

react@19

Overview

Hooks

>

Components

>

APIs

>

react-dom@19

Hooks

>

Components

>

APIs

>

Client APIs

>

Server APIs

>

Static APIs

>

Rules of React

Overview

>

React Server Components

Server Components

Server Functions

Directives

>

Legacy APIs

Legacy React APIs

>

API REFERENCE >

React Reference Overview

This section provides detailed reference documentation for working with React. For an introduction to React, please visit the [Learn](#) section.

The React reference documentation is broken down into functional subsections:

React

Programmatic React features:

- [Hooks](#) - Use different React features from your components.
- [Components](#) - Built-in components that you can use in your JSX.
- [APIs](#) - APIs that are useful for defining components.
- [Directives](#) - Provide instructions to bundlers compatible with React Server Components.

React DOM

React-dom contains features that are only supported for web applications (which run in the browser DOM environment). This section is broken into the following:

- [Hooks](#) - Hooks for web applications which run in the browser DOM environment.
- [Components](#) - React supports all of the browser built-in HTML and SVG components.
- [APIs](#) - The `react-dom` package contains methods supported only in web applications.
- [Client APIs](#) - The `react-dom/client` APIs let you render React components on the client (in the browser).
- [Server APIs](#) - The `react-dom/server` APIs let you render React components to HTML on the server.

Rules of React

React has idioms — or rules — for how to express patterns in a way that is easy to understand and yields high-quality applications:

- [Components and Hooks must be pure](#) – Purity makes your code easier to understand, debug, and allows React to automatically optimize your components and hooks correctly.
- [React calls Components and Hooks](#) – React is responsible for rendering components and hooks when necessary to optimize the user experience.
- [Rules of Hooks](#) – Hooks are defined using JavaScript functions, but they represent a special type of reusable UI logic with restrictions on where they

Is this page useful?

can be called.

Legacy APIs

- [Legacy APIs](#) - Exported from the `react` package, but not recommended for use in newly written code.

NEXT
[Hooks](#) >

Meta Open Source

Copyright © Meta Platforms, Inc
uwu?

Learn React

- Quick Start
- Installation
- Describing the UI
- Adding Interactivity
- Managing State
- Escape Hatches

API Reference

- React APIs
- React DOM APIs

Community

- Code of Conduct
- Meet the Team
- Docs Contributors
- Acknowledgements

More

- Blog
- React Native
- Privacy
- Terms



Is this page useful?