Browser

A Browser is created via browserType.launch(). An example of using a Browser to create a Page:

```
const { firefox } = require('playwright'); // Or 'chromium' or 'webkit'.

(async () => {
  const browser = await firefox.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await browser.close();
})();
```

Methods

browserType

Added in: v1.23

Get the browser type (chromium, firefox or webkit) that the browser belongs to.

Usage

```
browser.browserType();
```

Returns

BrowserType

close

Added before v1.9

In case this browser is obtained using browserType.launch(), closes the browser and all of its pages (if any were opened).

In case this browser is connected to, clears all created contexts belonging to this browser and disconnects from the browser server.



This is similar to force quitting the browser. Therefore, you should call <u>browserContext.close()</u> on any <u>BrowserContext</u>'s you explicitly created earlier with <u>browser.newContext()</u> **before** calling <u>browser.close()</u>.

The Browser object itself is considered to be disposed and cannot be used anymore.

Usage

```
await browser.close();
await browser.close(options);
```

Arguments

- options Object (optional)
 - o reason string (optional) Added in: v1.40

The reason to be reported to the operations interrupted by the browser closure.

Returns

Promise<void>

contexts

Added before v1.9

Returns an array of all open browser contexts. In a newly created browser, this will return zero browser contexts.

Usage

```
const browser = await pw.webkit.launch();
console.log(browser.contexts().length); // prints `0`

const context = await browser.newContext();
console.log(browser.contexts().length); // prints `1`
```

Returns

Array<BrowserContext>

isConnected

Added before v1.9

Indicates that the browser is connected.

Usage

```
browser.isConnected();
```

Returns

boolean

newBrowserCDPSession

Added in: v1.11



CDP Sessions are only supported on Chromium-based browsers.

Returns the newly created browser session.

Usage

```
await browser.newBrowserCDPSession();
```

Returns

• Promise<CDPSession>

newContext

Added before v1.9

Creates a new browser context. It won't share cookies/cache with other browser contexts.



If directly using this method to create <u>BrowserContexts</u>, it is best practice to explicitly close the returned context via <u>browserContext.close()</u> when your code is done with the <u>BrowserContext</u>, and before calling <u>browser.close()</u>. This will ensure the <u>context</u> is closed gracefully and any artifacts—like HARs and videos—are fully flushed and saved.

Usage

```
(async () => {
  const browser = await playwright.firefox.launch(); // Or 'chromium' or
'webkit'.
  // Create a new incognito browser context.
  const context = await browser.newContext();
  // Create a new page in a pristine context.
  const page = await context.newPage();
  await page.goto('https://example.com');

// Gracefully close up everything
  await context.close();
  await browser.close();
})();
```

Arguments

- options Object (optional)
 - acceptDownloads boolean (optional)

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

baseURL string (optional)

When using page.goto(), page.route(), page.waitForURL(), page.waitForRequest(), or page.waitForResponse() 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
- bypassCSP boolean (optional)

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

- clientCertificates Array<Object> (optional) Added in: 1.46
 - origin string

Exact origin that the certificate is valid for. Origin includes https protocol, a hostname and optionally a port.

certPath string (optional)

Path to the file with the certificate in PEM format.

cert Buffer (optional)

Direct value of the certificate in PEM format.

keyPath string (optional)

Path to the file with the private key in PEM format.

key Buffer (optional)

Direct value of the private key in PEM format.

pfxPath string (optional)

Path to the PFX or PKCS12 encoded private key and certificate chain.

pfx Buffer (optional)

Direct value of the PFX or PKCS12 encoded private key and certificate chain.

passphrase string (optional)

Passphrase for the private key (PEM or PFX).

TLS Client Authentication allows the server to request a client certificate and verify it.

Details

An array of client certificates to be used. Each certificate object must have either both certPath and keyPath, a single pfxPath, or their corresponding direct value equivalents (cert and key, or pfx). Optionally, passphrase property should be provided if the certificate is encrypted. The origin property should be provided with an exact match to the request origin that the certificate is valid for.

(i) NOTE

When using WebKit on macOS, accessing localhost will not pick up client certificates. You can make it work by replacing localhost with local.playwright.

colorScheme null | "light" | "dark" | "no-preference" (optional)

Emulates prefers-colors-scheme media feature, supported values are 'light' and 'dark'. See page.emulateMedia() for more details. Passing null resets emulation to system defaults. Defaults to 'light'.

deviceScaleFactor number (optional)

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

• (extraHTTPHeaders) Object<string, string> (optional)

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

forcedColors null | "active" | "none" (optional)

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

- geolocation Object (optional)
 - latitude number

Latitude between -90 and 90.

longitude number

Longitude between -180 and 180.

accuracy number (optional)

Non-negative accuracy value. Defaults to 0.

hasTouch boolean (optional)

Specifies if viewport supports touch events. Defaults to false. Learn more about mobile emulation.

- httpCredentials Object (optional)
 - username string
 - password string
 - origin string (optional)

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

send "unauthorized" | "always" (optional)

This option only applies to the requests sent from corresponding APIRequestContext and does not affect requests sent from the browser. 'always' - Authorization header with basic authentication credentials will be sent with the each API request. 'unauthorized - the credentials are only sent when 401 (Unauthorized) response with WWW-Authenticate header is received. Defaults to 'unauthorized'.

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

• ignoreHTTPSErrors boolean (optional)

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

• isMobile boolean (optional)

Whether the meta viewport tag is taken into account and touch events are enabled. is Mobile 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.

javaScriptEnabled boolean (optional)

Whether or not to enable JavaScript in the context. Defaults to true. Learn more about disabling JavaScript.

locale string (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.

• logger Logger (optional)

Logger sink for Playwright logging.

offline boolean (optional)

Whether to emulate network being offline. Defaults to false. Learn more about network emulation.

permissions Array<string> (optional)

A list of permissions to grant to all pages in this context. See browserContext.grantPermissions() for more details. Defaults to none.

- proxy Object (optional)
 - server string

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 string (optional)

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

username string (optional)

Optional username to use if HTTP proxy requires authentication.

password string (optional)

Optional password to use if HTTP proxy requires authentication.

Network proxy settings to use with this context. Defaults to none.

- recordHar Object (optional)
 - omitContent boolean (optional)

Optional setting to control whether to omit request content from the HAR. Defaults to false. Deprecated, use content policy instead.

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 or entries in the ZIP archive. If embed is specified, content is stored inline the HAR file as per HAR specification. Defaults to attach for zip output files and to embed for all other file extensions.

path string

Path on the filesystem to write the HAR file to. If the file name ends with .zip, content: 'attach' is used by default.

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.

urlFilter string | RegExp (optional)

A glob or regex pattern to filter requests that are stored in the HAR. When a baseURL via the context options was provided and the passed URL is a path, it gets merged via the new URL() constructor. Defaults to none.

Enables HAR recording for all pages into recordHar.path file. If not specified, the HAR is not recorded. Make sure to await browserContext.close() for the HAR to be saved.

- o recordVideo Object (optional)
 - dir string

Path to the directory to put videos into.

- size Object (optional)
 - width number

Video frame width.

height number

Video frame height.

Optional 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.

Enables video recording for all pages into recordVideo.dir directory. If not specified videos are not recorded. Make sure to await browserContext.close() for videos to be saved.

reducedMotion null | "reduce" | "no-preference" (optional)

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

- screen Object (optional)
 - width numberpage width in pixels.
 - height numberpage height in pixels.

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

o serviceWorkers "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.
- storageState string | Object (optional)
 - cookies Array<Object>
 - name string

- value string
- domain string

Domain and path are required. For the cookie to apply to all subdomains as well, prefix domain with a dot, like this: ".example.com"

path string

Domain and path are required

expires number

Unix time in seconds.

- httpOnly boolean
- secure boolean
- sameSite "Strict" | "Lax" | "None"

sameSite flag

Cookies to set for context

- origins Array<Object>
 - origin string
 - localStorage Array<Object>
 - name string
 - value string

localStorage to set for context

Learn more about storage state and auth.

Populates context with given storage state. This option can be used to initialize context with logged-in information obtained via browserContext.storageState().

• strictSelectors boolean (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.

timezoneId string (optional)

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

userAgent string (optional)

Specific user agent to use in this context.

videoSize Object (optional)



Use recordVideo instead.

width number

Video frame width.

height number

Video frame height.

videosPath string (optional)



A DEPRECATED

Use <u>recordVideo</u> instead.

- viewport null | Object (optional)
 - width number page width in pixels.
 - height number

page height in pixels.

Emulates consistent viewport for each page. Defaults to an 1280x720 viewport. Use null to disable the consistent viewport emulation. Learn more about viewport emulation.



The null value opts out from the default presets, makes viewport depend on the host window size defined by the operating system. It makes the execution of the tests non-deterministic.

Returns

Promise<BrowserContext>

newPage

Added before v1.9

Creates a new page in a new browser context. Closing this page will close the context as well.

This is a convenience API that should only be used for the single-page scenarios and short snippets. Production code and testing frameworks should explicitly create browser.newContext() followed by the browserContext.newPage() to control their exact life times.

Usage

```
await browser.newPage();
await browser.newPage(options);
```

Arguments

- options Object (optional)
 - acceptDownloads boolean (optional)

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

baseURL string (optional)

When using page.goto(), page.route(), page.waitForURL(), page.waitForRequest(), or page.waitForResponse() 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
- bypassCSP boolean (optional)

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

- clientCertificates Array<Object> (optional) Added in: 1.46
 - origin string

Exact origin that the certificate is valid for. Origin includes https protocol, a hostname and optionally a port.

certPath string (optional)

Path to the file with the certificate in PEM format.

cert Buffer (optional)

Direct value of the certificate in PEM format.

keyPath string (optional)

Path to the file with the private key in PEM format.

key Buffer (optional)

Direct value of the private key in PEM format.

pfxPath string (optional)

Path to the PFX or PKCS12 encoded private key and certificate chain.

pfx Buffer (optional)

Direct value of the PFX or PKCS12 encoded private key and certificate chain.

passphrase string (optional)

Passphrase for the private key (PEM or PFX).

TLS Client Authentication allows the server to request a client certificate and verify it.

Details

An array of client certificates to be used. Each certificate object must have either both certPath and keyPath, a single pfxPath, or their corresponding direct value equivalents (cert and key, or pfx). Optionally, passphrase property should be provided if the certificate is encrypted. The origin property should be provided with an exact match to the request origin that the certificate is valid for.

(i) NOTE

When using WebKit on macOS, accessing localhost will not pick up client certificates. You can make it work by replacing localhost with local.playwright.

colorScheme null | "light" | "dark" | "no-preference" (optional)

Emulates prefers-colors-scheme media feature, supported values are 'light' and 'dark'. See page.emulateMedia() for more details. Passing null resets emulation to system defaults. Defaults to 'light'.

deviceScaleFactor number (optional)

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

extraHTTPHeaders Object<string, string> (optional)

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

• forcedColors null | "active" | "none" (optional)

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

- o geolocation Object (optional)
 - latitude number

Latitude between -90 and 90.

longitude number

Longitude between -180 and 180.

accuracy number (optional)

Non-negative accuracy value. Defaults to 0.

hasTouch boolean (optional)

Specifies if viewport supports touch events. Defaults to false. Learn more about mobile emulation.

- httpCredentials Object (optional)
 - username string
 - password string
 - origin string (optional)

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

send "unauthorized" | "always" (optional)

This option only applies to the requests sent from corresponding APIRequestContext and does not affect requests sent from the browser. 'always' - Authorization header with basic authentication credentials will be sent with the each API request. 'unauthorized - the credentials are only sent when 401 (Unauthorized) response with WWW-Authenticate header is received. Defaults to 'unauthorized'.

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

• ignoreHTTPSErrors boolean (optional)

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

• isMobile boolean (optional)

Whether the meta viewport tag is taken into account and touch events are enabled. is Mobile 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.

• javaScriptEnabled boolean (optional)

Whether or not to enable JavaScript in the context. Defaults to true. Learn more about disabling JavaScript.

locale string (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.

• logger Logger (optional)

Logger sink for Playwright logging.

offline boolean (optional)

Whether to emulate network being offline. Defaults to false. Learn more about network emulation.

o permissions Array<string> (optional)

A list of permissions to grant to all pages in this context. See browserContext.grantPermissions() for more details. Defaults to none.

- o proxy Object (optional)
 - server string

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 string (optional)

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

username string (optional)

Optional username to use if HTTP proxy requires authentication.

password string (optional)

Optional password to use if HTTP proxy requires authentication.

Network proxy settings to use with this context. Defaults to none.

- recordHar Object (optional)
 - omitContent boolean (optional)

Optional setting to control whether to omit request content from the HAR. Defaults to false. Deprecated, use content policy instead.

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 or entries in the ZIP archive. If embed is specified, content is stored inline the HAR file as per HAR specification. Defaults to attach for .zip output files and to embed for all other file extensions.

path string

Path on the filesystem to write the HAR file to. If the file name ends with .zip, content: 'attach' is used by default.

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.

urlFilter string | RegExp (optional)

A glob or regex pattern to filter requests that are stored in the HAR. When a baseURL via the context options was provided and the passed URL is a path, it gets merged via the new URL() constructor. Defaults to none.

Enables HAR recording for all pages into recordHar.path file. If not specified, the HAR is not recorded. Make sure to await browserContext.close() for the HAR to be saved.

- recordVideo Object (optional)
 - dir string

Path to the directory to put videos into.

size Object (optional)

width number

Video frame width.

height number

Video frame height.

Optional 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.

Enables video recording for all pages into recordVideo.dir directory. If not specified videos are not recorded. Make sure to await browserContext.close() for videos to be saved.

reducedMotion null | "reduce" | "no-preference" (optional)

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

- screen Object (optional)
 - width number
 page width in pixels.
 - height number page height in pixels.

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

o serviceWorkers "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.
- storageState string | Object (optional)
 - cookies Array<Object>

- name string
- value string
- domain string

Domain and path are required. For the cookie to apply to all subdomains as well, prefix domain with a dot, like this: ".example.com"

path string

Domain and path are required

expires number

Unix time in seconds.

- httpOnly boolean
- secure boolean
- sameSite "Strict" | "Lax" | "None"

sameSite flag

Cookies to set for context

- origins Array<Object>
 - origin string
 - localStorage Array<Object>
 - name string
 - value string

localStorage to set for context

Learn more about storage state and auth.

Populates context with given storage state. This option can be used to initialize context with logged-in information obtained via browserContext.storageState().

• strictSelectors boolean (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.

timezoneId string (optional)

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

userAgent string (optional)

Specific user agent to use in this context.

videoSize Object (optional)



A DEPRECATED

Use <u>recordVideo</u> instead.

width number

Video frame width.

height number

Video frame height.

videosPath string (optional)



A DEPRECATED

Use <u>recordVideo</u> instead.

- viewport null | Object (optional)
 - width number

page width in pixels.

height number

page height in pixels.

Emulates consistent viewport for each page. Defaults to an 1280x720 viewport. Use null to disable the consistent viewport emulation. Learn more about viewport emulation.

(i) NOTE

The null value opts out from the default presets, makes viewport depend on the host window size defined by the operating system. It makes the execution of the tests non-deterministic.

Returns

• Promise<Page>

removeAllListeners

Added in: v1.47

Removes all the listeners of the given type (or all registered listeners if no type given). Allows to wait for async listeners to complete or to ignore subsequent errors from these listeners.

Usage

```
await browser.removeAllListeners();
await browser.removeAllListeners(type, options);
```

Arguments

- type string (optional)
- options Object (optional)
 - behavior "wait" | "ignoreErrors" | "default" (optional)

Specifies whether to wait for already running listeners and what to do if they throw errors:

- 'default' do not wait for current listener calls (if any) to finish, if the listener throws, it may result in unhandled error
- 'wait' wait for current listener calls (if any) to finish
- 'ignoreErrors' do not wait for current listener calls (if any) to finish, all errors thrown by the listeners after removal are silently caught

Returns

Promise<void>

startTracing

Added in: v1.11



This API controls <u>Chromium Tracing</u> which is a low-level chromium-specific debugging tool. API to control <u>Playwright Tracing</u> could be found <u>here</u>.

You can use browser.startTracing() and browser.stopTracing() to create a trace file that can be opened in Chrome DevTools performance panel.

Usage

```
await browser.startTracing(page, { path: 'trace.json' });
await page.goto('https://www.google.com');
await browser.stopTracing();
```

Arguments

• page Page (optional)

Optional, if specified, tracing includes screenshots of the given page.

- options Object (optional)
 - categories Array<string> (optional)

specify custom categories to use instead of default.

path string (optional)

A path to write the trace file to.

screenshots boolean (optional)

captures screenshots in the trace.

Returns

• Promise<void>

stopTracing

Added in: v1.11



This API controls <u>Chromium Tracing</u> which is a low-level chromium-specific debugging tool. API to control <u>Playwright Tracing</u> could be found <u>here</u>.

Returns the buffer with trace data.

Usage

```
await browser.stopTracing();
```

Returns

• Promise<Buffer>

version

Added before v1.9

Returns the browser version.

Usage

```
browser.version();
```

Returns

• string

Events

on('disconnected')

Added before v1.9

Emitted when Browser gets disconnected from the browser application. This might happen because of one of the following:

- Browser application is closed or crashed.
- The browser.close() method was called.

Usage

```
browser.on('disconnected', data => {});
```

Event data

• Browser