

# Package ‘diario’

April 7, 2026

**Type** Package

**Title** 'R' Interface to the 'Diariodeobras' Application

**Version** 0.1.1

**Date** 2026-04-07

**Description** Provides a set of functions for securely storing 'API' tokens and interacting with the <<https://diariodeobras.net>> system. Includes convenient wrappers around the 'httr2' package to perform authenticated requests, retrieve project details, tasks, reports, and more.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Depends** R (>= 4.1.0)

**Imports** cli, tibble, httr2, keyring

**URL** <https://github.com/StrategicProjects/diario>,  
<https://strategicprojects.github.io/diario/>

**BugReports** <https://github.com/StrategicProjects/diario/issues>

**Config/Needs/website** tidyverse/tidytemplate

**NeedsCompilation** no

**Author** Andre Leite [aut, cre],  
Hugo Vaconcelos [aut],  
Diogo Bezerra [aut],  
Marcos Wasilew [aut],  
Carlos Amorin [aut]

**Maintainer** Andre Leite <leite@castlab.org>

**Repository** CRAN

**Date/Publication** 2026-04-07 17:50:02 UTC

## Contents

diario_get_company . . . . .	2
diario_get_entities . . . . .	3
diario_get_projects . . . . .	3
diario_get_project_details . . . . .	4
diario_get_reports . . . . .	4
diario_get_report_details . . . . .	5
diario_get_task_details . . . . .	6
diario_get_task_list . . . . .	6
diario_perform_request . . . . .	7
diario_retrieve_token . . . . .	8
diario_store_token . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

diario_get_company	<i>Get company details</i>
--------------------	----------------------------

---

### Description

This function retrieves company details from the Diario API.

### Usage

```
diario_get_company()
```

### Value

A list containing company details.

### Examples

```
## Not run:
company <- diario_get_company()

## End(Not run)
```

---

diario\_get\_entities    *Get all registered entities (cadastrros)*

---

**Description**

This function retrieves all registered entities from the Diario API.

**Usage**

```
diario_get_entities()
```

**Value**

A list containing the entities data.

**Examples**

```
## Not run:  
entities <- diario_get_entities()  
  
## End(Not run)
```

---

diario\_get\_projects    *Get list of projects (obras)*

---

**Description**

This function retrieves a list of projects from the Diario API.

**Usage**

```
diario_get_projects()
```

**Value**

A tibble containing the projects data.

**Examples**

```
## Not run:  
projects <- diario_get_projects()  
  
## End(Not run)
```

diario\_get\_project\_details  
*Get details of a specific project*

---

**Description**

This function retrieves details of a specific project by its ID from the Diario API.

**Usage**

```
diario_get_project_details(project_id)
```

**Arguments**

project\_id      A valid non-empty string with the project ID.

**Value**

A list containing the project details.

**Examples**

```
## Not run:  
project <- diario_get_project_details("66face5fe26175e0a904d398")  
  
## End(Not run)
```

---

diario\_get\_reports      *Get reports of a specific project*

---

**Description**

This function retrieves reports of a specific project with optional parameters for limit and order.

**Usage**

```
diario_get_reports(project_id, limit = 50, order = "desc")
```

**Arguments**

project\_id      A valid non-empty string with the project ID.  
limit            An integer specifying the maximum number of reports to retrieve. Default is 50.  
order            A character string specifying the order of the reports (e.g., "asc" or "desc").  
                  Default is "desc".

**Value**

A tibble containing the reports.

**Examples**

```
## Not run:  
reports <- diario_get_reports("6717f864d163f517ae06e242", limit = 10, order = "asc")  
  
## End(Not run)
```

---

```
diario_get_report_details  
  Get details of a specific report
```

---

**Description**

This function retrieves details of a specific report by report ID within a project.

**Usage**

```
diario_get_report_details(project_id, report_id)
```

**Arguments**

project_id	A valid non-empty string with the project ID.
report_id	A valid non-empty string with the report ID.

**Value**

A list containing the report details.

**Examples**

```
## Not run:  
report <- diario_get_report_details("6717f864d163f517ae06e242", "67648080f0971de9d00324c2")  
  
## End(Not run)
```

---

`diario_get_task_details`*Get details of a specific task*

---

**Description**

This function retrieves details of a specific task by task ID within a project.

**Usage**

```
diario_get_task_details(project_id, task_id)
```

**Arguments**

`project_id` A valid non-empty string with the project ID.

`task_id` A valid non-empty string with the task ID.

**Value**

A list containing task details.

**Examples**

```
## Not run:  
task <- diario_get_task_details("66cf438223aa80386306e647", "66cf44209e4fedefb306bcd3")  
  
## End(Not run)
```

---

`diario_get_task_list` *Get the task list of a specific project*

---

**Description**

This function retrieves the task list of a specific project by its ID.

**Usage**

```
diario_get_task_list(project_id)
```

**Arguments**

`project_id` A valid non-empty string with the project ID.

**Value**

A tibble containing the task list.

## Examples

```
## Not run:
tasks <- diario_get_task_list("66cf438223aa80386306e647")

## End(Not run)
```

---

diario\_perform\_request

*Perform an API request to Diario*

---

## Description

This function performs an authenticated request to the specified endpoint of the Diario API.

## Usage

```
diario_perform_request(
  endpoint,
  query = list(),
  method = "GET",
  body = NULL,
  verbosity = 0
)
```

## Arguments

endpoint	A non-empty character string specifying the API endpoint.
query	A named list of query parameters (optional).
method	The HTTP method to use (e.g., "GET", "POST", "PUT", "DELETE"). Default is "GET".
body	A list representing the JSON body for request methods that support a body (e.g., POST).
verbosity	Verbosity level for the request (0 = none, 1 = minimal). Default is 0.

## Value

A list (by default) containing the response from the API. If the content is JSON, it will be returned as an R object. If not, an error is raised.

## Examples

```
## Not run:
diario_perform_request("v1/obras", query = list(status = "active"))

## End(Not run)
```

---

diario\_retrieve\_token *Retrieve the API token for Diario*

---

**Description**

This function retrieves the stored authentication token using the keyring package. If no valid token (or keyring) is found, it will return NULL and emit an informational message indicating that no valid token was found.

**Usage**

```
diario_retrieve_token()
```

**Value**

A character string containing the API token, or NULL if no valid token is found.

**Examples**

```
## Not run:  
token <- diario_retrieve_token()  
  
## End(Not run)
```

---

diario\_store\_token *Store the API token for Diario*

---

**Description**

This function stores the provided authentication token using the keyring package. If the token cannot be stored (for example, because the keyring is not accessible), it emits a warning instead of throwing an error, and returns FALSE.

**Usage**

```
diario_store_token(token)
```

**Arguments**

token            A character string containing the API token to be stored.

**Value**

TRUE (invisibly) if the token was stored successfully; FALSE otherwise.

**Examples**

```
## Not run:  
# Attempt to store a token:  
diario_store_token("your-api-token")  
  
## End(Not run)
```

# Index

[diario\\_get\\_company](#), 2  
[diario\\_get\\_entities](#), 3  
[diario\\_get\\_project\\_details](#), 4  
[diario\\_get\\_projects](#), 3  
[diario\\_get\\_report\\_details](#), 5  
[diario\\_get\\_reports](#), 4  
[diario\\_get\\_task\\_details](#), 6  
[diario\\_get\\_task\\_list](#), 6  
[diario\\_perform\\_request](#), 7  
[diario\\_retrieve\\_token](#), 8  
[diario\\_store\\_token](#), 8