

React CSS Issue — Why One CSS File Affects All Components?

Problem

Even after creating separate CSS files for each React component (e.g., `Header`, `Hero`, `Contact`), the styles still overlap. Changing something in one CSS file unintentionally affects others.

Example Code

`header.jsx`

```
import "../header.css";
import React from "react";

function Header() {
  return (
    <>
      old: <h1>Welcome to My Website</h1>
    </>
  );
}

export default Header;
```

`hero.jsx`

```
import "../hero.css";
import React from "react";

function Hero() {
  return (
    <>
      <h1>Welcome to Hero Section</h1>
    </>
  );
}

export default Hero;
```

`header.css`

```
h1 {
  color: blue;
}
```

`hero.css`

```
h1 {
  color: red;
}
```

What Happens Behind the Scenes?

React's bundler (Webpack) merges all CSS files into one bundle.

Combined Output (Webpack)

```
h1 {  
  color: blue;  
}  
  
h1 {  
  color: red;  
}
```

Why Styles Overlap?

- CSS is global by default in React.
- Both files target the same selector `h1`.
- The last loaded rule wins (red overrides blue).

Solutions

1. Basic Solution — Use Class Names

header.css

```
.header {  
  color: blue;  
}
```

hero.css

```
.hero {  
  color: red;  
}
```

header.jsx

```
<h1 className="header">Welcome to My Website</h1>
```

hero.jsx

```
<h1 className="hero">Welcome to Hero Section</h1>
```

Why It Works

- Classes scope styling to each component.
- No global conflict like using `h1`.

2. Advanced Solution — CSS Modules

File Naming

```
Header.module.css  
Hero.module.css
```

Importing

```
import styles from "../Header.module.css";
```

Using

```
<h1 className={styles.title}>Welcome to My Website</h1>
```

How CSS Modules Prevent Conflicts

CSS Modules auto-generate unique hashed class names, so identical names never clash.

Example Internal Conversion

```
styles.title → title__x8a2b  
styles.title → title__9f3c1
```

Even if both files contain:

```
.title { color: blue; }
```

and

```
.title { color: red; }
```

They remain fully isolated.

Why Use CSS Modules?

- No naming conflicts
- Component-level scoped styles
- Cleaner structure in large apps
- Safe reuse of class names across components