

Lesson 03 Demo 01

Test the HTML Forms

Objective: To test HTML forms in Cypress

Tools required: Visual Studio Code

Prerequisites: Angular

Steps to be followed:

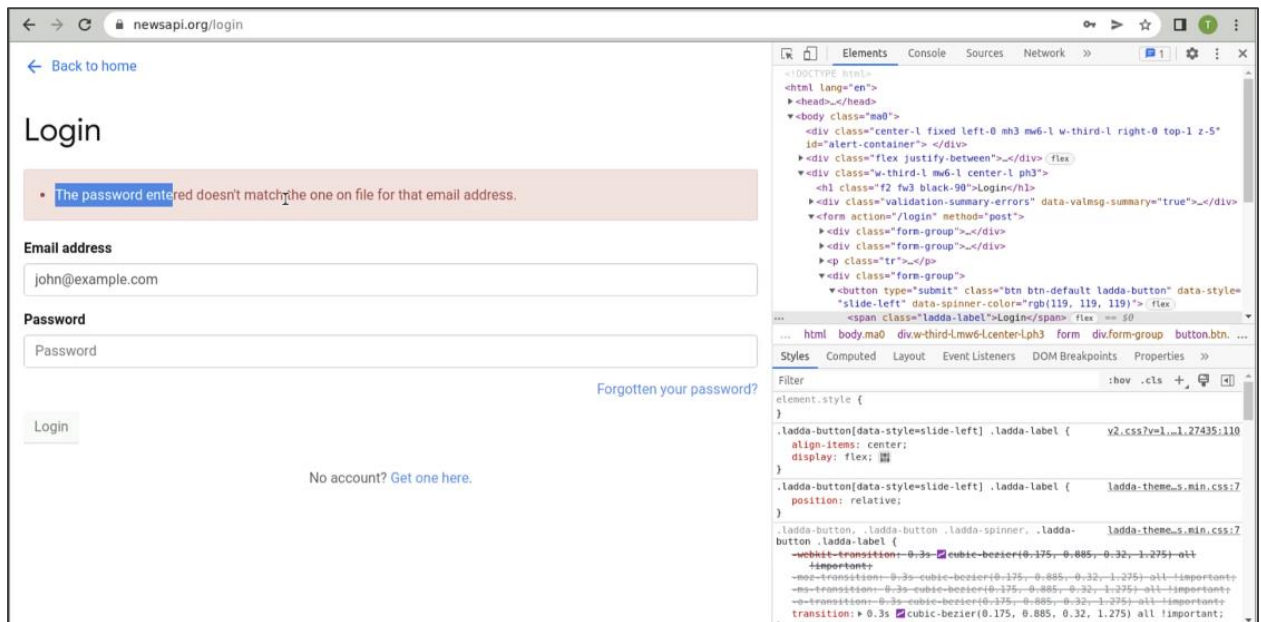
1. Test the HTML forms in the project
2. Change the test condition

Step 1: Test the HTML forms in the project

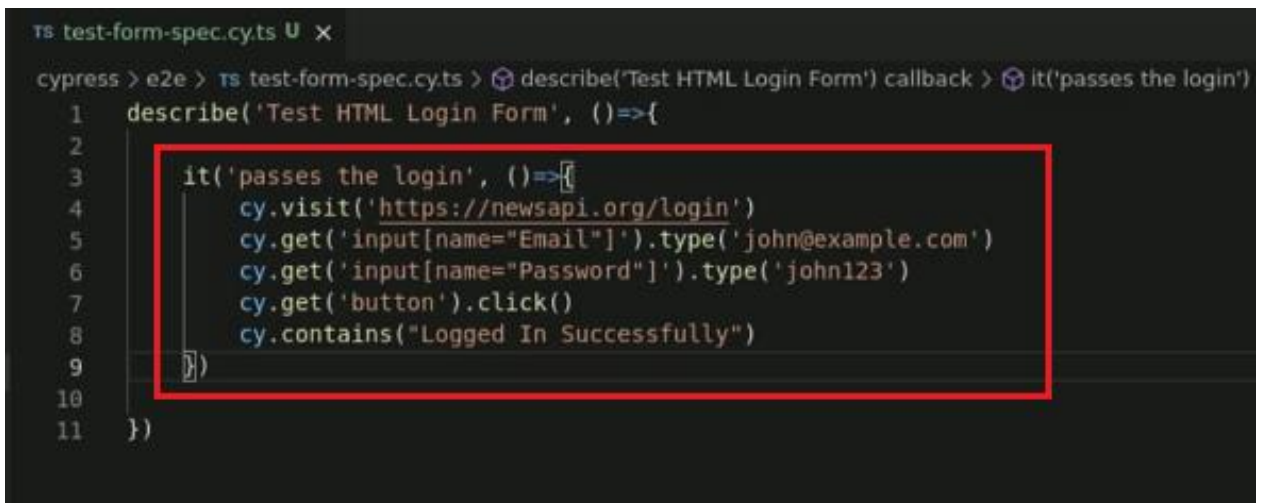
- 1.1 In the previously created **CYPRESSAPP**, create a new spec file **test-form-spec.cy.ts** and write **describe** function in it

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'CYPRESSAPP'. The 'e2e' folder is expanded, showing files 'search.cy.ts', 'spec.cy.ts', and 'test-form-spec.cy.ts'. The file 'test-form-spec.cy.ts' is selected and highlighted with a red rectangle. On the right, the editor window shows the content of 'test-form-spec.cy.ts'. It contains a single line of code: `describe('Test HTML Login Form', ()=>{`. A red rectangle highlights this line and the beginning of the next line, which is `it('')`. A tooltip is visible over the `it` function, showing its signature: `it(title: string, config: Cypress.TestConfigOverrides, fn?: Mocha.Func | undefined): Mocha.Test`. Below the signature, a description reads: 'Describe a specification or test-case with the given title, TestOptions, and callback fn acting as a thunk.'

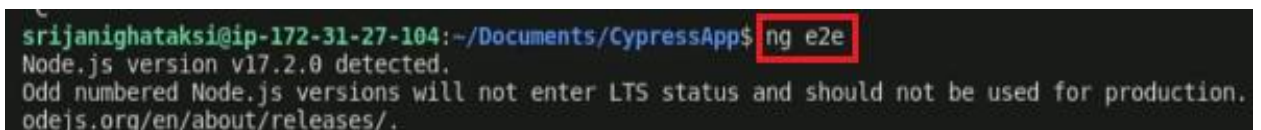
1.2 Visit **newsapi.org/login**. Enter some details in the form and copy the required **inspect** element details.



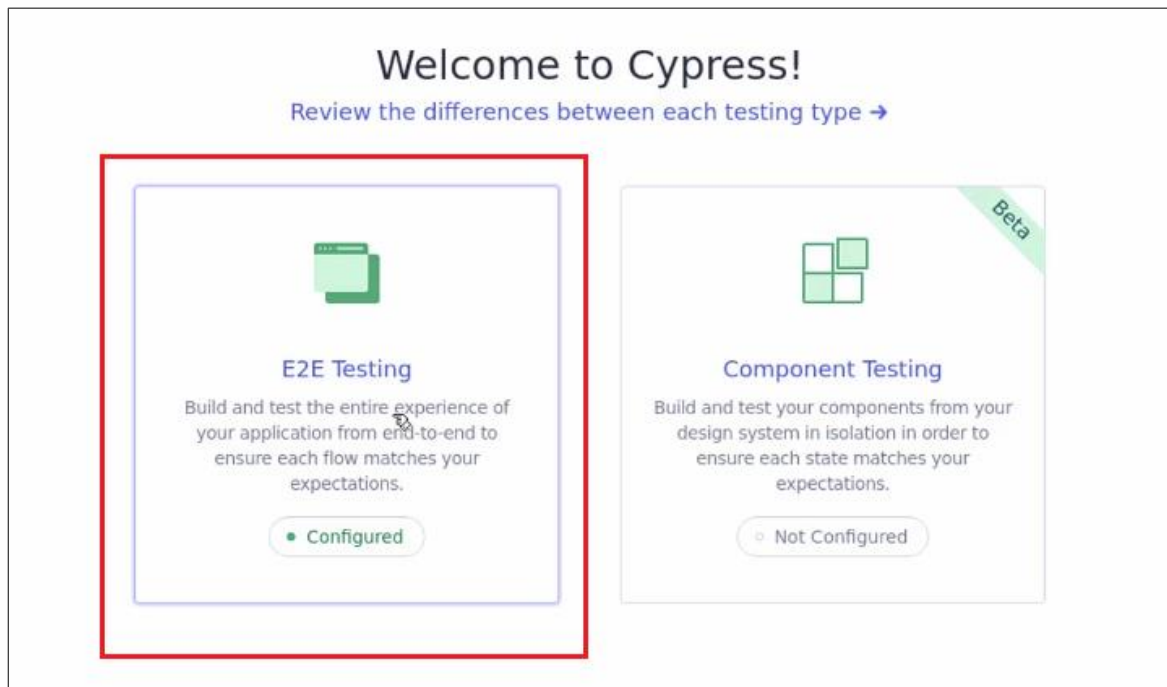
1.3 Paste the details in the test case for the complete form



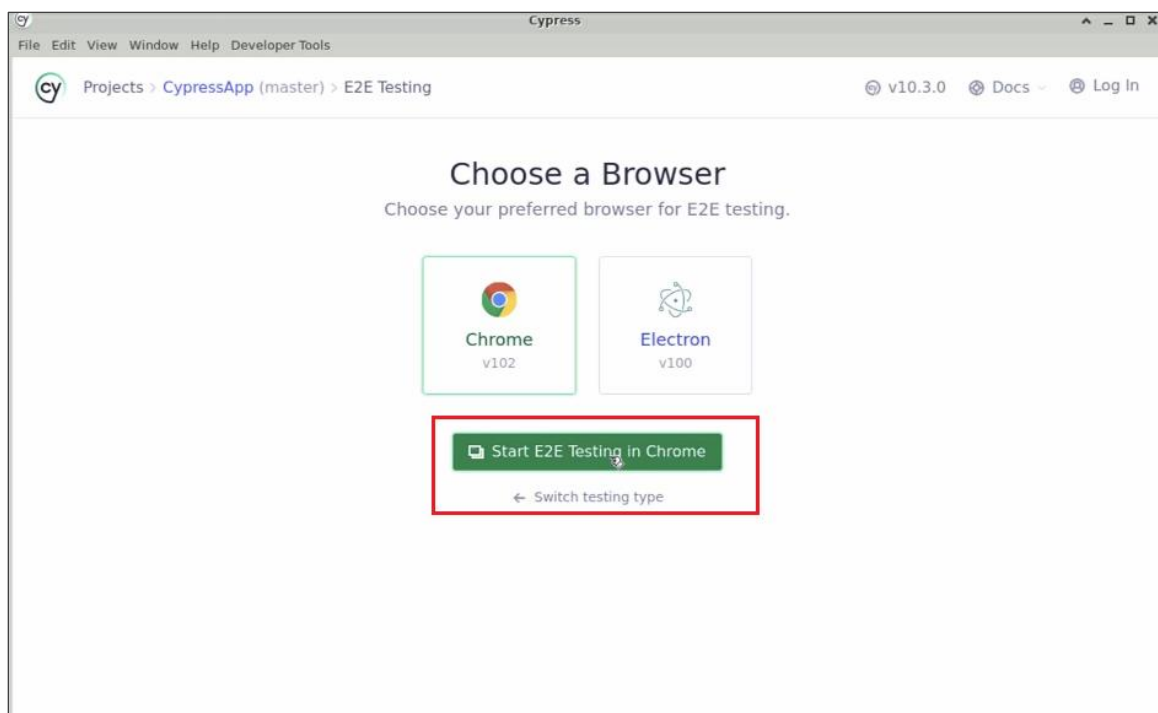
1.4 Run **ng e2e** command in the terminal to open the Cypress Launchpad



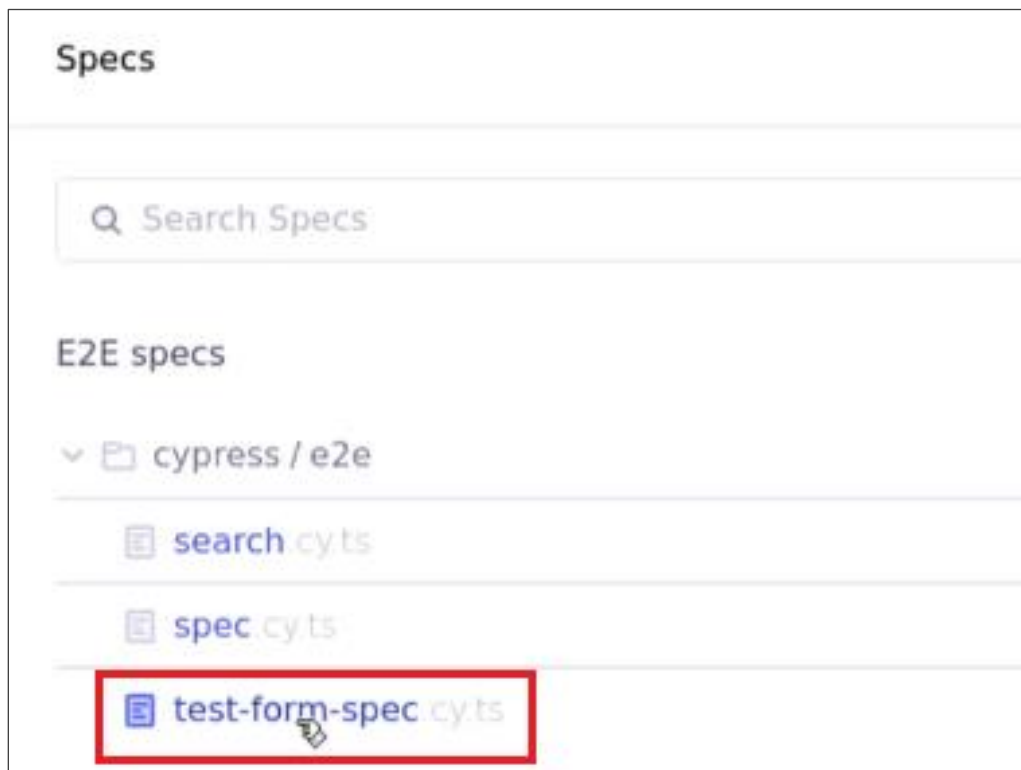
1.5 Click on **E2E Testing** in the popup



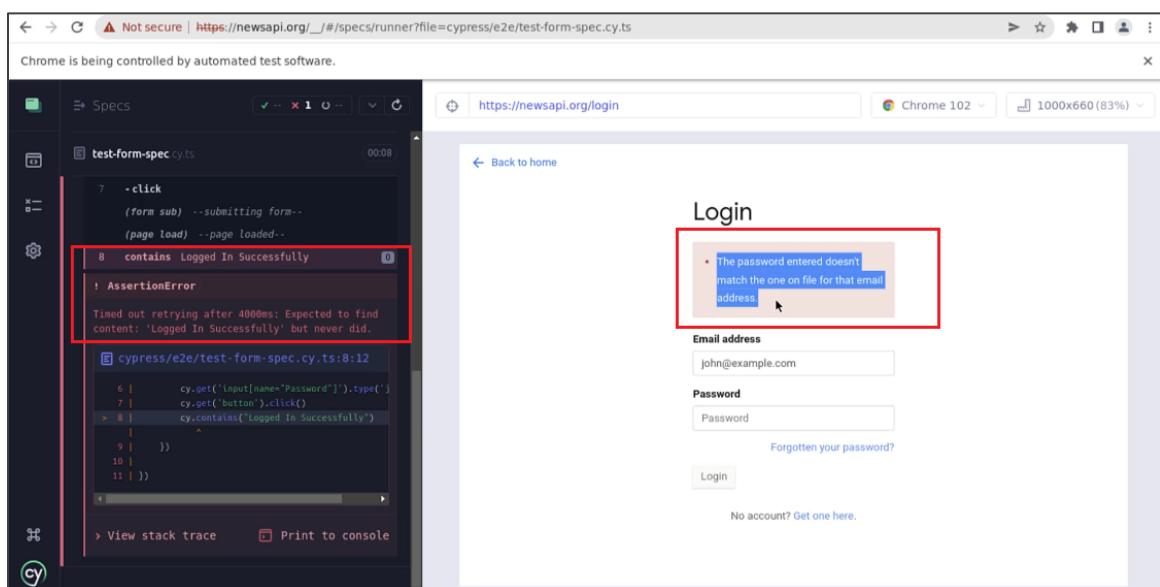
1.6 Click on **Start E2E Testing in Chrome** button



1.7 Select test-form-spec.cy.ts



1.8 The test case fails as the pop-up message is not the same.



Step 2: Change the test condition

2.1 Mark the previous contains as a comment. Change the condition as shown in the given screenshot:

```

1 describe('Test HTML Login Form', () => {
2
3   it('passes the login', () => {
4     cy.visit('https://newsapi.org/login')
5     cy.get('input[name="Email"]').type('john@example.com')
6     cy.get('input[name="Password"]').type('john123')
7     cy.get('button').click()
8     //cy.contains("Logged In Successfully")
9     cy.contains("The password entered doesn't match the one on file for that email address.")
10  })
11 })
12

```

contains(content: string | number | RegExp, options?: Partial<Cypress.Loggable & Cypress.Timeoutable & Cypress.CaseMatchable & Cypress.Shadow> | undefined): Cypress.Chainable<undefined>

Get the DOM element containing the text. DOM elements can contain more than the desired text and still match. Additionally, Cypress prefers some DOM elements over the deepest element found.

@see — <https://on.cypress.io/contains>

2.2 Check the test result in the Cypress Launchpad

Specs

test-form-spec.cy.ts 00:02

- Test HTML Login Form
 - passes the login

TEST BODY

```

1 visit https://newsapi.org/login
2 get input[name="Email"]
3 - type john@example.com

```

(xhr) POST 200 //collect? v=16_v=j966a=4510359356t=pageview6_s=1&dl=http s://newsapi.org/login&ul=en-us&de=UTF-8&dt=Login - News API&sd=24-bit&sr=1440x789&vp=1000x660&je=06_u=YEBAAUABAA AAAC-6jid=268560456&gid=203290136cid=858509336 .16575944306tid=UA-91285317-56_gid=669120351.16575944306_r=16gtm=2ou7b06z=165916656

(xhr) POST 200 https://www.google-analytics.com/j/collect? v=16_v=j966a=4510359356t=pageview6_s=1&dl=http s%3A%2F%2Fnewsapi.org%2Flogin&ul=en-us&de=UTF-8&dt=Login%20-%20News%20API&sd=24-bit&sr=1440x789&vp=1000x660&je=06_u=YEBAAUABAA AAAC-6jid=268560456&gid=203290136cid=858509336 .16575944306tid=UA-91285317-56_gid=669120351.16575944306_r=16gtm=2ou7b06z=165916656

https://newsapi.org/login Chrome 102

Back to home

Login

- The password entered doesn't match the one on file for that email address.

Email address
john@example.com

Password
Password

[Forgotten your password?](#)

Login

No account? [Get one here.](#)