JSXNode

August 17, 2026

JSXNode

Kind: Class

Source: `src/jsx/base.ts`

Part of: Jsx

`JSXNode` represents a JSX node and exposes conversion methods for rendering its contents. Call `toString()` to obtain rendered text, or call `toStringToBuffer()` when integrating with buffer-oriented output handling.

Implements: `HtmlEscaped`

Methods

Method	Signature	Returns
`toString`	`toString()`	`string
`toStringToBuffer`	`toStringToBuffer(buffer: StringBufferWithCallbacks)`	`void`

Properties

Property	Type
`tag`	`string
`props`	`Props`
`key`	`string`
`children`	`Child[]`
`isEscaped`	`true`

Where it refuses work

- `JSXNode` stops the work with `Error` when `typeof tag !== 'function' && !isValidTagName(tag)`.
- `JSXNode` stops the work with `Error` when `children.length > 0` — “Can only set one of `children` or `props.dangerouslySetInnerHTML`.”.
- `JSXNode` stops the work with `Error` when `!key.startsWith('on') && key !== 'ref'`.

Diagram

```
graph LR
  JSXNode --> ToString["toString(): string | Promise<string>"]
  JSXNode --> ToBuffer["toStringToBuffer(): void"]
  ToString --> Text["Rendered string"]
  ToBuffer --> BufferOutput["Buffer-oriented output"]
```

Usage

```
import { JSXNode } from "./jsx/base";

async function renderNode(node: JSXNode): Promise<string> {
  return await node.toString();
}

function writeNodeToBuffer(node: JSXNode): void {
  node.toStringToBuffer();
}
```

AI Coding Instructions

- Handle `toString()` as potentially asynchronous; use `await` when consuming its result.
- Keep string rendering and buffer-oriented output separate by calling the method that matches the caller's output path.
- Do not assume `toStringToBuffer()` returns rendered content; its return type is `void`.
- Pass `JSXNode` instances through rendering boundaries rather than converting them early when later output handling may differ.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `title` — `src/jsx/intrinsic-element/components.ts`:121
- `StreamingContext` — `src/jsx/streaming.ts`:30