



Tests E2E con TestCafé



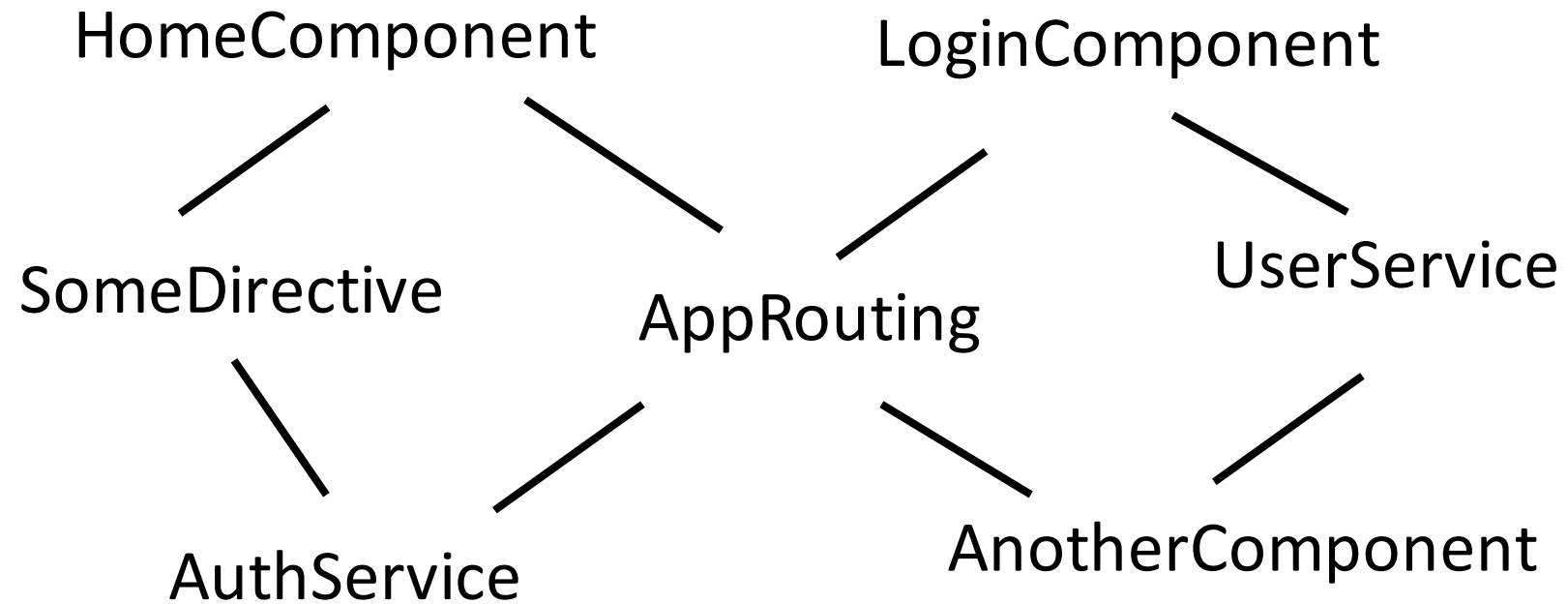
Jesus Rodriguez

<http://angular-tips.com>

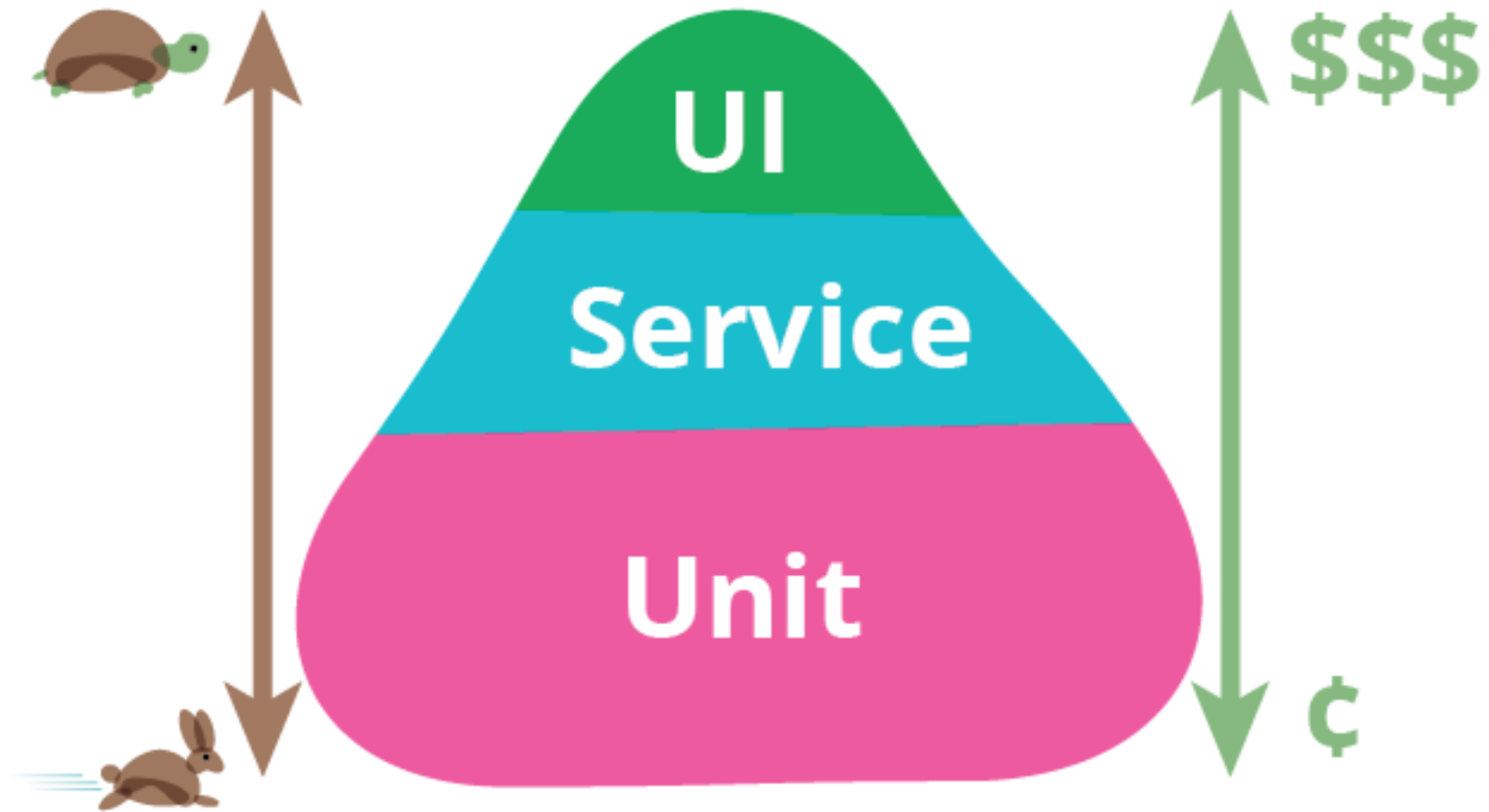
@foxandxss

¿Qué es End to End testing?

End 2 End



La pirámide de testing. M. Fowler



Selenium webdriver

- Es el framework “de facto” para hacer E2E.
- Se comunica con el navegador usando un driver específico.
- Permite ejecutar cualquier cosa que ejecutaría el usuario.
- Puede ser ejecutado en aplicaciones propias o de terceros.

Protractor

- Protractor es un wrapper de selenium para AngularJS.
- Se puede usar igualmente en Angular.

Selenium / Protractor – Lo malo

- Es lento.
- Es muy frágil.
- Te hará llorar por las noches.
- Y también durante el día.
- Necesita de muchos muchos hacks.
- Protractor no se actualiza lo que debería.
- Pasan de arreglar mucho de sus fallos.

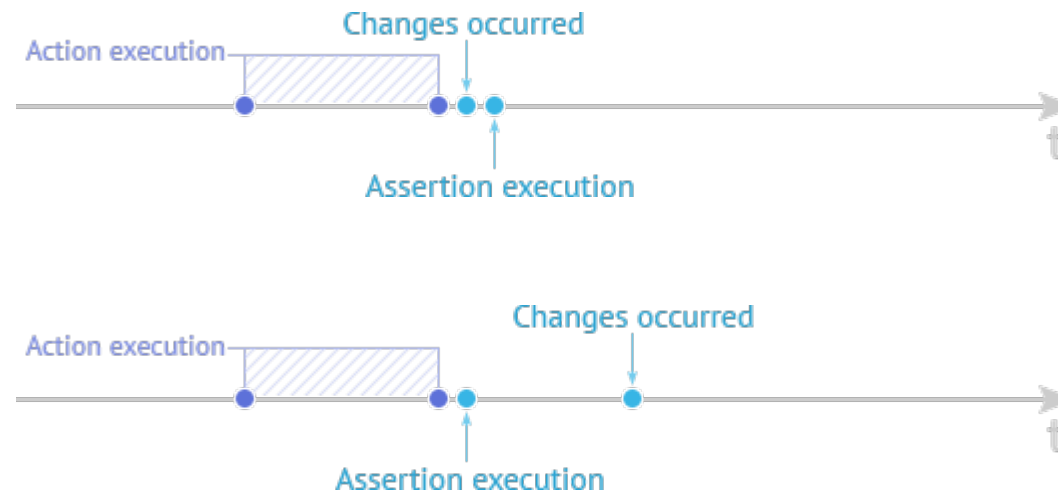


Pete Bacon Darwin 🍷 22:04

The failures are usually due to “flakes” in the E2E testing infrastructure. This has been a perpetual pain for us, since before Protractor was even a thing (~2014?).

¿Cuál es el problema principal?

Simplemente... No quiere esperar a nada



¿Cuál es el problema principal?

```
static WaitForElement(elementToWait: ElementFinder, errorMessage?: string) {  
    return browser.wait(  
        ExpectedConditions.visibilityOf(elementToWait),  
        30000,  
        `${errorMessage} || 'Element' taking too long to appear in the DOM`  
    );  
}  
  
static WaitForText(...) {}  
static WaitForElementToHide(...) {}  
static WaitForDxGridToLoad(...) {}  
static WaitForUrlToContain(...) {}  
static WaitForUrlNotToContain(...) {}
```

¿Cuál es el problema principal?

```
it('redirects to the facility device list by clicking the button', async () => {  
  const data = await facilityListPage.getRowData();  
  browser  
    .actions()  
    .mouseMove(data.rowEl)  
    .perform();  
  Utils.WaitForElement(element(By.css('app-toolbar')));  
  const button = element.all(By.css('app-toolbar button')).get(0);  
  button.click().then(() => {  
    Utils.WaitForDxGridToLoad();  
    Utils.WaitForText(deviceListPage.getFacilityName(), data.facilityName);  
    expect(deviceListPage.getFacilityName().getText()).toContain(data.facilityName);  
  });  
});
```

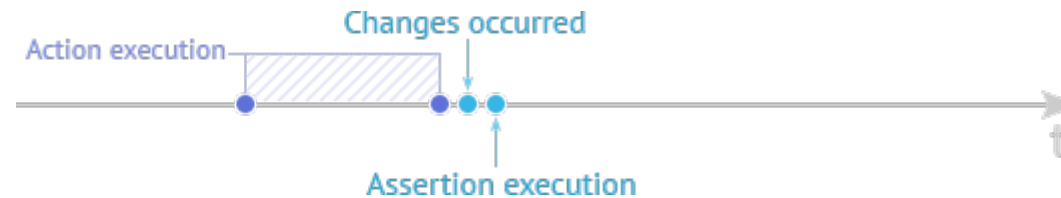
```
<div *ngIf="!isExpanded" class="mat-panel-title-text">Devices for:  
  {{selectedFacility ? selectedFacility.facilityDescription : "All Facilities" }}  
</div>
```

TestCafé

- Es un framework E2E open source creado por DevExpress.
- Sintaxis muy parecida a Selenium.
- 100% compatible con await / async y TypeScript.
- Sistema de Roles.
- Ejecución remota de tests.
- No más hacks 😊

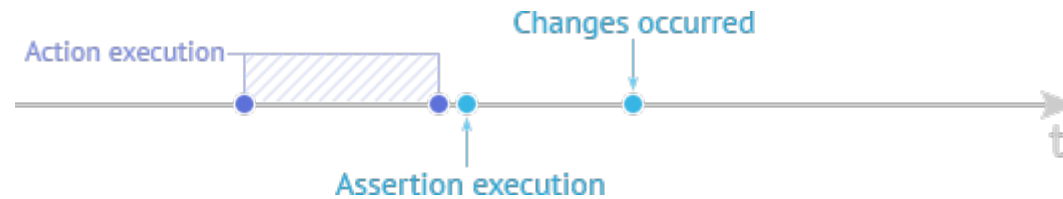
TestCafé - Smart Assertion Query Mechanism

En un mundo ideal (o síncrono)...



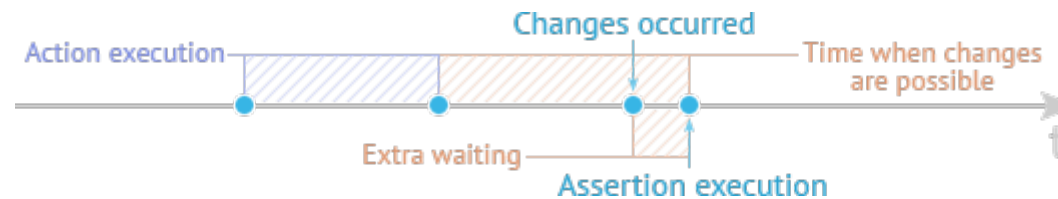
TestCafé - Smart Assertion Query Mechanism

Pero nunca es ideal (ni asíncrono)



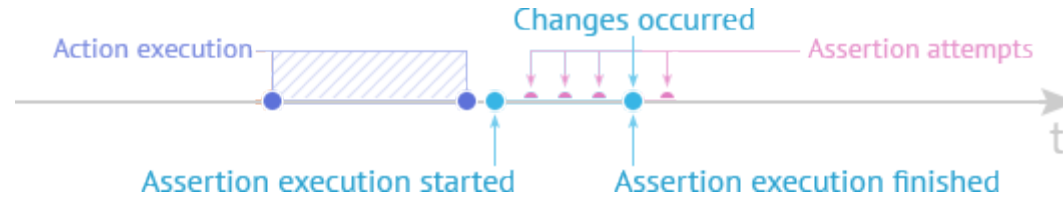
TestCafé - Smart Assertion Query Mechanism

Siempre se le puede añadir un timeout...



TestCafé - Smart Assertion Query Mechanism

TestCafé prefiere preguntar más veces



Comparando tests...

```
it('redirects to the facility device list by clicking the button', async () => {  
  const data = await facilityListPage.getRowData();  
  browser  
    .actions()  
    .mouseMove(data.rowEl)  
    .perform();  
  Utils.WaitForElement(element(By.css('app-toolbar')));  
  const button = element.all(By.css('app-toolbar button')).get(0);  
  button.click().then(() => {  
    Utils.WaitForDxGridToLoad();  
    Utils.WaitForText(deviceListPage.getFacilityName(), data.facilityName);  
    expect(deviceListPage.getFacilityName().getText()).toContain(data.facilityName);  
  });  
});
```

```
test('redirects to the facility device list by clicking the button', async t => {  
  const data = await facilityListPage.getRowData();  
  await t.hover(data.row)  
    .click(facilityListPage.deviceListButton)  
    .expect(deviceListPage.facilityName.innerText).contains(data.facilityName);  
});
```



Sistema de Roles

```
import { Role } from 'testcafe';

const regularAccUser = Role('http://example.com/login', async t => {
  await t
    .typeText('#login', 'TestUser')
    .typeText('#password', 'testpass')
    .click('#sign-in');
});

const facebookAccUser = Role('http://example.com/login', async t => {
  await t
    .click('#sign-in-with-facebook')
    .typeText('#email', 'testuser@mycompany.com')
    .typeText('#pass', 'testpass')
    .click('#submit');
});

const admin = Role('http://example.com/login', async t => {
  await t
    .typeText('#login', 'Admin')
    .typeText('#password', 'adminpass')
    .click('#sign-in');
});
```

- Crear diferentes roles para nuestro sitio
- Podemos cambiar entre roles en cualquier momento
- En cada rol se guardarán sus propias cookies

Manejo de errores

*** wont stay in a private page after logout**

- 1) A call to an async function is not awaited. Use the "await" keyword before actions, assertions or chains of them to ensure that they run in the right sequence.**

Browser: Chrome 67.0.3396 / Mac OS X 10.13.5

```
33 | t
34 |   .useRole(adminUser)
35 |   .click(navPage.navItemsLoggedIn.bet)
36 |   .expect(betPage.header.exists).ok()
37 |   .click(navPage.navItemsLoggedIn.logout)
> 38 |   .expect(betPage.header.exists).notOk();
39 | });
40 |
41 | test('contains a link for betting in the home page after
login', async t => {
42 |   await t
43 |     .expect(homePage.betButton.exists).notOk()

at Object.<anonymous>
```

Manejo de errores

* wont stay in a private page after logout

1) Action "selector" argument error:

Selector is expected to be initialized with a function, CSS selector string, another Selector, node snapshot or a Promise returned by a Selector, but undefined was passed.

Browser: Chrome 67.0.3396 / Mac OS X 10.13.5

```
32 | test('wont stay in a private page after logout', async t => {
33 |   await t
34 |     .useRole(adminUser)
35 |     .click(navPage.navItemsLoggedIn.bet)
36 |     .expect(betPage.header.exists).ok()
> 37 |     .click(navPage.navItemsLoggedIn.login)
38 |     .expect(betPage.header.exists).notOk();
39 | });
40 |
41 | test('contains a link for betting in the home page after login', async t => {
42 |   await t
```

Documentación

- <https://devexpress.github.io/testcafe/>