

[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](#)
add
 - [Custom components](#)

add

- [Utilities](#)
add
 - [Configuration](#)
add
 - TOOLS
-

- [App testing](#)
remove
 - [st.testing.v1.AppTest](#)
 - [Testing element classes](#)
- [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/](#)
- [App testing/](#)
- [st.testing.v1.AppTest](#)

The AppTest class



st.testing.v1.AppTest



Streamlit Version ▼

A simulated Streamlit app to check the correctness of displayed elements and outputs.

An instance of `AppTest` simulates a running Streamlit app. This class provides methods to set up, manipulate, and inspect the app contents via API instead of a browser UI. It can be used to write automated tests of an app in various scenarios. These can then be run using a tool like `pytest`.

`AppTest` can be initialized by one of three class methods:

- [st.testing.v1.AppTest.from_file](#) (recommended)
- [st.testing.v1.AppTest.from_string](#)
- [st.testing.v1.AppTest.from_function](#)

Once initialized, Session State and widget values can be updated and the script can be run. Unlike an actual live-running Streamlit app, you need to call `AppTest.run()` explicitly to re-run the app after changing a widget value. Switching pages also requires an explicit, follow-up call to `AppTest.run()`.

`AppTest` enables developers to build tests on their app as-is, in the familiar python test format, without major refactoring or abstracting out logic to be tested separately from the UI. Tests can run quickly with very low overhead. A typical pattern is to build a suite of tests for an app that ensure consistent functionality as the app evolves, and run the tests locally and/or in a CI environment like Github Actions.

Note

`AppTest` only supports testing a single page of an app per instance. For multipage apps, each page will need to be tested separately. `AppTest` is not yet compatible with multipage apps using `st.navigation` and `st.Page`.

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

`st.testing.v1.AppTest(script_path, *, default_timeout, args=None, kwargs=None)`

st.testing.v1.AppTest(script_path, *, default_timeout, args=None, kwargs=None)

Methods

[get](#)(element_type) Get elements or widgets of the specified type.

[run](#)(*, timeout=None) Run the script from the current state.

[switch_page](#)(page_path) Switch to another page of the app.

Attributes

secrets (dict[str, Any]) Dictionary of secrets to be used the simulated app. Use dict-like syntax to set secret values for the simulated app.

session_state (SafeSessionState) Session State for the simulated app. SafeSessionState object supports read and write operations as usual for Streamlit apps.

query_params (dict[str, Any]) Dictionary of query parameters to be used by the simulated app. Use dict-like syntax to set query_params values for the simulated app.

[button](#) Sequence of all `st.button` and `st.form_submit_button` widgets.

[button_group](#) Sequence of all `st.feedback` widgets.

[caption](#) Sequence of all `st.caption` elements.

[chat_input](#) Sequence of all `st.chat_input` widgets.

[chat_message](#) Sequence of all `st.chat_message` elements.

[checkbox](#) Sequence of all `st.checkbox` widgets.

[code](#) Sequence of all `st.code` elements.

[color_picker](#) Sequence of all `st.color_picker` widgets.

[columns](#) Sequence of all columns within `st.columns` elements.

[dataframe](#) Sequence of all `st.dataframe` elements.

st.testing.v1.AppTest(script_path, *, default_timeout, args=None, kwargs=None)

date_input	Sequence of all <code>st.date_input</code> widgets.
divider	Sequence of all <code>st.divider</code> elements.
error	Sequence of all <code>st.error</code> elements.
exception	Sequence of all <code>st.exception</code> elements.
expander	Sequence of all <code>st.expander</code> elements.
header	Sequence of all <code>st.header</code> elements.
info	Sequence of all <code>st.info</code> elements.
json	Sequence of all <code>st.json</code> elements.
latex	Sequence of all <code>st.latex</code> elements.
main	Sequence of elements within the main body of the app.
markdown	Sequence of all <code>st.markdown</code> elements.
metric	Sequence of all <code>st.metric</code> elements.
multiselect	Sequence of all <code>st.multiselect</code> widgets.
number_input	Sequence of all <code>st.number_input</code> widgets.
radio	Sequence of all <code>st.radio</code> widgets.
select_slider	Sequence of all <code>st.select_slider</code> widgets.
selectbox	Sequence of all <code>st.selectbox</code> widgets.
sidebar	Sequence of all elements within <code>st.sidebar</code> .

st.testing.v1.AppTest(script_path, *, default_timeout, args=None, kwargs=None)


slider	Sequence of all <code>st.slider</code> widgets.
status	Sequence of all <code>st.status</code> elements.
subheader	Sequence of all <code>st.subheader</code> elements.
success	Sequence of all <code>st.success</code> elements.
table	Sequence of all <code>st.table</code> elements.
tabs	Sequence of all tabs within <code>st.tabs</code> elements.
text	Sequence of all <code>st.text</code> elements.
text_area	Sequence of all <code>st.text_area</code> widgets.
text_input	Sequence of all <code>st.text_input</code> widgets.
time_input	Sequence of all <code>st.time_input</code> widgets.
title	Sequence of all <code>st.title</code> elements.
toast	Sequence of all <code>st.toast</code> elements.
toggle	Sequence of all <code>st.toggle</code> widgets.
warning	Sequence of all <code>st.warning</code> elements.

Initialize a simulated app using AppTest



AppTest.from_file



Streamlit Version 

Create an instance of `AppTest` to simulate an app page defined within a file.

This option is most convenient for CI workflows and testing of published apps. The script must be executable on its own and so must contain all necessary imports.

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

```
AppTest.from_file(cls, script_path, *, default_timeout=3)
```

Parameters

script_path (str | Path) Path to a script file. The path should be absolute or relative to the file calling .from_file.

default_timeout (float) Default time in seconds before a script run is timed out. Can be overridden for individual .run() calls.

Returns

(AppTest) A simulated Streamlit app for testing. The simulated app can be executed via .run().

AppTest.from_string



Streamlit Version

Create an instance of AppTest to simulate an app page defined within a string.

This is useful for testing short scripts that fit comfortably as an inline string in the test itself, without having to create a separate file for it. The script must be executable on its own and so must contain all necessary imports.

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

```
AppTest.from_string(cls, script, *, default_timeout=3)
```

Parameters

script (str) The string contents of the script to be run.

default_timeout (float) Default time in seconds before a script run is timed out. Can be overridden for individual .run() calls.

Returns

(AppTest) A simulated Streamlit app for testing. The simulated app can be executed via .run().

AppTest.from_function



Streamlit Version

Create an instance of AppTest to simulate an app page defined within a function.

This is similar to `AppTest.from_string()`, but more convenient to write with IDE assistance. The script must be executable on its own and so must contain all necessary imports.

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

`AppTest.from_function(cls, script, *, default_timeout=3, args=None, kwargs=None)`

Parameters

script (Callable)	A function whose body will be used as a script. Must be runnable in isolation, so it must include any necessary imports.
default_timeout (float)	Default time in seconds before a script run is timed out. Can be overridden for individual <code>.run()</code> calls.
args (tuple)	An optional tuple of args to pass to the script function.
kwargs (dict)	An optional dict of kwargs to pass to the script function.

Returns

(AppTest)	A simulated Streamlit app for testing. The simulated app can be executed via <code>.run()</code> .
-----------	--

Run an AppTest script



AppTest.run



Streamlit Version

Run the script from the current state.

This is equivalent to manually rerunning the app or the rerun that occurs upon user interaction. `AppTest.run()` must be manually called after updating a widget value or switching pages as script reruns do not occur automatically as they do for live-running Streamlit apps.

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

`AppTest.run(*, timeout=None)`

Parameters

timeout (float or None)	The maximum number of seconds to run the script. If <code>timeout</code> is <code>None</code> (default), Streamlit uses the default timeout set for the instance of <code>AppTest</code> .
-------------------------	--

Returns

(AppTest)	<code>self</code>
-----------	-------------------

AppTest.switch_page



Streamlit Version

Switch to another page of the app.

This method does not automatically rerun the app. Use a follow-up call to `AppTest.run()` to obtain the elements on the selected page.

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

AppTest.switch_page(page_path)

Parameters

page_path (str)	Path of the page to switch to. The path must be relative to the main script's location (e.g. "pages/my_page.py").
--------------------	---

Returns

(AppTest)	self
-----------	------

Get AppTest script elements



The main value of `AppTest` is providing an API to programmatically inspect and interact with the elements and widgets produced by a running Streamlit app. Using the `AppTest.<element type>` properties or `AppTest.get()` method returns a collection of all the elements or widgets of the specified type that would have been displayed by running the app.

Note that you can also retrieve elements within a specific container in the same way - first retrieve the container, then retrieve the elements just in that container.

AppTest.get



Streamlit Version

Get elements or widgets of the specified type.

This method returns the collection of all elements or widgets of the specified type on the current page. Retrieve a specific element by using its index (order on page) or key lookup.

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

AppTest.get(element_type)

Parameters

element_type
(str) An element attribute of `AppTest`. For example, "button", "caption", or "chat_input".

Returns

(Sequence of Elements) Sequence of elements of the given type. Individual elements can be accessed from a Sequence by index (order on the page). When getting and `element_type` that is a widget, individual widgets can be accessed by key. For example, `at.get("text")[0]` for the first `st.text` element or `at.get("slider")` (`key="my_key"`) for the `st.slider` widget with a given key.

AppTest.button



Streamlit Version ▼

Sequence of all `st.button` and `st.form_submit_button` widgets.

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

AppTest.button

Returns

(WidgetList of Button) Sequence of all `st.button` and `st.form_submit_button` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.button[0]` for the first widget or `at.button(key="my_key")` for a widget with a given key.

AppTest.caption



Streamlit Version ▼

Sequence of all `st.caption` elements.

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

AppTest.caption

Returns

(ElementList of Caption) Sequence of all `st.caption` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.caption[0]` for the first element. Caption is an extension of the Element class.

AppTest.chat_input



Streamlit Version

Sequence of all `st.chat_input` widgets.

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

AppTest.chat_input

Returns

(WidgetList of ChatInput) Sequence of all `st.chat_input` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.chat_input[0]` for the first widget or `at.chat_input(key="my_key")` for a widget with a given key.

AppTest.chat_message



Streamlit Version

Sequence of all `st.chat_message` elements.

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

AppTest.chat_message

Returns

(Sequence of ChatMessage) Sequence of all `st.chat_message` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.chat_message[0]` for the first element. ChatMessage is an extension of the Block class.

AppTest.checkbox



Streamlit Version

Sequence of all `st.checkbox` widgets.

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

AppTest.checkbox

Returns

(WidgetList of Checkbox) Sequence of all `st.checkbox` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.checkbox[0]` for the first widget or `at.checkbox(key="my_key")` for a widget with a given key.

AppTest.code



Streamlit Version

Version 1.41.0 ▼

Sequence of all `st.code` elements.

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

AppTest.code

Returns

(ElementList of Code) Sequence of all `st.code` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.code[0]` for the first element. Code is an extension of the Element class.

AppTest.color_picker



Streamlit Version

Version 1.41.0 ▼

Sequence of all `st.color_picker` widgets.

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

AppTest.color_picker

Returns

(WidgetList of ColorPicker) Sequence of all `st.color_picker` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.color_picker[0]` for the first widget or `at.color_picker(key="my_key")` for a widget with a given key.

AppTest.columns



Streamlit Version

Version 1.41.0 ▼

Sequence of all columns within `st.columns` elements.

Each column within a single `st.columns` will be returned as a separate Column in the Sequence.

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

AppTest.columns

Returns

(Sequence of Column) Sequence of all columns within `st.columns` elements. Individual columns can be accessed from an `ElementList` by index (order on the page). For example, `at.columns[0]` for the first column. Column is an extension of the `Block` class.

AppTest.dataframe



Streamlit Version

Sequence of all `st.dataframe` elements.

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

AppTest.dataframe

Returns

(ElementList of Dataframe) Sequence of all `st.dataframe` elements. Individual elements can be accessed from an `ElementList` by index (order on the page). For example, `at.dataframe[0]` for the first element. Dataframe is an extension of the `Element` class.

AppTest.date_input



Streamlit Version

Sequence of all `st.date_input` widgets.

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

AppTest.date_input

Returns

(WidgetList of DateInput) Sequence of all `st.date_input` widgets. Individual widgets can be accessed from a `WidgetList` by index (order on the page) or key. For example, `at.date_input[0]` for the first widget or `at.date_input(key="my_key")` for a widget with a given key.

AppTest.divider



Streamlit Version

Sequence of all `st.divider` elements.

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

AppTest.divider

Returns

(ElementList of Divider) Sequence of all `st.divider` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.divider[0]` for the first element. Divider is an extension of the Element class.

AppTest.error



Streamlit Version

Sequence of all `st.error` elements.

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

AppTest.error

Returns

(ElementList of Error) Sequence of all `st.error` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.error[0]` for the first element. Error is an extension of the Element class.

AppTest.exception



Streamlit Version

Sequence of all `st.exception` elements.

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

AppTest.exception

Returns

(ElementList of Exception) Sequence of all `st.exception` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.exception[0]` for the first element. Exception is an extension of the Element class.

AppTest.expander



Streamlit Version

Sequence of all `st.expander` elements.

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

AppTest.expander

Returns

(Sequence of
Expandable)

Sequence of all `st.expander` elements. Individual elements can be accessed from a Sequence by index (order on the page). For example, `at.expander[0]` for the first element. Expandable is an extension of the Block class.

AppTest.header



Streamlit Version ▼

Sequence of all `st.header` elements.

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

AppTest.header

Returns

(ElementList of
Header)

Sequence of all `st.header` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.header[0]` for the first element. Header is an extension of the Element class.

AppTest.info



Streamlit Version ▼

Sequence of all `st.info` elements.

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

AppTest.info

Returns

(ElementList of
Info)

Sequence of all `st.info` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.info[0]` for the first element. Info is an extension of the Element class.

AppTest.json



Streamlit Version ▼

Sequence of all `st.json` elements.

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

AppTest.json

Returns

(ElementList of Json) Sequence of all `st.json` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.json[0]` for the first element. Json is an extension of the Element class.

AppTest.latex



Streamlit Version ▼

Sequence of all `st.latex` elements.

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

AppTest.latex

Returns

(ElementList of Latex) Sequence of all `st.latex` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.latex[0]` for the first element. Latex is an extension of the Element class.

AppTest.main



Streamlit Version ▼

Sequence of elements within the main body of the app.

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

AppTest.main

Returns

(Block) A container of elements. Block can be queried for elements in the same manner as `AppTest`. For example, `Block.checkbox` will return all `st.checkbox` within the associated container.

AppTest.markdown



Streamlit Version ▼

Sequence of all `st.markdown` elements.

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

AppTest.markdown

Returns

(ElementList of Markdown) Sequence of all `st.markdown` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.markdown[0]` for the first element. Markdown is an extension of the Element class.

AppTest.metric



Streamlit Version ▼

Sequence of all `st.metric` elements.

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

AppTest.metric

Returns

(ElementList of Metric) Sequence of all `st.metric` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.metric[0]` for the first element. Metric is an extension of the Element class.

AppTest.multiselect



Streamlit Version ▼

Sequence of all `st.multiselect` widgets.

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

AppTest.multiselect

Returns

(WidgetList of Multiselect) Sequence of all `st.multiselect` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.multiselect[0]` for the first widget or `at.multiselect(key="my_key")` for a widget with a given key.

AppTest.number_input



Streamlit Version ▼

Sequence of all `st.number_input` widgets.

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

AppTest.number_input

Returns

(WidgetList of
NumberInput)

Sequence of all `st.number_input` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.number_input[0]` for the first widget or `at.number_input(key="my_key")` for a widget with a given key.

AppTest.radio



Streamlit Version

Sequence of all `st.radio` widgets.

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

AppTest.radio

Returns

(WidgetList of
Radio)

Sequence of all `st.radio` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.radio[0]` for the first widget or `at.radio(key="my_key")` for a widget with a given key.

AppTest.select_slider



Streamlit Version

Sequence of all `st.select_slider` widgets.

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

AppTest.select_slider

Returns

(WidgetList of
SelectSlider)

Sequence of all `st.select_slider` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.select_slider[0]` for the first widget or `at.select_slider(key="my_key")` for a widget with a given key.

AppTest.selectbox



Streamlit Version

Sequence of all `st.selectbox` widgets.

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

AppTest.selectbox

Returns

(WidgetList of Selectbox) Sequence of all `st.selectbox` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.selectbox[0]` for the first widget or `at.selectbox(key="my_key")` for a widget with a given key.

AppTest.sidebar



Streamlit Version ▼

Sequence of all elements within `st.sidebar`.

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

AppTest.sidebar

Returns

(Block) A container of elements. Block can be queried for elements in the same manner as `AppTest`. For example, `Block.checkbox` will return all `st.checkbox` within the associated container.

AppTest.slider



Streamlit Version ▼

Sequence of all `st.slider` widgets.

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

AppTest.slider

Returns

(WidgetList of Slider) Sequence of all `st.slider` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.slider[0]` for the first widget or `at.slider(key="my_key")` for a widget with a given key.

AppTest.subheader



Streamlit Version ▼

Sequence of all `st.subheader` elements.

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

AppTest.subheader

Returns

(ElementList of Subheader) Sequence of all `st.subheader` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.subheader[0]` for the first element. Subheader is an extension of the Element class.

AppTest.success



Streamlit Version ▼

Sequence of all `st.success` elements.

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

AppTest.success

Returns

(ElementList of Success) Sequence of all `st.success` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.success[0]` for the first element. Success is an extension of the Element class.

AppTest.status



Streamlit Version ▼

Sequence of all `st.status` elements.

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

AppTest.status

Returns

(Sequence of Status) Sequence of all `st.status` elements. Individual elements can be accessed from a Sequence by index (order on the page). For example, `at.status[0]` for the first element. Status is an extension of the Block class.

AppTest.table



Streamlit Version ▼

Sequence of all `st.table` elements.

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

AppTest.table

Returns

(ElementList of Table) Sequence of all `st.table` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.table[0]` for the first element. Table is an extension of the Element class.

AppTest.tabs



Streamlit Version

Sequence of all tabs within `st.tabs` elements.

Each tab within a single `st.tabs` will be returned as a separate Tab in the Sequence. Additionally, the tab labels are forwarded to each Tab element as a property. For example, `st.tabs("A", "B")` will yield two Tab objects, with `Tab.label` returning "A" and "B", respectively.

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

AppTest.tabs

Returns

(Sequence of Tab) Sequence of all tabs within `st.tabs` elements. Individual tabs can be accessed from an ElementList by index (order on the page). For example, `at.tabs[0]` for the first tab. Tab is an extension of the Block class.

AppTest.text



Streamlit Version

Sequence of all `st.text` elements.

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

AppTest.text

Returns

(ElementList of Text) Sequence of all `st.text` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, `at.text[0]` for the first element. Text is an extension of the Element class.

AppTest.text_area



Sequence of all `st.text_area` widgets.

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

`AppTest.text_area`

Returns

(WidgetList of
TextArea) Sequence of all `st.text_area` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.text_area[0]` for the first widget or `at.text_area(key="my_key")` for a widget with a given key.

AppTest.text_input



Sequence of all `st.text_input` widgets.

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

`AppTest.text_input`

Returns

(WidgetList of
TextInput) Sequence of all `st.text_input` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.text_input[0]` for the first widget or `at.text_input(key="my_key")` for a widget with a given key.

AppTest.time_input



Sequence of all `st.time_input` widgets.

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

`AppTest.time_input`

Returns

(WidgetList of
TimeInput) Sequence of all `st.time_input` widgets. Individual widgets can be accessed from a WidgetList by index (order on the page) or key. For example, `at.time_input[0]` for the first widget or `at.time_input(key="my_key")` for a widget with a given key.

AppTest.title



Sequence of all `st.title` elements.

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

AppTest.title

Returns

(ElementList of Title)

Sequence of all `st.title` elements. Individual elements can be accessed from an `ElementList` by index (order on the page). For example, `at.title[0]` for the first element. `Title` is an extension of the `Element` class.

AppTest.toast



Sequence of all `st.toast` elements.

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

AppTest.toast

Returns

(ElementList of Toast)

Sequence of all `st.toast` elements. Individual elements can be accessed from an `ElementList` by index (order on the page). For example, `at.toast[0]` for the first element. `Toast` is an extension of the `Element` class.

AppTest.toggle



Sequence of all `st.toggle` widgets.

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

AppTest.toggle

Returns

(WidgetList of Toggle)

Sequence of all `st.toggle` widgets. Individual widgets can be accessed from a `WidgetList` by index (order on the page) or key. For example, `at.toggle[0]` for the first widget or `at.toggle(key="my_key")` for a widget with a given key.

AppTest.warning



Sequence of all `st.warning` elements.

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

AppTest.warning

Returns

(ElementList of Warning)

Sequence of all `st.warning` elements. Individual elements can be accessed from an ElementList by index (order on the page). For example, at `.warning[0]` for the first element. Warning is an extension of the Element class.

←[Previous: App testing](#)[Next: Testing element classes](#)→

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