

[Documentation](#)

search

Search

- [rocket launch](#)

[Get started](#)

- [Installation](#)
add
- [Fundamentals](#)
add
- [First steps](#)
add
- [code](#)

[Develop](#)

- [Concepts](#)
add
- [API reference](#)
remove
 - PAGE ELEMENTS

 - [Write and magic](#)
add
 - [Text elements](#)
add
 - [Data elements](#)
add
 - [Chart elements](#)
add
 - [Input widgets](#)
add
 - [Media elements](#)
add
 - [Layouts and containers](#)
add
 - [Chat elements](#)
add
 - [Status elements](#)
add
 - [Third-party components](#)*open in new*
 - APPLICATION LOGIC

 - [Navigation and pages](#)
add
 - [Execution flow](#)
add
 - [Caching and state](#)
add
 - [Connections and secrets](#)
remove
 - SECRETS

- [st.secrets](#)
- [secrets.toml](#)
- CONNECTIONS

-
- [st.connection](#)
 - [SnowflakeConnection](#)
 - [SQLConnection](#)
 - [BaseConnection](#)
 - [SnowparkConnectiondelete](#)

- [Custom components](#)

add

- [Utilities](#)

add

- [Configuration](#)

add

- TOOLS

-
- [App testing](#)

add

- [Command line](#)

add

- [Tutorials](#)

add

- [Quick reference](#)

add

- [web_asset](#)

[Deploy](#)

- [Concepts](#)

add

- [Streamlit Community Cloud](#)

add

- [Snowflake](#)

- [Other platforms](#)

add

- [school](#)

[Knowledge base](#)

- [FAQ](#)

- [Installing dependencies](#)

- [Deployment issues](#)

- [Home/](#)

- [Develop/](#)

- [API reference/](#)

- [Connections and secrets/](#)

- [BaseConnection](#)

star

Tip

This page only contains information on the `st.connections.BaseConnection` class. For a deeper dive into creating and managing data connections within Streamlit apps, read [Connecting to data](#).

st.connections.BaseConnection



The abstract base class that all Streamlit Connections must inherit from.

This base class provides connection authors with a standardized way to hook into the `st.connection()` factory function: connection authors are required to provide an implementation for the abstract method `_connect` in their subclasses.

Additionally, it also provides a few methods/properties designed to make implementation of connections more convenient. See the docstrings for each of the methods of this class for more information

Note

While providing an implementation of `_connect` is technically all that's required to define a valid connection, connections should also provide the user with context-specific ways of interacting with the underlying connection object. For example, the first-party `SQLConnection` provides a `query()` method for reads and a `session` property for more complex operations.

Class description[\[source\]](#)

```
st.connections.BaseConnection(connection_name, **kwargs)
```

Methods

[reset\(\)](#) Reset this connection so that it gets reinitialized the next time it's used.

BaseConnection.reset



Reset this connection so that it gets reinitialized the next time it's used.

This method can be useful when a connection has become stale, an auth token has expired, or in similar scenarios where a broken connection might be fixed by reinitializing it. Note that some connection methods may already use `reset()` in their error handling code.

Function signature[\[source\]](#)

```
BaseConnection.reset()
```

Returns

(None) No description

Example

```
import streamlit as st

conn = st.connection("my_conn")

# Reset the connection before using it if it isn't healthy
# Note: is_healthy() isn't a real method and is just shown for example here.
if not conn.is_healthy():
    conn.reset()
```

Do stuff with conn...

←[Previous: SQLConnection](#)[Next: st.experimental_connection](#)→
forum

Still have questions?

Our [forums](#) are full of helpful information and Streamlit experts.

[Home](#)[Contact Us](#)[Community](#)



© 2025 Snowflake Inc. [Cookie policy](#)

forum Ask AI