

Advanced Reactjs Cheatsheet

Quick Reference for Advanced React Developers

Table of Contents

- [Advanced Hooks](#)
- [Performance Optimization](#)
- [Server-Side Rendering \(SSR\)](#)
- [Code Splitting](#)
- [React Patterns](#)
- [State Management](#)
- [Testing](#)
- [Security](#)

1. Advanced Hooks

useMemo: `const memoizedValue= useMemo(() => computeExpensiveValue(a, b), [a, b]);`
useCallback `const memoizedCallback= useCallback(() => { doSomething(a, b); }, [a, b]);`
useImperativeHandle: `useImperativeHandle(ref, () => ({ customMethod() { ... } }));`

2. Performance Optimization

React.memo: `export default React.memo(MyComponent);`
Lazy Loading: `const OtherComponent= React.lazy(() => import('./OtherComponent'));`

3. Server-Side Rendering (SSR)

Next.js: `import { GetServerSideProps } from 'next';`
Hydration: `ReactDOM.hydrate(<App />, document.getElementById('root'));`

4. Code Splitting

Dynamic Imports: `import('./module').then(module => ...);`
Reactlazy and Suspense `<Suspense fallback={<div>Loading...</div>}><MyComponent /></Suspense>`

5. React Patterns

Render Props: `<DataProvider render={data => <h1>{data}</h1>} />`
Higher Order Components (HOC): `const EnhancedComponent= withEnhancement(WrappedComponent);`

6. State Management

Redux Toolkit: `import { configureStore } from '@reduxjs/toolkit';`
MobX: `import { observable } from 'mobx';`

7. Testing

Jest `test('renders correctly', () => { ... });`
React Testing Library: `render(<MyComponent />);`

8. Security

Sanitizing Inputs: `import DOMPurify from 'dompurify';`
Secure API Calls `fetch(url, { method: 'GET', headers: { 'Authorization: 'Bearer token' } });`

