Request

Whenever the page sends a request for a network resource the following sequence of events are emitted by Page:

- page.on("request") emitted when the request is issued by the page.
- page.on("response") emitted when/if the response status and headers are received for the request.
- page.on("requestfinished") emitted when the response body is downloaded and the request is complete.

If request fails at some point, then instead of `'requestfinished'` event (and possibly instead of 'response' event), the page.on("requestfailed") event is emitted.

> **NOTE**
>
> HTTP Error responses, such as 404 or 503, are still successful responses from HTTP standpoint, so request will complete with `'requestfinished'` event.

If request gets a 'redirect' response, the request is successfully finished with the `requestfinished` event, and a new request is issued to a redirected url.

Methods

all_headers

Added in: v1.15

An object with all the request HTTP headers associated with this request. The header names are lower-cased.

Usage

```
request.all_headers()
```

Returns

- Dict[str, str]

header_value

Added in: v1.15

Returns the value of the header matching the name. The name is case insensitive.

Usage

```
request.header_value(name)
```

Arguments

- `name` str

  Name of the header.

Returns

- NoneType|str

headers_array

Added in: v1.15

An array with all the request HTTP headers associated with this request. Unlike request.all_headers(), header names are NOT lower-cased. Headers with multiple entries, such as `Set-Cookie`, appear in the array multiple times.

Usage

```
request.headers_array()
```

Returns

- List[Dict]
  - `name` str

Name of the header.

- `value` str

  Value of the header.

response

Added in: v1.8

Returns the matching Response object, or `null` if the response was not received due to error.

Usage

```
request.response()
```

Returns

- NoneType|Response

sizes

Added in: v1.15

Returns resource size information for given request.

Usage

```
request.sizes()
```

Returns

- Dict
  - `requestBodySize` int

    Size of the request body (POST data payload) in bytes. Set to 0 if there was no body.

  - `requestHeadersSize` int

Total number of bytes from the start of the HTTP request message until (and including) the double CRLF before the body.

- `responseBodySize` int

  Size of the received response body (encoded) in bytes.

- `responseHeadersSize` int

  Total number of bytes from the start of the HTTP response message until (and including) the double CRLF before the body.

Properties

failure

Added in: v1.8

The method returns `null` unless this request has failed, as reported by `requestfailed` event.

Usage

Example of logging of all the failed requests:

```
page.on("requestfailed", lambda request: print(request.url + " " +
request.failure))
```

Returns

- NoneType|str

frame

Added in: v1.8

Returns the Frame that initiated this request.

Usage

```
frame_url = request.frame.url
```

Returns

- Frame

Details

Note that in some cases the frame is not available, and this method will throw.

- When request originates in the Service Worker. You can use `request.serviceWorker()` to check that.
- When navigation request is issued before the corresponding frame is created. You can use request.is_navigation_request() to check that.

Here is an example that handles all the cases:

headers

Added in: v1.8

An object with the request HTTP headers. The header names are lower-cased. Note that this method does not return security-related headers, including cookie-related ones. You can use request.all_headers() for complete list of headers that include `cookie` information.

Usage

```
request.headers
```

Returns

- Dict[str, str]

is_navigation_request

Added in: v1.8

Whether this request is driving frame's navigation.

Some navigation requests are issued before the corresponding frame is created, and therefore do not have request.frame available.

Usage

```
request.is_navigation_request()
```

Returns

- bool

method

Added in: v1.8

Request's method (GET, POST, etc.)

Usage

```
request.method
```

Returns

- str

post_data

Added in: v1.8

Request's post body, if any.

Usage

```
request.post_data
```

Returns

- NoneType|str

post_data_buffer

Added in: v1.8

Request's post body in a binary form, if any.

Usage

```
request.post_data_buffer
```

Returns

- NoneType|bytes

post_data_json

Added in: v1.8

Returns parsed request's body for `form-urlencoded` and JSON as a fallback if any.

When the response is `application/x-www-form-urlencoded` then a key/value object of the values will be returned. Otherwise it will be parsed as JSON.

Usage

```
request.post_data_json
```

Returns

- NoneType|Serializable

redirected_from

Added in: v1.8

Request that was redirected by the server to this one, if any.

When the server responds with a redirect, Playwright creates a new Request object. The two requests are connected by `redirectedFrom()` and `redirectedTo()` methods. When multiple server redirects has happened, it is possible to construct the whole redirect chain by repeatedly calling `redirectedFrom()`.

Usage

For example, if the website `http://example.com` redirects to `https://example.com`:

Sync Async

```
response = page.goto("http://example.com")
print(response.request.redirected_from.url) # "http://example.com"
```

If the website `https://google.com` has no redirects:

Sync Async

```
response = page.goto("https://google.com")
print(response.request.redirected_from) # None
```

Returns

- NoneType|Request

redirected_to

Added in: v1.8

New request issued by the browser if the server responded with redirect.

Usage

This method is the opposite of request.redirected_from:

```
assert request.redirected_from.redirected_to == request
```

Returns

- NoneType|Request

resource_type

Added in: v1.8

Contains the request's resource type as it was perceived by the rendering engine. ResourceType will be one of the following: `document`, `stylesheet`, `image`, `media`, `font`, `script`, `texttrack`, `xhr`, `fetch`, `eventsource`, `websocket`, `manifest`, `other`.

Usage

```
request.resource_type
```

Returns

- str

timing

Added in: v1.8

Returns resource timing information for given request. Most of the timing values become available upon the response, `responseEnd` becomes available when request finishes. Find more information at Resource Timing API.

Usage

Sync **Async**

```
with page.expect_event("requestfinished") as request_info:
    page.goto("http://example.com")
request = request_info.value
print(request.timing)
```

Returns

- Dict
  - `startTime` float

    Request start time in milliseconds elapsed since January 1, 1970 00:00:00 UTC

- `domainLookupStart` float

  Time immediately before the browser starts the domain name lookup for the resource. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `domainLookupEnd` float

  Time immediately after the browser starts the domain name lookup for the resource. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `connectStart` float

  Time immediately before the user agent starts establishing the connection to the server to retrieve the resource. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `secureConnectionStart` float

  Time immediately before the browser starts the handshake process to secure the current connection. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `connectEnd` float

  Time immediately before the user agent starts establishing the connection to the server to retrieve the resource. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `requestStart` float

  Time immediately before the browser starts requesting the resource from the server, cache, or local resource. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `responseStart` float

  Time immediately after the browser receives the first byte of the response from the server, cache, or local resource. The value is given in milliseconds relative to `startTime`, -1 if not available.

- `responseEnd` float

  Time immediately after the browser receives the last byte of the resource or immediately before the transport connection is closed, whichever comes first. The value is given in

milliseconds relative to `startTime`, -1 if not available.

url

Added in: v1.8

URL of the request.

Usage

```
request.url
```

Returns

- str