





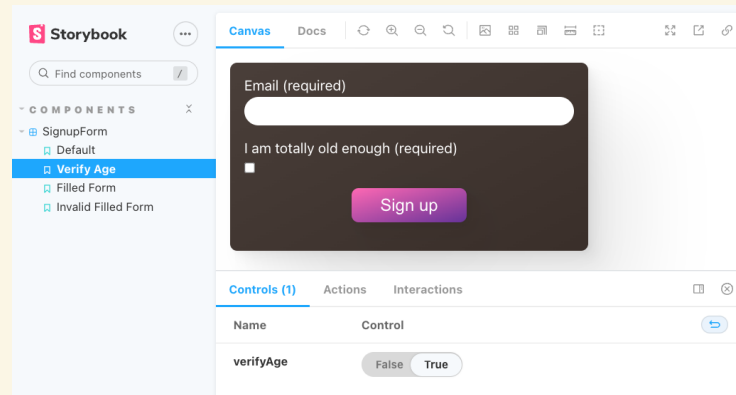


INTERACTION TESTING WITH STORYBOOK

STORYBOOK IN A NUTSHELL

A LIBRARY OF YOUR APP'S COMPONENTS.

-  Build components in isolation
-  Discover existing components
-  Serves as a style guide and documentation
-  Visually test edge-cases

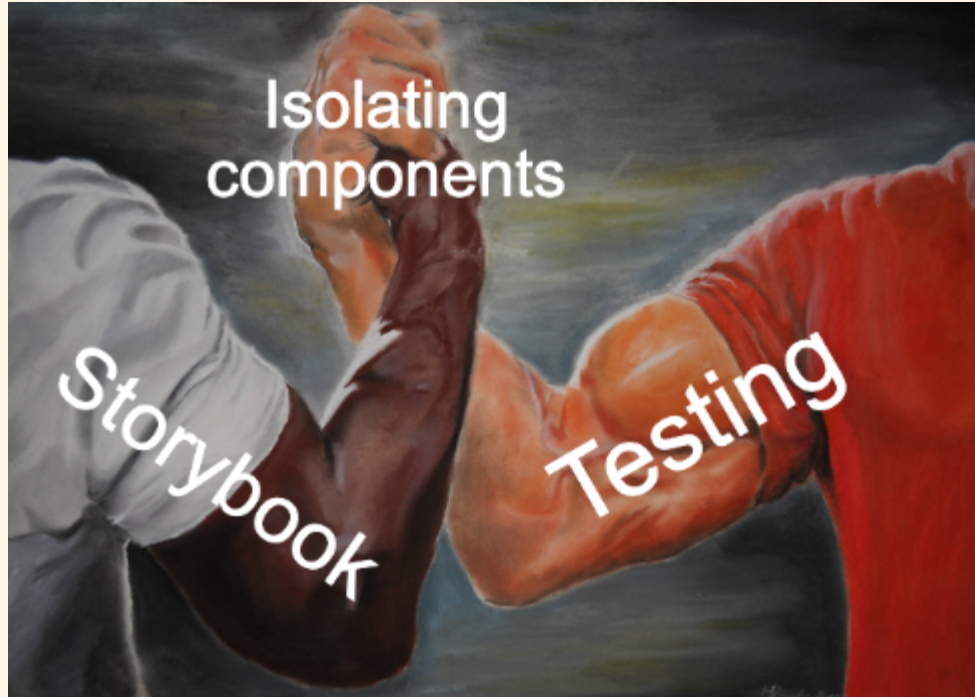


storybook.js.org



**WHERE ELSE DO WE
ISOLATE COMPONENTS?**





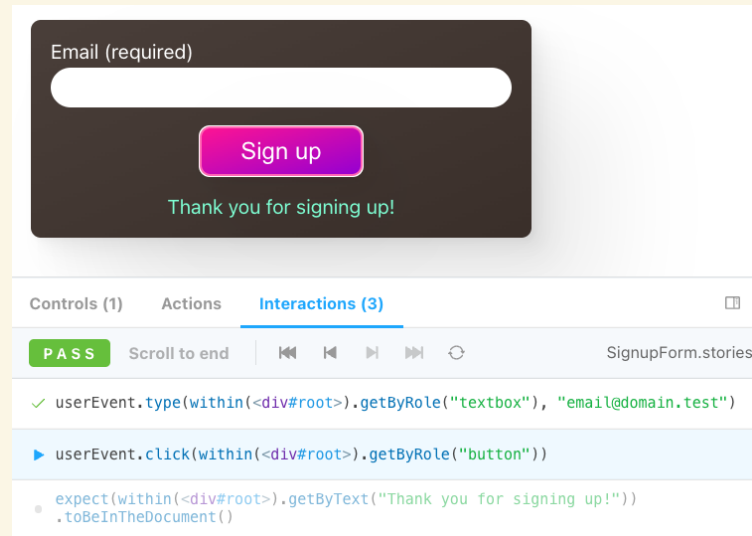
STORYBOOK ❤️ JEST BENEFITS

🧱 ARRANGE TESTS WITH STORIES

- Reuse your Stories to arrange tests.
- Check out the code repo at the end to learn more!

🧐 TEST INTERACTIONS WITH STORIES

- Useful for more complex components.
- Test entire pages!
- Visually 👁️ see what is happening, step-by-step.
- Helps debug errors.



JUST WRITE A `play` FUNCTION

```
FilledForm.play = async ({canvasElement}) => {  
  // Starts querying the component from its root element  
  const canvas = within(canvasElement);  
  
  // Simulate interactions with `user-event`  
  await userEvent.type(canvas.getByRole('textbox'), 'email@domain.test');  
  await userEvent.click(canvas.getByRole('button'));  
  
  // Assert that our success message is displayed  
  expect(canvas.getByText('Thank you for signing up!')).toBeInTheDocument();  
}
```

JUST WRITE A `play` FUNCTION

```
FilledForm.play = async ({canvasElement}) => {  
  // Starts querying the component from its root element  
  const canvas = within(canvasElement);  
  
  // Simulate interactions with `user-event`  
  await userEvent.type(canvas.getByRole('textbox'), 'email@domain.test');  
  await userEvent.click(canvas.getByRole('button'));  
  
  // Assert that our success message is displayed  
  expect(canvas.getByText('Thank you for signing up!')).toBeInTheDocument();  
}
```

JUST WRITE A `play` FUNCTION

```
FilledForm.play = async ({canvasElement}) => {  
  // Starts querying the component from its root element  
  const canvas = within(canvasElement);  
  
  // Simulate interactions with `user-event`  
  await userEvent.type(canvas.getByRole('textbox'), 'email@domain.test');  
  await userEvent.click(canvas.getByRole('button'));  
  
  // Assert that our success message is displayed  
  expect(canvas.getByText('Thank you for signing up!')).toBeInTheDocument();  
}
```


JUST WRITE A `play` FUNCTION

```
FilledForm.play = async ({canvasElement}) => {  
  // Starts querying the component from its root element  
  const canvas = within(canvasElement);  
  
  // Simulate interactions with `user-event`  
  await userEvent.type(canvas.getByRole('textbox'), 'email@domain.test');  
  await userEvent.click(canvas.getByRole('button'));  
  
  // Assert that our success message is displayed  
  expect(canvas.getByText('Thank you for signing up!')).toBeInTheDocument();  
}
```

SEE IT IN STORYBOOK

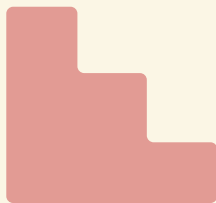
``localhost:6006`` **BY DEFAULT**





The screenshot displays a Storybook component titled 'SignupForm.stories'. The component is a dark-themed sign-up form with a text input labeled 'Email (required)', a 'Sign up' button, and a confirmation message 'Thank you for signing up!'. Below the component, the Storybook toolbar shows the 'Interactions (3)' tab selected. The interaction log contains three entries: a successful type event on the email input, a click event on the sign-up button, and an expect statement for the confirmation message.

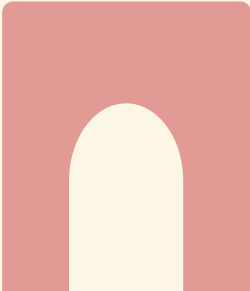

```
Controls (1)  Actions  Interactions (3)  [icon]  
PASS  Scroll to end  [play] [stop] [refresh]  SignupForm.stories  
✓ userEvent.type(within(<div#root>).getByRole("textbox"), "email@domain.test")  
▶ userEvent.click(within(<div#root>).getByRole("button"))  
◉ expect(within(<div#root>).getByText("Thank you for signing up!"))  
  .toBeInTheDocument()
```

Click on each step to see what happened!





BUT THAT'S NOT ALL!



Run `play` FUNCTIONS AS TESTS

- When you develop.
- In your CI pipeline.

```
npm run test-storybook
```

```
✓ ~/Projects/storybook-interaction-testing [main] 4...6
12:34 $ npm run test-storybook

> storybook-interaction-testing@0.1.0 test-storybook
> test-storybook

PASS browser: chromium src/components/SignupForm/SignupForm.stories.tsx
components/SignupForm
  Default
    ✓ smoke-test (46 ms)
  VerifyAge
    ✓ smoke-test (9 ms)
  FilledForm
    ✓ play-test (26 ms)
  InvalidFilledForm
    ✓ smoke-test (16 ms)

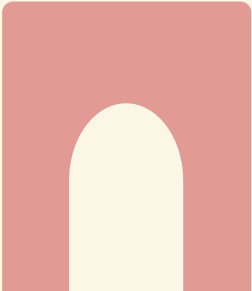

Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 1.91 s, estimated 2 s
Ran all test suites.
```



TRY IT YOURSELF!

Check out this repo I made for you

I'll post the link in the chat and Slack.





THANKS!