

# Package ‘ruODK’

April 20, 2022

**Type** Package

**Title** An R Client for the ODK Central API

**Version** 1.3.7

**Description** Access and tidy up data from the 'ODK Central' API.  
'ODK Central' is a clearinghouse for digitally captured data  
<<https://docs.getodk.org/central-intro/>>.  
The 'ODK Central' API is documented at <<https://odkcentral.docs.apiary.io/>>.

**License** GPL-3

**URL** <https://docs.ropensci.org/ruODK>,  
<https://github.com/ropensci/ruODK>

**BugReports** <https://github.com/ropensci/ruODK/issues>

**Depends** R (>= 4.0)

**Imports** clisymbols (>= 1.2.0),  
crayon (>= 1.4.1),  
dplyr (>= 1.0.7),  
fs (>= 1.5.0),  
glue (>= 1.4.2),  
here (>= 1.0.1),  
httr (>= 1.4.2),  
janitor (>= 2.1.0),  
lifecycle (>= 1.0.0),  
lubridate (>= 1.7.10),  
magrittr (>= 2.0.1),  
purrr (>= 0.3.4),  
readr (>= 2.0.1),  
rlang (>= 0.4.11),  
stringr (>= 1.4.0),  
tibble (>= 3.1.3),  
tidyr (>= 1.1.3),  
xml2 (>= 1.3.2)

**Suggests** covr (>= 3.5.1),  
DT (>= 0.18),

ggplot2 ( $\geq$  3.3.5),  
 knitr ( $\geq$  1.33),  
 lattice ( $\geq$  0.20-44),  
 leaflet ( $\geq$  2.0.4.1),  
 listviewer ( $\geq$  3.0.0),  
 leafpop ( $\geq$  0.1.0),  
 leafem ( $\geq$  0.1.7),  
 mapview ( $\geq$  2.10.1),  
 rmarkdown ( $\geq$  2.10),  
 roxygen2 ( $\geq$  7.1.1),  
 sf ( $\geq$  1.0-2),  
 terra ( $\geq$  1.5-21),  
 testthat ( $\geq$  3.0.4),  
 tmap ( $\geq$  3.3-2),  
 usethis ( $\geq$  2.0.1)

**VignetteBuilder** knitr

**RdMacros** lifecycle

**Encoding** UTF-8

**Language** en-AU

**LazyData** true

**RoxygenNote** 7.1.2

**X-schema.org-applicationCategory** Data Access

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

**Remotes** r-spatial/leafem,  
 r-spatial/mapview

**Roxygen** list(markdown = TRUE)

## R topics documented:

attachment_get . . . . .	4
attachment_link . . . . .	6
attachment_list . . . . .	8
audit_get . . . . .	9
drop_null_coords . . . . .	11
encryption_key_list . . . . .	13
form_detail . . . . .	15
form_list . . . . .	16
form_schema . . . . .	18
form_schema_ext . . . . .	21
form_schema_parse . . . . .	24
form_xml . . . . .	26
fq_attachments . . . . .	27
fq_data . . . . .	28
fq_data_strata . . . . .	29
fq_data_taxa . . . . .	30

fq_form_detail . . . . .	31
fq_form_list . . . . .	31
fq_form_schema . . . . .	32
fq_form_xml . . . . .	33
fq_meta . . . . .	33
fq_project_detail . . . . .	34
fq_project_list . . . . .	35
fq_raw . . . . .	35
fq_raw_strata . . . . .	36
fq_raw_taxa . . . . .	37
fq_submissions . . . . .	38
fq_submission_list . . . . .	38
fq_svc . . . . .	39
fq_zip_data . . . . .	40
fq_zip_strata . . . . .	40
fq_zip_taxa . . . . .	41
fs_v7 . . . . .	41
fs_v7_raw . . . . .	42
geo_fs . . . . .	43
geo_gj . . . . .	43
geo_gj88 . . . . .	44
geo_gj_raw . . . . .	45
geo_wkt . . . . .	45
geo_wkt88 . . . . .	46
geo_wkt_raw . . . . .	47
get_one_attachment . . . . .	47
get_one_submission . . . . .	49
get_one_submission_attachment_list . . . . .	51
handle_ru_attachments . . . . .	53
handle_ru_datetimes . . . . .	55
handle_ru_geopoints . . . . .	56
handle_ru_geoshapes . . . . .	58
handle_ru_geotraces . . . . .	59
odata_metadata_get . . . . .	61
odata_service_get . . . . .	62
odata_submission_get . . . . .	64
odata_submission_rectangle . . . . .	68
odata_svc_parse . . . . .	69
project_create . . . . .	70
project_detail . . . . .	71
project_list . . . . .	73
ru_msg_abort . . . . .	74
ru_msg_info . . . . .	75
ru_msg_noop . . . . .	75
ru_msg_success . . . . .	76
ru_msg_warn . . . . .	77
ru_settings . . . . .	77
ru_setup . . . . .	79

split_geopoint . . . . .	82
split_geoshape . . . . .	83
split_geotrace . . . . .	85
submission_detail . . . . .	88
submission_export . . . . .	89
submission_get . . . . .	92
submission_list . . . . .	94
user_list . . . . .	96

<b>Index</b>	<b>99</b>
--------------	-----------

---

attachment_get	<i>Download attachments and return the local path.</i>
----------------	--

---

**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(),  
    retries = get_retries(),  
    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 <a href="#">submission_export</a> , which keeps all attachment files together in a folder media. Enable this option if downloaded files collide on identical 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	<p>The numeric ID of the project, e.g.: 2.</p> <p>Default: <a href="#">get_default_pid</a>.</p> <p>Set default pid through <code>ru_setup(pid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
fid	<p>The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147".</p> <p>Default: <a href="#">get_default_fid</a>.</p> <p>Set default fid through <code>ru_setup(fid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
url	<p>The ODK Central base URL without trailing slash.</p> <p>Default: <a href="#">get_default_url</a>.</p> <p>Set default url through <code>ru_setup(url="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
un	<p>The ODK Central username (an email address). Default: <a href="#">get_default_un</a>. Set default un through <code>ru_setup(un="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
pw	<p>The ODK Central password. Default: <a href="#">get_default_pw</a>. Set default pw through <code>ru_setup(pw="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
retries	<p>The number of attempts to retrieve a web resource.</p> <p>This parameter is given to <code>RETRY(times = retries)</code>.</p> <p>Default: 3.</p>
verbose	<p>Whether to display debug messages or not.</p> <p>Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's verbosity can be set globally or per function.</p>

## 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_link()`, `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()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

## Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")

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	<i>Prefix attachment columns from CSV export with a local attachment file path.</i>
-----------------	---

---

## Description

[Stable]

## Usage

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

## Arguments

data_tbl	The downloaded submissions from <a href="#">submission_export</a> read into a tibble by <code>readr::read_csv</code> .
form_schema	The form_schema for the submissions. E.g. the output of <code>ruODK::form_schema()</code> .
att_path	A local path, default: "media" (as per .csv.zip export). Selected columns of the dataframe (containing attachment 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 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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geoshape\(\)](#), [split\\_geotrace\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

## Examples

```
## Not run:
t <- tempdir()
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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	<i>List all attachments for a list of submission instances.</i>
-----------------	---

---

## 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(),
  retries = get_retries()
)

```

## Arguments

iid	A list of submission instance IDs, e.g. from <code>submission_list\$instance_id</code> .
pid	The numeric ID of the project, e.g.: 2. Default: <code>get_default_pid</code> . Set default pid through <code>ru_setup(pid="...")</code> . See vignette("Setup", package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <code>get_default_fid</code> . Set default fid through <code>ru_setup(fid="...")</code> . See vignette("Setup", package = "ruODK").
url	The ODK Central base URL without trailing slash. Default: <code>get_default_url</code> . Set default url through <code>ru_setup(url="...")</code> . See vignette("Setup", package = "ruODK").
un	The ODK Central username (an email address). Default: <code>get_default_un</code> . Set default un through <code>ru_setup(un="...")</code> . See vignette("Setup", package = "ruODK").
pw	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See vignette("Setup", package = "ruODK").
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.



**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 submission-management: [encryption\\_key\\_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"),
    retries = get_retries()
  )

```

## Arguments

action	string. The action to filter the logs, e.g. "user.create". See <a href="https://odkcentral.docs.apiary.io/#reference/system-endpoints/server-audit-logs/">https://odkcentral.docs.apiary.io/#reference/system-endpoints/server-audit-logs/</a> 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: <code>get_default_url</code> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <code>get_default_un</code> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

## Details

Parameters to filter the audit logs: `action=form.create&start=2000-01-01z&end=2000-12-31T23%3A59.999z`

## 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: dtm. Time of action on server.

**See Also**

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

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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\\_link\(\)](#), [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\(\)](#), [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)

```

---

encryption\_key\_list     *List all encryption keys for a form.*

---

## Description

**[Maturing]**

## Usage

```

encryption_key_list(
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  retries = get_retries(),
  orders = c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd"),
  tz = get_default_tz()
)

```

## Arguments

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

<code>pw</code>	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
<code>retries</code>	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
<code>orders</code>	(vector of character) Orders of datetime elements for lubridate. Default: <code>c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd")</code> .
<code>tz</code>	A timezone to convert dates and times to. Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's timezone can be set globally or per function.

### Details

This endpoint provides a listing of all known encryption keys needed to decrypt all Submissions for a given Form. It will return at least the `base64RsaPublicKey` property (as column `public`) of all known versions of the form that have submissions against them. If managed keys are being used and a hint was provided, that will be returned as well.

### Value

A tibble of encryption keys.

### See Also

<https://odkcentral.docs.apiary.io/#reference/forms-and-submissions/submissions/listing-encryption-keys>

Other submission-management: `attachment_list()`, `submission_detail()`, `submission_export()`, `submission_get()`, `submission_list()`

### Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")

x <- encryption_key_list(
  pid = Sys.getenv("ODKC_TEST_PID_ENC"),
  fid = Sys.getenv("ODKC_TEST_FID_ENC"),
  url = get_test_url(),
  un = get_test_un(),
  pw = get_test_pw()
)

names(x)
# > [1] "id" "public" "managed" "hint" "created_at"

## End(Not run)
```

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(),
  retries = get_retries()
)
```

**Arguments**

pid	<p>The numeric ID of the project, e.g.: 2.</p> <p>Default: <a href="#">get_default_pid</a>.</p> <p>Set default pid through <code>ru_setup(pid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
fid	<p>The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147".</p> <p>Default: <a href="#">get_default_fid</a>.</p> <p>Set default fid through <code>ru_setup(fid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
url	<p>The ODK Central base URL without trailing slash.</p> <p>Default: <a href="#">get_default_url</a>.</p> <p>Set default url through <code>ru_setup(url="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
un	<p>The ODK Central username (an email address). Default: <a href="#">get_default_un</a>. Set default un through <code>ru_setup(un="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
pw	<p>The ODK Central password. Default: <a href="#">get_default_pw</a>. Set default pw through <code>ru_setup(pw="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
retries	<p>The number of attempts to retrieve a web resource.</p> <p>This parameter is given to <code>RETRY(times = retries)</code>.</p> <p>Default: 3.</p>

**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 form-management: `form_list()`, `form_schema_ext()`, `form_schema()`, `form_xml()`

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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" "published_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(),
  retries = get_retries()
)
```



**Arguments**

pid	The numeric ID of the project, e.g.: 2. Default: <code>get_default_pid</code> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <code>get_default_url</code> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <code>get_default_un</code> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

**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 form-management: `form_detail()`, `form_schema_ext()`, `form_schema()`, `form_xml()`

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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")

# Filter out draft forms (published_at=NA)
only_published_forms <- fl %>% dplyr::filter(is.na(published_at))

# Note: older ODK Central versions < 1.1 have published_at = NA for both
# published and draft forms. Drafts have NA for version and hash.
```

```
only_published_forms <- fl %>% dplyr::filter(is.na(version) & is.na(hash))

## End(Not run)
```

---

form\_schema

*Show the schema of one form.*


---

## Description

**[Stable]**

## Usage

```
form_schema(
  flatten = FALSE,
  odata = FALSE,
  parse = TRUE,
  draft = FALSE,
  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(),
  retries = get_retries(),
  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 <a href="#">form_schema_parse</a> internally. If used together with <code>flatten=TRUE</code> , <a href="#">form_schema</a> will raise a warning and return the unparsed, flattened form schema. Only applies to ODK Central version < 0.8. Default: TRUE.
draft	Whether the form is published (FALSE) or a draft (TRUE). Default: TRUE.
pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .

fid	<p>The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147".</p> <p>Default: <a href="#">get_default_fid</a>.</p> <p>Set default fid through <code>ru_setup(fid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
url	<p>The ODK Central base URL without trailing slash.</p> <p>Default: <a href="#">get_default_url</a>.</p> <p>Set default url through <code>ru_setup(url="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
un	<p>The ODK Central username (an email address). Default: <a href="#">get_default_un</a>. Set default un through <code>ru_setup(un="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
pw	<p>The ODK Central password. Default: <a href="#">get_default_pw</a>. Set default pw through <code>ru_setup(pw="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
odkc_version	<p>The ODK Central version as decimal number (major.minor). ruODK uses this parameter to adjust for breaking changes in ODK Central.</p> <p>Default: <a href="#">get_default_odkc_version</a> or 1.1 if unset.</p> <p>Set default <code>get_default_odkc_version</code> through <code>ru_setup(odkc_version=1.1)</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
retries	<p>The number of attempts to retrieve a web resource.</p> <p>This parameter is given to <code>RETRY(times = retries)</code>.</p> <p>Default: 3.</p>
verbose	<p>Whether to display debug messages or not.</p> <p>Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's verbosity can be set globally or per function.</p>

## 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, additionally cleaned with `make_clean_names` to match the cleaned column names from `odata_submission_rectangle`.

### 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 form-management: `form_detail()`, `form_list()`, `form_schema_ext()`, `form_xml()`

### Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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\_ext

*Show the extended schema of one form.***Description****[Experimental]****Usage**

```
form_schema_ext(
  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(),
  retries = get_retries(),
  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 <a href="#">form_schema_parse</a> internally. If used together with <code>flatten=TRUE</code> , <a href="#">form_schema</a> 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: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See vignette("Setup", package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See vignette("Setup", package = "ruODK").

url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
odkc_version	The ODK Central version as decimal number (major.minor). <code>ruODK</code> uses this parameter to adjust for breaking changes in ODK Central. Default: <a href="#">get_default_odkc_version</a> or 1.1 if unset. Set default <code>get_default_odkc_version</code> through <code>ru_setup(odkc_version=1.1)</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> to learn how <code>ruODK</code> '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. This list is returned by [form\\_schema](#).

However this still misses important elements, in particular labels and choice\_lists.

[form\\_schema\\_ext](#) returns the same object as [form\\_schema](#) adding labels and choice lists in all languages available. This is done by using the return object from [form\\_xml](#).

It has the exact function signature as [form\\_schema](#). In that sense, any call to [form\\_schema](#) can be replaced by [form\\_schema\\_ext](#).

This function, however, has been prepared with ODK Central version 0.8 or higher. If you use it with an earlier version, a warning will be given.

## Value

A tibble containing the form definition. 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, additionally cleaned with [make\\_clean\\_names](#) to match the cleaned column names from [odata\\_submission\\_rectangle](#).

- **label** The field label as given in the form schema. If specific languages are available, this column will return the default language or it will be empty if this is not specified.
- **label\_lang** The field label in language *\_lang* as given in the form schema.
- **choices** A list of lists containing at least values and, if available, labels of the choices as given in the form schema. If specific languages are available, this column will return the default language or it will be empty if this is not specified. Please notice that whenever choice filters are applied, this will return the unfiltered choice list.
- **choices\_lang** A list of lists containing at least values and, if available, labels of the choices in language *\_lang* as given in the form schema. Please notice that whenever choice filters are applied, this will return the unfiltered choice list.

### 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>

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

Other form-management: [form\\_detail\(\)](#), [form\\_list\(\)](#), [form\\_schema\(\)](#), [form\\_xml\(\)](#)

### Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")

# With current ODK Central (>0.7)
# get extended schema:
fsx <- form_schema_ext()

# print choice list in english:
fsx[fsx$name == "test_yn", "choices_english_(en)"][[1]]

# view the extended schema:
fsx

## 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". <a href="#">form_schema_parse</a> 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\\_link\(\)](#), [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\(\)](#), [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(),
  retries = get_retries()
)

```

## Arguments

parse	Whether to parse the XML into a nested list, default: TRUE
pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

**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 form-management: [form\\_detail\(\)](#), [form\\_list\(\)](#), [form\\_schema\\_ext\(\)](#), [form\\_schema\(\)](#)

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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	<i>Parsed submission data for an ODK Central form.</i>
---------	--

---

**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	<i>A tibble of form metadata.</i>
----------------	-----------------------------------

---

**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	<i>A tibble of forms.</i>
--------------	---------------------------

---

**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]****Usage**

fq\_form\_schema

**Format**

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

**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()`.

**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](#)



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	<i>A tibble of project metadata.</i>
-------------------	--------------------------------------

---

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	<i>A tibble of project metadata.</i>
-----------------	--------------------------------------

---

**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	<i>OData submission data for an ODK Central form.</i>
--------	---

---

**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	<i>OData submission data for a subgroup of an ODK Central form.</i>
---------------	---

---

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](#)

---

<code>fq_submissions</code>	<i>A nested list of submission data.</i>
-----------------------------	--

---

**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](#)

---

<code>fq_submission_list</code>	<i>A tibble of submission metadata.</i>
---------------------------------	---

---

**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	<i>A tibble of the main data table of records from a test form.</i>
-------------	---

---

**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	<i>A tibble of a repeated sub-group of records from a test form.</i>
---------------	--

---

**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	<i>The unparsed XML form_schema of a form from ODK Central v0.6 as nested list.</i>
-----------	---

---

**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	<i>The form_schema of a form containing geofields in GeoJSON.</i>
--------	---

---

**Description**

[Stable]

**Usage**

geo\_fs

**Format**

An object of class tbl\_df (inherits from tbl, data.frame) with 19 rows and 6 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	<i>The parsed submissions of a form containing geofields in GeoJSON.</i>
--------	--

---

**Description**

[Stable]

**Usage**

geo\_gj

**Format**

An object of class tbl\_df (inherits from tbl, data.frame) with 1 rows and 58 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	<i>The parsed submissions of a form containing geofields in GeoJSON with trailing empty coordinates present.</i>
----------	--

---

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 geotracess 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", pac = "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	<i>The unparsed submissions of a form containing geofields in GeoJSON.</i>
------------	--

---

**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	<i>The parsed submissions of a form containing geofields in WKT.</i>
---------	--

---

**Description**

[Stable]

**Usage**

geo\_wkt

**Format**

An object of class tbl\_df (inherits from tbl, data.frame) with 1 rows and 55 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 geotracess 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", pack = "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	<i>The unparsed submissions of a form containing geofields in WKT.</i>
-------------	--

---

**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	<i>Download one media attachment.</i>
--------------------	---------------------------------------

---

**Description**

[Stable]

**Usage**

```
get_one_attachment(  
  pth,  
  fn,  
  src,  
  url = get_default_url(),  
  un = get_default_un(),  
  pw = get_default_pw(),  
  retries = get_retries(),  
  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. If the file does not exist in ODK Central, a warning will be emitted.
src	The attachment's download URL, generated by <a href="#">attachment_url</a> . The src must contain the uuid: prefix. Note that the main Submissions table contains the submission id in the field id, whereas nested sub-tables contain the submission id in the field submissions_id.
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> 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\\_link\(\)](#), [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\(\)](#), [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)
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")

# 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	<i>Download one submission.</i>
--------------------	---------------------------------

---

## 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(),
  retries = get_retries()
)
```

## Arguments

<code>iid</code>	The instance_id, a UUID, as returned by <a href="#">submission_list</a> .
<code>pid</code>	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .

fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

## 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\\_link\(\)](#), [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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geoshape\(\)](#), [split\\_geotrace\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

## Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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(),
  retries = get_retries()
)

```

## Arguments

iid	The instance_id, a UUID, as returned by <a href="#">submission_list</a> .
pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See vignette("Setup", package = "ruODK").
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See vignette("Setup", package = "ruODK").

<code>url</code>	The ODK Central base URL without trailing slash. Default: <code>get_default_url</code> . Set default url through <code>ru_setup(url="...")</code> . See vignette("Setup", package = "ruODK").
<code>un</code>	The ODK Central username (an email address). Default: <code>get_default_un</code> . Set default un through <code>ru_setup(un="...")</code> . See vignette("Setup", package = "ruODK").
<code>pw</code>	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See vignette("Setup", package = "ruODK").
<code>retries</code>	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

## 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_link()`, `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()`, `split_geopoint()`, `split_geoshape()`, `split_geotrace()`, `strip_uuid()`, `tidyeval`, `unnest_all()`

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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(),
  retries = get_retries(),
  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 <code>ruODK::form_schema()</code> .
local_dir	The local folder to save the downloaded files to, default: "media".
pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> 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\\_link\(\)](#), [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\(\)](#), [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	<i>Parse datetimes of submission data according to a form schema.</i>
---------------------	---

---

**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: <code>c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd")</code> .
tz	A timezone to convert dates and times to. Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's timezone can be set globally or per function.
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> 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_link()`, `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()`, `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	<i>Split all geopoints of a submission tibble into their components.</i>
---------------------	--

---

## Description

[Stable]



**Usage**

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

**Arguments**

<code>data</code>	Submissions rectangled into a tibble. E.g. the output of <code>ruODK::odata_submission_get(parse = FALSE) %&gt;% ruODK::odata_submission_rectangle()</code>
<code>form_schema</code>	The <code>form_schema</code> for the submissions. E.g. the output of <code>ruODK::form_schema()</code> .
<code>wkt</code>	Whether geofields are GeoJSON (if <code>FALSE</code> ) or WKT strings (if <code>TRUE</code> ), default: <code>FALSE</code> .
<code>verbose</code>	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> to learn how <code>ruODK</code> '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\\_link\(\)](#), [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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geoshape\(\)](#), [split\\_geotrace\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

**Examples**

```
library(magrittr)
data("geo_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	<i>Split all geoshapes of a submission tibble into their components.</i>
---------------------	--

---

## 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  <pre>ruODK::odata_submission_get(parse = FALSE) %&gt;% ruODK::odata_submission_rectangle(form_schema = ...)</pre>
form_schema	The form_schema for the submissions. E.g. the output of <code>ruODK::form_schema()</code> .
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: <a href="#">get_default_odkc_version</a> or 1.1 if unset. Set default <code>get_default_odkc_version</code> through <code>ru_setup(odkc_version=1.1)</code> . See <code>vignette("Setup", package = "ruODK")</code> .
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> 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\\_link\(\)](#), [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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geoshape\(\)](#), [split\\_geotrace\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

**Examples**

```
## Not run:
library(magrittr)
data("geo_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	<i>Split all geotraces of a submission tibble into their components.</i>
---------------------	--

---

**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: <a href="#">get_default_odkc_version</a> or 1.1 if unset. Set default <a href="#">get_default_odkc_version</a> through <a href="#">ru_setup(odkc_version=1.1)</a> . See <a href="#">vignette("Setup", package = "ruODK")</a> .
verbose	Whether to display debug messages or not. Read <a href="#">vignette("setup", package = "ruODK")</a> 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\\_link\(\)](#), [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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geoshape\(\)](#), [split\\_geotrace\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

**Examples**

```
## Not run:
library(magrittr)
data("geo_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	<i>Retrieve metadata from an OData URL ending in .svc as list of lists.</i>
--------------------	---

---

**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(),
  retries = get_retries()
)
```

**Arguments**

pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
-----	--

fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through ru_setup(fid="..."). See vignette("Setup", package = "ruODK").
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through ru_setup(url="..."). See vignette("Setup", package = "ruODK").
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through ru_setup(un="..."). See vignette("Setup", package = "ruODK").
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through ru_setup(pw="..."). See vignette("Setup", package = "ruODK").
retries	The number of attempts to retrieve a web resource. This parameter is given to <a href="#">RETRY</a> (times = retries). Default: 3.

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:  
# See vignette("setup") for setup and authentication options  
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")  
  
meta <- odata_metadata_get()  
listviewer::jsonedit(meta)  
  
## End(Not run)
```

---

odata_service_get	<i>Retrieve service metadata from an OData URL ending in .svc as tibble.</i>
-------------------	--

---

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(),
  retries = get_retries()
)
```

**Arguments**

pid	<p>The numeric ID of the project, e.g.: 2.</p> <p>Default: <a href="#">get_default_pid</a>.</p> <p>Set default pid through <code>ru_setup(pid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
fid	<p>The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147".</p> <p>Default: <a href="#">get_default_fid</a>.</p> <p>Set default fid through <code>ru_setup(fid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
url	<p>The ODK Central base URL without trailing slash.</p> <p>Default: <a href="#">get_default_url</a>.</p> <p>Set default url through <code>ru_setup(url="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
un	<p>The ODK Central username (an email address). Default: <a href="#">get_default_un</a>. Set default un through <code>ru_setup(un="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
pw	<p>The ODK Central password. Default: <a href="#">get_default_pw</a>. Set default pw through <code>ru_setup(pw="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
retries	<p>The number of attempts to retrieve a web resource.</p> <p>This parameter is given to <a href="#">RETRY</a>(times = retries).</p> <p>Default: 3.</p>

**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:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")

svc <- odata_service_get()
svc

## End(Not run)
```

---

odata_submission_get	<i>Retrieve and rectangle form submissions, parse dates, geopoints, download and link attachments.</i>
----------------------	--

---

**Description****[Stable]****Usage**

```
odata_submission_get(
  table = "Submissions",
  skip = NULL,
  top = NULL,
  count = FALSE,
  wkt = FALSE,
  expand = FALSE,
  filter = NULL,
  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(),
  retries = get_retries(),
  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 <a href="#">odata_service_get</a> .
-------	--



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.
expand	If TRUE, all subtables will be expanded and included with column names containing the number of the repeat, the group name, and the field name. This parameter is supported from ODK Central v1.2 onwards and will be ignored on earlier versions of ODK Central. The version is inferred from the parameter odkc_version. Default: FALSE.
filter	If provided, will filter responses to those matching the query. For an odkc_version below 1.1, this parameter will be discarded. In ODK Central v1.1, only the fields system/submitterId and system/submissionDate are available to reference. In ODK Central v1.2, other fields may become available. The operators lt, lte, eq, neq, gte, gt, not, and, or are supported, and the built-in functions now, year, month, day, hour, minute, second. ruODK does not validate the query string given to filter. It is highly recommended to refer to the ODK Central API documentation as well as the <a href="#">OData spec on filters</a> . for filter options and capabilities.
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", "YmdHMS", "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: <a href="#">get_default_pid</a> . 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: <a href="#">get_default_fid</a> . Set default fid through ru_setup(fid="..."). See vignette("Setup", package = "ruODK").

url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
odkc_version	The ODK Central version as decimal number (major.minor). ruODK uses this parameter to adjust for breaking changes in ODK Central. Default: <a href="#">get_default_odkc_version</a> or 1.1 if unset. Set default <code>get_default_odkc_version</code> through <code>ru_setup(odkc_version=1.1)</code> . See <code>vignette("Setup", package = "ruODK")</code> .
tz	A timezone to convert dates and times to. Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's timezone can be set globally or per function.
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> 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.

Geotracess (lines) and gepshapess (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.

Entirely unpopulated form fields, as well as notes and form groups, will be excluded from the resulting tibble. Submitting at least one complete form instance will prevent the accidental exclusion of an otherwise mostly empty form field.

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:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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
)

# Filter submissions
data <- odata_submission_get(
  table = form_tables$url[1],
  filter = "year(__system/submissionDate) lt year(now())"
)
data <- odata_submission_get(
```

```

    table = form_tables$url[1],
    filter = "year(__system/submissionDate) lt 2020"
  )

  ## 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 <a href="#">odata_submission_get</a> .
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 <a href="#">form_schema</a> . 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_link()`, `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()`, `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	<i>Retrieve URL, project ID, and form ID from an ODK Central OData service URL.</i>
-----------------	---

---

**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	<i>Create a new project.</i>
----------------	------------------------------

---

**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: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .

**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 project-management: [project\\_detail\(\)](#), [project\\_list\(\)](#)

## Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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	<i>List all details of one project.</i>
----------------	---

---

## 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(),
  retries = get_retries()
)
```

## Arguments

pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .

un	The ODK Central username (an email address). Default: <code>get_default_un</code> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

## 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 project-management: `project_create()`, `project_list()`

## Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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(),
  retries = get_retries(),
  orders = c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd"),
  tz = get_default_tz()
)
```

## Arguments

url	The ODK Central base URL without trailing slash. Default: <code>get_default_url</code> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <code>get_default_un</code> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <code>get_default_pw</code> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
orders	(vector of character) Orders of datetime elements for lubridate. Default: <code>c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd")</code> .
tz	A timezone to convert dates and times to. Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's timezone can be set globally or per function.

## 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 project-management: [project\\_create\(\)](#), [project\\_detail\(\)](#)

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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 messaging: [ru\\_msg\\_info\(\)](#), [ru\\_msg\\_noop\(\)](#), [ru\\_msg\\_success\(\)](#), [ru\\_msg\\_warn\(\)](#)

## Examples

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

## End(Not run)
```

---

ru_msg_info	<i>Print a blue info message with an info symbol.</i>
-------------	---

---

## Description

[Stable]

## Usage

```
ru_msg_info(message, verbose = get_ru_verbose())
```

## Arguments

message	(chr) A message to print
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.

## See Also

Other messaging: [ru\\_msg\\_abort\(\)](#), [ru\\_msg\\_noop\(\)](#), [ru\\_msg\\_success\(\)](#), [ru\\_msg\\_warn\(\)](#)

## Examples

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

---

ru_msg_noop	<i>Print a green noop message with a filled circle symbol.</i>
-------------	--

---

## Description

[Stable]

## Usage

```
ru_msg_noop(message, verbose = get_ru_verbose())
```

**Arguments**

message	(chr) A message to print
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.

**See Also**

Other messaging: [ru\\_msg\\_abort\(\)](#), [ru\\_msg\\_info\(\)](#), [ru\\_msg\\_success\(\)](#), [ru\\_msg\\_warn\(\)](#)

**Examples**

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

---

ru_msg_success	<i>Print a green success message with a tick symbol.</i>
----------------	--

---

**Description**

**[Stable]**

**Usage**

```
ru_msg_success(message, verbose = get_ru_verbose())
```

**Arguments**

message	(chr) A message to print
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.

**See Also**

Other messaging: [ru\\_msg\\_abort\(\)](#), [ru\\_msg\\_info\(\)](#), [ru\\_msg\\_noop\(\)](#), [ru\\_msg\\_warn\(\)](#)

**Examples**

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

---

ru_msg_warn	<i>rlang::warn()</i> with a yellow warning message with a warning symbol.
-------------	---

---

**Description****[Stable]****Usage**

```
ru_msg_warn(message, verbose = get_ru_verbose())
```

**Arguments**

message	(chr) A message to print
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.

**See Also**

Other messaging: [ru\\_msg\\_abort\(\)](#), [ru\\_msg\\_info\(\)](#), [ru\\_msg\\_noop\(\)](#), [ru\\_msg\\_success\(\)](#)

**Examples**

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

---

ru_settings	<i>Get or set ruODK settings.</i>
-------------	-----------------------------------

---

**Description****[Stable]****Usage**

```
ru_settings()  
  
get_default_pid()  
  
get_default_fid()  
  
get_default_url()
```

```
get_default_un()
get_default_pw()
get_default_pp()
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_test_pp()
get_ru_verbose()
get_default_odkc_version()
get_test_odkc_version()
get_retries()
```

### Value

[ru\\_settings](#) prints your default ODK Central project ID, form ID, url, username, and password, corresponding optional test server as well as verbosity and HTTP request 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\\_pp](#), [get\\_default\\_tz](#), [get\\_default\\_odkc\\_version](#), [get\\_retries](#), [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\\_pp](#), [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	<i>Configure default <a href="#">ruODK</a> settings.</i>
----------	--

---

## 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,
  pp = NULL,
  tz = NULL,
  odkc_version = NULL,
  retries = NULL,
  verbose = 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_pp = NULL,
  test_odkc_version = 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).
pp	The passphrase (optional, character) for an encrypted form.
tz	Global default time zone. ruODK's time zone is determined in order of precedence: <ul style="list-style-type: none"> <li>• Function parameter: e.g. <code>odata_submission_get(tz = "Australia/Perth")</code></li> <li>• ruODK setting: <code>ru_setup(tz = "Australia/Perth")</code></li> <li>• Environment variable RU_TIMEZONE (e.g. set in <code>.Renviron</code>)</li> <li>• UTC (GMT+00)</li> </ul>
odkc_version	The ODK Central version as major/minor version, e.g. 1.1.
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
verbose	Global default for ruODK verbosity. ruODK verbosity is determined in order of precedence: <ul style="list-style-type: none"> <li>• Function parameter: e.g. <code>odata_submission_get(verbose = TRUE)</code></li> <li>• ruODK setting: <code>ru_setup(verbose = TRUE)</code></li> <li>• Environment variable RU_VERBOSE (e.g. set in <code>.Renviron</code>)</li> <li>• FALSE.</li> </ul>
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_pp	(optional, character) The valid passphrase to decrypt the data of encrypted project test_pid for download. Only used for tests.
test_odkc_version	The ODK Central test server's version as major/minor version, e.g. 1.1.

## 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 = "...",
  pp = "...",
  test_url = "https://sandbox.central.getodk.org",
  test_un = "me@email.com",
  test_pw = "...",
```

```

test_pp = "...",
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",
retries = 3,
verbose = TRUE
)

```

---

split_geopoint	<i>Annotate a dataframe containing a geopoint column with lon, lat, alt.</i>
----------------	--

---

## 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\\_link\(\)](#), [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\(\)](#), [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("geo_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 = get_default_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: <a href="#">get_default_odkc_version</a> or 1.1 if unset. Set default <a href="#">get_default_odkc_version</a> through <a href="#">ru_setup(odkc_version=1.1)</a> . See <a href="#">vignette("Setup", package = "ruODK")</a> .

**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\\_link\(\)](#), [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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geotrace\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

**Examples**

```
## Not run:
library(magrittr)
data("geo_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**

<code>data</code>	(dataframe) A dataframe with a geotrace column.
<code>colname</code>	(chr) The name of the geotrace column. This column will be retained.
<code>wkt</code>	Whether geofields are GeoJSON (if FALSE) or WKT strings (if TRUE), default: FALSE.
<code>odkc_version</code>	The ODK Central version as decimal number (major.minor). ruODK uses this parameter to adjust for breaking changes in ODK Central. Default: <a href="#">get_default_odkc_version</a> or 1.1 if unset. Set default <code>get_default_odkc_version</code> through <code>ru_setup(odkc_version=1.1)</code> . See <code>vignette("Setup", package = "ruODK")</code> .

**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\\_link\(\)](#), [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\(\)](#), [split\\_geopoint\(\)](#), [split\\_geoshape\(\)](#), [strip\\_uuid\(\)](#), [tidyeval](#), [unnest\\_all\(\)](#)

**Examples**

```
## Not run:
library(magrittr)
data("geo_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(),
  retries = get_retries()
)
```

## Arguments

iid	The instance_id, a UUID, as returned by <a href="#">submission_list</a> .
pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.



**Value**

A nested list of submission metadata.

**See Also**

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

Other submission-management: [attachment\\_list\(\)](#), [encryption\\_key\\_list\(\)](#), [submission\\_export\(\)](#), [submission\\_get\(\)](#), [submission\\_list\(\)](#)

**Examples**

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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 and the submitter
names(sub)
# > "instance_id" "submitter_id" "submitter_type" "submitter_display_name"
# > "submitter_created_at" "device_id" "created_at"

## End(Not run)
```

---

submission\_export

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

---

**Description**

[Maturing]

**Usage**

```
submission_export(
  local_dir = here::here(),
  overwrite = TRUE,
  media = TRUE,
  repeats = TRUE,
  pid = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
```

```

un = get_default_un(),
pw = get_default_pw(),
pp = get_default_pp(),
retries = get_retries(),
odkc_version = get_default_odkc_version(),
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
media	Whether to include media attachments, default: TRUE. This feature only has effect on ODK Central v1.1 and higher. Setting this feature to FALSE with an odkc_version < 1.1 and will display a verbose noop message, but still return all media attachments.
repeats	Whether to include repeat data (if TRUE), or whether to return the root table only (FALSE). Default: TRUE. Requesting repeats=FALSE will also omit any media, and override the parameter media. Setting this feature to FALSE with an odkc_version < 1.1 and will display a verbose noop message, but still include all repeat data.
pid	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . 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: <a href="#">get_default_fid</a> . Set default fid through ru_setup(fid="..."). See vignette("Setup", package = "ruODK").
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through ru_setup(url="..."). See vignette("Setup", package = "ruODK").
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through ru_setup(un="..."). See vignette("Setup", package = "ruODK").
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through ru_setup(pw="..."). See vignette("Setup", package = "ruODK").
pp	The passphrase for an encrypted form. Default: NULL. Passphrases can be stored e.g. as environment variables. See vignette("Setup", package = "ruODK").
retries	The number of attempts to retrieve a web resource. This parameter is given to <a href="#">RETRY</a> (times = retries). Default: 3.

odkc_version	<p>The ODK Central version as decimal number (major.minor). ruODK uses this parameter to adjust for breaking changes in ODK Central.</p> <p>Default: <a href="#">get_default_odkc_version</a> or 1.1 if unset.</p> <p>Set default <code>get_default_odkc_version</code> through <code>ru_setup(odkc_version=1.1)</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
verbose	<p>Whether to display debug messages or not.</p> <p>Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's verbosity can be set globally or per function.</p>

## Details

This function exports all the Submission data associated with a Form as one zip file containing one or more CSV files, as well as all multimedia attachments associated with the included Submissions.

For an incremental download of a subset of submissions, use [submission\\_list](#) or [odata\\_submission\\_get](#) with filter queries.

### Contents:

The inclusion of subtables (from repeating form groups) can be toggled through `repeats`, whereas the inclusion of media attachments can be toggled through `media`.

### Download location:

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).

### Encryption:

ODK Central supports two modes of encryption - learn about them [here](#). ruODK supports project managed encryption, however the support is limited to exactly one encryption key. The supplied passphrase will be used against the first returned encryption key. Remaining encryption keys are ignored by ruODK.

If an incorrect passphrase is given, the request is terminated immediately. It has been reported that multiple requests with incorrect passphrases can crash ODK Central.

## Value

The absolute path to the exported ZIP file named after the form ID. The exported ZIP file will have the extension `.zip` unless only the root table was requested (with `repeats=FALSE`), in which case the exported file will have the extension `.csv`. In contrast to ODK Central, which exports to `submissions.csv(.zip)`, the exported ZIP file is named after the form to avoid accidentally overwriting the ZIP export from another form.

## See Also

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

Other submission-management: [attachment\\_list\(\)](#), [encryption\\_key\\_list\(\)](#), [submission\\_detail\(\)](#), [submission\\_get\(\)](#), [submission\\_list\(\)](#)

## Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...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 = get_default_pid(),
  fid = get_default_fid(),
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  retries = get_retries()
)
```

## Arguments

<code>iid</code>	A list of submission instance IDs, e.g. from <code>submission_list\$instance_id</code> .
<code>pid</code>	The numeric ID of the project, e.g.: 2. Default: <a href="#">get_default_pid</a> . Set default pid through <code>ru_setup(pid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .

fid	The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147". Default: <a href="#">get_default_fid</a> . Set default fid through <code>ru_setup(fid="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
url	The ODK Central base URL without trailing slash. Default: <a href="#">get_default_url</a> . Set default url through <code>ru_setup(url="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
un	The ODK Central username (an email address). Default: <a href="#">get_default_un</a> . Set default un through <code>ru_setup(un="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
pw	The ODK Central password. Default: <a href="#">get_default_pw</a> . Set default pw through <code>ru_setup(pw="...")</code> . See <code>vignette("Setup", package = "ruODK")</code> .
retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.

## Details

Forms with submission audit enabled will also receive the submission audit as `audit.csv`. This will overwrite all previous `audit.csv` files. To get the combined submission audit logs as one single, concatenated `audit.csv` file, use `submission_export`.

## Value

A nested list of submission data.

## See Also

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

Other submission-management: [attachment\\_list\(\)](#), [encryption\\_key\\_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	<i>List all submissions of one form.</i>
-----------------	--

---

## 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(),
  retries = get_retries(),
  orders = c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd"),
  tz = get_default_tz()
)
```

## Arguments

pid	<p>The numeric ID of the project, e.g.: 2.</p> <p>Default: <a href="#">get_default_pid</a>.</p> <p>Set default pid through <code>ru_setup(pid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
fid	<p>The alphanumeric form ID, e.g. "build_Spotlighting-0-8_1559885147".</p> <p>Default: <a href="#">get_default_fid</a>.</p> <p>Set default fid through <code>ru_setup(fid="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
url	<p>The ODK Central base URL without trailing slash.</p> <p>Default: <a href="#">get_default_url</a>.</p> <p>Set default url through <code>ru_setup(url="...")</code>.</p> <p>See <code>vignette("Setup", package = "ruODK")</code>.</p>
un	<p>The ODK Central username (an email address). Default: <a href="#">get_default_un</a>. Set default un through <code>ru_setup(un="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
pw	<p>The ODK Central password. Default: <a href="#">get_default_pw</a>. Set default pw through <code>ru_setup(pw="...")</code>. See <code>vignette("Setup", package = "ruODK")</code>.</p>
retries	<p>The number of attempts to retrieve a web resource.</p> <p>This parameter is given to <code>RETRY(times = retries)</code>.</p> <p>Default: 3.</p>
orders	<p>(vector of character) Orders of datetime elements for lubridate.</p> <p>Default: <code>c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd")</code>.</p>

**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.

### 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, dtm
- \* `updated_at`: time of submission update on server, dtm or NA

### See Also

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

Other submission-management: [attachment\\_list\(\)](#), [encryption\\_key\\_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)
```

---

user_list	List all users.
-----------	-----------------

---

**Description**

[Maturing]

**Usage**

```
user_list(
  qry = NULL,
  url = get_default_url(),
  un = get_default_un(),
  pw = get_default_pw(),
  retries = get_retries(),
  orders = c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd"),
  tz = get_default_tz(),
  verbose = get_ru_verbose()
)
```

**Arguments**

qry	<p>A query string to filter users by. The query string is not case sensitive and can contain special characters, such as @.</p> <p>The query string must be at least 5 alphabetic characters long to return good enough matches. E.g. janet will match a user with display name Janette Doe. E.g., @dbca.wa will match users with an email from dbca.wa.gov.au, whereas @dbca.w or @dbca will return no matches.</p> <p>Default: NULL.</p>
url	<p>The ODK Central base URL without trailing slash.</p> <p>Default: <a href="#">get_default_url</a>.</p> <p>Set default url through ru_setup(url="...").</p> <p>See vignette("Setup", package = "ruODK").</p>
un	<p>The ODK Central username (an email address). Default: <a href="#">get_default_un</a>. Set default un through ru_setup(un="..."). See vignette("Setup", package = "ruODK").</p>
pw	<p>The ODK Central password. Default: <a href="#">get_default_pw</a>. Set default pw through ru_setup(pw="..."). See vignette("Setup", package = "ruODK").</p>



retries	The number of attempts to retrieve a web resource. This parameter is given to <code>RETRY(times = retries)</code> . Default: 3.
orders	(vector of character) Orders of datetime elements for lubridate. Default: <code>c("YmdHMS", "YmdHMSz", "Ymd HMS", "Ymd HMSz", "Ymd", "ymd")</code> .
tz	A timezone to convert dates and times to. Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's timezone can be set globally or per function.
verbose	Whether to display debug messages or not. Read <code>vignette("setup", package = "ruODK")</code> to learn how ruODK's verbosity can be set globally or per function.

## Details

Currently, there are no paging or filtering options, so listing Users will get you every User in the system, every time. Optionally, a `q` query string parameter may be provided to filter the returned users by any given string. The search is performed via a trigram similarity index over both the Email and Display Name fields, and results are ordered by match score, best matches first. If a `q` parameter is given, and it exactly matches an email address that exists in the system, that user's details will always be returned, even for actors who cannot `user.list`. The request must still authenticate as a valid Actor. This allows non-Administrators to choose a user for an action (e.g. grant rights) without allowing full search.

Actors who cannot `user.list` will always receive `[]` with a 200 OK response. ruODK does not (yet) warn if this is the case, and you (the requesting Actor) have no permission to `user.list`.

## Value

A tibble with user details as per the ODK Central API docs.

## See Also

<https://odkcentral.docs.apiary.io/#reference/accounts-and-users/users/listing-all-users>  
<https://www.postgresql.org/docs/9.6/pgtrgm.html>

## Examples

```
## Not run:
# See vignette("setup") for setup and authentication options
# ruODK::ru_setup(svc = "...svc", un = "me@email.com", pw = "...")

# All users
ul <- user_list()

# Search users
# Given a user with display name "Janette Doe" and email "@org.com.au"
user_list(qry = "jan") # no results, query string too short
user_list(qry = "jane") # no results, query string too short
user_list(qry = "janet") # returns Janette
```

```
user_list(qry = "@org") # no results, query string too short
user_list(qry = "@org.c") # no results, query string too short
user_list(qry = "@org.co") # returns all users matching "@org.co"

# Actor not allowed to user.list
user_list() # If this is empty, you might not have permissions to list users

## End(Not run)
```

# Index

## \* datasets

- [fq\\_attachments](#), 27
- [fq\\_data](#), 28
- [fq\\_data\\_strata](#), 29
- [fq\\_data\\_taxa](#), 30
- [fq\\_form\\_detail](#), 31
- [fq\\_form\\_list](#), 31
- [fq\\_form\\_schema](#), 32
- [fq\\_form\\_xml](#), 33
- [fq\\_meta](#), 33
- [fq\\_project\\_detail](#), 34
- [fq\\_project\\_list](#), 35
- [fq\\_raw](#), 35
- [fq\\_raw\\_strata](#), 36
- [fq\\_raw\\_taxa](#), 37
- [fq\\_submission\\_list](#), 38
- [fq\\_submissions](#), 38
- [fq\\_svc](#), 39
- [fq\\_zip\\_data](#), 40
- [fq\\_zip\\_strata](#), 40
- [fq\\_zip\\_taxa](#), 41
- [fs\\_v7](#), 41
- [fs\\_v7\\_raw](#), 42
- [geo\\_fs](#), 43
- [geo\\_gj](#), 43
- [geo\\_gj88](#), 44
- [geo\\_gj\\_raw](#), 45
- [geo\\_wkt](#), 45
- [geo\\_wkt88](#), 46
- [geo\\_wkt\\_raw](#), 47

## \* form-management

- [form\\_detail](#), 15
- [form\\_list](#), 16
- [form\\_schema](#), 18
- [form\\_schema\\_ext](#), 22
- [form\\_xml](#), 26

## \* included

- [fq\\_attachments](#), 27
- [fq\\_data](#), 28

- [fq\\_data\\_strata](#), 29
- [fq\\_data\\_taxa](#), 30
- [fq\\_form\\_detail](#), 31
- [fq\\_form\\_list](#), 31
- [fq\\_form\\_schema](#), 32
- [fq\\_form\\_xml](#), 33
- [fq\\_meta](#), 33
- [fq\\_project\\_detail](#), 34
- [fq\\_project\\_list](#), 35
- [fq\\_raw](#), 35
- [fq\\_raw\\_strata](#), 36
- [fq\\_raw\\_taxa](#), 37
- [fq\\_submission\\_list](#), 38
- [fq\\_submissions](#), 38
- [fq\\_svc](#), 39
- [fq\\_zip\\_data](#), 40
- [fq\\_zip\\_strata](#), 40
- [fq\\_zip\\_taxa](#), 41
- [fs\\_v7](#), 41
- [fs\\_v7\\_raw](#), 42
- [geo\\_fs](#), 43
- [geo\\_gj](#), 43
- [geo\\_gj88](#), 44
- [geo\\_gj\\_raw](#), 45
- [geo\\_wkt](#), 45
- [geo\\_wkt88](#), 46
- [geo\\_wkt\\_raw](#), 47

## \* messaging

- [ru\\_msg\\_abort](#), 74
- [ru\\_msg\\_info](#), 75
- [ru\\_msg\\_noop](#), 75
- [ru\\_msg\\_success](#), 76
- [ru\\_msg\\_warn](#), 77

## \* odata-api

- [odata\\_metadata\\_get](#), 61
- [odata\\_service\\_get](#), 62
- [odata\\_submission\\_get](#), 64

## \* project-management

- [project\\_create](#), 70

- project\_detail, [71](#)
- project\_list, [73](#)
- \* **ru\_settings**
  - odata\_svc\_parse, [69](#)
  - ru\_settings, [77](#)
  - ru\_setup, [79](#)
- \* **server-management**
  - audit\_get, [9](#)
- \* **submission-management**
  - attachment\_list, [8](#)
  - encryption\_key\_list, [13](#)
  - submission\_detail, [88](#)
  - submission\_export, [89](#)
  - submission\_get, [92](#)
  - submission\_list, [94](#)
- \* **user-management**
  - user\_list, [96](#)
- \* **utilities**
  - attachment\_get, [4](#)
  - attachment\_link, [6](#)
  - drop\_null\_coords, [11](#)
  - form\_schema\_parse, [24](#)
  - get\_one\_attachment, [47](#)
  - get\_one\_submission, [49](#)
  - get\_one\_submission\_attachment\_list, [51](#)
  - handle\_ru\_attachments, [53](#)
  - handle\_ru\_datetimes, [55](#)
  - handle\_ru\_geopoints, [56](#)
  - handle\_ru\_geoshapes, [58](#)
  - handle\_ru\_geotraces, [59](#)
  - odata\_submission\_rectangle, [68](#)
  - split\_geopoint, [82](#)
  - split\_geoshape, [83](#)
  - split\_geotrace, [85](#)
- attachment\_get, [4](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#), [57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)
- attachment\_link, [6](#), [6](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#), [57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)
- attachment\_list, [8](#), [14](#), [27](#), [89](#), [91](#), [93](#), [95](#)
- attachment\_url, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#), [57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)
- audit\_get, [9](#)
- drop\_null\_coords, [6](#), [7](#), [11](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#), [57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)
- encryption\_key\_list, [9](#), [13](#), [89](#), [91](#), [93](#), [95](#)
- form\_detail, [15](#), [17](#), [20](#), [24](#), [27](#), [31](#)
- form\_list, [16](#), [16](#), [20](#), [24](#), [27](#), [31](#)
- form\_schema, [16–18](#), [18](#), [19](#), [22–25](#), [27](#), [42](#), [43](#), [68](#), [82](#), [84](#), [86](#)
- form\_schema\_ext, [16](#), [17](#), [20](#), [21](#), [23](#), [27](#)
- form\_schema\_parse, [6](#), [7](#), [12](#), [18](#), [22](#), [24](#), [25](#), [42](#), [48](#), [50](#), [52](#), [54](#), [56](#), [57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)
- form\_xml, [16](#), [17](#), [20](#), [23](#), [24](#), [26](#), [33](#)
- fq\_attachments, [27](#), [28–47](#)
- fq\_data, [28](#), [28](#), [29–47](#)
- fq\_data\_strata, [28](#), [29](#), [30–47](#)
- fq\_data\_taxa, [28](#), [29](#), [30](#), [31–47](#)
- fq\_form\_detail, [28–30](#), [31](#), [32–47](#)
- fq\_form\_list, [28–31](#), [31](#), [32–47](#)
- fq\_form\_schema, [28–32](#), [32](#), [33–47](#)
- fq\_form\_xml, [28–32](#), [33](#), [34–47](#)
- fq\_meta, [28–33](#), [33](#), [34–47](#)
- fq\_project\_detail, [28–34](#), [34](#), [35–47](#)
- fq\_project\_list, [28–34](#), [35](#), [36–47](#)
- fq\_raw, [28–35](#), [35](#), [37–47](#)
- fq\_raw\_strata, [28–36](#), [36](#), [38–47](#)
- fq\_raw\_taxa, [28–37](#), [37](#), [38–47](#)
- fq\_submission\_list, [28–38](#), [38](#), [39–47](#)
- fq\_submissions, [28–38](#), [38](#), [39–47](#)
- fq\_svc, [28–39](#), [39](#), [40–47](#)
- fq\_zip\_data, [28–39](#), [40](#), [41–47](#)
- fq\_zip\_strata, [28–40](#), [40](#), [41–47](#)
- fq\_zip\_taxa, [28–41](#), [41](#), [42–47](#)
- fs\_v7, [28–41](#), [41](#), [42–47](#)
- fs\_v7\_raw, [28–42](#), [42](#), [43–47](#)
- geo\_fs, [28–42](#), [43](#), [44–47](#)
- geo\_gj, [28–43](#), [43](#), [44–47](#)
- geo\_gj88, [28–44](#), [44](#), [45–47](#)
- geo\_gj\_raw, [28–44](#), [45](#), [46](#), [47](#)
- geo\_wkt, [28–45](#), [45](#), [46](#), [47](#)
- geo\_wkt88, [28–46](#), [46](#), [47](#)
- geo\_wkt\_raw, [28–46](#), [47](#)
- get\_default\_fid, [5](#), [8](#), [13](#), [15](#), [19](#), [22](#), [26](#), [50](#), [51](#), [54](#), [62](#), [63](#), [65](#), [78](#), [88](#), [90](#), [93](#), [94](#)
- get\_default\_fid(ru\_settings), [77](#)
- get\_default\_odkc\_version, [19](#), [23](#), [58](#), [60](#), [66](#), [78](#), [84](#), [86](#), [91](#)
- get\_default\_odkc\_version(ru\_settings), [77](#)
- get\_default\_pid, [5](#), [8](#), [13](#), [15](#), [17](#), [18](#), [22](#), [26](#), [49](#), [51](#), [54](#), [61](#), [63](#), [65](#), [71](#), [78](#), [88](#), [90](#), [92](#), [94](#)

- get\_default\_pid(ru\_settings), 77
- get\_default\_pp, 78
- get\_default\_pp(ru\_settings), 77
- get\_default\_pw, 5, 8, 10, 14, 15, 17, 19, 23, 26, 48, 50, 52, 54, 62, 63, 66, 70, 72, 73, 78, 88, 90, 93, 94, 96
- get\_default\_pw(ru\_settings), 77
- get\_default\_tz, 78
- get\_default\_tz(ru\_settings), 77
- get\_default\_un, 5, 8, 10, 13, 15, 17, 19, 23, 26, 48, 50, 52, 54, 62, 63, 66, 70, 72, 73, 78, 88, 90, 93, 94, 96
- get\_default\_un(ru\_settings), 77
- get\_default\_url, 5, 8, 10, 13, 15, 17, 19, 23, 26, 48, 50, 52, 54, 62, 63, 66, 70, 71, 73, 78, 88, 90, 93, 94, 96
- get\_default\_url(ru\_settings), 77
- get\_one\_attachment, 6, 7, 12, 25, 47, 50, 52, 54, 56, 57, 59, 60, 69, 82, 84, 86
- get\_one\_submission, 6, 7, 12, 25, 48, 49, 52, 54, 56, 57, 59, 60, 69, 82, 84, 86, 92
- get\_one\_submission\_attachment\_list, 6, 7, 12, 25, 48, 50, 51, 54, 56, 57, 59, 60, 69, 82, 84, 86
- get\_retries, 78
- get\_retries(ru\_settings), 77
- get\_ru\_verbose, 78, 79
- get\_ru\_verbose(ru\_settings), 77
- get\_test\_fid, 78
- get\_test\_fid(ru\_settings), 77
- get\_test\_fid\_att, 78
- get\_test\_fid\_att(ru\_settings), 77
- get\_test\_fid\_gap, 78
- get\_test\_fid\_gap(ru\_settings), 77
- get\_test\_fid\_wkt, 78
- get\_test\_fid\_wkt(ru\_settings), 77
- get\_test\_fid\_zip, 78
- get\_test\_fid\_zip(ru\_settings), 77
- get\_test\_odkc\_version, 78
- get\_test\_odkc\_version(ru\_settings), 77
- get\_test\_pid, 78
- get\_test\_pid(ru\_settings), 77
- get\_test\_pp, 78
- get\_test\_pp(ru\_settings), 77
- get\_test\_pw, 78
- get\_test\_pw(ru\_settings), 77
- get\_test\_un, 78
- get\_test\_un(ru\_settings), 77
- get\_test\_url, 78
- get\_test\_url(ru\_settings), 77
- handle\_ru\_attachments, 5–7, 12, 25, 48, 50, 52, 53, 56, 57, 59, 60, 67, 69, 82, 84, 86
- handle\_ru\_datetimes, 6, 7, 12, 25, 48, 50, 52, 54, 55, 57, 59, 60, 67, 69, 82, 84, 86
- handle\_ru\_geopoints, 6, 7, 12, 25, 48, 50, 52, 54, 56, 56, 59, 60, 67, 69, 82, 84, 86
- handle\_ru\_geoshapes, 6, 7, 12, 25, 48, 50, 52, 54, 56, 57, 58, 60, 67, 69, 82, 84, 86
- handle\_ru\_geotraces, 6, 7, 12, 25, 48, 50, 52, 54, 56, 57, 59, 59, 67, 69, 82, 84, 86
- isodt\_to\_local, 6, 7, 12, 25, 48, 50, 52, 54, 56, 57, 59, 60, 69, 82, 84, 86
- make\_clean\_names, 20, 23
- odata\_metadata\_get, 61, 63, 67
- odata\_service\_get, 62, 62, 64, 67
- odata\_submission\_get, 20, 23, 28–30, 43–47, 62, 63, 64, 66, 68, 80, 91
- odata\_submission\_rectangle, 6, 7, 12, 20, 23, 25, 48, 50, 52, 54, 56, 57, 59, 60, 67, 68, 82, 84, 86
- odata\_svc\_parse, 69, 79, 81
- predict\_ruodk\_name, 6, 7, 12, 25, 48, 50, 52, 54, 56, 57, 59, 60, 69, 82, 84, 86
- prepend\_uuid, 6, 7, 12, 25, 48, 50, 52, 54, 56, 57, 59, 60, 69, 82, 84, 86
- project\_create, 70, 72, 74
- project\_detail, 34, 70, 71, 71, 74
- project\_list, 35, 70, 72, 73, 73
- RETRY, 5, 8, 10, 14, 15, 17, 19, 23, 26, 48, 50, 52, 54, 62, 63, 66, 72, 73, 80, 88, 90, 93, 94, 97
- ru\_msg\_abort, 74, 75–77
- ru\_msg\_info, 74, 75, 76, 77
- ru\_msg\_noop, 74, 75, 75, 76, 77
- ru\_msg\_success, 74–76, 76, 77
- ru\_msg\_warn, 74–76, 77
- ru\_settings, 70, 77, 78, 81

ru\_setup, [70](#), [78](#), [79](#), [79](#), [80](#), [81](#)  
ruODK, [79](#), [81](#)

split\_geopoint, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#),  
[56](#), [57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)  
split\_geoshape, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#),  
[56](#), [57](#), [59](#), [60](#), [66](#), [69](#), [82](#), [83](#), [86](#)  
split\_geotrace, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#),  
[56](#), [57](#), [59](#), [60](#), [66](#), [69](#), [82](#), [84](#), [85](#)  
strip\_uuid, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#),  
[57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)  
submission\_detail, [9](#), [14](#), [88](#), [91](#), [93](#), [95](#)  
submission\_export, [4](#), [7](#), [9](#), [14](#), [40](#), [41](#), [89](#),  
[89](#), [93](#), [95](#)  
submission\_get, [9](#), [14](#), [38](#), [89](#), [91](#), [92](#), [95](#)  
submission\_list, [8](#), [9](#), [14](#), [38](#), [39](#), [49](#), [51](#), [88](#),  
[89](#), [91–93](#), [94](#)

tidyeval, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#), [57](#),  
[59](#), [60](#), [69](#), [82](#), [84](#), [86](#)

unnest\_all, [6](#), [7](#), [12](#), [25](#), [48](#), [50](#), [52](#), [54](#), [56](#),  
[57](#), [59](#), [60](#), [69](#), [82](#), [84](#), [86](#)  
user\_list, [96](#)

yell\_if\_error, [70](#), [79](#), [81](#)  
yell\_if\_missing, [70](#), [79](#), [81](#)