static-components

Validates that components are static, not recreated every render. Components that are recreated dynamically can reset state and trigger excessive re-rendering.

Rule Details

Components defined inside other components are recreated on every render. React sees each as a brand new component type, unmounting the old one and mounting the new one, destroying all state and DOM nodes in the process.

Invalid

Examples of incorrect code for this rule:

```
// ❌ Component defined inside component
function Parent() {
  const ChildComponent = () => { // New component every render!
    const [count, setCount] = useState(0);
    return <button onClick={() => setCount(count + 1)}>{count}</button>;
  };

  return <ChildComponent />; // State resets every render
}

// ❌ Dynamic component creation
function Parent({type}) {
  const Component = type === 'button'
    ? () => <button>Click</button>
    : () => <div>Text</div>;

  return <Component />;
```

```
}
```

Valid

Examples of correct code for this rule:

```
// ✅ Components at module level
const ButtonComponent = () => <button>Click</button>;
const TextComponent = () => <div>Text</div>;

function Parent({type}) {
  const Component = type === 'button'
    ? ButtonComponent  // Reference existing component
    : TextComponent;

  return <Component />;
}
```

Troubleshooting

I need to render different components conditionally

You might define components inside to access local state:

```
// ❌ Wrong: Inner component to access parent state
function Parent() {
  const [theme, setTheme] = useState('light');

  function ThemedButton() { // Recreated every render!
    return (
      <button className={theme}>
        Click me
      </button>
    );
  }

  return <ThemedButton />;
```

Pass data as props instead:

```
// ✅ Better: Pass props to static component
function ThemedButton({theme}) {
  return (
    <button className={theme}>
      Click me
    </button>
  );
}

function Parent() {
  const [theme, setTheme] = useState('light');
  return <ThemedButton theme={theme} />;
}
```

중요합니다!

If you find yourself wanting to define components inside other components to access local variables, that's a sign you should be passing props instead. This makes components more reusable and testable.

이전
set-state-in-render

다음
unsupported-syntax

Meta Open Source

uwu?

React 학습하기

- 빠르게 시작하기
- 설치하기
- UI 표현하기
- 상호작용성 더하기
- State 관리하기
- 탈출구

API 참고서

- React APIs
- React DOM APIs

커뮤니티

- 행동 강령
- 팀 소개
- 문서 기여자
- 감사의 말

더 보기

- 블로그
- React Native
- 개인 정보 보호
- 약관