ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Playwright Workshop

4 - Advanced II

Agenda

- Miscellaneous
- Network stubbing
- Session storage
- Project Dependencies

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Miscellaneous / Step

- Suited for larger Tests
- Element to Structure Tests
- Asynchronous

ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Miscellaneous / Step

```
test('order holiday', async ({ page }) => {
  await test.step('select holiday', async () => {
    await page.getByRole('link', { name: 'Holidays', exact: true }).click();
    // …
  });

  await test.step('order brochure', async () => {
    // …
  });
});
```

Miscellaneous / Soft Assertion

- For Assertions (expect)
- Continues the test on failed assertion
- Test is still marked as failed
- Useful when multiple assertions are used
- Customised expect with soft enabled possible

Miscellaneous / Soft Assertion (Flavour I)

```
test('order holiday', async ({ page }) => {
  await expect.soft(page.getByText('Welcome to Eternal')).toBeVisible();
  await page.getByRole('link', { name: 'Holidays', exact: true }).click();
  await expect
    .soft(page.getByText('Holidays', { exact: true }))
    .toBeVisible();

  // ...
});
```

Miscellaneous / Soft Assertion (Flavour II)

```
import { expect as baseExpect, test } from '@playwright/test';


const expect = baseExpect.configure({ soft: true });


test('order holiday', async ({ page }) => {
  await expect(page.getByText('Welcome to Eternal')).toBeVisible();
  await page.getByRole('link', { name: 'Holidays', exact: true }).click();
  await expect(page.getByText('Holidays', { exact: true })).toBeVisible();


 // ...
});
```

ANGULAR ARCHITECTS
INSIDE KNOWLEDGE

Miscellaneous / Sequential (Serial)

- `test.describe.configure({ mode: 'serial' });`
- Edge Case
- Tests (complete file) run in sequential order
- Complete sequence stops on failure
- Retried altogether
- Runs in same worker process (?)
  - allows for state between tests

Miscellaneous / Failing

- ```
  test.fail('feature not implemented yet')
  ```

- ```
  test.fail('failing because it is weekend', () => new Date().getDay() === 0);
  ```
  - Conditional Version

Network stubbing

- Intercept the network
- Verify requests
- Abort them
- Modify them
- Response: main use case

Use case: logging

- No impact on the application at all
-
```
page.on("request", (req) => console.log(req.url()));
```

Use case: Waiting for a certain request

```
test('holidays should be called', async ({ page }) => {
  const holidaysRequest = page.waitForResponse((response) => {
    return response.url() === 'https://api.eternal-holidays.net/holiday';
  });
  await page.goto('');
  await page.getByTestId('btn-holidays').click();
  await expect(page.getByTestId('holiday-card')).toHaveCount(9);
  await holidaysRequest;
});
```

Use case: Requesting from test itself (API Test)

```
test('should count the holidays and verify', async ({ page, request }) => {
  const response = await request.get(
    'https://api.eternal-holidays.net/holiday'
  );
  expect(response.ok()).toBeTruthy();
  const holidays = (await response.json()) as unknown[];
  const holidayCount = holidays.length;

  await page.goto('');
  await page.getByRole('link', { name: 'Holidays', exact: true }).click();
  await expect(page.getByTestId('holiday-card')).toHaveCount(holidayCount);
});
```

Use case: stubbing

```
page.route("https://api.eternal-holidays.net/holidays", (route) =>
  route.fulfill({
    status: 200,
     json: [
      {
        id: 1,
        title: "Cambodia",
        teaser: "Discover old temples and the old Khmer Empire",
        imageUrl:
          "https://eternal-app.s3.eu-central-1.amazonaws.com/assets/AngkorWatSmall.jpg",
        description:
          "Cambodia is known for Angkor Wat which is the most visited attraction in whole Asia. Started out
as an Hindu temple, it was gradually transformed to a Buddhist one. Beware, it can be very humid there.",
      },
    ],
  })
);
```


Demo

Caching authentication

1. Global script that runs one time at the start
2. Signs in
3. Persists localStorage, cookies in a file
4. Reload on demand

Browser Context: Storing

```
await page.context().storageState({ path: 'session.json' });
```

- Persists Cookies and localStorage
- Multiple States possible (via separate files)
- Usually executed in
  - Global Setup Function
  - Setup Project Type

Browser Context: Loading

- Project-Level
  - ```
    {
      name: 'chromium',
      use: { ...devices['Desktop Chrome'], storageState: 'session.json' }
    }
    ```
- Test-Level
  - ```
    test.use({ storageState: 'session.json' });
    ```


Demo

Project Types

- Project Types can depend on each other
- Use Cases
  - Setup
  - Teardown
- "Not-Main" Types usually run with a single browser

Project Dependencies

```
projects: [
  {
    name: 'smoke',
    use: { browserName: 'chromium' },
    testMatch: 'smoke.ts',
  },
  {
    name: 'chromium',
    use: { ...devices['Desktop Chrome'] },
    dependencies: ['smoke'],
  }
]
```


Lab Time