



# Events

## Introduction

Playwright allows listening to various types of events happening on the web page, such as network requests, creation of child pages, dedicated workers etc. There are several ways to subscribe to such events such as waiting for events or adding or removing event listeners.

## Waiting for event

Most of the time, scripts will need to wait for a particular event to happen. Below are some of the typical event awaiting patterns.

Wait for a request with the specified url using `page.expect_request()`:

**Sync**    Async

```
with page.expect_request("**/*logo*.png") as first:
    page.goto("https://wikipedia.org")
print(first.value.url)
```

Wait for popup window:

**Sync**    Async

```
with page.expect_popup() as popup:
    page.get_by_text("open the popup").click()
popup.value.goto("https://wikipedia.org")
```

# Adding/removing event listener

Sometimes, events happen in random time and instead of waiting for them, they need to be handled. Playwright supports traditional language mechanisms for subscribing and unsubscribing from the events:

**Sync**    Async

---

```
def print_request_sent(request):
    print("Request sent: " + request.url)

def print_request_finished(request):
    print("Request finished: " + request.url)

page.on("request", print_request_sent)
page.on("requestfinished", print_request_finished)
page.goto("https://wikipedia.org")

page.remove_listener("requestfinished", print_request_finished)
page.goto("https://www.openstreetmap.org/")
```

## Adding one-off listeners

If a certain event needs to be handled once, there is a convenience API for that:

**Sync**    Async

---

```
page.once("dialog", lambda dialog: dialog.accept("2021"))
page.evaluate("prompt('Enter a number: ')")
```