APIRequestContext

This API is used for the Web API testing. You can use it to trigger API endpoints, configure micro-services, prepare environment or the service to your e2e test.

Each Playwright browser context has associated with it APIRequestContext instance which shares cookie storage with the browser context and can be accessed via browser_context.request or page.request. It is also possible to create a new APIRequestContext instance manually by calling api_request.new_context().

Cookie management

APIRequestContext returned by browser_context.request and page.request shares cookie storage with the corresponding BrowserContext. Each API request will have `Cookie` header populated with the values from the browser context. If the API response contains `Set-Cookie` header it will automatically update BrowserContext cookies and requests made from the page will pick them up. This means that if you log in using this API, your e2e test will be logged in and vice versa.

If you want API requests to not interfere with the browser cookies you should create a new APIRequestContext by calling api_request.new_context(). Such `APIRequestContext` object will have its own isolated cookie storage.

Sync **Async**

```
import os
from playwright.sync_api import sync_playwright

REPO = "test-repo-1"
USER = "github-username"
API_TOKEN = os.getenv("GITHUB_API_TOKEN")

with sync_playwright() as p:
    # This will launch a new browser, create a context and page. When making HTTP
    # requests with the internal APIRequestContext (e.g. `context.request` or
`page.request`)
    # it will automatically set the cookies to the browser page and vice versa.
    browser = p.chromium.launch()
    context = browser.new_context(base_url="https://api.github.com")
    api_request_context = context.request
    page = context.new_page()

    # Alternatively you can create a APIRequestContext manually without having a browser context attached:
    # api_request_context = p.request.new_context(base_url="https://api.github.com")


    # Create a repository.
    response = api_request_context.post(
        "/user/repos",
        headers={
            "Accept": "application/vnd.github.v3+json",
            # Add GitHub personal access token.
            "Authorization": f"token {API_TOKEN}",
        },
        data={"name": REPO},
    )
    assert response.ok
    assert response.json()["name"] == REPO

    # Delete a repository.
    response = api_request_context.delete(
        f"/repos/{USER}/{REPO}",
        headers={
            "Accept": "application/vnd.github.v3+json",
            # Add GitHub personal access token.
            "Authorization": f"token {API_TOKEN}",
        },
    )
    assert response.ok
    assert await response.body() == '{"status": "ok"}'
```

Methods

delete

Added in: v1.16

Sends HTTP(S) DELETE request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects.

Usage

```
api_request_context.delete(url)
api_request_context.delete(url, **kwargs)
```

Arguments

- `url` str

  Target URL.

- `data` str|bytes|Serializable *(optional)* Added in: v1.17

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set. Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

  Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)* Added in: v1.17

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)* Added in: v1.17

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

    File content

  Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

  Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

dispose

Added in: v1.16

All responses returned by api_request_context.get() and similar methods are stored in the memory, so that you can later call api_response.body(). This method discards all stored responses, and makes api_response.body() throw "Response disposed" error.

Usage

```
api_request_context.dispose()
```

fetch

Added in: v1.16

Sends HTTP(S) request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects. JSON objects can be passed directly to the request.

Usage

```
data = {
    "title": "Book Title",
    "body": "John Doe",
}
api_request_context.fetch("https://example.com/api/createBook", method="post",
data=data)
```

The common way to send file(s) in the body of a request is to encode it as form fields with `multipart/form-data` encoding. You can achieve that with Playwright API like this:

```
api_request_context.fetch(
  "https://example.com/api/uploadScrip'",
  method="post",
  multipart={
    "fileField": {
      "name": "f.js",
      "mimeType": "text/javascript",
      "buffer": b"console.log(2022);",
    },
  })
```

Arguments

- `url_or_request` str|Request

  Target URL or Request to get all parameters from.

- `data` str|bytes|Serializable *(optional)*

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set.

Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

  Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)*

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `method` str *(optional)*

  If set changes the fetch method (e.g. PUT or POST). If not specified, GET method is used.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)*

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

File content

Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

  Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

get

Added in: v1.16

Sends HTTP(S) GET request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects.

Usage

Request parameters can be configured with `params` option, they will be serialized into the URL search parameters:

```
query_params = {
  "isbn": "1234",
  "page": "23"
}
api_request_context.get("https://example.com/api/getText", params=query_params)
```

Arguments

- `url` str

  Target URL.

- `data` str|bytes|Serializable *(optional)* Added in: v1.26

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set. Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

  Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)* Added in: v1.26

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)* Added in: v1.26

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

    File content

  Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

  Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

head

Added in: v1.16

Sends HTTP(S) HEAD request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects.

Usage

```
api_request_context.head(url)
api_request_context.head(url, **kwargs)
```

Arguments

- `url` str

  Target URL.

- `data` str|bytes|Serializable *(optional)* Added in: v1.26

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set.

Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

  Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)* Added in: v1.26

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)* Added in: v1.26

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

    File content

  Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

  Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

patch

Added in: v1.16

Sends HTTP(S) PATCH request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects.

Usage

```
api_request_context.patch(url)
api_request_context.patch(url, **kwargs)
```

Arguments

- `url` str

  Target URL.

- `data` str|bytes|Serializable *(optional)*

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set. Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)*

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)*

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

    File content

  Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

post

Added in: v1.16

Sends HTTP(S) POST request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects.

Usage

JSON objects can be passed directly to the request:

```
data = {
    "title": "Book Title",
    "body": "John Doe",
}
api_request_context.post("https://example.com/api/createBook", data=data)
```

To send form data to the server use `form` option. Its value will be encoded into the request body with `application/x-www-form-urlencoded` encoding (see below how to use `multipart/form-data` form encoding to send files):

```
formData = {
    "title": "Book Title",
    "body": "John Doe",
}
api_request_context.post("https://example.com/api/findBook", form=formData)
```

The common way to send file(s) in the body of a request is to upload them as form fields with `multipart/form-data` encoding. You can achieve that with Playwright API like this:

```
api_request_context.post(
  "https://example.com/api/uploadScrip'",
  multipart={
    "fileField": {
      "name": "f.js",
      "mimeType": "text/javascript",
      "buffer": b"console.log(2022);",
    },
  })
```

Arguments

- `url` str

  Target URL.

- `data` str|bytes|Serializable *(optional)*

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set. Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

  Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)*

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)*

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

    File content

  Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

  Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

put

Added in: v1.16

Sends HTTP(S) PUT request and returns its response. The method will populate request cookies from the context and update context cookies from the response. The method will automatically follow redirects.

Usage

```
api_request_context.put(url)
api_request_context.put(url, **kwargs)
```

Arguments

- `url` str

  Target URL.

- `data` str|bytes|Serializable *(optional)*

  Allows to set post data of the request. If the data parameter is an object, it will be serialized to json string and `content-type` header will be set to `application/json` if not explicitly set. Otherwise the `content-type` header will be set to `application/octet-stream` if not explicitly set.

- `fail_on_status_code` bool *(optional)*

  Whether to throw on response codes other than 2xx and 3xx. By default response object is returned for all status codes.

- `form` Dict[str, str|float|bool] *(optional)*

  Provides an object that will be serialized as html form using `application/x-www-form-urlencoded` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `application/x-www-form-urlencoded` unless explicitly provided.

- `headers` Dict[str, str] *(optional)*

  Allows to set HTTP headers. These headers will apply to the fetched request as well as any redirects initiated by it.

- `ignore_https_errors` bool *(optional)*

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

- `max_redirects` int *(optional)* Added in: v1.26

  Maximum number of request redirects that will be followed automatically. An error will be thrown if the number is exceeded. Defaults to `20`. Pass `0` to not follow redirects.

- `multipart` Dict[str, str|float|bool|[ReadStream]|Dict] *(optional)*

  - `name` str

    File name

  - `mimeType` str

    File type

  - `buffer` bytes

    File content

  Provides an object that will be serialized as html form using `multipart/form-data` encoding and sent as this request body. If this parameter is specified `content-type` header will be set to `multipart/form-data` unless explicitly provided. File values can be passed either as `fs.ReadStream` or as file-like object containing file name, mime-type and its content.

- `params` Dict[str, str|float|bool] *(optional)*

  Query parameters to be sent with the URL.

- `timeout` float *(optional)*

  Request timeout in milliseconds. Defaults to `30000` (30 seconds). Pass `0` to disable timeout.

Returns

- APIResponse

storage_state

Added in: v1.16

Returns storage state for this request context, contains current cookies and local storage snapshot if it was passed to the constructor.

Usage

```
api_request_context.storage_state()
api_request_context.storage_state(**kwargs)
```

Arguments

- `path` Union[str, pathlib.Path] *(optional)*

  The file path to save the storage state to. If `path` is a relative path, then it is resolved relative to current working directory. If no path is provided, storage state is still returned, but won't be saved to the disk.

Returns

- Dict
  - `cookies` List[Dict]
    - `name` str
    - `value` str
    - `domain` str
    - `path` str
    - `expires` float

      Unix time in seconds.
    - `httpOnly` bool
    - `secure` bool
    - `sameSite` "Strict"|"Lax"|"None"
  - `origins` List[Dict]
    - `origin` str
    - `localStorage` List[Dict]
      - `name` str
      - `value` str