



# Response

[Response](#) class represents responses which are received by page.

## Methods

### **all\_headers**

Added in: v1.15

An object with all the response HTTP headers associated with this response.

#### **Usage**

```
response.all_headers()
```

#### **Returns**

- [Dict\[str, str\]](#)

### **body**

Added in: v1.8

Returns the buffer with response body.

#### **Usage**

```
response.body()
```

#### **Returns**

- [bytes](#)

# finished

Added in: v1.8

Waits for this response to finish, returns always `null`.

## Usage

```
response.finished()
```

## Returns

- `NoneType|str`

# header\_value

Added in: v1.15

Returns the value of the header matching the name. The name is case insensitive. If multiple headers have the same name (except `set-cookie`), they are returned as a list separated by `,`. For `set-cookie`, the `\n` separator is used. If no headers are found, `null` is returned.

## Usage

```
response.header_value(name)
```

## Arguments

- `name` `str`

Name of the header.

## Returns

- `NoneType|str`

# header\_values

Added in: v1.15

Returns all values of the headers matching the name, for example `set-cookie`. The name is case insensitive.

## Usage

```
response.header_values(name)
```

## Arguments

- `name` `str`

Name of the header.

## Returns

- `List[str]`

# headers\_array

Added in: v1.15

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

## Usage

```
response.headers_array()
```

## Returns

- `List[Dict]`
  - `name` `str`
  - `value` `str`

Name of the header.

Value of the header.

# json

Added in: v1.8

Returns the JSON representation of response body.

This method will throw if the response body is not parsable via `JSON.parse`.

## Usage

```
response.json()
```

## Returns

- `Serializable`

# security\_details

Added in: v1.13

Returns SSL and other security information.

## Usage

```
response.security_details()
```

## Returns

- `NoneType|Dict`

- `issuer` `str` (*optional*)

Common Name component of the Issuer field. from the certificate. This should only be used for informational purposes. Optional.

- `protocol` `str` (*optional*)

The specific TLS protocol used. (e.g. `TLS 1.3`). Optional.

- `subjectName` `str` (*optional*)

Common Name component of the Subject field from the certificate. This should only be used for informational purposes. Optional.

- `validFrom` `float` (*optional*)

Unix timestamp (in seconds) specifying when this cert becomes valid. Optional.

- `validTo` `float` (*optional*)

Unix timestamp (in seconds) specifying when this cert becomes invalid. Optional.

## server\_addr

Added in: v1.13

Returns the IP address and port of the server.

### Usage

```
response.server_addr()
```

### Returns

- `NoneType|Dict`

- `ipAddress` `str`

IPv4 or IPV6 address of the server.

- `port` `int`

## text

Added in: v1.8

Returns the text representation of response body.

### Usage

```
response.text()
```

## Returns

- `str`

# Properties

## frame

Added in: v1.8

Returns the `Frame` that initiated this response.

## Usage

```
response.frame
```

## Returns

- `Frame`

## from\_service\_worker

Added in: v1.23

Indicates whether this Response was fulfilled by a Service Worker's Fetch Handler (i.e. via `FetchEvent.respondWith`).

## Usage

```
response.from_service_worker
```

## Returns

- `bool`

## headers

Added in: v1.8

An object with the response 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 `response.all_headers()` for complete list of headers that include `cookie` information.

## Usage

```
response.headers
```

## Returns

- `Dict[str, str]`

# ok

Added in: v1.8

Contains a boolean stating whether the response was successful (status in the range 200-299) or not.

## Usage

```
response.ok
```

## Returns

- `bool`

# request

Added in: v1.8

Returns the matching `Request` object.

## Usage

```
response.request
```

## Returns

- [Request](#)

## status

Added in: v1.8

Contains the status code of the response (e.g., 200 for a success).

### Usage

```
response.status
```

### Returns

- [int](#)

## status\_text

Added in: v1.8

Contains the status text of the response (e.g. usually an "OK" for a success).

### Usage

```
response.status_text
```

### Returns

- [str](#)

## url

Added in: v1.8

Contains the URL of the response.

### Usage

```
response.url
```



## Returns

- `str`