

# Package: snowflakeauth (via r-universe)

May 18, 2026

**Title** Authentication Helpers for 'Snowflake'

**Version** 0.2.2.9000

**Description** Authentication helpers for 'Snowflake'. It provides compatibility with authentication approaches supported by the 'Snowflake Connector for Python' <<https://pypi.org/project/snowflake-connector-python>> and the 'Snowflake CLI' <<https://pypi.org/project/snowflake-cli>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** cli, curl, jsonlite, RcppTOML, rlang, jose, openssl

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

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Config/testthat/edition** 3

**URL** <https://posit-dev.github.io/snowflakeauth/>,  
<https://github.com/posit-dev/snowflakeauth>

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

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

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

**Date/Publication** 2026-05-18 15:53:37 UTC

**RemoteUrl** <https://github.com/posit-dev/snowflakeauth>

**RemoteRef** HEAD

**RemoteSha** 4bbc9d3173f05337c7db3442e5721a91c14da6a1

## Contents

has_a_default_connection . . . . .	2
snowflake_connection . . . . .	2
snowflake_credentials . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

has\_a\_default\_connection

*Reports whether a default connection is available*

---

### Description

Reports whether a default connection is available

### Usage

```
has_a_default_connection(...)
```

### Arguments

... arguments passed to [snowflake\\_connection\(\)](#)

### Value

Logical value indicating whether a default connection is available.

### Examples

```
has_a_default_connection()
```

---

snowflake\_connection *Snowflake connection parameter configuration*

---

### Description

Reads Snowflake connection parameters from the `connections.toml` and `config.toml` files used by the [Snowflake Connector for Python](#) and the [Snowflake CLI](#), or specifies them for a connection manually.

### Usage

```
snowflake_connection(name = NULL, ..., .config_dir = NULL, .verbose = FALSE)
```

### Arguments

`name` A named connection. Defaults to `$SNOWFLAKE_DEFAULT_CONNECTION_NAME` if set, the `default_connection_name` from the `config.toml` file (if present), and finally the `[default]` section of the `connections.toml` file, if any. See [Snowflake's documentation](#) for details.

... Additional connection parameters. See **Common parameters**.

<code>.config_dir</code>	The directory to search for a <code>connections.toml</code> and <code>config.toml</code> file. Defaults to <code>\$SNOWFLAKE_HOME</code> or <code>~/ .snowflake</code> if that directory exists, otherwise it falls back to a platform-specific default. See <a href="#">Snowflake's documentation</a> for details.
<code>.verbose</code>	Logical; if <code>TRUE</code> , prints detailed information about configuration loading, including which files are read and how connection parameters are resolved. Defaults to <code>FALSE</code> .

### Value

An object of class "snowflake\_connection".

### Common parameters

The following is a list of common connection parameters. A more complete list can be found in the [documentation for the Snowflake Connector for Python](#):

- `account`: A Snowflake account identifier.
- `host`: An optional hostname for the Snowflake endpoint. When provided, this is used instead of the default `{account}.snowflakecomputing.com`. Useful for private endpoints or non-standard deployments.
- `user`: A Snowflake username.
- `role`: The role to use for the connection.
- `schema`: The default schema to use for the connection.
- `database`: The default database to use for the connection.
- `warehouse`: The default warehouse to use for the connection.
- `authenticator`: The authentication method to use for the connection.
- `private_key` or `private_key_file`: A path to a PEM-encoded private key for key-pair authentication.
- `private_key_file_pwd`: The passphrase for the private key, if any.
- `token`: The OAuth token to use for authentication.
- `token_file_path`: A path to an OAuth token to use for authentication.
- `password`: The user's Snowflake password.

### Examples

```
# Read the default connection parameters from an existing
# connections.toml file:
conn <- snowflake_connection()

# Read a named connection from an existing connections.toml file:
conn <- snowflake_connection(name = "default")

# Override specific parameters for a connection:
conn <- snowflake_connection(
  schema = "myschema",
```

```

    warehouse = "mywarehouse"
  )

# Pass connection parameters manually, which is useful if there is no
# connections.toml file. For example, to use key-pair authentication:
conn <- snowflake_connection(
  account = "myaccount",
  user = "me",
  private_key = "rsa_key.p8"
)

```

---

snowflake\_credentials *Get credentials for a Snowflake connection*

---

### Description

Get credentials for a Snowflake connection

### Usage

```
snowflake_credentials(params, role = NULL, spcs_endpoint = NULL, ...)
```

### Arguments

params	a list of connection parameters from ‘ <a href="#">snowflake_connection()</a> ’
role	a snowflake entity
spcs_endpoint	a Snowpark Container Services ingress URL, formatted (*-accountname.snowflakecomputing.app)
...	Additional Snowflake connection parameters

### Value

A list of HTTP headers.

### Examples

```

# Obtain authentication headers for accessing Snowflake APIs
snowflake_credentials(
  snowflake_connection()
)

# If the application is in Snowpark Container Services,
# a different collection of headers are returned:
snowflake_credentials(
  snowflake_connection(),
  spcs_endpoint = "https://example-accountname.snowflakecomputing.app"
)

```

# Index

`has_a_default_connection`, [2](#)

`snowflake_connection`, [2](#)

`snowflake_connection()`, [2](#), [4](#)

`snowflake_credentials`, [4](#)