

События (Events)

События в `page.goto`

В переменной `wait_until`.

- `load`: загрузка всего контента
- `domcontentloaded`: загрузка `dom`
- `commit`: при получении ответа от сервера
- `networkidle`: до завершения всех событий сети. Для динамического контента не меньше чем `load`.

Можно считать время загрузки.

```
from playwright.sync_api import sync_playwright
from time import perf_counter

with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=False, slow_mo=500)
    page = browser.new_page()
    print('Page loading...')
    start = perf_counter()
    page.goto("https://playwright.dev/python", wait_until='load')
    delta = perf_counter() - start
    print(f'Page loaded in {delta} s.')

    browser.close()
```

События динамического контента (React, ...)

Находим динамический элемент, кликаем по нему и при помощи `wait_for()` ждем.

```
from playwright.sync_api import sync_playwright
from time import perf_counter

with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=False, slow_mo=500)
    page = browser.new_page()
    page.goto("https://www.scrapethissite.com/pages/ajax-javascript/", timeout=60_000)
```

```
mylink = page.get_by_role("link", name="2014")
mylink.click()
print('Loading movie...')
start = perf_counter()
loadedcont = page.locator("td.film-title").first
loadedcont.wait_for()
delta = perf_counter() - start
print(f'Movie loaded in {delta} s.')

browser.close()
```

Ожидание события.

Указываем тип события и функцию, выполняемую при наступлении события.

```
from playwright.sync_api import sync_playwright

def onload(page):
    print("Page loaded", page)

with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=False, slow_mo=500)
    page = browser.new_page()
    page.on("load", onload)
    page.goto("https://bootswatch.com/default")
    browser.close()
```

Пример просмотра событий запросов

```
from playwright.sync_api import sync_playwright

def onrequest(request):
    print("Request send: ", request)

with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=False, slow_mo=500)
    page = browser.new_page()
    page.on("request", onrequest)
    page.goto("https://bootswatch.com/default")
    browser.close()
```

Вывод:

```
Request send: <Request url='https://bootswatch.com/default' method='GET'>
Request send: <Request url='http://bootswatch.com/default/' method='GET'>
Request send: <Request url='https://bootswatch.com/default/' method='GET'>
Request send: <Request url='https://bootswatch.com/_vendor/bootstrap/dist/css/bootstrap.css' method='GET'>
Request send: <Request url='https://bootswatch.com/_vendor/bootstrap-icons/font/bootstrap-icons.min.css'
method='GET'>
Request send: <Request url='https://bootswatch.com/_vendor/prismjs/themes/prism-okaidia.css'
method='GET'>
Request send: <Request url='https://bootswatch.com/_assets/css/custom.min.css' method='GET'>
Request send: <Request url='https://www.googletagmanager.com/gtag/js?id=G-KGDJB EFF3W' method='GET'>
Request send: <Request
url='https://cdn.carbonads.com/carbon.js?serve=CKYIE23N&placement=bootswatchcom' method='GET'>
Request send: <Request url='https://bootswatch.com/_vendor/bootstrap/dist/js/bootstrap.bundle.min.js'
method='GET'>
Request send: <Request url='https://bootswatch.com/_vendor/prismjs/prism.js' method='GET'>
Request send: <Request url='https://bootswatch.com/_assets/js/custom.js' method='GET'>
Request send: <Request url='https://bootswatch.com/_vendor/bootstrap-icons/font/fonts/bootstrap-
icons.woff2?1fa40e8900654d2863d011707b9fb6f2' method='GET'>
Request send: <Request url='https://www.google-analytics.com/g/collect?v=2&tid=G-
KGDJB EFF3W&gtm=45je54g0v9135688085za200&_p=1744912703288&gcd=13131313111&npa=0&dma=0&tag_
exp=102509682~102803279~102813109~102887800~102926062~103027016~103051953~103055465~10
3077950~103106314~103106316~103130495~103130497&cid=128797986.1744912704&ul=ru-
ru&sr=1280x720&uaa=x86&uab=64&uafvl=Not%253AA-
Brand%3B24.0.0.0%7CChromium%3B134.0.6998.35&uamb=0&uam=&uap=Windows&uapv=10.0.0&uaw=0&ar
e=1&frm=0&pscdl=noapi&_s=1&sid=1744912703&sct=1&seg=0&dl=https%3A%2F%2Fbootswatch.com%2Fde
fault%2F&dt=Bootswatch%3A%20Default&en=page_view&_fv=1&_nsi=1&_ss=1&_ee=1&tfd=2018'
method='POST'>
Request send: <Request
url='https://srv.carbonads.net/ads/CKYIE23N.json?segment=placement:bootswatchcom&v=true'
method='GET'>
```

Есть событие при выборе файла.

Удаление прослушивания события: `page.remove_listener("name_event", func)`

События всплывающих окон (`alert`, `confirm`, `prompt`)

Событие `page.on("dialog", func)`, устанавливаем ожидание до возможного появления окна.

Функция:

```
def on_dialog(dialog)
    dialog.accept()
    dialog.dismiss()
```

Если диалог prompt, то для заполнения поля ввода в функции ассерт нужно добавить строковую переменную.

```
def on_dialog(dialog)
    dialog.accept("Text for enter")
```

Событие скачивания файла

```
from playwright.sync_api import sync_playwright

with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=False, slow_mo=500)
    page = browser.new_page()
    page.goto("https://bootswatch.com/default")
    btn = page.get_by_role("link", name="Download me")
    with page.expect_download() as download_info:
        btn.click()
    download = download_info.value
    download.save_as(fname)
    browser.close()
```

Второй вариант: добавить listener

```
from playwright.sync_api import sync_playwright

def saving_func(download):
    fname = "first.jpg"
    download.save_as(fname)

with sync_playwright() as playwright:
    browser = playwright.chromium.launch(headless=False, slow_mo=500)
    page = browser.new_page()
    page.goto("https://bootswatch.com/default")
    page.once("download", saving_func)
    btn = page.get_by_role("link", name="Download me")
    with page.expect_download() as download_info:
        btn.click()
```

browser.close()

Revision #4

Created 17 April 2025 16:36:17 by Admin

Updated 18 April 2025 07:05:22 by Admin