# Cypress Component Testing with Storybook

React + Vite + JavaScript Setup and Documentation

## 1. Project Stack

• Framework: React (JavaScript)

• Bundler: Vite

• Testing Tool: Cypress (Component Testing)

• UI Explorer: Storybook

## 2. Installation

### Install Storybook

Run the following command to initialize Storybook in your project:

***npx storybook init***

This command will set up the .storybook/ folder and install required dependencies.

### Install Cypress

Install Cypress as a dev dependency with:

***npm install --save-dev cypress***

## 3. Configure Cypress for Component Testing with Vite

Add or update your cypress.config.js file with the following configuration:

***const { defineConfig } = require("cypress");  
  
module.exports = defineConfig({  
 component: {  
 devServer: {  
 framework: "react",  
 bundler: "vite",  
 },  
 specPattern: [  
 'cypress/component/\*\*/\*.cy.{js,jsx,ts,tsx}',  
 'src/\*\*/\*.cy.{js,jsx,ts,tsx}',  
 ],  
 },  
});***

## 4. Writing a Cypress Component Test

Use the following folder structure:

***src/  
└── components/  
 └── LoginForm/  
 ├── LoginForm.jsx  
 ├── LoginForm.css  
 └── LoginForm.stories.jsx  
  
cypress/  
└── component/  
 └── LoginForm.cy.jsx***

Example test file `LoginForm.cy.jsx`:

***import React from 'react';  
import { mount } from 'cypress/react';  
import LoginForm from '../../src/components/LoginForm/LoginForm';  
  
describe('LoginForm', () => {  
 it('shows error when fields are empty', () => {  
 mount(<LoginForm />);  
 cy.get('button[type="submit"]').click();  
 cy.get('.error').should('contain', 'Username and password are required.');  
 });  
  
 it('logs in with valid credentials', () => {  
 mount(<LoginForm />);  
 cy.get('input[type="text"]').type('admin');  
 cy.get('input[type="password"]').type('admin123');  
 cy.get('button[type="submit"]').click();  
 cy.get('.success').should('contain', 'Login successful!');  
 });  
});***

## 5. Running Component Tests

Use the following command to open Cypress in component testing mode:

***npx cypress open --component***

Then select the test file and run it. Cypress will mount your component isolated in a Vite dev server.

## 6. Storybook + Cypress

Storybook and Cypress are complementary tools. While Storybook is a powerful UI explorer and design documentation tool, Cypress Component Testing is used to test React components directly in isolation using the Vite dev server.  
- Component tests do not depend on Storybook and mount components directly.  
- For testing Storybook stories directly (like visual regression or iframe-based behavior), use Cypress in E2E mode with `cy.visit()` pointing to Storybook iframe URLs.

For component testing, Cypress mounts components directly and does not require Storybook or cy.visit().

## 7. Benefits of Component Testing

• Fast and isolated UI tests  
• Integrated with modern dev servers like Vite  
• Ideal for testing React component logic and UI states  
• Enables testing UI with real props and user interactions

## 8. Best Practices

• Keep tests close to components  
• Test happy paths and edge cases  
• Use data-testid attributes for reliable selectors  
• Use component testing primarily for UI and logic validation

## 9. Related Links

• Cypress Component Testing Docs: https://docs.cypress.io/guides/component-testing/introduction

• Storybook Docs: https://storybook.js.org/docs/react/get-started/introduction

• React + Vite Integration Guide: https://vitejs.dev/guide/

# 10. Spike Task Summary

Spike Title: Integration of Cypress Component Testing with Storybook in React + Vite Project  
  
**Goal**: Evaluate feasibility and setup of Cypress component testing in a project that uses Storybook for UI development.  
  
**Findings**:  
- Cypress Component Testing works independently of Storybook.  
- Storybook is helpful for UI dev & design system previews, not required for component test rendering.  
- Vite is compatible as a dev server for Cypress CT.  
- A working LoginForm component test was created and successfully executed.  
  
**Outcome**:  
- Minimal setup completed.  
- Sample component tested.  
- Storybook and Cypress can co-exist in the project, each serving different purposes.  
- Documentation.