

# Package: hakaiApi (via r-universe)

October 14, 2026

**Title** Authenticated HTTP Request Client for the 'Hakai' API

**Version** 1.0.5.9000

**Description** Initializes a class that obtains API credentials and provides a method to use those credentials to make GET requests to the 'Hakai' API server. Usage instructions are documented at <https://hakaiinstitute.github.io/hakai-api/>.

**License** MIT + file LICENSE

**URL** <https://github.com/HakaiInstitute/hakai-api-client-r>,  
<https://hakaiinstitute.github.io/hakai-api-client-r/>

**BugReports** <https://github.com/HakaiInstitute/hakai-api-client-r/issues>

**Depends** R (>= 4.2.0)

**Imports** dplyr, httr2, R6, readr, tibble

**Suggests** knitr, markdown, rmarkdown, sf, withr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-CA

**Config/testthat/edition** 3

**Config/Needs/website** rmarkdown

**Config/roxygen2/version** 8.0.0

**Config/pak/sysreqs** libssl-dev libx11-dev

**Repository** <https://hakaiinstitute.r-universe.dev>

**Date/Publication** 2026-10-14 21:03:35 UTC

**RemoteUrl** <https://github.com/HakaiInstitute/hakai-api-client-r>

**RemoteRef** HEAD

**RemoteSha** af2a0b655d9d9b51e2d316532125720d3522eeb0

**RemoteSubdir** hakaiApi

## Contents

|                  |          |
|------------------|----------|
| Client . . . . . | 2        |
| <b>Index</b>     | <b>7</b> |

---

|        |                                   |
|--------|-----------------------------------|
| Client | <i>The Hakai API Client Class</i> |
|--------|-----------------------------------|

---

### Description

Class to use to make authenticated API requests for Hakai data. Credentials can be provided via the HAKAI\_API\_TOKEN environment variable or through a credentials file.

### Public fields

`api_root` The `api_root` you are logged in to

### Methods

#### Public methods:

- `Client$new()`
- `Client$get()`
- `Client$get_stations()`
- `Client$post()`
- `Client$put()`
- `Client$patch()`
- `Client$remove_credentials()`
- `Client$clone()`

`Client$new()`: Log into Google to gain credential access to the API

#### Usage:

```
Client$new(
  api_root = "https://portal.hakai.org/api",
  login_page = NULL,
  credentials_file = NULL
)
```

#### Arguments:

`api_root` Optional API base url to fetch data. Defaults to "https://portal.hakai.org/api"

`login_page` Optional API login page url to display to user. When NULL (the default) it is derived from the `api_root` host, so a custom `api_root` authenticates against the matching server.

`credentials_file` Optional path to the credentials cache file. Defaults to a file in the user's data directory as determined by `tools::R_user_dir()`

*Details:* Credentials can be provided in two ways: 1. Via the HAKAI\_API\_TOKEN environment variable (contains query string: "token\_type=Bearer&access\_token=...") 2. Via a credentials file (default: in user data directory via tools::R\_user\_dir()) The environment variable takes precedence if both are available.

*Returns:* A client instance

*Examples:*

```
try(
  client <- Client$new()
)
# Using environment variable
Sys.setenv(HAKAI_API_TOKEN = "token_type=Bearer&access_token=TOKEN")
try(
  client <- Client$new()
)
# Using custom credentials file
try(
  client <- Client$new(credentials_file = "/path/to/creds")
)
```

`Client$get()`: Send a GET request to the API

*Usage:*

```
Client$get(endpoint_url, col_types = NULL, verbose = FALSE)
```

*Arguments:*

`endpoint_url` The API endpoint url - can be a full URL or a relative path that will be appended to the `api_root`

`col_types` a readr type specification

`verbose` When TRUE, print the request and response details (including the resolved url) to aid debugging

*Returns:* A dataframe of the requested data

*Examples:*

```
try(client$get("/aco/views/projects"))
try(client$get("https://portal.hakai.org/api/aco/views/projects"))
try(client$get("/aco/views/projects", verbose = TRUE))
```

`Client$get_stations()`: Get recent sensor nodes and their locations as an sf object

*Usage:*

```
Client$get_stations(verbose = FALSE)
```

*Arguments:*

`verbose` When TRUE, print the request and response details (including the resolved url) to aid debugging

*Returns:* An sf object of sensor nodes and their locations

*Examples:*

```
try(client$get_stations())
```

**Client\$post():** Send a POST request to the API

*Usage:*

```
Client$post(endpoint_url, rec_data, verbose = FALSE)
```

*Arguments:*

**endpoint\_url** The API endpoint url - can be a full URL or a relative path that will be appended to the `api_root`

**rec\_data** dataframe, list, or other R data structure to send as part of the post request payload

**verbose** When TRUE, print the request and response details (including the resolved url) to aid debugging

*Returns:* post request response status code and description

**Client\$put():** Send a PUT request to the API

*Usage:*

```
Client$put(endpoint_url, rec_data, verbose = FALSE)
```

*Arguments:*

**endpoint\_url** The API endpoint url - can be a full URL or a relative path that will be appended to the `api_root`

**rec\_data** dataframe, list, or other R data structure to send as part of the post request payload

**verbose** When TRUE, print the request and response details (including the resolved url) to aid debugging

*Returns:* PUT request response status code and description

**Client\$patch():** Send a PATCH request to the API

*Usage:*

```
Client$patch(endpoint_url, rec_data, verbose = FALSE)
```

*Arguments:*

**endpoint\_url** The API endpoint url - can be a full URL or a relative path that will be appended to the `api_root`

**rec\_data** dataframe, list, or other R data structure to send as part of the post request payload

**verbose** When TRUE, print the request and response details (including the resolved url) to aid debugging

*Returns:* PATCH request response status code and description

**Client\$remove\_credentials():** Remove your cached login credentials to logout of the client

*Usage:*

```
Client$remove_credentials()
```

*Examples:*

```
try(  
  client$remove_credentials()  
)
```

**Client\$clone():** The objects of this class are cloneable with this method.

*Usage:*

```
Client$clone(deep = FALSE)
```

*Arguments:*

**deep** Whether to make a deep clone.

**Examples**

```
## Not run:
# Initialize a new client
try(
  client <- Client$new()
)

# Or use environment variable for token
Sys.setenv(HAKAI_API_TOKEN = "token_type=Bearer&access_token=TOKEN")
try(
  client <- Client$new()
)

# Follow authorization prompts to log in

# Retrieve some data. See <https://hakaiinstitute.github.io/hakai-api/> for options.
try(
  projects_2020 <- client$get("/aco/views/projects?project_year=2020&fields=project_name")
)

try(
  print(projects_2020)
)
# # A tibble: 20 x 1
#   project_name
#   <chr>
# 1 Fountain FN
# 2 Haig Glacier
# 3 Fraser River - Chimney Creek West William Canyon
# 4 Cruickshank WS
# ...

## End(Not run)

## -----
## Method `Client$new()`
## -----

try(
  client <- Client$new()
)
# Using environment variable
Sys.setenv(HAKAI_API_TOKEN = "token_type=Bearer&access_token=TOKEN")
try(
  client <- Client$new()
)
# Using custom credentials file
try(
  client <- Client$new(credentials_file = "/path/to/creds")
)

## -----
```

```
## Method `Client$get()`  
## -----  
  
try(client$get("/aco/views/projects"))  
try(client$get("https://portal.hakai.org/api/aco/views/projects"))  
try(client$get("/aco/views/projects", verbose = TRUE))  
  
## -----  
## Method `Client$get_stations()`  
## -----  
  
try(client$get_stations())  
  
## -----  
## Method `Client$remove_credentials()`  
## -----  
  
try(  
  client$remove_credentials()  
)
```

# Index

Client, [2](#)