Getting started with Cypress

A complete step by step guide
by **M Haseeb Jamil**

LinkedIn Profile Link

What is Cypress

Cypress is a JavaScript-based end-to-end testing tool designed for modern web test automation. This developer-friendly tool operates directly in the browser using a DOM manipulation technique and enables front-end developers and QA engineers to write automated web tests while eliminating pain points.

Prerequisite:

Install any javascript base IDE:

- We can install any JavaScript IDE to write our script e.g., VS Code ,Sublime ,Web Storm.

Here is the link of VS Code installation.

- **VS Code Installation link** for Mac, Windows, & Linux.

Install Node.js:

- **Node.js installation link** for Mac, Windows, & Linux.

Note:

Follow the given first **2 steps** only if you want to Install Cypress in the **Standalone Project** otherwise you can skip them and start from step **3** if you are integrating it into the **Front-end Project.**

Stepwise guide

Open Terminal and move to your workspace

1. Create Project Directory

```
$ mkdir CypressAutomation
```

```
$ cd CypressAutomation
```

2. Install package.json

Write command “npm init -y” to download package.json.

After installing type **yes** if shows in terminal **y/n**.

```
$ npm init -y
```

```
$ yes
```

3. Installing Cypress

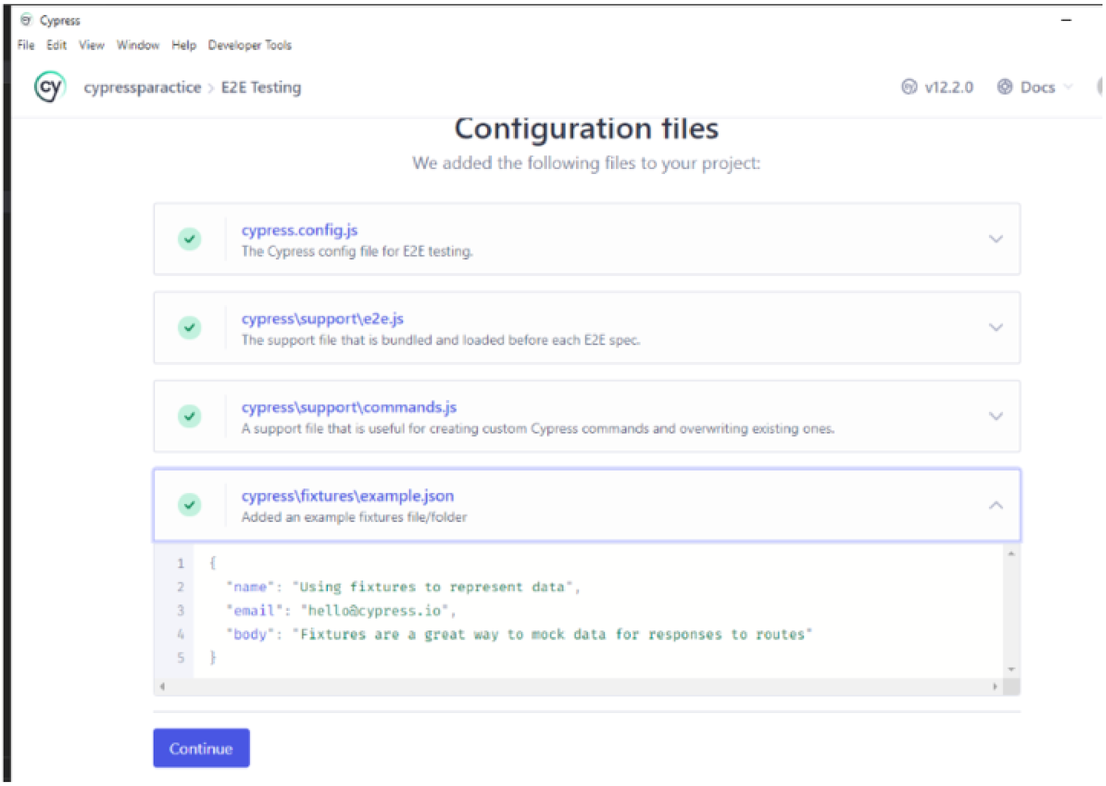
```
$ npm install cypress --save-dev
```

4. Opening Cypress

```
$ npx cypress open
```

Step 3:

- Select all the configuration files if these are not selected by default and click on the **Continue** button.


Cypress
File Edit View Window Help Developer Tools
cypresspractice › E2E Testing
v12.2.0
Docs
Continue

Configuration files

We added the following files to your project:

cypress.config.js
The Cypress config file for E2E testing.

cypress\support\e2e.js
The support file that is bundled and loaded before each E2E spec.

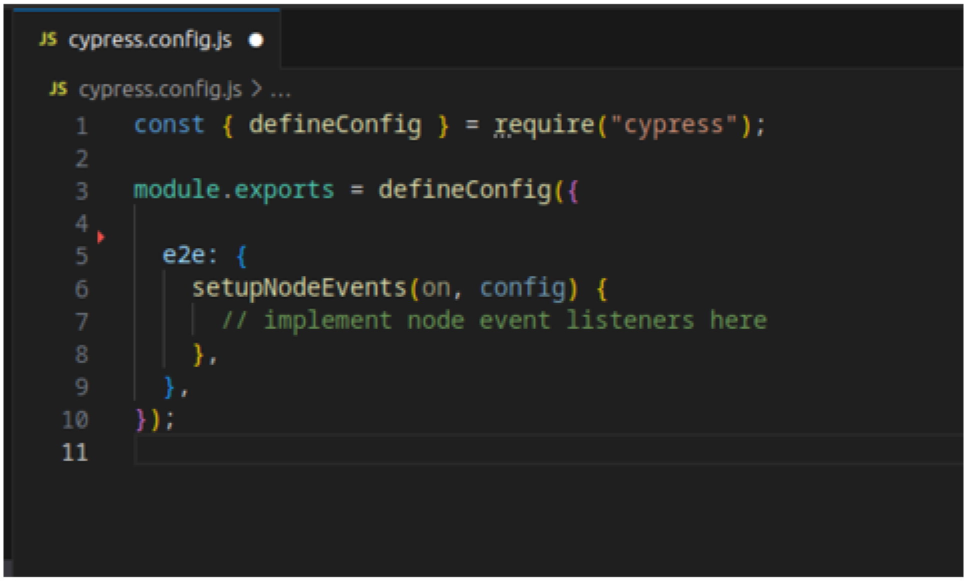
cypress\support\commands.js
A support file that is useful for creating custom Cypress commands and overwriting existing ones.

cypress\fixtures\example.json
Added an example fixtures file/folder

```
{
  "name": "Using fixtures to represent data",
  "email": "hello@cypress.io",
  "body": "Fixtures are a great way to mock data for responses to routes"
}
```

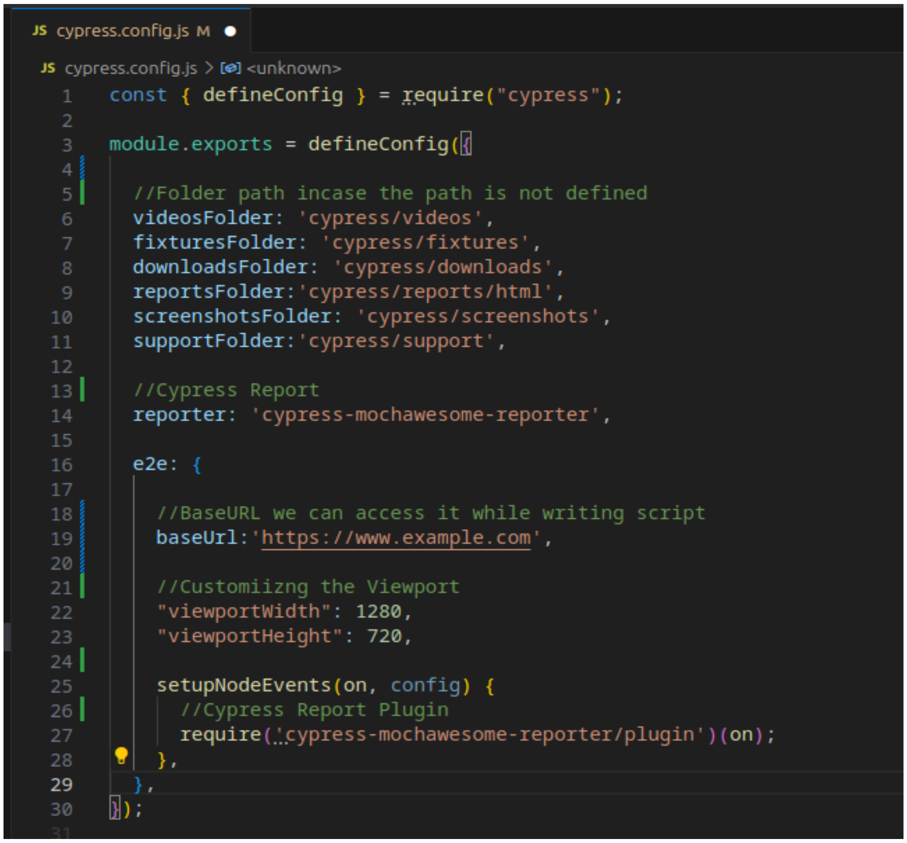
Step 4: Configuration File

- This is the Configuration file by default and you can customize it according to your requirements.



```
const { defineConfig } = require("cypress");

module.exports = defineConfig({

  e2e: {
    setupNodeEvents(on, config) {
      // implement node event listeners here
    },
  },
});
```

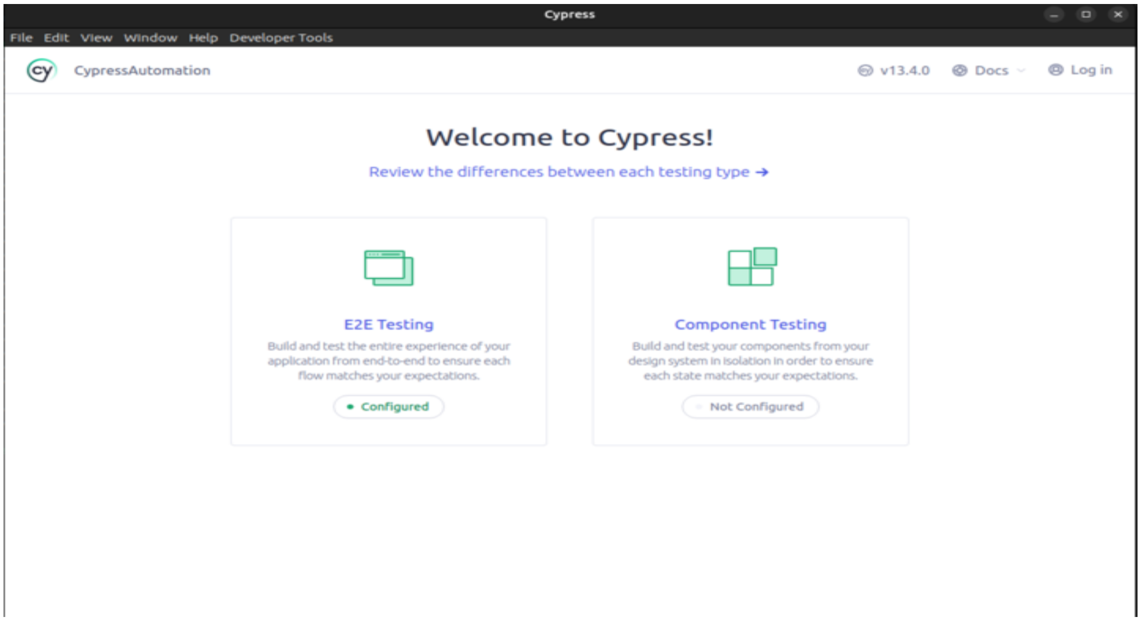
- Here is an example of **Customized Config File**



```
const { defineConfig } = require("cypress");

module.exports = defineConfig({

  //Folder path incase the path is not defined
  videosFolder: 'cypress/videos',
  fixturesFolder: 'cypress/fixtures',
  downloadsFolder: 'cypress/downloads',
  reportsFolder:'cypress/reports/html',
  screenshotsFolder: 'cypress/screenshots',
  supportFolder:'cypress/support',

  //Cypress Report
  reporter: 'cypress-mochawesome-reporter',

  e2e: {

    //BaseURL we can access it while writing script
    baseUrl:'https://www.example.com',

    //Customiizng the Viewport
    "viewportWidth": 1280,
    "viewportHeight": 720,

    setupNodeEvents(on, config) {
      //Cypress Report Plugin
      require('cypress-mochawesome-reporter/plugin')(on);
    },
  },
});
```

Step 5:

Click on **E2E Testing**



Welcome to Cypress!

Review the differences between each testing type →

E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

● Configured

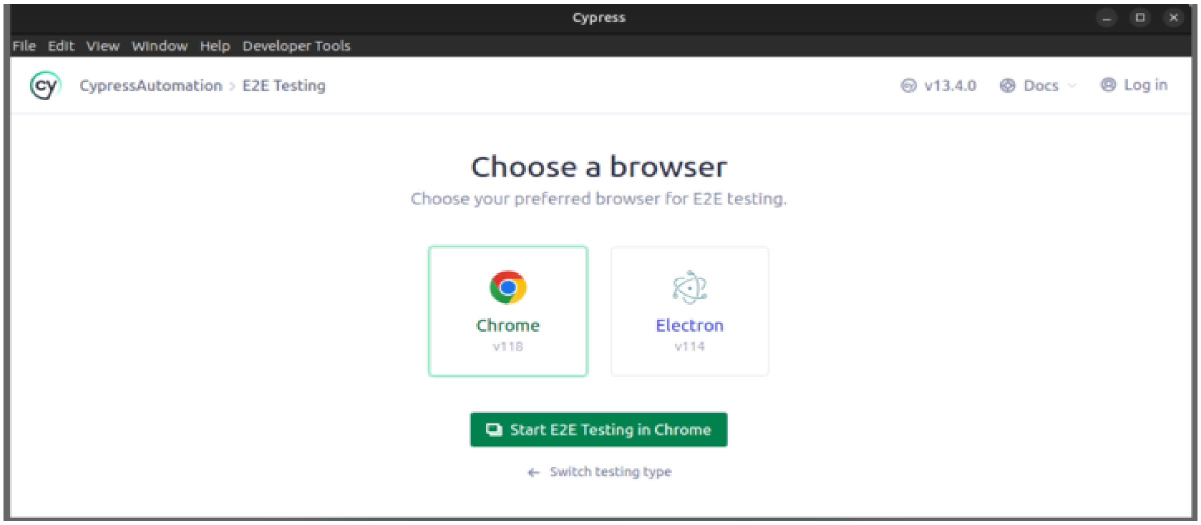
Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

Not Configured

Step 6: Choose a browser:

- Select the desired browser and click on **Start E2E Testing**


Cypress
File Edit View Window Help Developer Tools
CypressAutomation > E2E Testing
v13.4.0
Docs
Log in

Choose a browser

Choose your preferred browser for E2E testing.

Chrome
v118

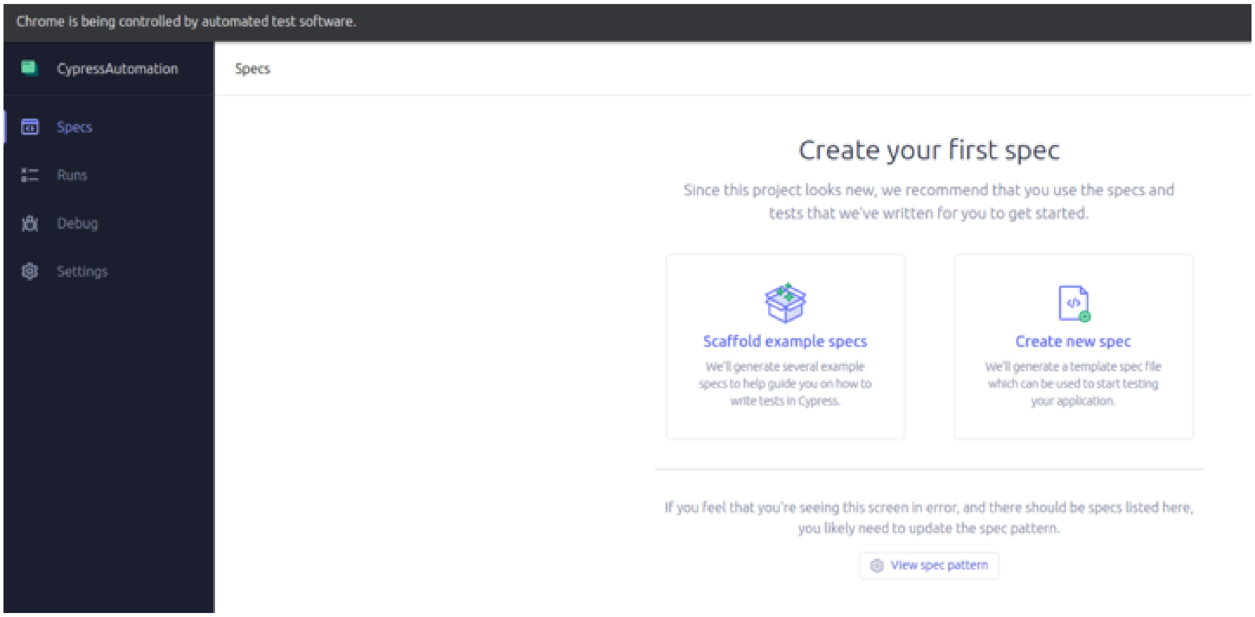
Electron
v114

Start E2E Testing in Chrome

← Switch testing type

Step 7: Create First Spec:

- Click on the **Create new spec.**



Chrome is being controlled by automated test software.

Create your first spec

Since this project looks new, we recommend that you use the specs and tests that we've written for you to get started.

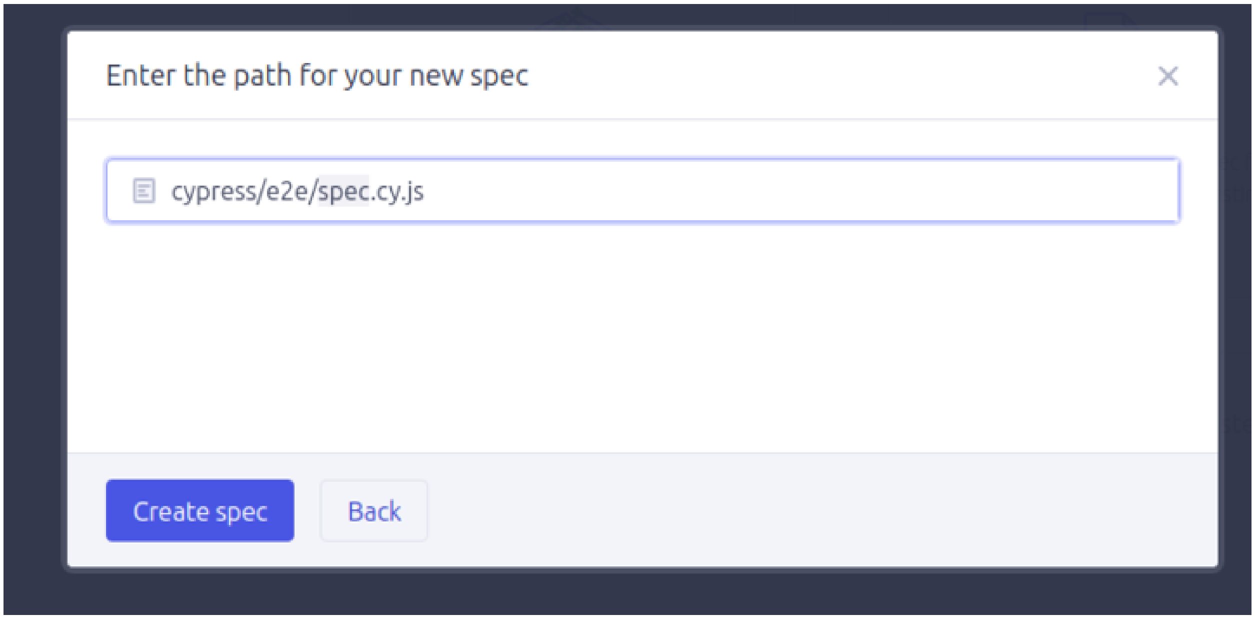
We'll generate several example specs to help guide you on how to write tests in Cypress.

We'll generate a template spec file which can be used to start testing your application.

If you feel that you're seeing this screen in error, and there should be specs listed here, you likely need to update the spec pattern.

Step 8:

- Click on the **Create spec**

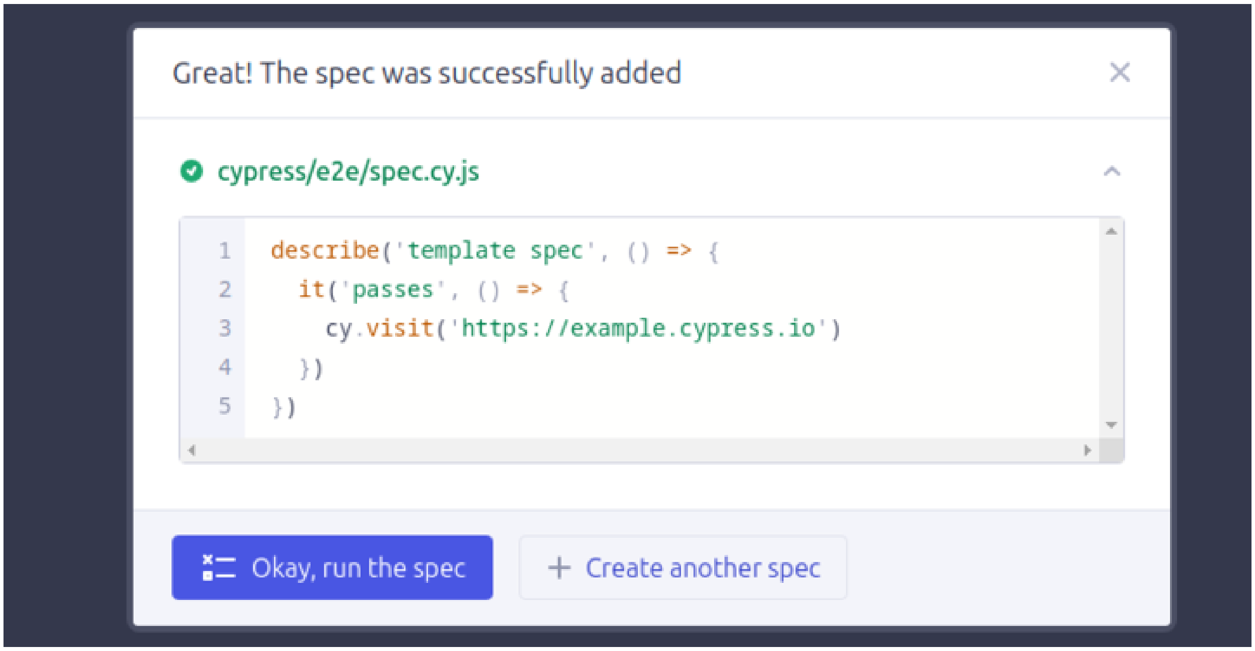


Enter the path for your new spec

cypress/e2e/spec.cy.js

Step 9:

- Click on the **Okay, run the spec**



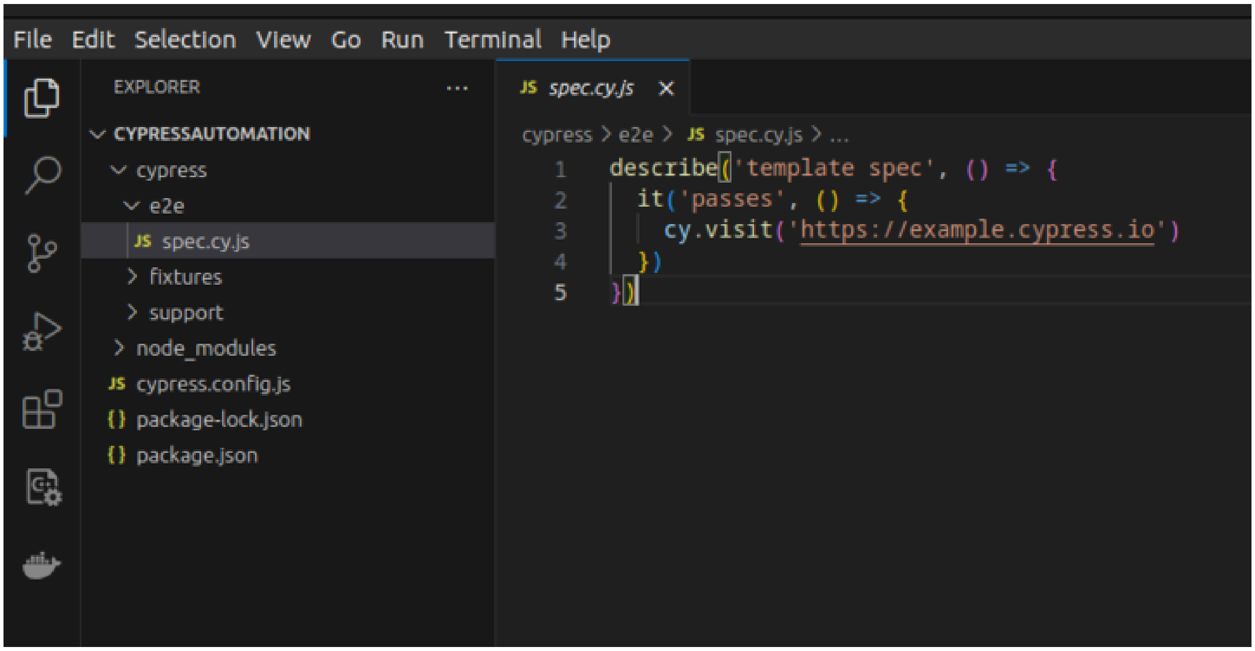
Great! The spec was successfully added

cypress/e2e/spec.cy.js

```
describe('template spec', () => {
  it('passes', () => {
    cy.visit('https://example.cypress.io')
  })
})
```

Congrats..! You have successfully run your first e2e Test.

- Now go to IDE Create New spec Files **cypress/e2e/testcasename.cy.js** and write test cases as many as you want



```
describe('template spec', () => {
  it('passes', () => {
    cy.visit('https://example.cypress.io')
  })
})
```