**CYPRESS**

Task:

* Study and Install Cypress
* Write test case for login website and test in cypress

Cypress is an open-source and free test automation tool written in [*JavaScript*](https://www.toolsqa.com/javascript/what-is-javascript/). It can be used to test front end and perform API Testing as well. Cypress provides a visual interface to indicate what all tests and which all commands are running, passed, or failed.

It allows us to test highly interactive applications and carry out different tests such as manipulating the DOM, asserting that if some element is available or present on the screen, reading or writing data into/from fields, submitting forms and even redirecting to a different page without actually making direct modifications to your code.

## How to install cypress?

**Prerequisites:**

1. Node Js installed
2. Visual studio code or vim installed

**Steps:**

**1.**Create a folder

mkdir cypress

cd cypress

**2**. Set up a Node project using **npm init -y** command using the terminal. we will get the package.json file created with default values.

**3.** Open the **package.json** file in the root of the project and update the scripts property to the following:

"scripts": {

"test": "npx cypress run",

"cypress:open": "npx cypress open" },

Save and close the file, then in the root folder of the project, create a file called **cypress.json**, which is the configuration file where we can customise Cypress’ behaviour for this specific project. Add the following code to the file and save it:

{ "chromeWebSecurity": false }

**3.** Run the below command to install cypress

npm install cypress

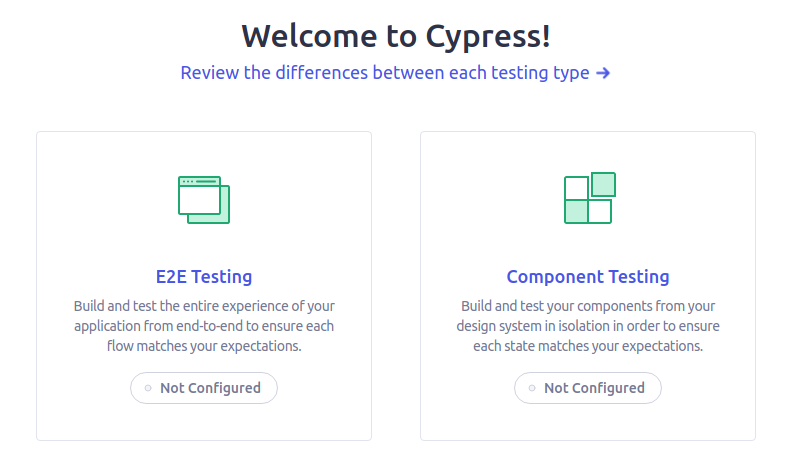
4. Now run the command to open the cypress

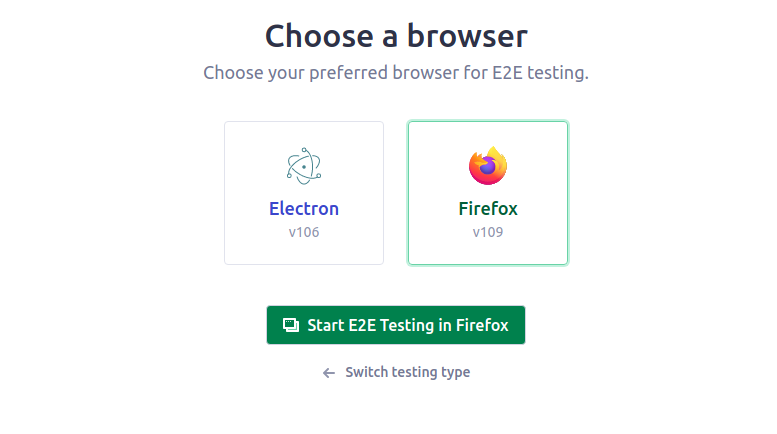
npx cypress open

This command will create a folder structure and will automatically open a cypress runner window.

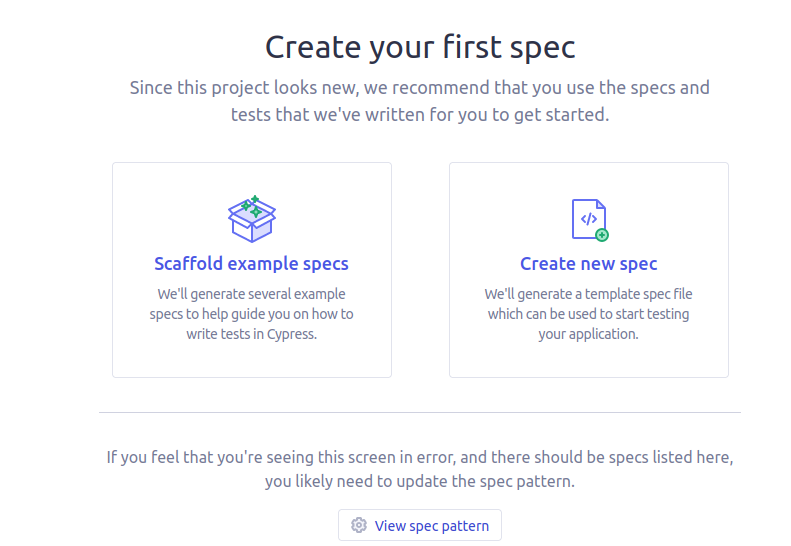
In this runner window we need to configure two types of tests

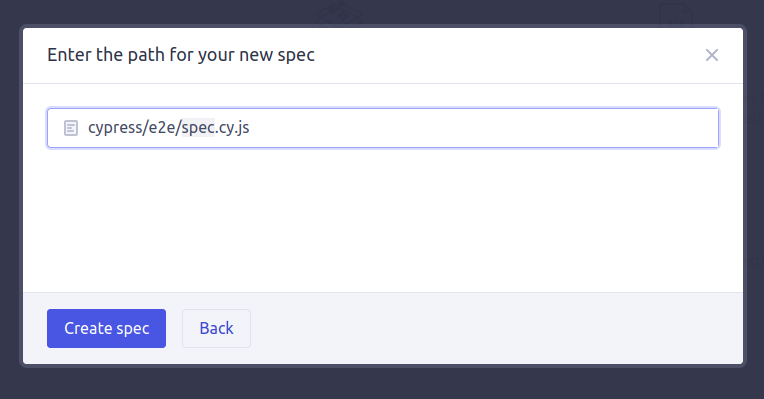
1. End to End test configuration
2. Component level test configuration.



When we click configure then we will get the option to run tests on different browsers. Select any browser. 

Click create new spec file. It will have a default example test . just click create spec.



****

Once configured those configurations will get added in the folder section.

Let's look what are the uses of folders created:

**Fixtures--->** Fixtures folder is used to store test data.

**Supports---->** supports folder is used to write and store reusable code and use it in our tests.

**e2e--->**This folder contains all the end to end test files.

**downloads-->**This folder is automatically created when we test downloads any file and stores all the downloaded files.

If we need to run the tests, we can run the tests in the test runner itself using the below browsers:

* Chrome
* Electron
* Edge
* Firefox

Test Case:

it('mp login check', () => {

cy.visit('<https://mp.mygov.in>');

cy.get('.en > .language-link').click();

cy.contains('Login').click();

cy.get('#edit-name').type('yogendra.yadav@fosteringlinux.com', {delay: 100})

cy.get('#edit-pass').type('password', { log: false })

cy.get('#edit-submit').click();

cy.get('.views-field-name > .field-content').click();

cy.wait(500)

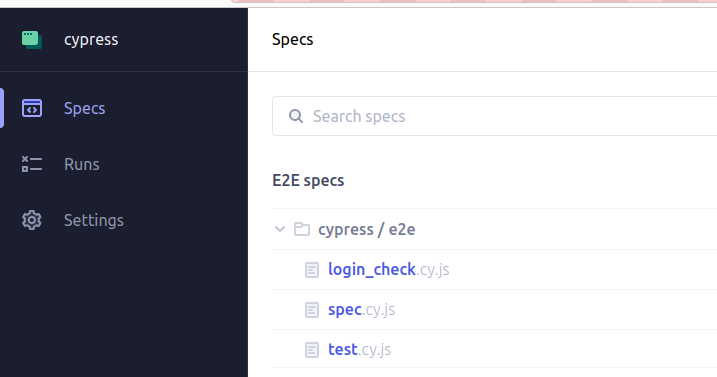
cy.get('.no-bdr > .capital').click();

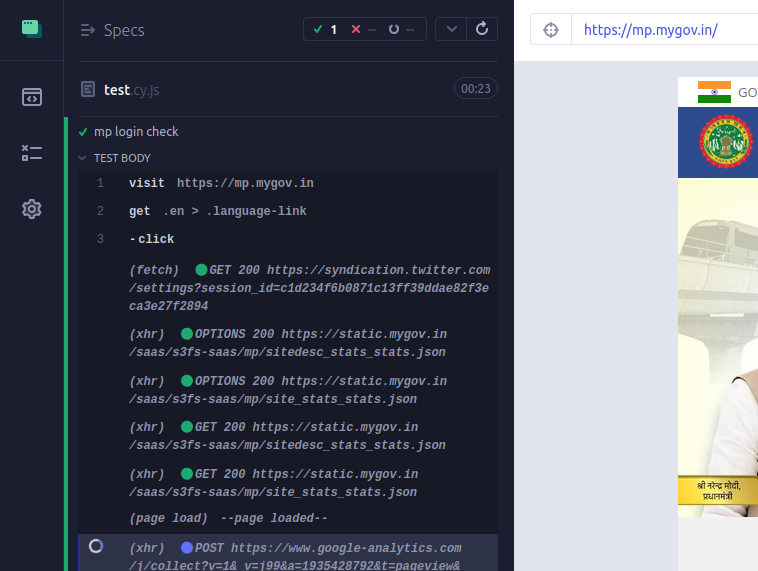
cy.wait(100)

cy.get('#edit-submit').click();

})

To test the test case open the cypress and click specs and select the code file. This will run the testing of the code.





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*