colname>_altitude. The three new columns are prefixed with the original colname to avoid naming conflicts with any other geoshape columns.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geotrace(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
library(magrittr)
data("gep_fs")
data("geo_wkt_raw")
data("geo_gj_raw")

# Find variable names of geoshapes
geo_fields <- geo_fs %>%
  dplyr::filter(type == "geoshape") %>%
  magrittr::extract2("ruodk_name")
geo_fields[1] # First geoshape in data: shape_location_shape_gps

# Rectangle but don't parse submission data (GeoJSON and WKT)
geo_gj_rt <- geo_gj_raw %>%
  odata_submission_rectangle(form_schema = geo_fs)
geo_wkt_rt <- geo_wkt_raw %>%
  odata_submission_rectangle(form_schema = geo_fs)

# Data with first geoshape split
gj_first_gt <- split_geoshape(geo_gj_rt, geo_fields[1], wkt = FALSE)
cn_gj <- names(gj_first_gt)
testthat::expect_true("shape_location_shape_gps_longitude" %in% cn_gj)
testthat::expect_true("shape_location_shape_gps_latitude" %in% cn_gj)
testthat::expect_true("shape_location_shape_gps_altitude" %in% cn_gj)
testthat::expect_true(
  is.numeric(gj_first_gt$shape_location_shape_gps_longitude)
)
testthat::expect_true(
  is.numeric(gj_first_gt$shape_location_shape_gps_latitude)
)
testthat::expect_true(
  is.numeric(gj_first_gt$shape_location_shape_gps_altitude)
)

wkt_first_gt <- split_geoshape(geo_wkt_rt, geo_fields[1], wkt = TRUE)
cn_wkt <- names(wkt_first_gt)
testthat::expect_true("shape_location_shape_gps_longitude" %in% cn_wkt)
testthat::expect_true("shape_location_shape_gps_latitude" %in% cn_wkt)
testthat::expect_true("shape_location_shape_gps_altitude" %in% cn_wkt)
testthat::expect_true(
  is.numeric(wkt_first_gt$shape_location_shape_gps_longitude)
)
testthat::expect_true(
  is.numeric(wkt_first_gt$shape_location_shape_gps_latitude)
)
testthat::expect_true(
  is.numeric(wkt_first_gt$shape_location_shape_gps_altitude)
)

## End(Not run)
```

split_geotrace *Annotate a dataframe containing a geotrace column with lon, lat, alt of the geotrace's first point.*

Description

Stable

Usage

```
split_geotrace(
  data,
  colname,
  wkt = FALSE,
  odkc_version = get_default_odkc_version()
)
```

Arguments

`data`	(dataframe) A dataframe with a geotrace column.
`colname`	(chr) The name of the geotrace column. This column will be retained.
`wkt`	Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.
`odkc_version`	The ODK Central version as decimal number (major.minor). 'ruODK' uses this parameter to adjust for breaking changes in ODK Central. Default: `get_default_odkc_version` or 0.8 if unset. Set default `get_default_odkc_version` through `ru_setup(odkc_version=0.8)`. See `vignette("Setup",package = "ruODK")`.

Details

This function is used by `handle_ru_geopoints` on all `geopoint` fields as per `form_schema`.

The format of the geotrace (GeoJSON, WKT, ODK Linestring) is determined via parameters 'wkt' and 'odkc_version', rather than inferred from the class of the column. ODK Linestrings are character vectors without a leading "LINESTRING (", WKT are character vectors with a leading "LINESTRING (", and GeoJSON are list columns.

Value

The given dataframe with the geotrace column <colname>, plus three new columns, `<colname>_longitude`, `<colname>_latitude`, `<colname>_altitude`. The three new columns are prefixed with the original `colname` to avoid naming conflicts with any other geotrace columns.

See Also

Other utilities: attachment_get(), attachment_url(), drop_null_coords(), form_schema_parse(), get_one_attachment(), get_one_submission_attachment_list(), get_one_submission(), handle_ru_attachments(), handle_ru_datetimes(), handle_ru_geopoints(), handle_ru_geoshapes(), handle_ru_geotraces(), isodt_to_local(), odata_submission_rectangle(), predict_ruodk_name(), prepend_uuid(), ru_msg_abort(), ru_msg_info(), ru_msg_noop(), ru_msg_success(), ru_msg_warn(), split_geopoint(), split_geoshape(), strip_uuid(), tidyeval, unnest_all()

Examples

```
## Not run:
library(magrittr)
data("gep_fs")
data("geo_wkt_raw")
data("geo_gj_raw")

# Find variable names of geotraces
geo_fields <- geo_fs %>%
  dplyr::filter(type == "geotrace") %>%
  magrittr::extract2("ruodk_name")
geo_fields[1] # First geotrace in data: path_location_path_gps

# Rectangle but don't parse submission data (GeoJSON and WKT)
geo_gj_rt <- geo_gj_raw %>%
  odata_submission_rectangle(form_schema = geo_fs)
geo_wkt_rt <- geo_wkt_raw %>%
  odata_submission_rectangle(form_schema = geo_fs)

# Data with first geotrace split
gj_first_gt <- split_geotrace(geo_gj_rt, geo_fields[1], wkt = FALSE)
testthat::expect_true(
  "path_location_path_gps_longitude" %in% names(gj_first_gt)
)
testthat::expect_true(
  "path_location_path_gps_latitude" %in% names(gj_first_gt)
)
testthat::expect_true(
  "path_location_path_gps_altitude" %in% names(gj_first_gt)
)
testthat::expect_true(
  is.numeric(gj_first_gt$path_location_path_gps_longitude)
)
testthat::expect_true(
  is.numeric(gj_first_gt$path_location_path_gps_latitude)
)
testthat::expect_true(
  is.numeric(gj_first_gt$path_location_path_gps_altitude)
)

wkt_first_gt <- split_geotrace(geo_wkt_rt, geo_fields[1], wkt = TRUE)
testthat::expect_true(
  "path_location_path_gps_longitude" %in% names(wkt_first_gt)
)
testthat::expect_true(
  "path_location_path_gps_latitude" %in% names(wkt_first_gt)
)
testthat::expect_true(
  "path_location_path_gps_altitude" %in% names(wkt_first_gt)
)
testthat::expect_true(
  is.numeric(wkt_first_gt$path_location_path_gps_longitude)
)
testthat::expect_true(
  is.numeric(wkt_first_gt$path_location_path_gps_latitude)
)
testthat::expect_true(
  is.numeric(wkt_first_gt$path_location_path_gps_altitude)
)

## End(Not run)
```

---

submission_detail *Show metadata for one submission.*

Description

Stable

Usage

```
submission_detail(
  iid,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

iid The ‘instance_id‘, a UUID, as returned by submission_list.

pid The numeric ID of the project, e.g.: 2. Default: get_default_pid. Set default pid through ru_setup(pid="..."). See vignette("Setup",package = "ruODK").

fid The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: get_default_fid. Set default fid through ru_setup(fid="..."). See vignette("Setup",package = "ruODK").

`url` The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.

`un` The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.

`pw` The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.

Value

A nested list of submission metadata.

See Also

`https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/submissions/getting-submission-details`

Other restful-api: `attachment_link()`, `attachment_list()`, `audit_get()`, `form_detail()`, `form_list()`, `form_schema()`, `form_xml()`, `project_create()`, `project_detail()`, `project_list()`, `submission_export()`, `submission_get()`, `submission_list()`

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

sl <- submission_list()

sub <- submission_detail(sl$instance_id[[1]])

# The details for one submission return exactly one row
nrow(sub)
# > 1

# The columns are metadata about the submission
names(sub)
# > "instance_id" "submitter_id" "submitter" "created_at" "updated_at"

## End(Not run)
```

submission_export *Export all form submissions including repeats and attachments to CSV.*

Description

To export all the Submission data associated with a Form, just add .csv.zip to the end of the listing URL. The response will be a zip file containing one or more CSV files, as well as all multimedia attachments associated with the included Submissions.

Usage

```
submission_export(
  local_dir = here::here(),
  overwrite = TRUE,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  verbose = get_ru_verbose()
)
```

Arguments

`local_dir`	The local folder to save the downloaded files to, default: `here::here`.
`overwrite`	Whether to overwrite previously downloaded zip files, default: FALSE
`pid`	The numeric ID of the project, e.g.: 2. Default: `get_default_pid`. Set default `pid` through `ru_setup(pid="...")`. See `vignette("Setup",package = "ruODK")`.
`fid`	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: `get_default_fid`. Set default `fid` through `ru_setup(fid="...")`. See `vignette("Setup",package = "ruODK")`.
`url`	The ODK Central base URL without trailing slash. Default: `get_default_url`. Set default `url` through `ru_setup(url="...")`. See `vignette("Setup",package = "ruODK")`.
`un`	The ODK Central username (an email address). Default: `get_default_un`. Set default un through `ru_setup(un="...")`. See `vignette("Setup",package = "ruODK")`.
`pw`	The ODK Central password. Default: `get_default_pw`. Set default pw through `ru_setup(pw="...")`. See `vignette("Setup",package = "ruODK")`.
`verbose`	Whether to display debug messages or not. Read ‘vignette("setup", package = "ruODK")‘ to learn how ‘ruODK‘’s verbosity can be set globally or per function.

Details

The file will be downloaded to the project root unless specified otherwise (via ‘local_dir‘). Subsequently, the zip file can be extracted. Attachment filenames (e.g. "12345.jpg") should be prepended with ‘media‘ (resulting in e.g. ‘media/12345.jpg‘) in order to represent the relative path to the actual attachment file (as extracted from the zip file).

This function downloads all submissions and attachments in one go. For incremental download of a subset of submissions, use submission_list, choose the submissions of interest (e.g. by submission date), and use their uuids to download them one by one via submission_get. Download attachments as listed for each submission (attachment_list).

Stable

Value

The absolute path to the zip file named "‘fid‘.zip" containing submissions as CSV, plus separate CSVs for any repeating groups, plus any attachments in a subfolder ‘media‘.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/submissions/exporting-form-submissions-to-csv

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_get(), submission_list()

Examples

```
## Not run:
# Set default credentials, see vignette "setup"
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

se <- submission_export()

# Unzip and inspect the loot
t <- tempdir()
f <- unzip(se, exdir = t)
fs::dir_ls(t)
fid <- get_test_fid()
sub <- fs::path(t, glue::glue("{fid}.csv")) %>% readr::read_csv()
sub %>% knitr::kable(.)

## End(Not run)
```

submission_get *Get submissions for a list of submission instance IDs.*

Description

Uses get_one_submission on a list of submission instance IDs (‘iid‘) as returned from submission_list$instance_id. By giving the list of ‘iid‘ to download explicitly, that list can be modified using information not accessible to ‘ruODK‘, e.g. ‘iid‘ can be restricted to "only not already downloaded submissions".

Usage

```
submission_get(
  iid,
  pid = ruODK::get_test_pid(),
  fid = ruODK::get_test_fid(),
  url = ruODK::get_test_url(),
  un = ruODK::get_test_un(),
  pw = ruODK::get_test_pw()
)
```

Arguments

iid	A list of submission instance IDs, e.g. from submission_list$instance_id.
pid	The numeric ID of the project, e.g.: 2. Default: get_default_pid. Set default pid through ru_setup(pid="..."). See vignette("Setup",package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: get_default_fid. Set default fid through ru_setup(fid="..."). See vignette("Setup",package = "ruODK").
url	The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").
un	The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").
pw	The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Value

A nested list of submission data.

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/submissions/retrieving-submission-xml

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_list()

Examples

```
## Not run:
# Step 1: Setup ruODK with OData Service URL (has url, pid, fid)
ruODK::ru_setup(svc = "...")

# Step 2: List all submissions of form
sl <- submission_list()

# Step 3: Get submissions
subs <- submission_get(sl$instance_id)

## End(Not run)
```

---

submission_list   *List all submissions of one form.*

Description

Stable

Usage

```
submission_list(
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw()
)
```

Arguments

pid	The numeric ID of the project, e.g.: 2. Default: get_default_pid. Set default pid through ru_setup(pid="..."). See vignette("Setup",package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: get_default_fid. Set default fid through ru_setup(fid="..."). See vignette("Setup",package = "ruODK").

url — The ODK Central base URL without trailing slash. Default: get_default_url. Set default url through ru_setup(url="..."). See vignette("Setup",package = "ruODK").

un — The ODK Central username (an email address). Default: get_default_un. Set default un through ru_setup(un="..."). See vignette("Setup",package = "ruODK").

pw — The ODK Central password. Default: get_default_pw. Set default pw through ru_setup(pw="..."). See vignette("Setup",package = "ruODK").

Value

A tibble containing some high-level details of the form submissions. One row per submission, columns are submission attributes:

* instance_id: uuid, string. The unique ID for each submission. * submitter_id: user ID, integer. * created_at: time of submission upload, dttm * updated_at: time of submission update on server, dttm or NA

See Also

https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/submissions/listing-all-submissions-on-a-form

Other restful-api: attachment_link(), attachment_list(), audit_get(), form_detail(), form_list(), form_schema(), form_xml(), project_create(), project_detail(), project_list(), submission_detail(), submission_export(), submission_get()

Examples

```
## Not run:
# Set default credentials, see vignette("setup")
ruODK::ru_setup(
  svc = paste0(
    "https://sandbox.central.getodk.org/v1/projects/14/",
    "forms/build_Flora-Quadrat-0-2_1558575936.svc"
  ),
  un = "me@email.com",
  pw = "..."
)

sl <- submission_list()
sl %>% knitr::kable(.)

fl <- form_list()

# submission_list returns a tibble
class(sl)
# > c("tbl_df", "tbl", "data.frame")

# Submission attributes are the tibble's columns
names(sl)
# > "instance_id" "submitter_id" "device_id" "created_at" "updated_at"
# Number of submissions (rows) is same as advertised in form_list
form_list_nsub <- fl %>%
  filter(fid == get_test_fid()) %>%
  magrittr::extract2("submissions") %>%
  as.numeric()
nrow(sl) == form_list_nsub
# > TRUE

## End(Not run)
```

Index