



BrowserType

BrowserType provides methods to launch a specific browser instance or connect to an existing one. The following is a typical example of using Playwright to drive automation:

Sync **Async**

```
from playwright.sync_api import sync_playwright, Playwright

def run(playwright: Playwright):
    chromium = playwright.chromium
    browser = chromium.launch()
    page = browser.new_page()
    page.goto("https://example.com")
    # other actions...
    browser.close()

with sync_playwright() as playwright:
    run(playwright)
```

Methods

connect

Added in: v1.8

This method attaches Playwright to an existing browser instance. When connecting to another browser launched via `BrowserType.launchServer` in Node.js, the major and minor version needs to match the client version (1.2.3 → is compatible with 1.2.x).

Usage

```
browser_type.connect(ws_endpoint)
browser_type.connect(ws_endpoint, **kwargs)
```

Arguments

- `ws_endpoint` **str** Added in: v1.10

A browser websocket endpoint to connect to.

- `expose_network` **str** (*optional*) Added in: v1.37

This option exposes network available on the connecting client to the browser being connected to. Consists of a list of rules separated by comma.

Available rules:

- Hostname pattern, for example: `example.com`, `*.org:99`, `x.*.y.com`, `*foo.org`.
- IP literal, for example: `127.0.0.1`, `0.0.0.0:99`, `:::1`, `[0:0:::1]:99`.
- `<loopback>` that matches local loopback interfaces: `localhost`, `*.localhost`, `127.0.0.1`, `:::1`.

Some common examples:

- `"*"` to expose all network.
- `"<loopback>"` to expose localhost network.
- `"*.test.internal-domain,*.staging.internal-domain,<loopback>"` to expose test/staging deployments and localhost.

- `headers` **Dict[str, str]** (*optional*) Added in: v1.11

Additional HTTP headers to be sent with web socket connect request. Optional.

- `slow_mo` **float** (*optional*) Added in: v1.10

Slows down Playwright operations by the specified amount of milliseconds. Useful so that you can see what is going on. Defaults to 0.

- `timeout` **float** (*optional*) Added in: v1.10

Maximum time in milliseconds to wait for the connection to be established. Defaults to `0` (no timeout).

Returns

- **Browser**

connect_over_cdp

Added in: v1.9

This method attaches Playwright to an existing browser instance using the Chrome DevTools Protocol.

The default browser context is accessible via [browser.contexts](#).

NOTE

Connecting over the Chrome DevTools Protocol is only supported for Chromium-based browsers.

Usage

Sync **Async**

```
browser = playwright.chromium.connect_over_cdp("http://localhost:9222")
default_context = browser.contexts[0]
page = default_context.pages[0]
```

Arguments

- `endpoint_url` [str](#) Added in: v1.11

A CDP websocket endpoint or http url to connect to. For example `http://localhost:9222/` or `ws://127.0.0.1:9222/devtools/browser/387adf4c-243f-4051-a181-46798f4a46f4`.

- `headers` [Dict\[str, str\]](#) (*optional*) Added in: v1.11

Additional HTTP headers to be sent with connect request. Optional.

- `slow_mo` [float](#) (*optional*) Added in: v1.11

Slows down Playwright operations by the specified amount of milliseconds. Useful so that you can see what is going on. Defaults to 0.

- `timeout` [float](#) (*optional*) Added in: v1.11

Maximum time in milliseconds to wait for the connection to be established. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- [Browser](#)

launch

Added in: v1.8

Returns the browser instance.

Usage

You can use `ignore_default_args` to filter out `--mute-audio` from default arguments:

Sync **Async**

```
browser = playwright.chromium.launch( # or "firefox" or "webkit".
    ignore_default_args=["--mute-audio"]
)
```

Chromium-only Playwright can also be used to control the Google Chrome or Microsoft Edge browsers, but it works best with the version of Chromium it is bundled with. There is no guarantee it will work with any other version. Use `executable_path` option with extreme caution.

If Google Chrome (rather than Chromium) is preferred, a [Chrome Canary](#) or [Dev Channel](#) build is suggested.

Stock browsers like Google Chrome and Microsoft Edge are suitable for tests that require proprietary media codecs for video playback. See [this article](#) for other differences between Chromium and Chrome. [This article](#) describes some differences for Linux users.

Arguments

- `args` [List\[str\]](#) (*optional*)

Additional arguments to pass to the browser instance. The list of Chromium flags can be found [here](#).

- `channel` [str](#) (*optional*)

Browser distribution channel. Supported values are "chrome", "chrome-beta", "chrome-dev", "chrome-canary", "msedge", "msedge-beta", "msedge-dev", "msedge-canary". Read more about using [Google Chrome and Microsoft Edge](#).

- `chromium_sandbox` `bool` (*optional*)

Enable Chromium sandboxing. Defaults to `false`.

- `devtools` `bool` (*optional*)

Chromium-only Whether to auto-open a Developer Tools panel for each tab. If this option is `true`, the `headless` option will be set `false`.

- `downloads_path` `Union[str, pathlib.Path]` (*optional*)

If specified, accepted downloads are downloaded into this directory. Otherwise, temporary directory is created and is deleted when browser is closed. In either case, the downloads are deleted when the browser context they were created in is closed.

- `env` `Dict[str, str|float|bool]` (*optional*)

Specify environment variables that will be visible to the browser. Defaults to `process.env`.

- `executable_path` `Union[str, pathlib.Path]` (*optional*)

Path to a browser executable to run instead of the bundled one. If `executable_path` is a relative path, then it is resolved relative to the current working directory. Note that Playwright only works with the bundled Chromium, Firefox or WebKit, use at your own risk.

- `firefox_user_prefs` `Dict[str, str|float|bool]` (*optional*)

Firefox user preferences. Learn more about the Firefox user preferences at [about:config](#).

- `handle_sighup` `bool` (*optional*)

Close the browser process on SIGHUP. Defaults to `true`.

- `handle_sigint` `bool` (*optional*)

Close the browser process on Ctrl-C. Defaults to `true`.

- `handle_sigterm` `bool` (*optional*)

Close the browser process on SIGTERM. Defaults to `true`.

- `headless` `bool` (*optional*)

Whether to run browser in headless mode. More details for `Chromium` and `Firefox`. Defaults to `true` unless the `devtools` option is `true`.

- `ignore_default_args` `bool|List[str]` (*optional*)

If `true`, Playwright does not pass its own configurations args and only uses the ones from `args`. If an array is given, then filters out the given default arguments. Dangerous option; use with care. Defaults to `false`.

- `proxy` `Dict` (*optional*)

- `server` `str`

Proxy to be used for all requests. HTTP and SOCKS proxies are supported, for example `http://myproxy.com:3128` or `socks5://myproxy.com:3128`. Short form `myproxy.com:3128` is considered an HTTP proxy.

- `bypass` `str` (*optional*)

Optional comma-separated domains to bypass proxy, for example `".com, chromium.org, .domain.com"`.

- `username` `str` (*optional*)

Optional username to use if HTTP proxy requires authentication.

- `password` `str` (*optional*)

Optional password to use if HTTP proxy requires authentication.

Network proxy settings.

- `slow_mo` `float` (*optional*)

Slows down Playwright operations by the specified amount of milliseconds. Useful so that you can see what is going on.

- `timeout` `float` (*optional*)

Maximum time in milliseconds to wait for the browser instance to start. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

- `traces_dir` `Union[str, pathlib.Path]` (*optional*)

If specified, traces are saved into this directory.

Returns

- `Browser`

launch_persistent_context

Added in: v1.8

Returns the persistent browser context instance.

Launches browser that uses persistent storage located at `user_data_dir` and returns the only context. Closing this context will automatically close the browser.

Usage

```
browser_type.launch_persistent_context(user_data_dir)
browser_type.launch_persistent_context(user_data_dir, **kwargs)
```

Arguments

- `user_data_dir` `Union[str, pathlib.Path]`

Path to a User Data Directory, which stores browser session data like cookies and local storage. More details for `Chromium` and `Firefox`. Note that Chromium's user data directory is the **parent** directory of the "Profile Path" seen at `chrome://version`. Pass an empty string to use a temporary directory instead.

- `accept_downloads` `bool` (*optional*)

Whether to automatically download all the attachments. Defaults to `true` where all the downloads are accepted.

- `args` `List[str]` (*optional*)

Additional arguments to pass to the browser instance. The list of Chromium flags can be found [here](#).

- `base_url` `str` (*optional*)

When using `page.goto()`, `page.route()`, `page.wait_for_url()`, `page.expect_request()`, or `page.expect_response()` it takes the base URL in consideration by using the `URL()` constructor for building the corresponding URL. Unset by default. Examples:

- `baseURL`: `http://localhost:3000` and navigating to `/bar.html` results in `http://localhost:3000/bar.html`
- `baseURL`: `http://localhost:3000/foo/` and navigating to `./bar.html` results in `http://localhost:3000/foo/bar.html`
- `baseURL`: `http://localhost:3000/foo` (without trailing slash) and navigating to `./bar.html` results in `http://localhost:3000/bar.html`

- `bypass_csp` `bool` (*optional*)

Toggles bypassing page's Content-Security-Policy. Defaults to `false`.

- `channel` `str` (*optional*)

Browser distribution channel. Supported values are "chrome", "chrome-beta", "chrome-dev", "chrome-canary", "msedge", "msedge-beta", "msedge-dev", "msedge-canary". Read more about using [Google Chrome and Microsoft Edge](#).

- `chromium_sandbox` `bool` (*optional*)

Enable Chromium sandboxing. Defaults to `false`.

- `color_scheme` `"light"|"dark"|"no-preference"|"null"` (*optional*)

Emulates `'prefers-colors-scheme'` media feature, supported values are `'light'`, `'dark'`, `'no-preference'`. See `page.emulate_media()` for more details. Passing `'null'` resets emulation to system defaults. Defaults to `'light'`.

- `device_scale_factor` `float` (*optional*)

Specify device scale factor (can be thought of as dpr). Defaults to `1`. Learn more about [emulating devices with device scale factor](#).

- `devtools` `bool` (*optional*)

Chromium-only Whether to auto-open a Developer Tools panel for each tab. If this option is `true`, the `headless` option will be set `false`.

- `downloads_path` `Union[str, pathlib.Path]` (*optional*)

If specified, accepted downloads are downloaded into this directory. Otherwise, temporary directory is created and is deleted when browser is closed. In either case, the downloads are deleted when the browser context they were created in is closed.

- `env` `Dict[str, str|float|bool]` (*optional*)

Specify environment variables that will be visible to the browser. Defaults to `process.env`.

- `executable_path` `Union[str, pathlib.Path]` (*optional*)

Path to a browser executable to run instead of the bundled one. If `executable_path` is a relative path, then it is resolved relative to the current working directory. Note that Playwright only works with the bundled Chromium, Firefox or WebKit, use at your own risk.

- `extra_http_headers` `Dict[str, str]` (*optional*)

An object containing additional HTTP headers to be sent with every request. Defaults to none.

- `forced_colors` `"active"|"none"|"null"` (*optional*)

Emulates `'forced-colors'` media feature, supported values are `'active'`, `'none'`. See [page.emulate_media\(\)](#) for more details. Passing `'null'` resets emulation to system defaults. Defaults to `'none'`.

- `geolocation` `Dict` (*optional*)

- `latitude` `float`

Latitude between -90 and 90.

- `longitude` `float`

Longitude between -180 and 180.

- `accuracy` `float` (*optional*)

Non-negative accuracy value. Defaults to `0`.

- `handle_sighup` `bool` (*optional*)

Close the browser process on SIGHUP. Defaults to `true`.

- `handle_sigint` `bool` (*optional*)

Close the browser process on Ctrl-C. Defaults to `true`.

- `handle_sigterm` `bool` (*optional*)

Close the browser process on SIGTERM. Defaults to `true`.

- `has_touch` `bool` (*optional*)

Specifies if viewport supports touch events. Defaults to false. Learn more about [mobile emulation](#).

- `headless` `bool` (*optional*)

Whether to run browser in headless mode. More details for [Chromium](#) and [Firefox](#). Defaults to `true` unless the `devtools` option is `true`.

- `http_credentials` `Dict` (*optional*)

- `username` `str`
- `password` `str`
- `origin` `str` (*optional*)

Restrain sending http credentials on specific origin (scheme://host:port).

Credentials for [HTTP authentication](#). If no origin is specified, the username and password are sent to any servers upon unauthorized responses.

- `ignore_default_args` `bool|List[str]` (*optional*)

If `true`, Playwright does not pass its own configurations args and only uses the ones from `args`. If an array is given, then filters out the given default arguments. Dangerous option; use with care. Defaults to `false`.

- `ignore_https_errors` `bool` (*optional*)

Whether to ignore HTTPS errors when sending network requests. Defaults to `false`.

- `is_mobile` `bool` (*optional*)

Whether the `meta viewport` tag is taken into account and touch events are enabled. `isMobile` is a part of device, so you don't actually need to set it manually. Defaults to `false` and is not supported in Firefox. Learn more about [mobile emulation](#).

- `java_script_enabled` `bool` (*optional*)

Whether or not to enable JavaScript in the context. Defaults to `true`. Learn more about [disabling JavaScript](#).

- `locale` `str` (*optional*)

Specify user locale, for example `en-GB`, `de-DE`, etc. Locale will affect `navigator.language` value, `Accept-Language` request header value as well as number and date formatting rules. Defaults to the system default locale. Learn more about emulation in our [emulation guide](#).

- `no_viewport` `bool` (*optional*)

Does not enforce fixed viewport, allows resizing window in the headed mode.

- `offline` `bool` (*optional*)

Whether to emulate network being offline. Defaults to `false`. Learn more about [network emulation](#).

- `permissions` `List[str]` (*optional*)

A list of permissions to grant to all pages in this context. See [browser_context.grant_permissions\(\)](#) for more details. Defaults to none.

- `proxy` `Dict` (*optional*)

- `server` `str`

Proxy to be used for all requests. HTTP and SOCKS proxies are supported, for example `http://myproxy.com:3128` or `socks5://myproxy.com:3128`. Short form `myproxy.com:3128` is considered an HTTP proxy.

- `bypass` `str` (*optional*)

Optional comma-separated domains to bypass proxy, for example `".com, chromium.org, .domain.com"`.

- `username` `str` (*optional*)

Optional username to use if HTTP proxy requires authentication.

- `password` `str` (*optional*)

Optional password to use if HTTP proxy requires authentication.

Network proxy settings.

- `record_har_content` `"omit"|"embed"|"attach"` (*optional*)

Optional setting to control resource content management. If `omit` is specified, content is not persisted. If `attach` is specified, resources are persisted as separate files and all of these files are archived along with the HAR file. Defaults to `embed`, which stores content inline the HAR file as per HAR specification.

- `record_har_mode` `"full"|"minimal"` (*optional*)

When set to `minimal`, only record information necessary for routing from HAR. This omits sizes, timing, page, cookies, security and other types of HAR information that are not used when replaying from HAR. Defaults to `full`.

- `record_har_omit_content` `bool` (*optional*)

Optional setting to control whether to omit request content from the HAR. Defaults to `false`.

- `record_har_path` `Union[str, pathlib.Path]` (*optional*)

Enables `HAR` recording for all pages into the specified HAR file on the filesystem. If not specified, the HAR is not recorded. Make sure to call `browser_context.close()` for the HAR to be saved.

- `record_har_url_filter` `str|Pattern` (*optional*)

- `record_video_dir` `Union[str, pathlib.Path]` (*optional*)

Enables video recording for all pages into the specified directory. If not specified videos are not recorded. Make sure to call `browser_context.close()` for videos to be saved.

- `record_video_size` `Dict` (*optional*)

- `width` `int`

Video frame width.

- `height` `int`

Video frame height.

Dimensions of the recorded videos. If not specified the size will be equal to `viewport` scaled down to fit into 800x800. If `viewport` is not configured explicitly the video size defaults to 800x450. Actual picture of each page will be scaled down if necessary to fit the specified size.

- `reduced_motion` `"reduce"|"no-preference"|"null"` (*optional*)

Emulates `'prefers-reduced-motion'` media feature, supported values are `'reduce'`, `'no-preference'`. See `page.emulate_media()` for more details. Passing `'null'` resets emulation to system defaults. Defaults to `'no-preference'`.

- `screen` `Dict` (*optional*)

- `width` `int`

page width in pixels.

- `height` `int`

page height in pixels.

Emulates consistent window screen size available inside web page via `window.screen`. Is only used when the `viewport` is set.

- `service_workers` `"allow"|"block"` (*optional*)

Whether to allow sites to register Service workers. Defaults to `'allow'`.

- `'allow'`: **Service Workers** can be registered.
- `'block'`: Playwright will block all registration of Service Workers.

- `slow_mo` `float` (*optional*)

Slows down Playwright operations by the specified amount of milliseconds. Useful so that you can see what is going on.

- `strict_selectors` `bool` (*optional*)

If set to true, enables strict selectors mode for this context. In the strict selectors mode all operations on selectors that imply single target DOM element will throw when more than one element matches the selector. This option does not affect any Locator APIs (Locators are always strict). Defaults to `false`. See [Locator](#) to learn more about the strict mode.

- `timeout` `float` (*optional*)

Maximum time in milliseconds to wait for the browser instance to start. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

- `timezone_id` `str` (*optional*)

Changes the timezone of the context. See [ICU's metaZones.txt](#) for a list of supported timezone IDs. Defaults to the system timezone.

- `traces_dir` `Union[str, pathlib.Path]` (*optional*)

If specified, traces are saved into this directory.

- `user_agent` `str` (*optional*)

Specific user agent to use in this context.

- `viewport` `NoneType|Dict` (*optional*)

- `width` `int`

page width in pixels.

- `height` `int`

page height in pixels.

Sets a consistent viewport for each page. Defaults to an 1280x720 viewport. `no_viewport` disables the fixed viewport. Learn more about [viewport emulation](#).

Returns

- [BrowserContext](#)

Properties

executable_path

Added in: v1.8

A path where Playwright expects to find a bundled browser executable.

Usage

```
browser_type.executable_path
```

Returns

- `str`

name

Added in: v1.8

Returns browser name. For example: `'chromium'`, `'webkit'` or `'firefox'`.

Usage

```
browser_type.name
```

Returns

- `str`