Random Test data generation – Faker

You can use the faker-js library to generate random test data in your Cypress automation tests. Here’s how you can integrate it into your Cypress test suite:

**Step 1: Install Faker**

Run the following command to install Faker:

npm install @faker-js/faker --save-dev

**Step 2: Import Faker in Cypress Test**

In your Cypress test file (.cy.js or .cy.ts), import faker and use it to generate fake data.

import { faker } from '@faker-js/faker';

describe('Automation Test with Faker', () => {

it('Should fill out a form with fake data', () => {

cy.visit('https://example.com/signup'); // Change to your test URL

// Generate fake data

const firstName = faker.person.firstName();

const lastName = faker.person.lastName();

const email = faker.internet.email();

const password = faker.internet.password();

// Fill out the form

cy.get('#firstName').type(firstName);

cy.get('#lastName').type(lastName);

cy.get('#email').type(email);

cy.get('#password').type(password);

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

// Assertions (Modify based on your application)

cy.contains('Welcome').should('be.visible');

});

});

**Step 3: Using Faker in Fixtures or Commands**

**Using Faker in Fixtures**

Instead of generating random data inside the test, you can create a fixture and use Faker dynamically.

1. **Create a fixture file (cypress/fixtures/user.json):**

{

"firstName": "",

"lastName": "",

"email": "",

"password": ""

}

1. **Modify the fixture before the test runs:**

import { faker } from '@faker-js/faker';

beforeEach(() => {

cy.fixture('user').then((user) => {

user.firstName = faker.person.firstName();

user.lastName = faker.person.lastName();

user.email = faker.internet.email();

user.password = faker.internet.password();

cy.writeFile('cypress/fixtures/user.json', user);

});

});

**Using Faker in Custom Cypress Commands**

You can also define a Cypress command for generating random user data:

1. **Add this in cypress/support/commands.js:**

import { faker } from '@faker-js/faker';

Cypress.Commands.add('generateFakeUser', () => {

return {

firstName: faker.person.firstName(),

lastName: faker.person.lastName(),

email: faker.internet.email(),

password: faker.internet.password(),

};

});

1. **Use the command in tests:**

describe('Test with custom Faker command', () => {

it('Should create a user with fake data', () => {

cy.generateFakeUser().then((user) => {

cy.visit('/signup');

cy.get('#firstName').type(user.firstName);

cy.get('#lastName').type(user.lastName);

cy.get('#email').type(user.email);

cy.get('#password').type(user.password);

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

});

});

});

**Step 4: Running the Test**

Run your Cypress tests using:

npx cypress open

or

npx cypress run

This will execute your test cases with dynamically generated fake data.