

# Handling Different Types of Alerts in Playwright

In Selenium, we refer to alerts, but in Playwright, these are known as **dialogs**. Playwright supports three types of dialogs:

1. **Alert**
2. **Confirm**
3. **Prompt**

## Default Behaviour of Dialogs

By default, dialogs are automatically dismissed in Playwright. This means when a dialog appears, Playwright will automatically handle it. However, for validation purposes, we still need to **handle dialogs manually** when performing tests.

To handle dialogs, you must register a **dialog handler** before performing the action that triggers the dialog. This allows you to either accept, dismiss, or perform any validation on the dialog.

## How Dialogs Work in Playwright

When performing actions like clicking a button, a dialog might appear. Before clicking the button, you need to register the dialog handler to handle the dialog after it's triggered. This handler will allow you to accept, dismiss, or validate the dialog's contents.

---

## Types of Dialogs

1. **Alert Dialog**
  - Contains just an **OK** button.
2. **Confirm Dialog**
  - Contains **OK** and **Cancel** buttons.
3. **Prompt Dialog**
  - Contains an **input box** where we can enter text.

---

## 1. Alert Dialog with an OK Button

Although Playwright handles dialogs by default, you might want to manage the dialog manually. Here's how you can enable the dialog handler.

## How to Enable Dialog Window Handler

Before performing an action (like clicking a button), use the `page.on()` function to register the dialog handler. This function takes two parameters:

1. **Dialog Type:** Specify the type of dialog (alert, confirm, prompt).
2. **Anonymous Function:** Inside this function, use the `async` keyword to return a promise, and pass the `dialog` object.

You can then write assertions or handle the dialog inside this function.

### Example Code: Handling an Alert Dialog

```
import { test, expect } from '@playwright/test';

test('Dialog with OK button', async ({ page }) => {
  page.on('dialog', async (dialog) => {
    await expect(dialog.type()).toContain('alert'); // Check dialog type
    await expect(dialog.message()).toContain('This is a normal alert box');
    // Validate the message
    await dialog.accept(); // Accept the dialog
  });

  // Trigger the dialog by clicking the button
  await page.locator('//button[@id="alertbutton"]').click();
});
```

## 2. Confirm Dialog with OK and Cancel Buttons

For dialogs with both **OK** and **Cancel** buttons, you can choose to either accept or dismiss the dialog.

### Example Code: Handling a Confirm Dialog

```
const { test, expect } = require('@playwright/test');

test('Confirmation dialog with OK and Cancel buttons', async ({ page }) => {
  {
    page.on('dialog', async (dialog) => {
      await expect(dialog.type()).toContain('confirm'); // Check dialog type
      await expect(dialog.message()).toContain('Press Ok or cancel'); //
      // Validate the message
      await dialog.accept(); // Accept the dialog (you can also use
      // dialog.dismiss() to cancel)
    });

    // Trigger the confirmation dialog by clicking the button
    await page.click('//button[@id="confirmation_dialog"]');

    // After accepting the dialog, validate the success message displayed on
    // the page
    await
    expect(page.locator('//p[@id="dialogSuccessfulMessage"]')).toHaveText('You
    Pressed Ok');
  }
});
```

---

### 3. Prompt Dialog with Input Box

Prompt dialogs contain an **input box**, where you can enter text. You can validate the default value, input a new value and either accept or dismiss the dialog.

#### Example Code: Handling a Prompt Dialog

```
const { test, expect } = require('@playwright/test');

test('Prompt dialog box', async ({ page }) => {
  page.on('dialog', async (dialog) => {
    await expect(dialog.type()).toContain('prompt'); // Check dialog type
    await expect(dialog.message()).toContain('Please enter your name'); //
    Validate the message
    await expect(dialog.defaultValue()).toContain('Friends'); // Validate
    the default value

    await dialog.accept('Spidey'); // Accept the dialog with a new value
    // Alternatively, you can use dialog.dismiss() to cancel the prompt
  });

  // Trigger the prompt dialog by clicking the button
  await page.click('//button[@id="confirmation_dialog"]');

  // After accepting the dialog, validate the success message displayed on
  the page
  await
  expect(page.locator('//p[@id="dialogSuccessfullMessage"]')).toHaveText('Hey
  Spidey');
});
```

---

#### Summary of Dialog Handling Functions

1. **dialog.type()**: Returns the type of the dialog (alert, confirm, prompt).
  2. **dialog.message()**: Returns the message displayed in the dialog.
  3. **dialog.defaultValue()**: Returns the default value in a prompt dialog (if available).
  4. **dialog.accept()**: Accepts the dialog (useful for OK or confirmation buttons).
  5. **dialog.dismiss()**: Dismisses the dialog (useful for Cancel buttons).
- 

#### Conclusion

In Playwright, dialogs can be handled easily by registering a dialog handler before the action that triggers the dialog. This allows you to perform various operations like validating the dialog's type and message, accepting or dismissing the dialog, and handling prompt dialogs by providing input values.