

# Package ‘valdr’

April 22, 2026

**Title** Access and Analyse 'VALD' Data via Our External 'APIs'

**Version** 3.0.0

**Description** Provides helper functions and wrappers to simplify authentication, data retrieval, and result processing from the 'VALD' 'APIs'.

Designed to streamline integration for analysts and researchers working with 'VALD's external 'APIs'.

For further documentation on integrating with 'VALD' 'APIs', see:

<<https://support.vald.com/hc/en-au/articles/23415335574553-How-to-integrate-with-VALD-APIs>>.

For a step-by-step guide to using this package, see:

<<https://support.vald.com/hc/en-au/articles/48730811824281-A-guide-to-using-the-valdr-R-package>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** httr, base64enc, keyring, jsonlite, dplyr, readr

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Kieran Harrison [aut, cre],  
VALD Support [ctb],  
VALD [cph] (Copyright holder)

**Maintainer** Kieran Harrison <k.harrison@vald.com>

**Repository** CRAN

**Date/Publication** 2026-04-22 07:10:02 UTC

## Contents

|                                 |   |
|---------------------------------|---|
| .retry_key_get . . . . .        | 2 |
| export_forcedecks_csv . . . . . | 3 |
| get_config . . . . .            | 3 |

|  |    |
|--|----|
| get_forcedecks_data . . . . .                    | 4  |
| get_forcedecks_result_definitions_only . . . . . | 5  |
| get_forcedecks_tests_only . . . . .              | 5  |
| get_forcedecks_tests_trials . . . . .            | 6  |
| get_forcedecks_trials_only . . . . .             | 7  |
| get_forceframe_data . . . . .                    | 8  |
| get_forceframe_repetitions_by_id . . . . .       | 8  |
| get_forceframe_tests_only . . . . .              | 9  |
| get_forceframe_test_by_id . . . . .              | 10 |
| get_nordbord_data . . . . .                      | 11 |
| get_nordbord_tests_only . . . . .                | 11 |
| get_nordbord_test_by_id . . . . .                | 12 |
| get_profiles_groups_categories_mapping . . . . . | 13 |
| get_profiles_only . . . . .                      | 14 |
| get_smartspeed_data . . . . .                    | 14 |
| get_smartspeed_tests_only . . . . .              | 15 |
| get_start_date . . . . .                         | 16 |
| set_credentials . . . . .                        | 16 |
| set_start_date . . . . .                         | 17 |

## Index 18

---

|                |  |
|----------------|--|
| .retry_key_get | <i>Internal helper function to retry reading a key from the system keyring. This handles potential intermittent delays after writing to the keyring.</i> |
|----------------|--|

---

### Description

Internal helper function to retry reading a key from the system keyring. This handles potential intermittent delays after writing to the keyring.

### Usage

```
.retry_key_get(service, username, retries = 5, delay = 0.5)
```

### Arguments

|          |   |
|----------|---|
| service  | Character. Keyring service name (e.g., "valdr_credentials").    |
| username | Character. Keyring username (e.g., "client_id").                |
| retries  | Integer. Number of retry attempts before giving up (default 5). |
| delay    | Numeric. Delay in seconds between retry attempts (default 0.5). |

### Value

Character scalar. The retrieved credential value. Will throw an error if unable to retrieve a valid, non-blank value after all retries.

---

export\_forcedecks\_csv *Export ForceDecks data for dashboards*

---

### Description

Queries the ForceDecks API, joins trials/tests/profiles/result definitions, and writes a CSV file to the user's Downloads/VALD\_Exports folder (by default).

### Usage

```
export_forcedecks_csv(export_dir = NULL)
```

### Arguments

export\_dir      Optional directory to write the CSV to. If NULL, uses ~/Downloads/VALD\_Exports.

### Details

This export is ForceDecks-specific and uses the stored start date from `get_start_date()`, which is automatically updated by the ForceDecks data retrieval functions.

### Value

Invisibly, the full path to the created CSV file.

---

get\_config      *Retrieve stored VALD configuration*

---

### Description

Returns the configuration list previously set by `set_credentials()`. If credentials have not been set, this function will raise an error.

### Usage

```
get_config(safe = TRUE, quiet = FALSE)
```

### Arguments

safe            Logical; if TRUE (default), sensitive values are redacted in printed output (only when not quiet).

quiet          Logical; if TRUE, suppresses printed output (default FALSE).

### Value

A named list containing the stored VALD configuration values for the current user. Sensitive values are redacted when printed with `safe = TRUE`. Returned invisibly.

---

|                     |   |
|---------------------|---|
| get_forcedecks_data | <i>Run full initial data fetch from the VALD ForceDecks API and External Profiles API</i> |
|---------------------|---|

---

## Description

This function is intended for first-time use or a full refresh of all datasets. It retrieves profiles, result definitions, tests (from a specified start date), and trials.

## Usage

```
get_forcedecks_data(start_date = NULL, include_attributes = FALSE)
```

## Arguments

|                    |   |
|--------------------|---|
| start_date         | In ISO 8601 UTC format (e.g., "2025-06-25T00:00:00Z") indicating the start of the test retrieval window.  |
| include_attributes | Logical; if TRUE, the returned list will include an additional data frame test_attributes containing a long-format mapping of test attributes. Defaults to FALSE. |

## Value

A named list with data frames: profiles, result\_definitions, tests, and trials. If include\_attributes = TRUE, the list also contains test\_attributes.

## Examples

```
## Not run:  
# Fetch all data (profiles, results, tests, trials)  
data <- get_forcedecks_data()  
View(data$profiles)  
  
# Fetch all data including test attributes mapping  
data_with_attrs <- get_forcedecks_data(include_attributes = TRUE)  
View(data_with_attrs$test_attributes)  
  
## End(Not run)
```

---

```
get_forcedecks_result_definitions_only
```

*Get only ForceDecks result definitions*

---

**Description**

Wrapper around `get_forcedecks_result_definitions()` to refresh result definitions. Typically called infrequently unless definitions are known to have changed.

**Usage**

```
get_forcedecks_result_definitions_only()
```

**Value**

A data frame where each row corresponds to a ForceDecks result definition retrieved from the API. Returned invisibly.

**Examples**

```
## Not run:
# Fetch result definitions
results <- get_forcedecks_result_definitions_only()
View(results)

## End(Not run)
```

---

```
get_forcedecks_tests_only
```

*Get only ForceDecks test data*

---

**Description**

Wrapper around `get_forcedecks_tests()` if the user only wants test-level data without trials.

**Usage**

```
get_forcedecks_tests_only(start_date = NULL, include_attributes = FALSE)
```

**Arguments**

|                                 |   |
|---------------------------------|---|
| <code>start_date</code>         | In ISO 8601 UTC format (e.g., "2025-06-25T00:00:00Z") indicating the start of the test retrieval window.  |
| <code>include_attributes</code> | Logical; if TRUE, returns a named list with <code>tests</code> (the test data frame) and <code>attributes</code> (the long-format attributes mapping table). If FALSE (default), returns only the tests data frame. |

**Value**

Either a data frame containing ForceDecks test-level data, or a list with components tests and attributes if include\_attributes = TRUE.

**Examples**

```
## Not run:
# Fetch only tests
tests <- get_forcedecks_tests_only()

# Fetch tests plus attributes mapping
res <- get_forcedecks_tests_only(include_attributes = TRUE)
tests <- res$tests
attrs <- res$attributes

## End(Not run)
```

---

```
get_forcedecks_tests_trials
```

*Run a standard session to get new ForceDecks tests and trials only*

---

**Description**

Use this when profiles and result definitions have already been downloaded previously.

**Usage**

```
get_forcedecks_tests_trials(start_date = NULL, include_attributes = FALSE)
```

**Arguments**

|                    |   |
|--------------------|---|
| start_date         | In ISO 8601 UTC format (e.g., "2025-06-25T00:00:00Z") indicating the start of the test retrieval window.  |
| include_attributes | Logical; if TRUE, the returned list will include an additional data frame test_attributes containing a long-format mapping of test attributes. Defaults to FALSE. |

**Value**

A named list with data frames: tests and trials. If include\_attributes = TRUE, the list also contains test\_attributes.

### Examples

```
## Not run:  
# Fetch tests and trials only  
session <- get_forcedecks_tests_trials()  
View(session$tests)  
View(session$trials)  
  
# Fetch tests, trials, and test attributes mapping  
session2 <- get_forcedecks_tests_trials(include_attributes = TRUE)  
View(session2$test_attributes)  
  
## End(Not run)
```

---

`get_forcedecks_trials_only`

*Get trials for an existing test data frame*

---

### Description

Useful for re-running or extending analysis without re-downloading tests.

### Usage

```
get_forcedecks_trials_only(tests_df)
```

### Arguments

`tests_df`            A data frame of tests (from `get_forcedecks_tests()` or a previous session)

### Value

A data frame containing trial-level results corresponding to the input tests.

### Examples

```
## Not run:  
# Fetch trials based on existing tests  
tests <- get_forcedecks_tests_only()  
trials <- get_forcedecks_trials_only(tests)  
View(trials)  
  
## End(Not run)
```

---

get\_forceframe\_data    *Get ForceFrame test data*

---

### Description

Wrapper around `get_forceframe_tests()` that also calls `get_profiles()` to ensure profile data is available before retrieving ForceFrame test data. Intended for use when you want to fetch Profiles information as well as ForceFrame tests from the API.

### Usage

```
get_forceframe_data(start_date = NULL, profile_id = NULL)
```

### Arguments

`start_date`        (Optional) A UTC ISO 8601 datetime string (e.g. "2025-06-25T00:00:00Z") used to filter results by modification time. Input as a string.

`profile_id`        (Optional) A specific profile ID to filter results for a single athlete. Input as a string.

### Value

A data frame containing Profiles information and a data frame containing ForceFrame test data.

### Examples

```
## Not run:  
# Fetch all recent ForceFrame tests along with Profiles information  
tests <- get_forceframe_data()  
View(tests)  
  
# Fetch ForceFrame tests for a specific profile  
get_forceframe_data(profile_id = "abc123")  
  
## End(Not run)
```

---

get\_forceframe\_repetitions\_by\_id  
                                  *Get ForceFrame repetitions for a single test by ID*

---

### Description

Wrapper around `get_forceframe_reps_by_id()` to retrieve repetition-level ForceFrame data for a specific test.

**Usage**

```
get_forceframe_repetitions_by_id(test_id)
```

**Arguments**

`test_id` (Required) The unique test ID for the ForceFrame test you want to retrieve repetitions for.

**Value**

A data frame with one row per repetition corresponding to the requested ForceFrame test. Returned invisibly.

**Examples**

```
## Not run:  
# Fetch ForceFrame repetitions by test ID  
reps <- get_forceframe_repetitions_by_id("abcd1234-ab12-cd34-ef56-abcdef123456")  
View(reps)  
  
## End(Not run)
```

---

```
get_forceframe_tests_only  
  Get only ForceFrame test data
```

---

**Description**

Wrapper around `get_forceframe_tests()` to retrieve ForceFrame test data. Intended for use when you want to fetch multiple ForceFrame tests.

**Usage**

```
get_forceframe_tests_only(start_date = NULL, profile_id = NULL)
```

**Arguments**

`start_date` (Optional) A UTC ISO 8601 datetime string (e.g. "2025-06-25T00:00:00Z") used to filter results by modification time. Input as a string.

`profile_id` (Optional) A specific profile ID to filter results for a single athlete. Input as a string.

**Value**

A data frame where each row corresponds to a ForceFrame test retrieved from the API. Returned invisibly.

## Examples

```
## Not run:
# Fetch all recent ForceFrame tests
tests <- get_forceframe_tests_only()
View(tests)

# Fetch ForceFrame tests for a specific profile
get_forceframe_tests_only(profile_id = "abcd1234-ab12-cd34-ef56-abcdef123456")

## End(Not run)
```

---

get\_forceframe\_test\_by\_id

*Get a single ForceFrame test by ID*

---

## Description

Wrapper around `get_forceframe_tests_by_id()` to retrieve a specific ForceFrame test. Intended for use when you want to fetch a single ForceFrame test record.

## Usage

```
get_forceframe_test_by_id(test_id)
```

## Arguments

`test_id` (Required) The unique test ID for the ForceFrame test you want to retrieve.

## Value

A data frame with one row corresponding to the requested ForceFrame test. Returned invisibly.

## Examples

```
## Not run:
# Fetch a ForceFrame test by ID
test <- get_forceframe_test_by_id("abcd1234-ab12-cd34-ef56-abcdef123456")
View(test)

## End(Not run)
```

---

get\_nordbord\_data      *Get NordBord test data*

---

### Description

Wrapper around `get_nordbord_tests()` that also calls `get_profiles()` to ensure profile data is available before retrieving NordBord test data. Intended for use when you want to fetch Profiles information as well as NordBord tests from the API.

### Usage

```
get_nordbord_data(start_date = NULL, profile_id = NULL)
```

### Arguments

`start_date`      (Optional) A UTC ISO 8601 datetime string (e.g. "2025-06-25T00:00:00Z") used to filter results by modification time. Input as a string.

`profile_id`      (Optional) A specific profile ID to filter results for a single athlete. Input as a string.

### Value

A data frame containing Profiles information and a data frame containing NordBord test data.

### Examples

```
## Not run:
# Fetch all recent NordBord tests along with Profiles information
tests <- get_nordbord_data()
View(tests)

# Fetch NordBord tests for a specific profile
get_nordbord_data(profile_id = "abcd1234-ab12-cd34-ef56-abcdef123456")

## End(Not run)
```

---

get\_nordbord\_tests\_only  
*Get only NordBord test data*

---

### Description

Wrapper around `get_nordbord_tests()` to retrieve NordBord test data. Intended for use when you want to fetch multiple NordBord tests.

**Usage**

```
get_nordbord_tests_only(start_date = NULL, profile_id = NULL)
```

**Arguments**

|            |   |
|------------|---|
| start_date | (Optional) A UTC ISO 8601 datetime string (e.g. "2025-06-25T00:00:00Z") used to filter results by modification time. Input as a string. |
| profile_id | (Optional) A specific profile ID to filter results for a single athlete. Input as a string.   |

**Value**

A data frame where each row corresponds to a NordBord test retrieved from the API. Returned invisibly.

**Examples**

```
## Not run:
# Fetch all recent NordBord tests
tests <- get_nordbord_tests_only()
View(tests)

# Fetch NordBord tests for a specific profile
get_nordbord_tests_only(profile_id = "abcd1234-ab12-cd34-ef56-abcdef123456")

## End(Not run)
```

---

```
get_nordbord_test_by_id
```

*Get a single NordBord test by ID*

---

**Description**

Wrapper around `get_nordbord_tests_by_id()` to retrieve a specific NordBord test. Intended for use when you want to fetch a single NordBord test record.

**Usage**

```
get_nordbord_test_by_id(test_id)
```

**Arguments**

|         |   |
|---------|---|
| test_id | (Required) The unique test ID for the NordBord test you want to retrieve. |
|---------|---|

**Value**

A data frame with one row corresponding to the requested NordBord test. Returned invisibly.

**Examples**

```
## Not run:  
# Fetch a NordBord test by ID  
test <- get_nordbord_test_by_id("abcd1234-ab12-cd34-ef56-abcdef123456")  
View(test)  
  
## End(Not run)
```

---

```
get_profiles_groups_categories_mapping  
Get profile-group-category mappings
```

---

**Description**

Wrapper around `get_profiles_groups_categories()` to retrieve the long-format mapping of profiles to their groups and categories.

**Usage**

```
get_profiles_groups_categories_mapping()
```

**Details**

Intended for use when you want a relational view of profile metadata rather than the one-row-per-profile structure returned by `get_profiles()`.

**Value**

A data frame with one row per profile-group-category combination, containing the columns:

- `profileId`
- `categoryId`
- `categoryName`
- `groupId`
- `groupName`

Returned invisibly.

**Examples**

```
## Not run:  
# Fetch profile-group-category mappings for the current tenant  
mappings <- get_profiles_groups_categories_mapping()  
head(mappings)  
  
## End(Not run)
```

---

get\_profiles\_only      *Get only VALD profiles*

---

**Description**

Wrapper around get\_profiles() to retrieve VALD profiles. Useful if profile data needs to be refreshed independently.

**Usage**

```
get_profiles_only()
```

**Value**

A data frame containing VALD profiles information for the stored tenant\_id.

**Examples**

```
## Not run:  
# Fetch profiles  
profiles <- get_profiles_only()  
View(profiles)  
  
## End(Not run)
```

---

get\_smartspeed\_data      *Get SmartSpeed test data*

---

**Description**

Wrapper around get\_smartspeed\_tests() that also calls get\_profiles() to ensure profile data is available before retrieving SmartSpeed test data. Intended for use when you want to fetch Profiles information as well as SmartSpeed tests from the API.

**Usage**

```
get_smartspeed_data(start_date = NULL)
```

**Arguments**

start\_date      (Optional) A UTC ISO 8601 datetime string (e.g. "2025-06-25T00:00:00Z") used to filter results by modification time. Input as a string.

**Value**

A named list containing Profiles information and SmartSpeed test data.

**Examples**

```
## Not run:  
# Fetch all recent SmartSpeed tests along with Profiles information  
data <- get_smartspeed_data()  
View(data$profiles)  
View(data$tests)  
  
## End(Not run)
```

---

```
get_smartspeed_tests_only  
    Get only SmartSpeed test data
```

---

**Description**

Wrapper around `get_smartspeed_tests()` to retrieve SmartSpeed test data. Intended for use when you want to fetch multiple SmartSpeed tests.

**Usage**

```
get_smartspeed_tests_only(start_date = NULL)
```

**Arguments**

`start_date` (Optional) A UTC ISO 8601 datetime string (e.g. "2025-06-25T00:00:00Z") used to filter results by modification time. Input as a string.

**Value**

A data frame where each row corresponds to a SmartSpeed test summary retrieved from the API. Returned invisibly.

**Examples**

```
## Not run:  
# Fetch all recent SmartSpeed tests  
tests <- get_smartspeed_tests_only()  
View(tests)  
  
# Fetch SmartSpeed tests using a one-off start date override  
tests <- get_smartspeed_tests_only(start_date = "2025-07-01T00:00:00Z")  
View(tests)  
  
## End(Not run)
```

---

|                |  |
|----------------|--|
| get_start_date | <i>Retrieve the start date from config</i> |
|----------------|--|

---

**Description**

Returns the saved start date string from the config file.

**Usage**

```
get_start_date()
```

**Value**

A character scalar containing the start date string in ISO 8601 format previously saved to configuration.

---

|                 |  |
|-----------------|--|
| set_credentials | <i>Set and Save VALD API Credentials</i> |
|-----------------|--|

---

**Description**

Securely stores VALD API credentials in the system keyring and saves non-sensitive configuration to a JSON config file in the user's home directory for reuse across sessions.

**Usage**

```
set_credentials(client_id, client_secret, tenant_id, region)
```

**Arguments**

|               |  |
|---------------|--|
| client_id     | Your VALD API Client ID (stored securely in keyring)     |
| client_secret | Your VALD API Client Secret (stored securely in keyring) |
| tenant_id     | Your VALD Tenant ID                                      |
| region        | The VALD data region code (e.g., "aue", "use", "euw")    |

**Details**

Sensitive values (client ID and secret) are never written to disk and are retrieved securely from the keyring when needed.

**Value**

Invisibly returns TRUE if credentials and configuration were saved successfully.

---

|                |                                       |
|----------------|---------------------------------------|
| set_start_date | <i>Set and persist the start date</i> |
|----------------|---------------------------------------|

---

**Description**

Saves the provided ISO 8601 start date to the config file.

**Usage**

```
set_start_date(start_date)
```

**Arguments**

|            |   |
|------------|---|
| start_date | Date in ISO 8601 UTC format, e.g., "2025-06-25T00:00:00Z" |
|------------|---|

**Value**

A logical scalar (TRUE), returned invisibly, indicating that the start date was saved and persisted successfully.

# Index

.retry\_key\_get, [2](#)  
export\_forcedecks\_csv, [3](#)  
  
get\_config, [3](#)  
get\_forcedecks\_data, [4](#)  
get\_forcedecks\_result\_definitions\_only,  
[5](#)  
get\_forcedecks\_tests\_only, [5](#)  
get\_forcedecks\_tests\_trials, [6](#)  
get\_forcedecks\_trials\_only, [7](#)  
get\_forceframe\_data, [8](#)  
get\_forceframe\_repetitions\_by\_id, [8](#)  
get\_forceframe\_test\_by\_id, [10](#)  
get\_forceframe\_tests\_only, [9](#)  
get\_nordbord\_data, [11](#)  
get\_nordbord\_test\_by\_id, [12](#)  
get\_nordbord\_tests\_only, [11](#)  
get\_profiles\_groups\_categories\_mapping,  
[13](#)  
get\_profiles\_only, [14](#)  
get\_smartspeed\_data, [14](#)  
get\_smartspeed\_tests\_only, [15](#)  
get\_start\_date, [16](#)  
  
set\_credentials, [16](#)  
set\_start\_date, [17](#)