

# ReactDOMServer

**ReactDOMServer** オブジェクトはコンポーネントを静的なマークアップとして変換できるようにします。これは、一般的に Node サーバで使われます。

```
// ES modules
import ReactDOMServer from 'react-dom/server';
// CommonJS
var ReactDOMServer = require('react-dom/server');
```

## 概要

以下のメソッドはサーバとブラウザの両方の環境で使用できます：

- [renderToString\(\)](#)
  - [renderToStaticMarkup\(\)](#)
- 以下の追加のメソッドは**サーバでのみ利用可能な**パッケージ (stream) に依存しているため、ブラウザでは動作しません。
- [renderToNodeStream\(\)](#)
  - [renderToStaticNodeStream\(\)](#)

## リファレンス

### renderToString()

ReactDOMServer.[renderToString](#)(element)

React 要素を初期状態の HTML へと変換します。React は HTML 文字列を返します。このメソッドにより、サーバ上で HTML を生成して最初のリクエストに対してマークアップを送信してページ読み込み速度を向上させたり、また SEO 目的で検索エンジンがページを巡回することを可能にします。

このようにしてサーバ側で変換されたマークアップをあらかじめ持つノード上で [ReactDOM.hydrate\(\)](#) を呼び出した場合、React はマークアップを保持しつつイベントハンドラのみを追加するので、非常にパフォーマンスの高い初回ページロードの体験が得られます。

### renderToStaticMarkup()

ReactDOMServer.[renderToStaticMarkup](#)(element)

React が内部的に使用する `data-reactroot` のような追加の DOM 属性を作成しないことを除いて、[renderToString](#) と同様の動作をします。このメソッドは React を単純な静的サイトジェネレータとして使用したい場合に便利で、追加の属性を省略することでバイト数を削減できます。

マークアップをインタラクティブなものにするために、クライアントで React を導入しようとしている場合は、このメソッドを使用しないでください。代わりに、サーバで [renderToString](#) を、そしてクライアントで [ReactDOM.hydrate\(\)](#) を使用してください。

### renderToNodeStream()

ReactDOMServer.[renderToNodeStream](#)(element)

React 要素を初期状態の HTML へと変換します。HTML の文字列を出力する [Readable ストリーム](#) を返します。このストリームによる HTML 出力は [ReactDOMServer.renderToString](#) が返すものと全く同じです。このメソッドにより、サーバ上で HTML を生成して最初のリクエストに対してマークアップを送信してページ読み込み速度を向上させたり、また SEO 目的で検索エンジンがページを巡回することを可能にします。

このようにしてサーバ側で変換されたマークアップをあらかじめ持つノード上で [ReactDOM.hydrate\(\)](#) を呼び出した場合、React はマークアップを保持しつつイベントハンドラのみを追加するので、非常にパフォーマンスの高い初回ページロードの体験が得られます。

#### 補足：

この API はサーバ専用です。ブラウザでは利用できません。

このメソッドが返すストリームは UTF-8 でエンコードされたバイトストリームを返します。別の方式でエンコードされたストリームが必要な場合、テキストのトランスコーディングのためにストリーム変換を提供している [iconv-lite](#) のようなプロジェクトを参照してください。

### renderToStaticNodeStream()

`ReactDOMServer.renderToStaticNodeStream(element)`

React が内部的に使用する `data-reactroot` のような追加の DOM 属性を作成しないことを除いて、`renderToNodeStream` と同様の動作をします。このメソッドは React を単純な静的サイトジェネレータとして使用したい場合に便利で、追加の属性を省略することでバイト数を削減できます。

このストリームによる HTML 出力は `ReactDOMServer.renderToStaticMarkup` が返すものと全く同じです。

マークアップをインタラクティブなものにするために、クライアントで React を導入しようとしている場合は、このメソッドを使用しないでください。代わりに、サーバで `renderToNodeStream` を、そしてクライアントで `ReactDOM.hydrate()` を使用してください。

#### 補足：

この API はサーバ専用です。ブラウザでは利用できません。

このメソッドが返すストリームは UTF-8 でエンコードされたバイトストリームを返します。別の方式でエンコードされたストリームが必要な場合、テキストのトランスコーディングのためにストリーム変換を提供している [iconv-lite](#) のようなプロジェクトを参照してください。

[このページを編集する](#)