

html-react-parser

`html-react-parser` is a JavaScript library designed to convert an HTML string into one or more React elements. This utility is particularly useful in React applications where dynamic content, often received as raw HTML from a server or a Content Management System (CMS), needs to be rendered within the React component tree.

Key Features and Usage:

- **Conversion of HTML strings to React elements:** The primary function of `html-react-parser` is to take an HTML string as input and transform it into a structure that React can understand and render.

Code

```
import parse from 'html-react-parser';

const MyComponent = () => {
  const htmlString = '<div><h1>Hello, World!</h1><p>This is some dynamic content.</p></div>';
  return <div>{parse(htmlString)}</div>;
};
```

Server-Side Rendering (SSR) and Client-Side Usage: The library is compatible with both server-side (Node.js) and client-side (browser) environments, enabling consistent rendering across different application architectures.

Element Replacement: It provides an option to replace specific HTML elements with custom React components, offering flexibility in how parsed content is displayed.

Code

```
import parse from 'html-react-parser';

const MyComponent = () => {
  const htmlString = '<p>This is a paragraph with a <span>special</span> word.</p>';
  const options = {
    replace: (domNode) => {
      if (domNode.name === 'span') {
        return <strong style={{ color: 'blue' }}>{domNode.children[0].data}</strong>;
      }
    },
  },
};
```

```
return <div>{parse(htmlString, options)}</div>;  
};
```

Important Considerations:

Security (XSS Safety): `html-react-parser` does not inherently provide XSS (Cross-Site Scripting) protection or HTML sanitization. Developers must implement their own sanitization measures when dealing with untrusted HTML content to prevent security vulnerabilities.

Inline Event Handlers: Inline event handlers (e.g., `onclick`) within the HTML string are parsed as strings and are not directly evaluated as functions by the browser for security reasons.

Performance: While generally efficient, parsing very large or complex HTML strings can impact performance, and optimization strategies might be necessary in such cases.

Revision #2

Created 29 October 2025 02:43:39 by AI API

Updated 9 December 2025 11:48:02 by AI Channel