# Package: matos (via r-universe)

November 5, 2024

**Type** Package

**Title** Interface with the Mid-Atlantic Acoustic Telemetry Observing System (MATOS)

**Version** 0.4.0.9008

**Description** This package provides HTML-scaping functions to interact with the MATOS website (https://matos.asascience.com/), allowing aggregation of data and the downloading and posting of files.

**Imports** cli, curl, getPass, httr, memoise, otndo, rvest

**Suggests** future.apply, testthat (>= 3.0.0), knitr, rmarkdown, rvdat

**Remotes** mhpob/otndo, mhpob/rvdat

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**URL** https://github.com/mhpob/matos, https://matos.obrien.page

**BugReports** https://github.com/mhpob/matos/issues

**Depends** R (>= 4.1.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Roxygen** list(markdown = TRUE)

**Config/pak/sysreqs** libfontconfig1-dev libfreetype6-dev libgdal-dev gdal-bin libgeos-dev make libicu-dev libpng-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

**Repository** https://benjaminhlina.r-universe.dev

**RemoteUrl** https://github.com/mhpob/matos

**RemoteRef** HEAD

**RemoteSha** 93a6a05c56b7b09fb71718173d55518625d87bfa

# Contents



---

`CI_utilities` *Miscellaneous functions for package checking, building, and CI*

---

## Description

`skip_example_on_ci` and `skip_example_on_runiverse` check the environment for variables called "CI" and "MY_UNIVERSE", respectively, and return `TRUE` if it does not exist. Used to run examples if the package is being built locally and there's a chance that `vdat.exe` exists. If the package is being built on a continuous integration platform like GitHub Actions, the "CI" variable will be `TRUE` and `skip_example_on_ci` will return `FALSE`. If it is being built locally, "CI" will be "" and `skip_example_on_ci` will return `TRUE`. Similarly, if the package is being built on R-Universe, the "MY_UNIVERSE" variable will have your universe's name.

## Usage

```
skip_example_on_ci()

skip_example_on_runiverse()

skip_example_on_cran()
```

get_extract_file *Download OTN data extraction files from the MATOS website*

## Description

get_extract_file downloads files from the MATOS website. This is best-used in conjunction with list_extract_files.

## Usage

```
get_extract_file(
  file = NULL,
  project = NULL,
  url = NULL,
  out_dir = getwd(),
  overwrite = FALSE,
  to_vue = FALSE,
  quiet = FALSE
)
```

## Arguments

| | |
|---|---|
| file | A character vector listing the name of the file, or a numeric listing the index as found from list_extract_files. |
| project | A character vector listing the full name of the project, or a numeric listing the project number. |
| url | The URL of the file to be downloaded. |
| out_dir | Character. What directory/folder do you want the file saved into? Default is the current working directory. |
| overwrite | Logical. Do you want a file with the same name overwritten? Passed to httr::write_disk. |
| to_vue | Logical. Convert to VUE export format? |
| quiet | Logical. Do you want to silence matos' updates? Default is FALSE. |

## Examples

```
# If you know the direct URL to your file, you don't need the file or project names:
get_extract_file(url = "https://matos.asascience.com/projectfile/download/327")

get_extract_file(file = 1, project = 87)
```

`get_extract_updates` *Download all data extraction files that were updated after a certain date*

## Description

This is a loop around `get_extract_file`.

## Usage

```
get_extract_updates(
  project,
  since = NULL,
  detection_type = c("all", "matched", "external", "qualified", "sentinel",
    "unqualified"),
  out_dir = getwd(),
  overwrite = FALSE,
  to_vue = FALSE,
  progress = TRUE,
  quiet = TRUE
)
```

## Arguments

| | |
|---|---|
| `project` | Either the project code, the project number (the number in your project page URL), or the full name of the project (the big name in bold on your project page, *not* the "Project Title"). |
| `since` | Only list files uploaded after this date. Optional, but must be in YYYY-MM-DD format. |
| `detection_type` | one of, or a vector of, "all" (default), "matched", "external", "qualified", "sentinel_tag", or "unqualified". Partial matching is allowed, and will repair to the correct argument if spaces or the words detection(s)" are included. More information on data types can be found on OTN's website. |
| `out_dir` | Character. What directory/folder do you want the file saved into? Default is the current working directory. Passed to `httr::write_disk` via `get_extract_file`. |
| `overwrite` | Logical. Do you want a file with the same name overwritten? Passed to `httr::write_disk` via `get_extract_file`. |
| `to_vue` | Logical. Should the data be converted to match that of VUE's CSV export? |
| `progress` | Logical. Do you want a progress bar? Default is TRUE. |
| `quiet` | Logical. Do you want to silence matos' updates? Default is TRUE. |

## Value

Saves the requested files to your computer and provides a list of file paths of the downloaded files.

## Examples

```
# Download files from the MDWEA project updated in the November 2023 data push
#   (you'll need to use a project for which you have permissions).
get_extract_updates(project = "MDWEA", since = "2023-11-01")

# Match the VUE CSV export style
get_extract_updates(project = 160, to_vue = TRUE)
```

---

# get_otn_template *Download Ocean-Tracking-Network-style metadata templates*

---

## Description

Download Ocean-Tracking-Network-style metadata templates

## Usage

```
get_otn_template(
  template_type = c("receiver", "tag", "glider"),
  out_dir = NULL,
  quiet = FALSE
)
```

## Arguments

| | |
|---|---|
| template_type | Character string. One of: "receiver" (default), the deployment data submittal template; "tag", the tagging data submittal template; or "glider", the wave and Slocum glider metadata template. |
| out_dir | Optional character string noting where you would like the file to be downloaded. Defaults to the working directory. |
| quiet | suppress status messages from `download.file` |

## Value

Ocean Tracking Network metadata template in XLSX format.

## Examples

```
# Tag metadata template downloaded to working directory
get_otn_template()

# Glider metadata template downloaded to temporary directory
get_otn_template("glider", out_dir = tempdir())
```

`get_project_file` *Download project files from the MATOS website*

## Description

`get_project_file` downloads files from the MATOS website. This is best-used in conjunction with `list_project_files`.

## Usage

```
get_project_file(
  file = NULL,
  project = NULL,
  url = NULL,
  out_dir = getwd(),
  overwrite = FALSE,
  quiet = FALSE
)
```

## Arguments

| | |
|---|---|
| `file` | A character vector listing the name of the file, or a numeric listing the index as found from `list_project_files`. |
| `project` | A character vector listing the full name of the project, or a numeric listing the project number. |
| `url` | The URL of the file to be downloaded. |
| `out_dir` | Character. What directory/folder do you want the file saved into? Default is the current working directory. |
| `overwrite` | Logical. Do you want a file with the same name overwritten? Passed to httr::write_disk. |
| `quiet` | Logical. Do you want to silence matos' updates? Default is FALSE. |

## Examples

```
# If you know the index of the file, you can provide some numbers
get_project_file(file = 1, project = 87)

# If you know the direct URL to your file, you don't need the file or project names:
get_project_file(url = "https://matos.asascience.com/projectfile/download/327")
```

## get_tag_search *Search for tags on the MATOS website*

### Description

This function is an interface to MATOS' tag search page, with the result of a CSV downloaded into your working directory. Be aware: these downloads can take a *long* time, especially if you have many tags or are searching over a long period of time.

### Usage

```
get_tag_search(tags, start_date, end_date, import = F)
```

### Arguments

| | |
|---|---|
| tags | Character vector of tags. Will be coerced into CSV when POSTing to the website. |
| start_date | Character string listing the start date in MM/DD/YYYY format. If no dates are provided, all tag detections are returned. |
| end_date | Character string listing the end date in MM/DD/YYYY format. If no dates are provided, all tag detections are returned. |
| import | Should the downloaded data be imported into R as a data frame? Default is FALSE. |

### Examples

```
## Not run:
get_tag_search(
  tags = paste0("A69-1601-254", seq(60, 65, 1)),
  start_date = "03/01/2016",
  end_date = "04/01/2016"
)

## End(Not run)
```

## list_extract_files *List OTN data extraction files*

### Description

This function list the file names, types, upload date, and URLs of OTN data extraction files – basically everything you see in the *Data Extraction Files* section of your project page. Because it is from your project page, you will be prompted to log in.

## Usage

```
list_extract_files(
  project,
  detection_type = c("all", "matched", "external", "qualified", "sentinel",
    "unqualified"),
  since = NULL
)
```

## Arguments

| | |
|---|---|
| project | Either the project code, the project number (the number in your project page URL), or the full name of the project (the big name in bold on your project page, *not* the "Project Title"). |
| detection_type | one of, or a vector of, "all" (default), "matched", "external", "qualified", "sentinel_tag", or "unqualified". Partial matching is allowed, and will repair to the correct argument if spaces or the words detection(s)" are included. More information on data types can be found on OTN's website. |
| since | Only list files uploaded after this date. Optional, but must be in YYYY-MM-DD format. |

## Value

A data frame with columns of "project", "file_type", "detection_type", 'detection_year', 'upload_date', 'file_name', and "url".

## Details

`list_extract_files` is a wrapper around a web-scraping routine:

1. find the project number if not provided, 2) download the HTML table, 3) parse the URL for each file, and 4) combine the table and URLs into a data frame. This function is most useful when investigating what files you have available, and then downloading them with `get_extract_file`.

`list_extract_files` lists files associated with the ACT_MATOS OTN node. These are files listed on the *Data Extraction Files* page.

## Examples

```
# List all extraction files using project number
list_extract_files(87)

# Or, just grab the matched receiver detections
list_extract_files(project = 87, detection_type = "qualified")

# OR list files using the project name
list_extract_files("umces boem offshore wind energy")
```

## list_my_projects *List personal MATOS projects*

### Description

This function lists the functions for which the logged-on user has permissions.

### Usage

```
list_my_projects(read_access = TRUE, force = FALSE, warn_multimatch = TRUE)
```

### Arguments

| | |
|---|---|
| read_access | Do you want to only list projects for which you have file-read permission? Defaults to TRUE, though there is significant speed up if switched to FALSE. |
| force | Do you want to reset the cache and re-ping the database? Defaults to false. |
| warn_multimatch | Warn you if there have been multiple project matches? Defaults to TRUE. |

### Examples

```
# After logging in, just type the following:
list_my_projects()
```

## list_projects *List MATOS projects*

### Description

This function scrapes the table found at https://matos.asascience.com/project and combines it with project metadata stored on the Ocean Tracking Network Geoserver. This table provides the full name of the project, collection code, MATOS project number, MATOS project page URL, project status, full name, citation, website, project type, area, and abstract. You do not need to log in via matos_login or have any permissions to view/download this table.

### Usage

```
list_projects(
  what = c("all", "mine"),
  read_access = TRUE,
  quiet = FALSE,
  force = FALSE,
  warn_multimatch = TRUE
)
```

### Arguments

| | |
|---|---|
| what | What list of projects do you want returned: all projects ("all", default) or your projects ("mine")? |
| read_access | If listing your projects, do you want to only list projects for which you have file-read permission? Defaults to TRUE, though there is significant speed up if switched to FALSE. |
| quiet | Do you want to suppress messages regarding matched projects? Defaults to FALSE. |
| force | Do you want to reset the cache and re-ping the database? Defaults to FALSE. |
| warn_multimatch | Warn you if there have been multiple project matches? Defaults to TRUE. |

### Examples

```
# List all projects, the default:
list_projects()

# List your projects (which may contain some for which you do not have read access):
list_projects("mine", read_access = F)
```

---

## list_project_files *List MATOS project files*

---

### Description

This function lists the file names, types, upload date, and URLs of MATOS project files – basically everything you see in the *Project Files* section of your project page. Because it is from your project page, you will be prompted to log in.

### Usage

```
list_project_files(
  project,
  file_type = c("all", "detections", "receiver_metadata", "tag_metadata"),
  since = NULL
)
```

### Arguments

| | |
|---|---|
| project | Either the project number (the number in your project page URL) or the full name of the project (the big name in bold on your project page, *not* the "Project Title"). |
| file_type | one of, or a vector of, "all" (default), "detections", "receiver_metadata", or "tag_metadata". Partial matching is allowed. |
| since | Only list files uploaded after this date. Optional, but must be in YYYY-MM-DD format. |

## Value

A data frame with columns of "project", "file_type", "upload_date", and "file_name".

## Details

`list_project_files` is a wrapper around a web-scraping routine:

1. find the project number if not provided, 2) download the HTML table, 3) parse the URL for each file, and 4) combine the table and URLs into a data frame. This function is most useful when investigating what files you have available, and then downloading them with `get_project_file`.

`list_project_files` lists tag and receiver metadata files that have been uploaded by the user. These are the files listed on the *Project Files* section of your project page.

## Examples

```
# List files using project number:
list_project_files(87)

# Or using the project name
list_project_files("umces boem offshore wind energy")

# List only the receiver deployment metadata files
List_project_files(87, "receiver_metadata")

# List both the receiver and transmitter deployment metadata files
List_project_files(87, c("receiver_metadata", "tag_metadata"))

# Cheat and use shorter names
List_project_files(87, c("receiver", "tag"))
```

---

# `matos_login` *Log in to your MATOS account*

---

## Description

This function prompts you for the username (email) and password associated with your MATOS account. This is necessary so that you may interface with any project-specific files. If you don't have a MATOS account you can sign up for one here.

## Usage

```
matos_login(credentials = NULL)
```

### Arguments

| | |
|---|---|
| credentials | list with names "UserName" and "Password". This argument only exists for testing purposes and should not be used! It will store your credentials in your R history, which is definitely not good. |

### Details

A pop up will appear asking for your username and password. If everything works out, your credentials will be kept in the sessions' cookies. Your username/password will not be saved – this was done intentionally so that you don't accidentally save credentials in a public script.

### Examples

```
## Not run:
# Type:
matos_login()
# ...then follow the on-screen prompts

## End(Not run)
```

---

## matos_logoff *Log out of your MATOS account*

---

### Description

This function takes no arguments – just tells MATOS that you want to log out. Useful if you're changing users or on a public computer and would like to protect your projects.

### Usage

```
matos_logoff()
```

### Examples

```
## Not run:
matos_logoff()

## End(Not run)
```

`matos_receiver_summary` *Create summary reports of receiver project data from the OTN data push*

## Description

Create summary reports of receiver project data from the OTN data push

## Usage

```
matos_receiver_summary(
  matos_project = NULL,
  qualified = NULL,
  unqualified = NULL,
  deployment = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `matos_project` | MATOS project number or name that you wish to have summarized |
| `qualified, unqualified` | Default is NULL: OTN qualified or unqualified detections will be downloaded from MATOS and unzipped. If you do not wish to download your files (or you're not a member of ACT), this argument also accepts a character vector of file paths of your qualified/unqualified detections. These can be CSVs or zipped folders. |
| `deployment` | File path of user-supplied master OTN receiver deployment metadata. |
| `...` | Arguments passed to `otndo::make_receiver_push_summary` |

## No files provided

If you only provide your ACT project number or title and leave all of the arguments as their defaults, this function will ask you to log in then proceed to download all of the necessary files. If you provide already-downloaded files you can speed up this process substantially.

## Output

This function creates an HTML report that can be viewed in your web browser.

## Examples

```
## Not run:
# Using only the ACT/MATOS project number:
matos_receiver_summary(87)

# Providing a local file:
```

```
matos_receiver_summary(87, deployment = "my_master_deployment_metadata.xlsx")

# Get a summary fo what has changed since a particular date:
matos_receiver_summary(87, since = "2022-05-01")

## End(Not run)
```

---

`matos_tag_summary` *Create summary reports of receiver project data from the OTN data push*

---

## Description

Create summary reports of receiver project data from the OTN data push

## Usage

```
matos_tag_summary(matos_project = NULL, matched = NULL, ...)
```

## Arguments

| | |
|---|---|
| `matos_project` | MATOS project number or name that you wish to have summarized |
| `matched` | Default is NULL: OTN matched detections will be downloaded from MATOS and unzipped. If you do not wish to download your files, this argument also accepts a character vector of file paths of your matched detections. These can be CSVs or zipped folders. |
| `...` | Arguments passed to `otndo::make_tag_push_summary` |

## Examples

```
## Not run:
# You can just use your ACT project number
matos_tag_summary(87)

# Or provide an optional date to summarize "What's New".
matos_tag_summary(87, since = "2018-11-01")

## End(Not run)
```

## set_matos_credentials *Install your MATOS username and password in your* .Renviron *File for repeated use*

### Description

This code was adapted from `tidycensus::census_api_key`. Note that this saves your credentials in your .Renviron, meaning that anyone who is using your computer can theoretically access what your MATOS username and password are. So... use this carefully!

### Usage

```
set_matos_credentials(overwrite = FALSE)
```

### Arguments

| | |
|---|---|
| `overwrite` | Logical. Overwrite previously-stored MATOS credentials? |

### Examples

```
## Not run:
set_matos_credentials()

## End(Not run)
# Yup, that's it!
```

## upload_file *Upload files to MATOS*

### Description

This function lets you post files to your projects. You will need to log in before uploading.

### Usage

```
upload_file(
  project,
  file,
 data_type = c("new_tags", "receivers", "detections", "events", "gps", "glider"),
  print_response = F
)
```

## Arguments

| | |
|---|---|
| project | The name (character) or number (numeric) of the project to which you wish post your file. |
| file | The file(s) you wish to upload. If the file is located in your working directory, this can be just the filename and extension. You will need to provide the full file location if it is located elsewhere. |
| data_type | Character string. The data type that you are uploading. One of: "new_tags" (default), "receivers", "detections", "events", "gps", or "glider". |
| print_response | Logical. Do you want the POST response to be printed? Mostly useful for diagnostic purposes. Default is FALSE. |

## Details

upload_file takes a project name or number, the file you want to upload, and the type of file you want to upload (defaulting to VRL transmitter detections).

If data_type is "new_tags", "receivers", or "glider", only CSV and XLS/XLSX files are accepted; if "detections", only VRL and CSV files are accepted; if "events" or "gps", only CSV is accepted.

Multiple files can be uploaded at once, but they must all be the same data type and posted to the same project.

## Value

A notification of the success of your file upload is returned.

## Examples

```
## Not run:
# Newly tagged fish, the default
upload_file(87, "your_tagged_fish.xls")
upload_file(87, "your_tagged_fish.xls", "new_tags")

# Transmitter detections
upload_file(
  "umces boem offshore wind energy",
  "c:/wherever/your_CSV_detections.csv",
  "detections"
)
upload_file(
  "umces boem offshore wind energy",
  "c:/wherever/your_VRL_detections.vrl",
  "detections"
)

# Receiver metadata
upload_file(
  "umces boem offshore wind energy",
```

```
  "your_receiver_metadata.xlsx", "receivers"
)

## End(Not run)
```

---

`utilities` *Internal functions used by* `matos`

---

## Description

Non-exported utility functions used by other functions in `matos`.

## Usage

```
get_file_list(project_number, data_type, force = FALSE)

get_file_list_mem(project_number, data_type)

get_project_number(project_name, matos_projects = NULL)

get_project_name(project_number, matos_projects = NULL)

html_table_to_df(html_file_list)

login_check(url = "https://matos.asascience.com/report/submit")

project_check(project, return_projects = FALSE)

scrape_file_urls(html_file_list)

download_process(url, out_dir, overwrite, to_vue = FALSE, quiet = FALSE)
```

## Arguments

| | |
|---|---|
| `project_number` | Number of the project |
| `data_type` | one of "dataextractionfiles" for OTN detection extracts or "downloadfiles" for the uploaded project files. |
| `force` | Do you want to reset the cache and re-ping the database? Defaults to false. |
| `project_name` | Character string of the full MATOS project name. This will be the big name in bold at the top of your project page, not the "Project Title" below it. Will be coerced to all lower case, so capitalization doesn't matter. |
| `matos_projects` | Data frame. Used to pass the MATOS project list from `project_check`. |
| `html_file_list` | Listed files in HTML form. Always the result of `get_file_list` |
| `url` | The (protected) URL that the overlapping function is trying to call. |
| `project` | MATOS project ID. Can be the name or number of the project. |

return_projects
: Logical. Do you want project_check to return the list of projects? Used to not ping the website too much in one function call.

out_dir
: Character. To what directory would you like your files downloaded? Defaults to the current working directory.

overwrite
: Logical. Do you want to overwrite existing files that have the same name (TRUE) or protect yourself against doing this (FALSE, the default)?

to_vue
: Logical. Should the data be converted to match that of VUE's CSV export? Defaults to FALSE.

quiet
: Logical. Do you want to silence matos' updates? Default is FALSE.

## Details

get_file_list checks to see if it should re-evaluate itself, then wraps get_file_list_mem which is the actual workhorse.

get_file_list_mem memoised function which scrapes the HTML associated with the project or data extraction files page provided with a given project.

get_project_number finds the internal MATOS number associated with each project by scraping the HTML of the main MATOS projects page.

get_project_name finds the MATOS project name associated with the given project number by scraping the HTML of the main MATOS projects page.

html_table_to_df converts the HTML table provided by get_file_list into a R-usable data frame.

login_check pings protected URLs and calls matos_login when referred to the login page.

project_check

scrape_file_urls is used internally by html_table_to_df to extract the URLs associates with each "Download" link.

download_process is used internally by get_project_file and get_extract_file

# Index