

[Home](#) > [puppeteer](#) > [Page](#) > [setRequestInterception](#)

Page.setRequestInterception() method

Signature:

```
setRequestInterception(value: boolean): Promise<void>;
```

Parameters

Parameter	Type	Description
value	boolean	Whether to enable request interception.

Returns:

Promise<void>

Remarks

Activating request interception enables [HTTPRequest.abort\(\)](#), [HTTPRequest.continue\(\)](#) and [HTTPRequest.respond\(\)](#) methods. This provides the capability to modify network requests that are made by a page.

Once request interception is enabled, every request will stall unless it's continued, responded or aborted; or completed using the browser cache.

Example

An example of a naïve request interceptor that aborts all image requests:

```
const puppeteer = require('puppeteer');
(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.setRequestInterception(true);
  page.on('request', interceptedRequest => {
    if (interceptedRequest.url().endsWith('.png') ||
        interceptedRequest.url().endsWith('.jpg'))
      interceptedRequest.abort();
    else
      interceptedRequest.continue();
  });
  await page.goto('https://example.com');
  await browser.close();
})();
```

NOTE: Enabling request interception disables page caching.