

Accessibility

The Accessibility class provides methods for inspecting Chromium's accessibility tree. The accessibility tree is used by assistive technology such as [screen readers](#) or [switches](#).

Accessibility is a very platform-specific thing. On different platforms, there are different screen readers that might have wildly different output.

Rendering engines of Chromium, Firefox and WebKit have a concept of "accessibility tree", which is then translated into different platform-specific APIs. Accessibility namespace gives access to this Accessibility Tree.

Most of the accessibility tree gets filtered out when converting from internal browser AX Tree to Platform-specific AX-Tree or by assistive technologies themselves. By default, Playwright tries to approximate this filtering, exposing only the "interesting" nodes of the tree.

Deprecated

snapshot

Added in: v1.8

DEPRECATED

This method is deprecated. Please use other libraries such as [Axe](#) if you need to test page accessibility. See our [Node.js guide](#) for integration with Axe.

Captures the current state of the accessibility tree. The returned object represents the root accessible node of the page.

NOTE

The Chromium accessibility tree contains nodes that go unused on most platforms and by most screen readers. Playwright will discard them as well for an easier to process tree, unless `interesting_only` is set to `false`.

Usage

An example of dumping the entire accessibility tree:

Sync **Async**

```
snapshot = page.accessibility.snapshot()
print(snapshot)
```

An example of logging the focused node's name:

Sync **Async**

```
def find_focused_node(node):
    if node.get("focused"):
        return node
    for child in (node.get("children") or []):
        found_node = find_focused_node(child)
        if found_node:
            return found_node
    return None

snapshot = page.accessibility.snapshot()
node = find_focused_node(snapshot)
if node:
    print(node["name"])
```

Arguments

- `interesting_only` **bool** (*optional*)

Prune uninteresting nodes from the tree. Defaults to `true`.

- `root` **ElementHandle** (*optional*)

The root DOM element for the snapshot. Defaults to the whole page.

Returns

- **NoneType|Dict**

- `role` `str`

The `role`.

- `name` `str`

A human readable name for the node.

- `value` `str|float`

The current value of the node, if applicable.

- `description` `str`

An additional human readable description of the node, if applicable.

- `keyshortcuts` `str`

Keyboard shortcuts associated with this node, if applicable.

- `roledescription` `str`

A human readable alternative to the role, if applicable.

- `valuetext` `str`

A description of the current value, if applicable.

- `disabled` `bool`

Whether the node is disabled, if applicable.

- `expanded` `bool`

Whether the node is expanded or collapsed, if applicable.

- `focused` `bool`

Whether the node is focused, if applicable.

- `modal` `bool`

Whether the node is `modal`, if applicable.

- `multiline` `bool`

Whether the node text input supports multiline, if applicable.

- `multiselectable` `bool`

Whether more than one child can be selected, if applicable.

- `readonly` `bool`

Whether the node is read only, if applicable.

- `required` `bool`

Whether the node is required, if applicable.

- `selected` `bool`

Whether the node is selected in its parent node, if applicable.

- `checked` `bool`|"mixed"

Whether the checkbox is checked, or "mixed", if applicable.

- `pressed` `bool`|"mixed"

Whether the toggle button is checked, or "mixed", if applicable.

- `level` `int`

The level of a heading, if applicable.

- `valuemin` `float`

The minimum value in a node, if applicable.

- `valuemax` `float`

The maximum value in a node, if applicable.

- `autocomplete` `str`

What kind of autocomplete is supported by a control, if applicable.

- `haspopup` `str`

What kind of popup is currently being shown for a node, if applicable.

- `invalid` `str`

Whether and in what way this node's value is invalid, if applicable.

- `orientation` `str`

Whether the node is oriented horizontally or vertically, if applicable.

- `children` `List[Dict]`

Child nodes, if any, if applicable.