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
      remove
      - CALLOUTS

      - st.success
      - st.info
      - st.warning
      - st.error
      - st.exception
      - OTHER

      - st.progress
      - st.spinner
      - st.status

- st.toast
- st.balloons
- st.snow
- Third-party components*open_in_new*
- APPLICATION LOGIC

- Navigation and pages
  add
- Execution flow
  add
- Caching and state
  add
- Connections and secrets
  add
- 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

st.status

Streamlit Version Version 1.41.0

Insert a status container to display output from long-running tasks.

Inserts a container into your app that is typically used to show the status and details of a process or task. The container can hold multiple elements and can be expanded or collapsed by the user similar to `st.expander`. When collapsed, all that is visible is the status icon and label.

The label, state, and expanded state can all be updated by calling `.update()` on the returned object. To add elements to the returned container, you can use `with` notation (preferred) or just call methods directly on the returned object.

By default, `st.status()` initializes in the "running" state. When called using `with` notation, it automatically updates to the "complete" state at the end of the "with" block. See examples below for more details.

Function signature[source]

st.status(label, *, expanded=False, state="running")

Parameters	
label (str)	The initial label of the status container. The label can optionally contain GitHub-flavored Markdown of the following types: Bold, Italics, Strikethroughs, Inline Code, Links, and Images. Images display like icons, with a max height equal to the font height. Unsupported Markdown elements are unwrapped so only their children (text contents) render. Display unsupported elements as literal characters by backslash-escaping them. E.g., `"1\. Not an ordered list"`. See the `body` parameter of `st.markdown` for additional, supported Markdown directives.
expanded (bool)	If True, initializes the status container in "expanded" state. Defaults to False (collapsed).
state ("running", "complete", or "error")	The initial state of the status container which determines which icon is shown: • `running` (default): A spinner icon is shown. • `complete`: A checkmark icon is shown. • `error`: An error icon is shown.

Returns	
(StatusContainer)	A mutable status container that can hold multiple elements. The label, state, and expanded state can be updated after creation via `.update()`.

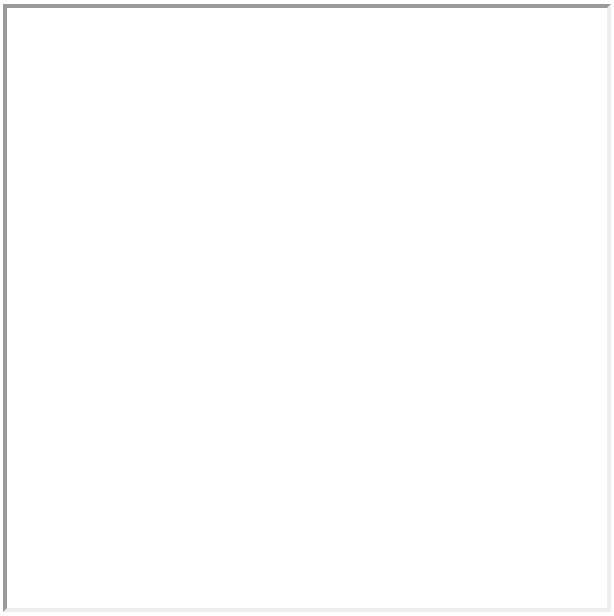
Examples

You can use the `with` notation to insert any element into an status container:

```
import time
import streamlit as st

with st.status("Downloading data..."):
    st.write("Searching for data...")
    time.sleep(2)
    st.write("Found URL.")
    time.sleep(1)
    st.write("Downloading data...")
    time.sleep(1)
```

```
st.button("Rerun")
```



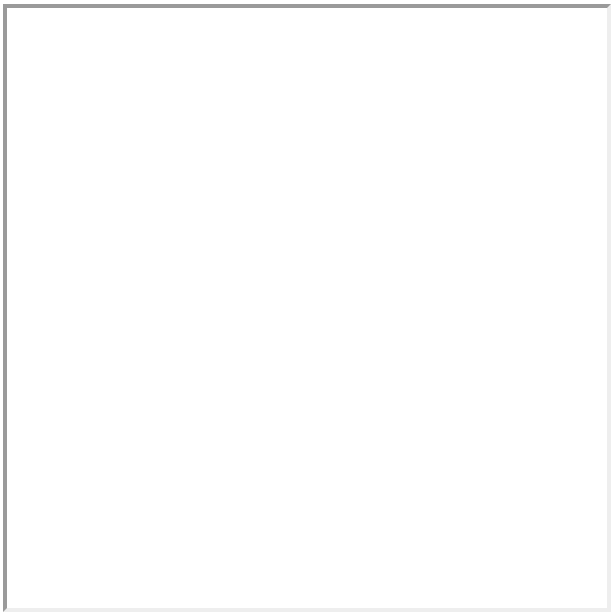
Built with Streamlit 🎈
Fullscreen *open_in_new*

You can also use `.update()` on the container to change the label, state, or expanded state:

```
import time
import streamlit as st

with st.status("Downloading data...", expanded=True) as status:
    st.write("Searching for data...")
    time.sleep(2)
    st.write("Found URL.")
    time.sleep(1)
    st.write("Downloading data...")
    time.sleep(1)
    status.update(
        label="Download complete!", state="complete", expanded=False
    )

st.button("Rerun")
```



Built with Streamlit 🎈
Fullscreen *open_in_new*

StatusContainer.update

Streamlit Version Version 1.41.0

Update the status container.

Only specified arguments are updated. Container contents and unspecified arguments remain unchanged.

Function signature[source]

StatusContainer.update(*, label=None, expanded=None, state=None)

Parameters	
label (str or None)	A new label of the status container. If None, the label is not changed.
expanded (bool or None)	The new expanded state of the status container. If None, the expanded state is not changed.
state ("running", "complete", "error", or None)	The new state of the status container. This mainly changes the icon. If None, the state is not changed.

←Previous: st.spinnerNext: st.toast→

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.
