

# **Downloads**

## Introduction

For every attachment downloaded by the page, page.on("download") event is emitted. All these attachments are downloaded into a temporary folder. You can obtain the download url, file name and payload stream using the Download object from the event.

You can specify where to persist downloaded files using the downloads\_path option in browser\_type.launch().



Downloaded files are deleted when the browser context that produced them is closed.

Here is the simplest way to handle the file download:

#### Sync Async

```
# Start waiting for the download
with page.expect_download() as download_info:
    # Perform the action that initiates download
    page.get_by_text("Download file").click()
download = download_info.value

# Wait for the download process to complete and save the downloaded file somewhere
download.save_as("/path/to/save/at/" + download.suggested_filename)
```

#### **Variations**

If you have no idea what initiates the download, you can still handle the event:

### Sync Async

```
page.on("download", lambda download: print(download.path()))
```

Note that handling the event forks the control flow and makes the script harder to follow. Your scenario might end while you are downloading a file since your main control flow is not awaiting for this operation to resolve.



For uploading files, see the <u>uploading files</u> section.