

[Home](#) > [puppeteer](#) > [Frame](#) > [goto](#)

Frame.goto() method

Signature:

```
goto(url: string, options?: {  
  referer?: string;  
  timeout?: number;  
  waitUntil?: PuppeteerLifeCycleEvent | PuppeteerLifeCycleEvent[];  
}): Promise<HTTPResponse | null>;
```

Parameters

Parameter	Type	Description
url	string	the URL to navigate the frame to. This should include the scheme, e.g. <code>https://</code> .
options	{ referer?: string; timeout?: number; waitUntil?: PuppeteerLifeCycleEvent PuppeteerLifeCycleEvent []; }	navigation options. <code>waitUntil</code> is useful to define when the navigation should be considered successful - see the docs for PuppeteerLifeCycleEvent for more details.

Returns:

Promise<[HTTPResponse](#) | null>

A promise which resolves to the main resource response. In case of multiple redirects, the navigation will resolve with the response of the last redirect.

Remarks

`frame.goto` will throw an error if: - there's an SSL error (e.g. in case of self-signed certificates).

- target URL is invalid.
- the `timeout` is exceeded during navigation.
- the remote server does not respond or is unreachable.
- the main resource failed to load.

`frame.goto` will not throw an error when any valid HTTP status code is returned by the remote server, including 404 "Not Found" and 500 "Internal Server Error". The status code for such responses can be retrieved by calling [HTTPResponse.status\(\)](#).

NOTE: `frame.goto` either throws an error or returns a main resource response. The only exceptions are navigation to `about:blank` or navigation to the same URL with a different hash, which would succeed and return `null`.

NOTE: Headless mode doesn't support navigation to a PDF document. See the [upstream issue](#).