resumeAndPrerender

`resumeAndPrerender` continues a prerendered React tree to a static HTML string using a Web Stream.

```
const { prelude,postpone } = await resumeAndPrerender(reactNode, postpone
```

중요합니다!

This API depends on Web Streams. For Node.js, use `resumeAndPrerenderToNodeStream` instead.

Reference

`resumeAndPrerender(reactNode, postponedState, options?)`

Call `resumeAndPrerender` to continue a prerendered React tree to a static HTML string.

```
import { resumeAndPrerender } from 'react-dom/static';
import { getPostponedState } from 'storage';

async function handler(request, response) {
  const postponedState = getPostponedState(request);
  const { prelude } = await resumeAndPrerender(<App />, postponedState, {
    bootstrapScripts: ['/main.js']
  });
  return new Response(prelude, {
    headers: { 'content-type': 'text/html' },
  });
}
```

On the client, call `hydrateRoot` to make the server-generated HTML interactive.

See more examples below.

Parameters

- `reactNode` : The React node you called `prerender` (or a previous `resumeAndPrerender`) with. For example, a JSX element like `<App />` . It is expected to represent the entire document, so the `App` component should render the `<html>` tag.
- `postponedState` : The opaque `postpone` object returned from a prerender API, loaded from wherever you stored it (e.g. redis, a file, or S3).
- **optional** `options` : An object with streaming options.
    - **optional** `signal` : An abort signal that lets you abort server rendering and render the rest on the client.
    - **optional** `onError` : A callback that fires whenever there is a server error, whether recoverable or not. By default, this only calls `console.error` . If you override it to log crash reports, make sure that you still call `console.error` .

Returns

`prerender` returns a Promise:

- If rendering the is successful, the Promise will resolve to an object containing:

    - `prelude`: a Web Stream of HTML. You can use this stream to send a response in chunks, or you can read the entire stream into a string.
    - `postponed`: an JSON-serializeable, opaque object that can be passed to `resume` or `resumeAndPrerender` if `prerender` is aborted.
- If rendering fails, the Promise will be rejected. Use this to output a fallback shell.

Caveats

`nonce` is not an available option when prerendering. Nonces must be unique per request and if you use nonces to secure your application with CSP it would be inappropriate and insecure to include the nonce value in the prerender itself.

중요합니다!

When should I use `resumeAndPrerender`?

The static `resumeAndPrerender` API is used for static server-side generation (SSG). Unlike `renderToString`, `resumeAndPrerender` waits for all data to load before resolving. This makes it suitable for generating static HTML for a full page, including data that needs to be fetched using Suspense. To stream content as it loads, use a streaming server-side render (SSR) API like renderToReadableStream.

`resumeAndPrerender` can be aborted and later either continued with another `resumeAndPrerender` or resumed with `resume` to support partial pre-rendering.

Usage

Further reading

`resumeAndPrerender` behaves similarly to `prerender` but can be used to continue a previously started prerendering process that was aborted.

For more information about resuming a prerendered tree, see the resume documentation.

이전
prerenderToNodeStream

다음
resumeAndPrerenderToNodeStream

React 학습하기

빠르게 시작하기
설치하기
UI 표현하기
상호작용성 더하기
State 관리하기
탈출구

API 참고서

React APIs
React DOM APIs

커뮤니티

행동 강령
팀 소개
문서 기여자
감사의 말

더 보기

블로그
React Native
개인 정보 보호
약관