



DEV ACADEMY
TE KURA HANGARAU
O AOTEAROA

Testing Simple React applications

Week 2, Day 1





Arrange, Act and Assert with React

1. Arrange - Render our component
2. Act - Not much for now
3. Assert - Observe properties of the DOM



Render

We're using a library called: `@testing-library/react`

This provides a function `render` that renders our component into a fake-dom

```
1 import { render } from '@testing-library/react'
```



Getting ready for DOM based tests

We need to make sure that our test file using the `jest-dom`` environment

```
1 // @vitest-environment jsdom
```

We also need to make sure that package is installed

```
1 npm i -D jsdom
```



DOM oriented assertions

We're going to use assertions from `@testing-library/jest-dom``

```
1 expect(...).toBeVisible()
2 expect(...).toHaveAccessibleName('...')
3 expect(...).toHaveAttribute('...')
```



Connecting these libraries into vitest

Since these libraries were created for ``jest`` we need to do a few manual steps to make them work well with ``vitest``



Connecting @testing-library/react

We need to make sure our DOM is clean before every test

```
1  import { cleanup } from '@testing-library/react'
2  import { beforeEach } from 'vitest'
3
4  beforeEach(cleanup)
```



Connecting @testing-library/jest-dom

We need to extend our `expect` with the new matchers:

```
1 import { expect } from 'vitest'
2 import * as matchers from '@testing-library/jest-dom/matchers'
3
4 expect.extend(matchers)
```




Setting it up all together

We can pop our setup into a `test/setup.tsx` file

```
1 import { expect, beforeEach } from 'vitest'
2 import { cleanup } from '@testing-library/react'
3 import * as matchers from '@testing-library/jest-dom/matchers'
4 import '@testing-library/jest-dom/vitest'
5
6 beforeEach(cleanup)
7 expect.extend(matchers)
```

We can then import as needed in our component test files after we include the essentials

```
1 // @vitest-environment jsdom
2 import { render, screen } from '@testing-library/react'
3 import { it, describe, expect } from 'vitest'
4 import '../test/setup.tsx'
```

Testing





Making sure our markup is accessible

In our component instead of a `` for the dog name we can use a heading

```
1 <span class="dog-name">James</span>
2
3 <!-- role: heading level: 3 -->
4 <h3 class="dog-name">James</h3>
```

Making sure our markup is accessible

Make sure our images have alt-text

```
1 
```

1. Add alt text to all non-decorative images.
2. Keep it short and descriptive, like a tweet.
3. Don't include "image of" or "photo of".
4. Leave alt text blank if the image is purely decorative

<https://accessibility.huit.harvard.edu/describe-content-images>