

# Package ‘connectcreds’

January 31, 2025

**Title** Manage 'OAuth' Credentials from 'Posit Connect'

**Version** 0.1.0

**Description** A toolkit for making use of credentials mediated by 'Posit Connect'. It handles the details of communicating with the Connect API correctly, 'OAuth' token caching, and refresh behaviour.

**License** MIT + file LICENSE

**Imports** cli, httr2 (>= 1.1.0), rlang (>= 1.1.0)

**Suggests** testthat (>= 3.0.0), withr

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Config/testthat/edition** 3

**URL** <https://github.com/posit-dev/connectcreds>

**BugReports** <https://github.com/posit-dev/connectcreds/issues>

**NeedsCompilation** no

**Author** Aaron Jacobs [aut, cre],  
Posit Software, PBC [cph, fnd]

**Maintainer** Aaron Jacobs <aaron.jacobs@posit.co>

**Repository** CRAN

**Date/Publication** 2025-01-31 11:00:11 UTC

## Contents

connect_viewer_token . . . . .	2
with_mocked_connect_responses . . . . .	3
<b>Index</b>	<b>5</b>

---

connect\_viewer\_token *Viewer-based credentials on Posit Connect*

---

## Description

Request an OAuth access token for a third-party resource belonging to the user associated with a given Shiny session. This works by exchanging a short-lived session credential for OAuth tokens issued to the client managed by the Connect server, without the Shiny app in question having to manage the user's authentication flow (or the associated client credentials) itself.

## Usage

```
connect_viewer_token(  
  resource = NULL,  
  scope = NULL,  
  session = get_connect_session(),  
  server_url = Sys.getenv("CONNECT_SERVER"),  
  api_key = Sys.getenv("CONNECT_API_KEY")  
)  
  
has_viewer_token(..., session = get_connect_session())
```

## Arguments

resource	The URI that identifies the resource that the client is trying to access, if applicable.
scope	Scopes to be requested from the resource owner.
session	A Shiny session object. By default, this grabs the Shiny session of the parent environment (if any), provided we are also running on Connect.
server_url	The Connect server to exchange credentials with. Defaults to the value of the CONNECT_SERVER environment variable, which is set automatically when running on Connect.
api_key	An API key for the Connect server. Defaults to the value of the CONNECT_API_KEY environment variable, which is set automatically when running on Connect.
...	Further arguments passed on to <a href="#">connect_viewer_token()</a> .

## Details

`connect_viewer_token()` handles caching automatically.

## Value

`connect_viewer_token()` returns an `httr2::oauth_token`.

`has_viewer_token()` returns TRUE if the session has a viewer token and FALSE otherwise.

## Examples

```
token <- "default-token"
if (has_viewer_token()) {
  token <- connect_viewer_token()
}
```

---

```
with_mocked_connect_responses
```

*Mock responses from the Posit Connect server*

---

## Description

These functions can be used to temporarily mock responses from the Connect server, which is useful for writing tests that verify the behaviour of viewer-based credentials.

## Usage

```
with_mocked_connect_responses(
  code,
  mock = NULL,
  token = NULL,
  error = FALSE,
  env = caller_env()
)

local_mocked_connect_responses(
  mock = NULL,
  token = NULL,
  error = FALSE,
  env = caller_env()
)
```

## Arguments

code	Code to execute in the temporary environment.
mock	A function, a list, or NULL. <ul style="list-style-type: none"><li>• NULL disables mocking and returns htr2 to regular operation.</li><li>• A list of responses will be returned in sequence. After all responses have been used up, will return 503 server errors.</li><li>• For maximum flexibility, you can supply a function that takes a single argument, req, and returns either NULL (if it doesn't want to handle the request) or a <a href="#">response</a> (if it does).</li></ul>
token	When not NULL, return this token from the Connect server.
error	When TRUE, return an error from the Connect server.
env	Environment to use for scoping changes.

**Value**

`with_mocked_connect_responses()` returns the result of evaluating code.

**Examples**

```
with_mocked_connect_responses(  
    connect_viewer_token(),  
    token = "test"  
)
```

# Index

`connect_viewer_token`, [2](#)  
`connect_viewer_token()`, [2](#)

`has_viewer_token`  
    (`connect_viewer_token`), [2](#)  
`has_viewer_token()`, [2](#)  
`httr2::oauth_token`, [2](#)

`local_mocked_connect_responses`  
    (`with_mocked_connect_responses`),  
    [3](#)

`response`, [3](#)

`with_mocked_connect_responses`, [3](#)  
`with_mocked_connect_responses()`, [4](#)