



# Getting started with Cypress 10 and Cucumber



Amna Bibi · Follow

Published in Emumba · 6 min read · Sep 21, 2022



111



5



## Cypress 10 is here! 🎉

Among the most popular test automation frameworks for the web, Cypress saves one of the top spots. As mentioned in the [official docs](#) of cypress, it provides time travel, debuggability, automatic waiting, flake-free and reliable tests, network traffic control, cross-browser testing, and much more! making it the perfect choice for web automation testers.

Not long ago, cypress 10 had been released with breaking changes so it's better for the folks who are just planning to get their hands dirty on cypress to install the latest version. The process of installation of cypress is quite simple and easy but we'll look at how to install and configure cucumber with it.

Wait a minute...!!

We haven't yet discussed what cucumber is and why we are using it here. Before starting, let's quickly discuss cucumber.

### What is cucumber BDD?



Based on Behavior Driven Development, cucumber is a testing framework that uses simple English sentences to define the behavior of any application. With the help of a Gherkin script, it brings all the technical

and non-technical people to the same page about the product hence making it simple for the team members to understand what's going to be created.

You can further read about cucumber [here](#)

### Installation steps

Now, let's move on to the installation of cypress and cucumber step by step. Starting with the prerequisites, you need to have these tools to initiate project setup

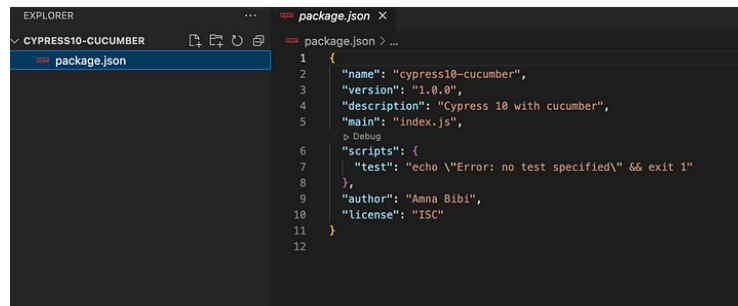
- [VS Code](#)
- [Node.js](#) 12 or 14 and above

#### 1. Create a project and install cypress

- Create an empty project folder in your local machine and open that project in VS Code
- Open the terminal and enter the command

```
npm init
```

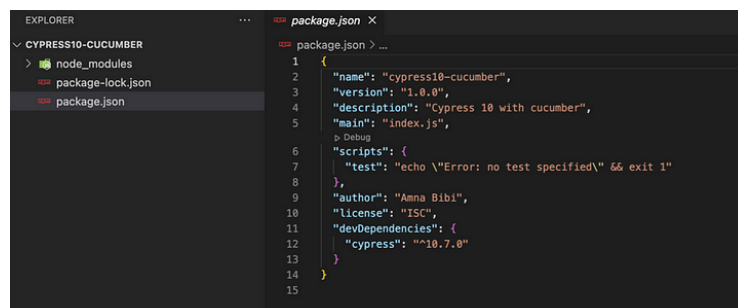
Answer the questions that it asks or just keep pressing Enter. After the last step, it'll add a package.json file to your project with the information you have added so far, like this



package.json

- Type the following command on the terminal to install the latest version of cypress

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



package.json after installation of cypress

At this point, add the script to run cypress. You can directly open cypress Launchpad from here

```
"cypress:open": "cypress open"
```

package.json after adding cypress:open script

It'll give you the option to select either E2E Testing or Component Testing. We'll go with the E2E Testing here

Cypress Launchpad

For first-time users, there's an onboarding screen, you can read and click the Continue button which takes you to the selection of Browser screen. We'll go with Chrome, click "Start E2E Testing" in Chrome and it'll take you to this screen

After setting up cucumber and creating new feature files, we'll return to this screen, so let's close it for now.

You'll notice that the above step has automatically added the cypress folder and cypress.config.js file into your root directory.

## 2. Install cucumber

- Type this command on the terminal to install cucumber

```
npm i @badeball/cypress-cucumber-preprocessor
npm i cypress @bahmutov/cypress-esbuild-preprocessor esbuild
```

package.json after installation of cucumber libraries

- The next step is to set the configurations in cypress.config.js file

```
const createBundler = require("@bahmutov/cypress-esbuild-preprocessor");

const addCucumberPreprocessorPlugin = require("@badeball/cypress-cucumber-preprocessor").addCucumberPreprocessorPlugin;

const createEsBuildPlugin = require("@badeball/cypress-cucumber-preprocessor/esbuild").createEsbuildPlugin;

module.exports = defineConfig({
  e2e: {
    async setupNodeEvents(on, config) {
      // implement node event listeners here
      const bundler = createBundler({
        plugins: [createEsBuildPlugin(config)],
      });
      on('file:preprocessor', bundler);
      await addCucumberPreprocessorPlugin(on, config);
      return config;
    },
    specPattern: 'cypress/e2e/features/*.feature'
  },
});
```

Add your feature file path in the **specPattern**

cypress.config.js

- Create a new file named `.cypress-cucumber-preprocessorrc.json` inside the root directory of your project and specify the location of step definitions as follows

```
{
  "stepDefinitions": [
    "cypress/stepDefinitions/*.js"
  ]
}
```

`.cypress-cucumber-preprocessorrc.json`

Installing a cucumber support plugin in VS Code will come in handy as it'll make you write the features and step definitions files more effortlessly. I'm using the one shown in the screenshot below

Cucumber (Gherkin) plugin

### 3. Write scripts

Now, the project is ready to add features and step definitions. I have used <https://automationexercise.com/> for practice purposes and automated the Login test case for demonstration.

Here's the login.feature file snippet and screenshot

```
Feature: Login
Scenario: Login user with correct email and password
Given I navigate to automation exercise website
```

```
When I enter login credentials
Then I should be logged in
```

login.feature

and below is the loginSteps snippet with screenshot

```
/// <reference types="Cypress" />
import {Given, When, Then} from '@badeball/cypress-cucumber-preprocessor';

Given('I navigate to automation exercise website', () => {
  cy.visit('https://automationexercise.com/');
})

When('I enter login credentials', () => {
  cy.get('a[href="/login"]').click();
  cy.contains('Login to your account').should('be.visible');
  cy.get('input[data-qa="login-email"]').type('example1@test.com');
  cy.get('input[data-qa="login-password"]').type('foobar');
  cy.get('button[data-qa="login-button"]').click();
})

Then('I should be logged in', () => {
  cy.contains(' Logged in as ').should('be.visible');
})
```

loginSteps.js

#### 4. Run scripts

Run the below script from package.json file directly using

```
npm run cypress:open
```

and select login.feature from your feature files inside the Spec Explorer

login.feature file in specExplorer

Now, you can see your test running in the test runner.

One point to note here is that when you run the test you'll see all of the xhr calls that are not necessary to view so it's better to hide them as they'll cause a nuisance while viewing your individual test actions.

xhr calls in test runner

In order to hide them, simply copy the following code in *cypress/support/e2e.js*

```
// Hide fetch/XHR requests
const app = window.top;
if (!app.document.head.querySelector('[data-hide-command-log-request]')) {
  const style = app.document.createElement('style');
  style.innerHTML =
    '.command-name-request, .command-name-xhr { display: none }';
  style.setAttribute('data-hide-command-log-request', '');
  app.document.head.appendChild(style);
}
```

e2e.js

Then again run the test and notice that there are no xhr calls anymore

Test runner without xhr calls

I hope you found this blog helpful. Follow the above steps and run your first test case with cypress and let me know how it went in the comments.

Happy Testing!

Cypress

Cucumber

Web Automation Testing

👏 111

💬 5



**Written by Amna Bibi**

15 Followers · Writer for Emumba

SQA Engineer at Emumba

Follow



More from Amna Bibi and Emumba



Amna Bibi in Emumba

### Setup Xcode and Appium for iOS Automation (Real device)

Passing software through rounds of automation testing, before it is released into...

7 min read · Jan 24, 2022

👏 55    💬 1



Usman Ghani in Emumba

### JS/TS Automation Best Practices

Tips to make your automation suite effective

8 min read · Jan 4

👏 19    💬 1



Mah Noor in Emumba

### Signing a JWT in JMeter for load testing

How to create a JWT in JMeter using JSR223 Sampler?

4 min read · Jan 6, 2022

👏 73    💬



Amna Bibi in Emumba

### Things to Add in your Mobile App Testing Arsenal

Testing plays an integral role in the software development life cycle. Detecting bugs early...

5 min read · Oct 25, 2021

👏 56    💬 2



See all from Amna Bibi

See all from Emumba

## Recommended from Medium

Miktađ Öztürk

### ChatGPT + Cypress Tests

In this article, I would like to tell you how I made the Cypress tests by asking ChatGPT.

🌟 · 2 min read · Jan 18

👏 61    💬 2



Dragos Bogdan in ILLUMINATION'S MIRROR

### Streamlining Web Application Testing with Cypress: A Beginner'...

A web application must be tested to ensure that it is functioning properly and providing...

🌟 · 6 min read · Jan 24

👏 196    💬 1



#### Staff Picks

300 stories · 62 saves

#### Self-Improvement 101

20 stories · 44 saves

#### Stories to Help You Level-Up at Work

19 stories · 19 saves

#### Productivity 101

20 stories · 42 saves

Andreea-QA in The QA LEAD

#### API Testing Interview Questions

According to a SlashData 2020 survey, almost 90% of the developers interviewed were...

★ · 7 min read · Feb 17

👏 61



S Chathuranga Jayasinghe in Geek Culture

#### Testing QR/Bar Codes with Cypress

Did you know that you can test the QR codes and Bar codes in your web application using...

★ · 4 min read · Mar 15

👏 28



S Chathuranga Jayasinghe in Geek Culture

#### Mind Maps: Keep you a few steps ahead in Quality Engineering

Mind maps do magic! It can be super useful if you use it the right way..

★ · 6 min read · Jan 12

👏 59



Sanghamitra Bhowmick

#### Cypress Code Coverage for Cypress—Typescript Framework

Code coverage in Cypress refers to the measurement of how much of your...

2 min read · Feb 19

👏



See more recommendations