

playwright-test: test that input is required

Asked 2 years, 10 months ago Modified 6 months ago Viewed 8k times



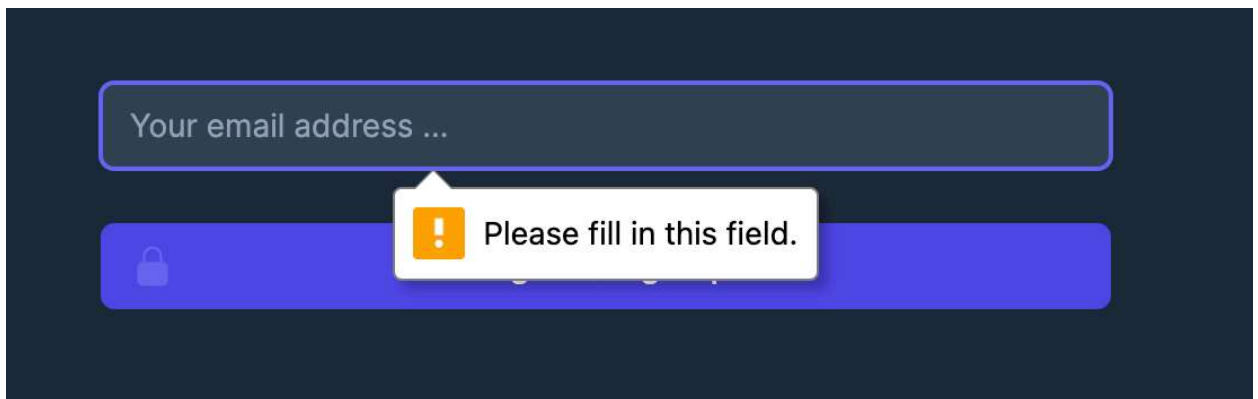
I have an input for a login form that is required:

8



```
<input
  autoComplete="email"
  defaultValue={email}
  disabled={state === 'submitting'}
  id="email"
  name="email"
  placeholder={t('user-authentication:email-placeholder')}
  required // HERE!
  type="email"
/>
```

How can I test that the input is required with Playwright? My native browser warning in Chromium is: "Please fill in this field."



Is there a way to check / assert for this warning in Playwright? Or another way to test that the input is required?

I tried writing nothing in the input with `.fill('')` and then pressing enter with `page.keyboard.press('Enter')`, but those did not cause the warning to show up.

[playwright](#) [required](#) [html-input](#) [playwright-test](#)

[Share](#) [Improve this question](#) [Follow](#)

asked Nov 11, 2022 at 20:52



[J. Hesters](#)

15k 36 156 271

Try this. Add empty space in input field `fill(' ')`. Then press **Backspace**
`page.keyboard.press('Backspace')` . – [Muhammad Nouman Rafique](#) Nov 11, 2022 at 23:10

What about pressing Enter directly, without using fill()? The pop up should appear – [jopp](#) Nov 12, 2022 at 20:21



11



If the message is displayed by the browser and not something added to the DOM, you would struggle to test it directly in Playwright.

You could check the the `required` attribute is present, and trust that the browser is going to display the warning.

```
await page.locator('input#email[required]')
```

There's also a [Constraint validation](#) you could apply

If the element is required and the element's value is the empty string, then the element is suffering from `valueMissing` and the element will match the `:invalid` pseudo class.

```
await page.locator('input#email[required]:invalid')
```

Share Improve this answer Follow

answered Nov 12, 2022 at 19:46

user19615870

1 Comment

Add a comment



Jaky Ruby Over a year ago

I am agree with Layla. If the field is required, the browser should show the alert, you have to trust on the browser, if the browser is not working you will not be able to solve yourself, so I think it is a good validation. If it is not enough for you maybe you need to talk with developers to add some other kind of errors.

▲ 0

Reply



7



Here is the working Playwright code of testing browser native warning message (tested for Chrome):

```
import { expect, test } from '@playwright/test';

test('has warning message presented', async ({ page }) => {
  await page.goto('https://q94u5.csb.app/');
  const email = page.getByRole('textbox');
  const submit = page.getByRole('button', { name: 'Submit' });

  await email.fill('x');
  await submit.click();

  const validationMessage = await email.evaluate((element) => {
```

```
const input = element<as HTMLInputElement>
return input.validationMessage
})

expect(validationMessage).toContain("Please include an '@' in the email
address. 'x' is missing an '@'.")
});
```

As you can see, you can catch `validationMessage` in Playwright but is not recommended to test it. It is browser responsibility to deliver this element (and it is not responsibility of the webpage itself).

Read more about build-in validation here: <https://www.w3.org/WAI/tutorials/forms/validation/>

More elements information can be extracted this way: https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation and go to chapter **Validating forms using JavaScript** to check available methods

Share Improve this answer Follow

edited Sep 22, 2023 at 3:59

answered Sep 17, 2023 at 6:37



pbaranski

25.3k 23 123 134

1 Comment ▾

Add a comment



crashbus Over a year ago

It's working for Chrome and other browser, too. But the problem with cross browser testing is that the validation messages vary from browser to browser. So when it's only Chrome really nice way :)



1



Reply



0

Adding on to the other answer here: <https://stackoverflow.com/a/74416040/9889773>

I used `waitForSelector` to solve this for myself. i.e:



```
await page.waitForSelector('input#email[required]:invalid')
```



And using `:valid` after filling in valid input.



Share Improve this answer Follow

answered Mar 10 at 1:06



nivlac

431 4 12

Comments

Add a comment

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

[playwright](#)

[required](#)

[html-input](#)

[playwright-test](#)

See similar questions with these tags.