

Networking

Jetzt wird's interaktiv

Einführung

Interfaces:

- Response
- Request

Functions:

- status()
- allHeaders()
- text()

Einführung

Hier eine Einführung zu den Netzwerk-Aktivitäten

```
1 // 11_test_runner/tests/01_intro/intro.test.ts
2 import { test, expect } from '@playwright/test';
3
4 test("Response / Request", async ({ page }) => {
5
6     const response = await page.goto('');
7
8     if (response === null) return;
9
10    console.log(response.url());
11    console.log(response.status());
12    console.log(response.ok());
13
14    expect(response.ok()).toBeTruthy();
15
16    console.log(await response.allHeaders());
17    console.log(await response.headersArray());
18
19    console.log(await response.body());
20
21 });
```

API-Tests mit Github

Nun weiter mit einem etwas aufwändigeren Beispiel:

```
1 // 11_test_runner/tests/02_api/api.test.ts
2
3 import { test, expect } from '@playwright/test';
4
5 const REPO = 'play';
6 const USER = 'dpinezich';
7
8 test('should create a bug report', async ({ request }) => {
9   const newIssue = await request.post(`/repos/${USER}/${REPO}/issues`, {
10     data: {
11       title: '[Bug] report 1',
12       body: 'Bug description',
13     }
14   });
15   expect(newIssue.ok()).toBeTruthy();
```

API-Tests mit Github

Nun weiter mit einem etwas aufwändigeren Beispiel:

```
1 // 11_test_runner/playwright.config.test
2
3 import { defineConfig } from '@playwright/test';
4 export default defineConfig({
5   use: {
6     // All requests we send go to this API endpoint.
7     baseURL: 'https://api.github.com',
8     extraHTTPHeaders: {
9       // We set this header per GitHub guidelines.
10       'Accept': 'application/vnd.github.v3+json',
11       // Add authorization token to all requests.
12       // Assuming personal access token available in the environment.
13       'Authorization': `token xxx`,
14     },
15   }
16 });
```

Ende

Das war alles für dieses Kapitel
