Trainer guide
Examples and exercises (React)

React-6 – Component-scoped styling

Contents

Trainer demo | CSS Modules

Link to environment

In this demo, you will explain the benefits of component-scoped styling, and how to achieve this using CSS Modules.

src/App.jsx

- Here we have a simple app with two components, PageTitle and MainContent.
- Notice the different colours of the page title and the main content title.

src/style.css

- We apply these different colours in this CSS file called style.css.
- We have one CSS class for the page title, giving it this slate blue colour, and one class for the main content title, giving it this salmon colour.
- Now imagine that our app had dozens of components. It wouldn't be sustainable to keep adding more and more CSS into this single file.
- This would be problematic for a few reasons:
    - The CSS file would become very large and hard to maintain.
    - All the styles for the different components would be mixed together, and it would become difficult to tell at a glance where the styling for one component ends and the styling for the next component begins.
- So, what if instead of thinking in terms of styling our application, we could style *individual components*? In other words, for each React component, have a CSS file associated with it.
- This would have some advantages:
    - Although we'd have more CSS files, they would be much smaller.
    - We would know exactly what the purpose of each CSS file is, because the styles only apply to that one component and nothing else in the application.
    - Our code would be more organised.
- All of these benefits are provided by CSS Modules.

Demonstrate the solution, explaining as you go along.

- *Create file: `src/components/PageTitle.module.css` (right click components folder -> New file)*
- *Cut the CSS class `pageTitle` from `style.css` and paste it into `PageTitle.module.css`*
- *Navigate to `src/components/PageTitle.jsx`*
    - *Add import statement: `import styles from './PageTitle.module.css'`*

    - *In the h1, replace* `className="pageTitle"` *with* `className={styles.pageTitle}`

- Now we're going to do the same with the MainContent component.
- *Create file:* `src/components/MainContent.module.css` *(right click components folder -> New file)*
- *Cut the CSS class* `mainContentTitle` *from* `style.css` *and paste it into* `MainContent.module.css`
- *Navigate to* `src/components/MainContent.jsx`
    - *Add import statement:* `import styles from './MainContent.module.css'`
    - *In the h2, replace* `className="mainContentTitle"` *with* `className={styles.mainContentTitle}`

src/style.css

- Now our main style file only contains styles that are global to the application. Any component styles are now in separate CSS Modules.

Scope

- CSS Modules have a trick up their sleeve: class names are *locally scoped*. This means class names in one CSS Module will never conflict with class names in another CSS Module. Let's see an example.

- *Navigate to* `PageTitle.module.css`.
    - *Change the class name to* `title`.
- *Navigate to* `PageTitle.jsx`
    - *Update* `className` *to* `styles.title`

- *Navigate to* `MainContent.module.css`.
    - *Change the class name to* `title`.
- *Navigate to* `MainContent.jsx`
    - *Update* `className` *to* `styles.title`

- Now it looks like both title elements have the same CSS class name, but they have different styles applied to them.
- This shows that classes are locally scoped in CSS Modules.
- How is this possible?
    - *Open dev tools*.
    - *Inspect the title elements and point out that the class names have random characters added to the end*.
    - These nonsense characters are automatically generated. This is how CSS Modules prevent class names from conflicting with each other.

Project work

The CSS Modules topic precedes the first block of project work.

Using CSS Modules is optional for the upcoming React practical project, but if you feel comfortable trying them out, you've been given a link to the example we've just seen, which you can use as a reference.

Solution reference