Implementing automated UI testing for SharePoint solutions

Elio Struyf, Valo Intranet
A Huge “Thank You!” To Our Sponsors ...
Testing your code
Code analysis

• Manual
• JSLint / TSLint → Define code/analysis rules
• Static analysis like SonarQube
Unit Tests

```
21:43:14  /nodejs/spfx/spfx-testing-jest  / master  npm test

> testing@0.0.1 test /Users/eliostruyf/nodejs/spfx/spfx-testing-jest
> jest

PASS  src/webparts/helloWorld/tests/HelloWorld.test.tsx
<HelloWorld />
  ✓ Should do something (30ms)
  ✓ <HelloWorld /> should render something (6ms)
  ✓ <HelloWorld /> should render the description (5ms)
  ✓ <HelloWorld /> should render an ul (4ms)
  ✓ <HelloWorld /> state results should not be null (3ms)
  ✓ <HelloWorld /> should call componentDidMount only once (3ms)
  ✓ <HelloWorld /> should render an ul with 3 items (using the mock data) (1008ms)
  ✓ <HelloWorld /> should render 10 list items (triggering setState from the test) (8ms)

Test Suites: 1 passed, 1 total
Tests: 8 passed, 8 total
Snapshots: 0 passed, 0 total
Time: 2.786s
Ran all test suites.
```
Visual UI Tests
Automated UI Tests

- Unit tests validate your code, UI tests validate that it actually works
- Test your solution functionality
- Test out time consuming actions
- Check if your solution gets rendered correctly

- Use tools like:
  - Coded UI Tests in Visual Studio
  - Selenium
  - Robot Framework
  - ...

Puppeteer

• High-level API to control Chrome / Chromium
• Runs Chrome in headless mode (can be non-headless)
• Full access over the DevTools Protocol
• Can automate all manual browsers steps
  • Navigating
  • Taking screenshots / generating a PDF
  • Clicking on links / buttons / menu / ...
• Documentation: https://github.com/GoogleChrome/puppeteer
• Install: npm i puppeteer
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://www.eliostruyf.com');
  await page.screenshot({path: 'example.png', fullPage: true});

  await browser.close();
})();
Make tests easier with a recorder: https://github.com/checkly/puppeteer-recorder
Authenticate against SharePoint
Authenticate against SharePoint

• Go through the whole authentication process
• Use a library like: `node-sp-auth`
  • https://www.npmjs.com/package/node-sp-auth

```typescript
import * as spauth from 'node-sp-auth';
const data = await spauth.getAuth(pageUrl, {
  username: username,
  password: password
});
```
Finding differences in your pictures
What changed between today and yesterday
Pixel matching

https://github.com/mapbox/pixelmatch

```javascript
pixelmatch(img1.data, img2.data, diff.data, img1.width, img1.height, { threshold: 0.1 });
```
Jest testing framework

- Zero configuration testing platform
- Testing JavaScript based applications
- Created by Facebook

Install: `npm install jest @types/jest -D`
Combining Puppeteer and Jest

test('Should load the page', async (done) => {
  if (page) {
    await page.goto(configObj.pageUrl, {
      waitUntil: 'networkidle0'
    });
  }

  expect(page).not.toBeNull();
  expect(await page.title()).not.toBeNull();
  expect(await page.title()).toBe("Automated UI Tests - Home");

  done();
});
Combining Puppeteer and Jest

```javascript
/**
 * Start of the other page tests
 */

test('Check if caption element is present in the web part', async (done) => {
    const elm = await page.$('.caption');
    expect(elm).not.toBeNull();
    done();
}, 5000);
```
Wait!
It’s not so easy!
Solution: work with data-* attributes
<div className={styles.caption} data-ui-test-id="caption">
  <p data-ui-test-id="caption-title">
    <b>{item.title}</b>
  </p>
  <p data-ui-test-id="caption-description">
    <i>{item.description}</i>
  </p>
</div>
Simple element retrieval

/**
 * Start of the other page tests
 */

test('Check if caption element is present in the web part', async (done) => {
  const caption = await page.$('div[data-ui-test-id="caption"]');
  expect(caption).not.toBeNull();
  done();
}, 5000);
What’s next?
Next steps

• Where to run this?
  • Docker
  • Website

• Link it in your Azure DevOps pipelines

• Automated reports to Teams
  • …
Questions?
Elio Struyf
Architect

@eliostruyf
www.eliostruyf.com
info@estruyf.be

Office Servers & Services MVP
Azure / Office 365 / SharePoint
Slides

https://www.eliostruyf.com/speaking