Playwright Advanced Features

Target Locators
Alert

Testleaf
Always Ahead

The Golden Circle

What

What is an Alert?

An alert is a message or notification from a web application that requires user interaction.

Why

Why do we need to handle it?

Alerts block interaction with other elements on the page until they are addressed.

How

How to handle alert?

Playwright provides an event-driven approach to handling alerts.

More about Alert..

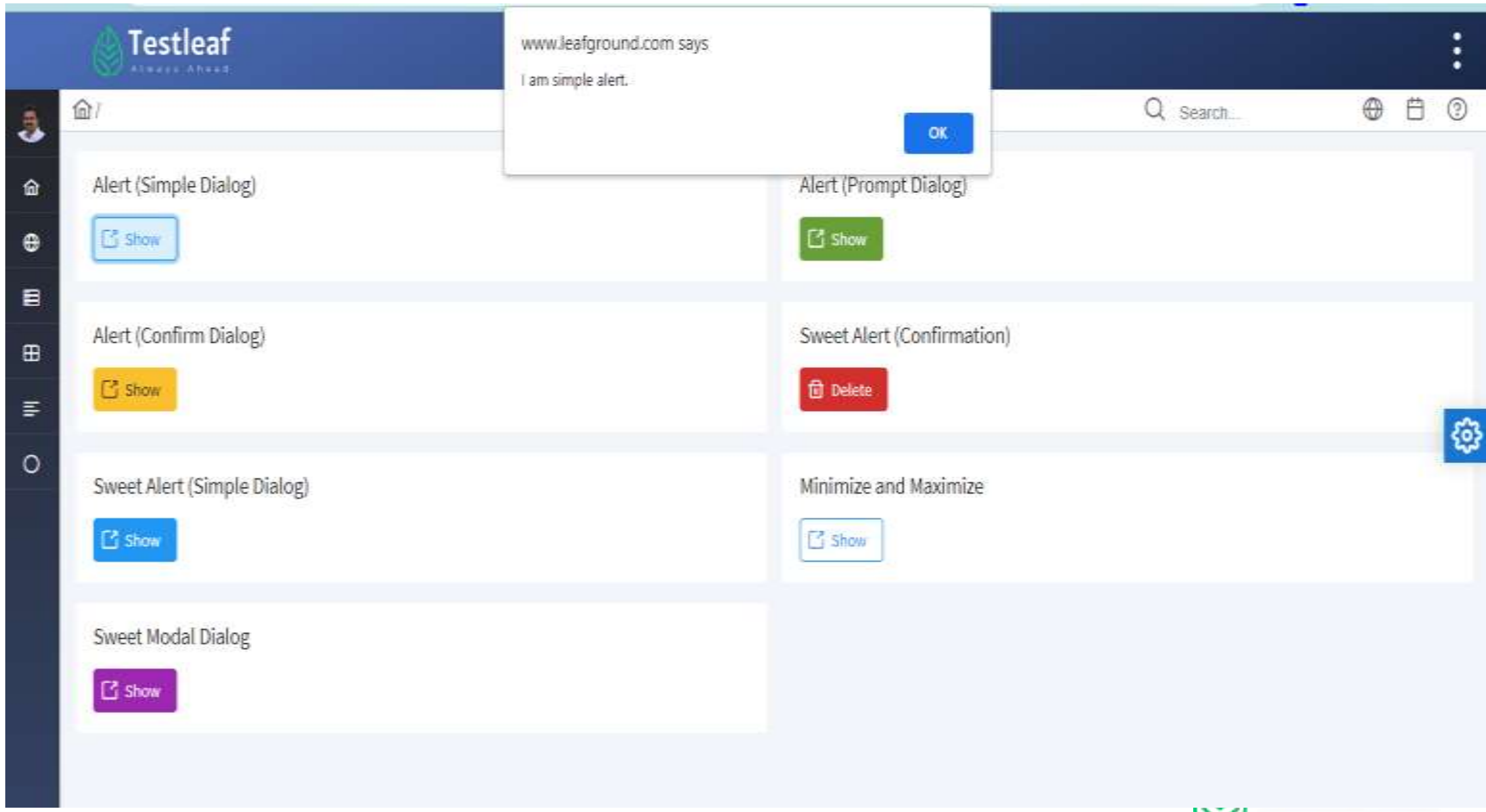
Alert is an application pop up/ dialog box

Alerts are JavaScript function

To confirm the pop up as an alert, check the element that

-is not inspectable ..

-cannot be ignored



How to Handle it ?

To handle alert

- ✓ In Playwright, alert handling is event-driven.
- ✓ Instead of manually switching focus, we use the page.on('dialog', callback) event listener to listen for alert dialogs and handle them dynamically.

Syntax:

```
await page.goto(`https://leafground.com/alert.xhtml`);

    //Event Listener
    page.on('dialog', async (dialog:any) => {
        dialog.accept();
      })

    //Confirm Dialog
    const confirmAlert = await page.locator(".card").filter({hasText:"Confirm Dialog"});
    await confirmAlert.locator("//span[text()='Show']").click();
    await page.waitForTimeout(5000);
```

Testleaf
Always Ahead

Types of Alert

Modal Alert

- ❑ Simple Alert
- ❑ Confirmation Alert
- ❑ Prompt Alert

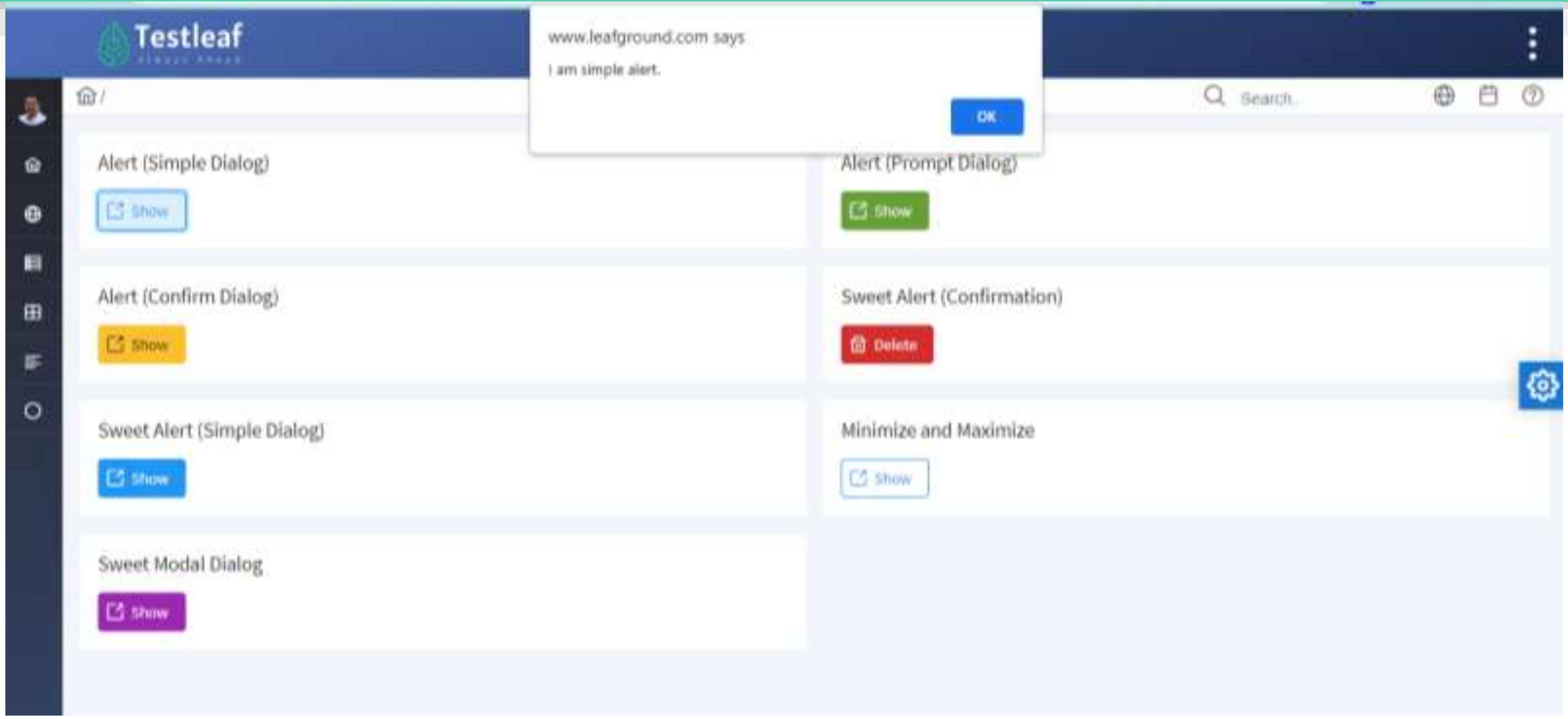
Non –Modal Alert

- ❑ Sweet Alert

Simple Alert

-Displays a message and requires acknowledgment.

-Allows fetching text from the alert box.


www.leafground.com says

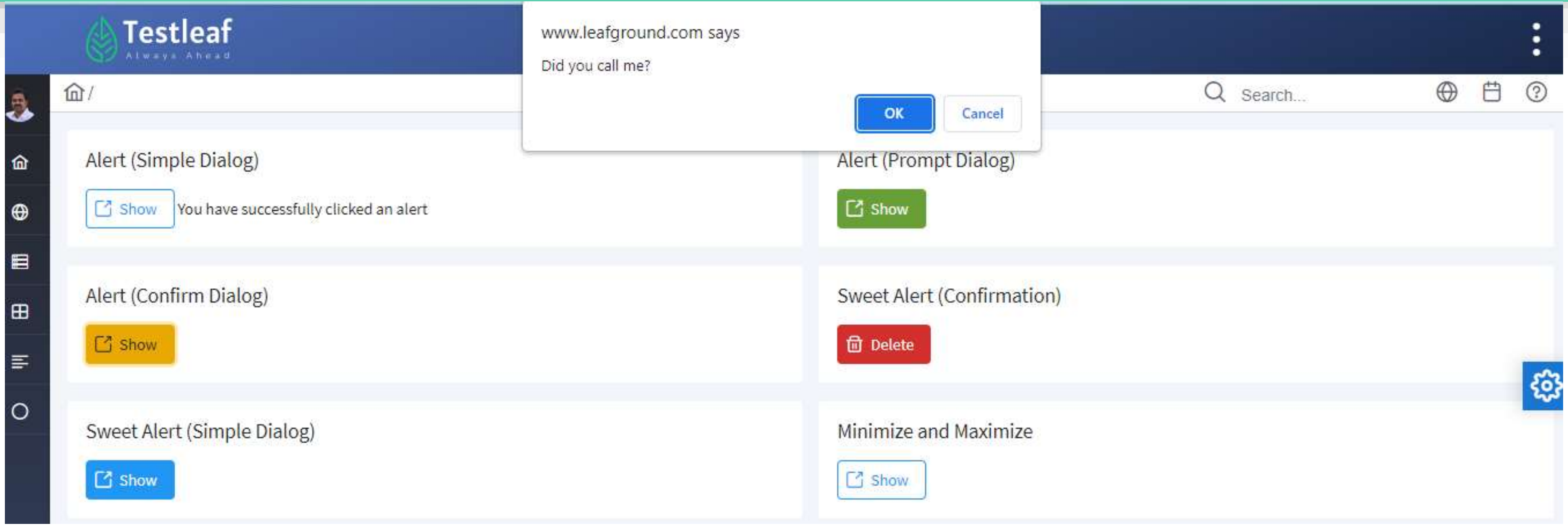
I am simple alert.

```
//Simple Alerts

page.on('dialog', async dialog => {
    console.log(`Alert message: ${dialog.message()}`);
    await dialog.accept();
});
await page.click('#simpleAlertButton');
```

Confirmation Alert

-Offers options to accept or dismiss the dialog.

-Allows fetching text from the alert box.



www.leafground.com says

Did you call me?

You have successfully clicked an alert

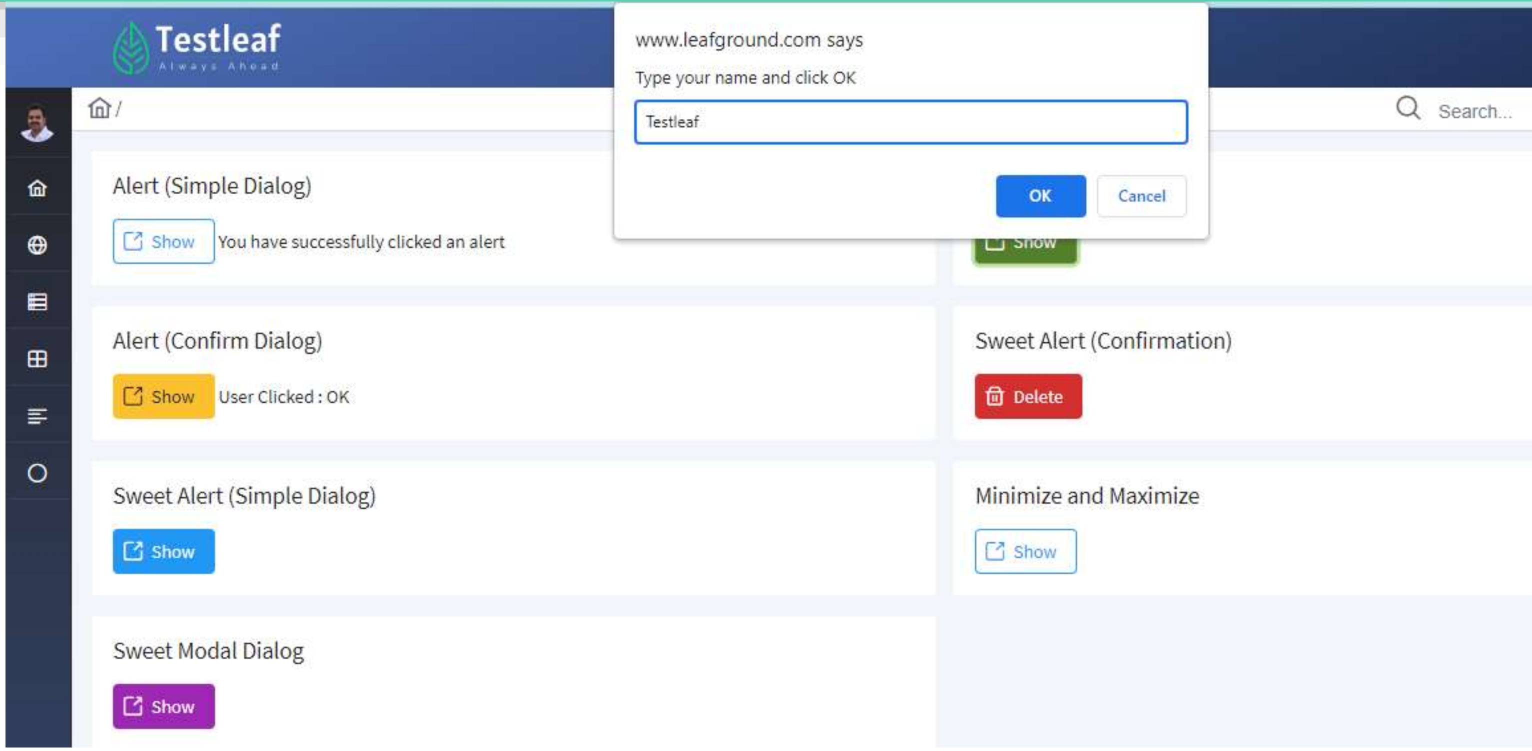
```
//Handling Confirmation Alerts

page.on('dialog', async dialog => {
    console.log(`Confirmation message: ${dialog.message()}`);
    await dialog.dismiss(); // or dialog.accept()
});
await page.click('#confirmAlertButton');
```

Prompt Alert

-Offers options to accept or dismiss the dialog.

-Allows to get the text from the dialog box

-Allows entering user input into the alert box.


www.leafground.com says

Type your name and click OK

Testleaf

Alert (Simple Dialog)

You have successfully clicked an alert

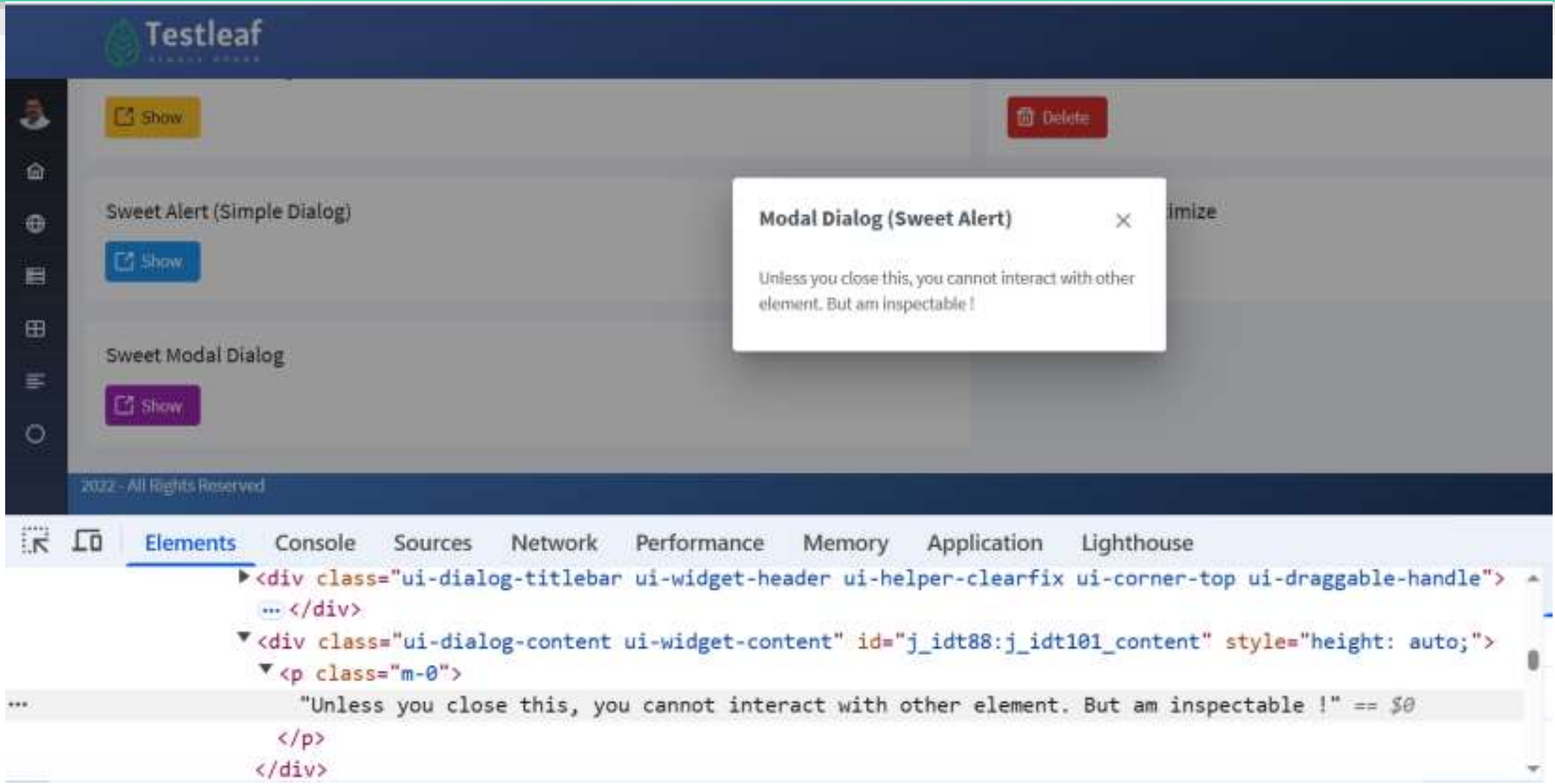
Alert (Confirm Dialog)

User Clicked : OK

```
//Handling Prompt Alerts

page.on('dialog', async dialog => {
    console.log(`Prompt message: ${dialog.message()}`);
    await dialog.accept(‘Testleaf');
});
await page.click('#promptAlertButton');
```

Sweet Alert

- Inspectable and interacts like normal elements.



Modal Dialog (Sweet Alert)

Unless you close this, you cannot interact with other element. But am inspectable !

```
▶<div class="ui-dialog-titlebar ui-widget-header ui-helper-clearfix ui-corner-top ui-draggable-handle">
 ⋯ </div>
▼<div class="ui-dialog-content ui-widget-content" id="j_idt88:j_idt101_content" style="height: auto;">
  ▼<p class="m-0">
     "Unless you close this, you cannot interact with other element. But am inspectable !" == $0
   </p>
 </div>
```

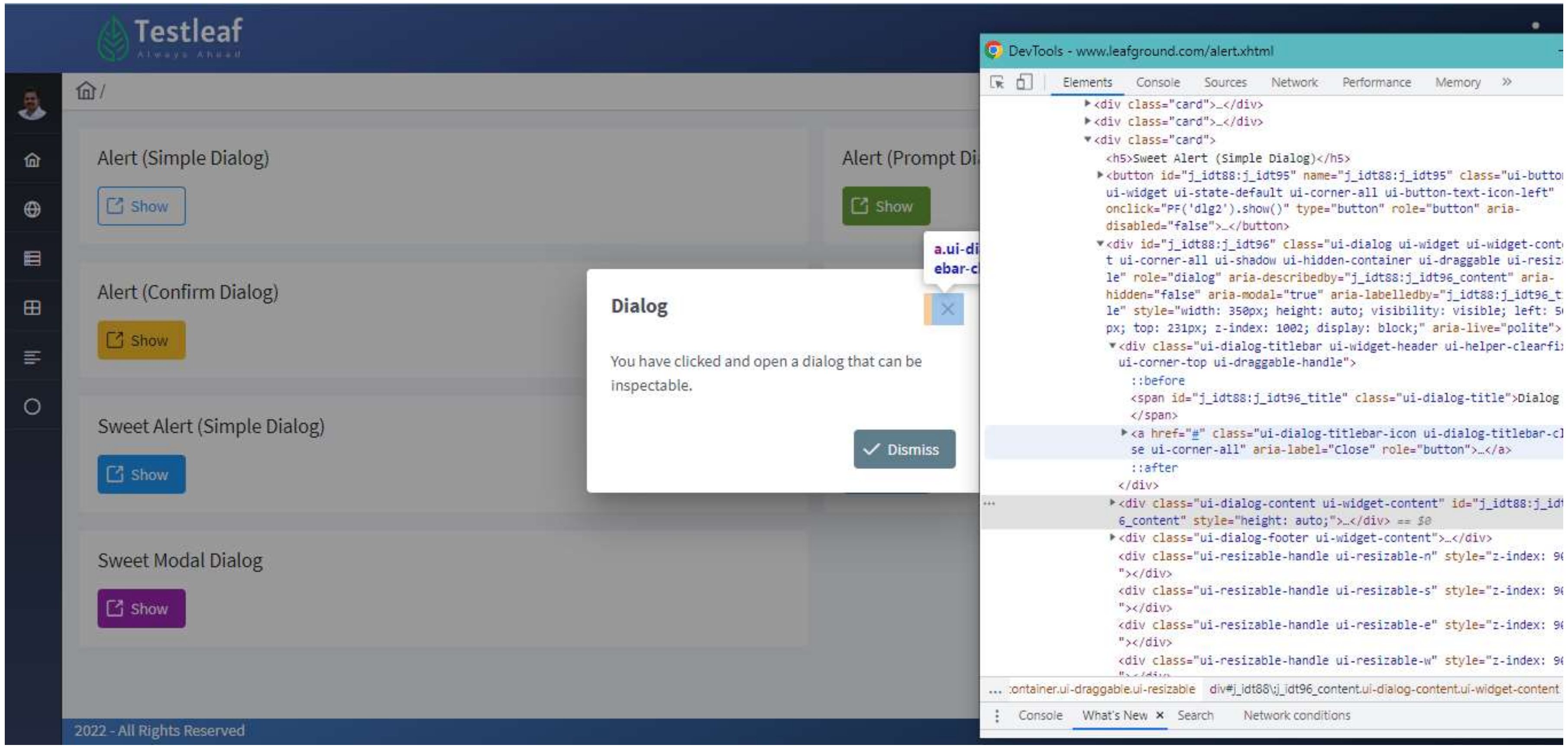
```
//Handling Sweet Alerts

await page.click('#sweetAlertButton');
await page.waitForSelector('.swal2-popup');
await page.click('.swal2-confirm');
```

Sweet Alert

- handled as like normal web element

- it is inspectable

- cannot be ignored



You have clicked and open a dialog that can be inspectable.

```
▶<div class="card">…</div>
▶<div class="card">…</div>
▼<div class="card">
  <h5>Sweet Alert (Simple Dialog)</h5>
 ▶<button id="j_idt88:j_idt95" name="j_idt88:j_idt95" class="ui-butto
  ui-widget ui-state-default ui-corner-all ui-button-text-icon-left"
  onclick="PF('dlg2').show()" type="button" role="button" aria-
  disabled="false">…</button>
 ▼<div id="j_idt88:j_idt96" class="ui-dialog ui-widget ui-widget-cont
  t ui-corner-all ui-shadow ui-hidden-container ui-draggable ui-resiz
  le" role="dialog" aria-describedby="j_idt88:j_idt96_content" aria-
  hidden="false" aria-modal="true" aria-labelledby="j_idt88:j_idt96_t
  le" style="width: 350px; height: auto; visibility: visible; left: 5
  px; top: 231px; z-index: 1002; display: block;" aria-live="polite">
  ▼<div class="ui-dialog-titlebar ui-widget-header ui-helper-clearfi
   ui-corner-top ui-draggable-handle">
     ::before
     <span id="j_idt88:j_idt96_title" class="ui-dialog-title">Dialog
     </span>
    ▶<a href="#" class="ui-dialog-titlebar-icon ui-dialog-titlebar-cl
     se ui-corner-all" aria-label="Close" role="button">…</a>
     ::after
   </div>
...▶<div class="ui-dialog-content ui-widget-content" id="j_idt88:j_idt
   6_content" style="height: auto;">…</div> == $0
  ▶<div class="ui-dialog-footer ui-widget-content">…</div>
   <div class="ui-resizable-handle ui-resizable-n" style="z-index: 9
   "></div>
   <div class="ui-resizable-handle ui-resizable-s" style="z-index: 9
   "></div>
   <div class="ui-resizable-handle ui-resizable-e" style="z-index: 9
   "></div>
   <div class="ui-resizable-handle ui-resizable-w" style="z-index: 9
```

… :ontainer.ui-draggable.ui-resizable div#j_idt88\:j_idt96_content.ui-dialog-content.ui-widget-content

Actions allowed in Playwright Alert Handling

```
dialog.accept();
    -to accept the alert

dialog.dismiss();
    -to cancel the alert

dialog.message();
   -Retrieves the text from the alert box.

dialog.accept('Your Text');
   -Enters text into a prompt alert before accepting it.
```

Testleaf
Always Ahead

Actions allowed in Playwright Alert Handling

```
dialog.type();
    -to get the type of the dialog.
```

Classroom

Launch the url:
https://www.leafground.com/alert.xhtml
Cick on prompt dialog
Switch to alert
Use fill() to type in the alert box
Dismiss the alert

Summary

- Alerts -> Pop up from Web application
- Cannot be inspected
- 3 types – Simple, Prompt and Confirmation Alert
- Sweet alert can be inspectable.