

useRequestContext

August 17, 2026

useRequestContext

Kind: Function

Source: `src/middleware/jsx-renderer/index.ts`

Part of: [Middleware](#)

useRequestContext for Hono.

`useRequestContext` returns the current Hono request context inside a JSX component rendered by the JSX renderer middleware. Use it to read request data, headers, parameters, and environment bindings while generating JSX output.

Signature

```
function useRequestContext(): Context<E, P, I>
```

Returns: `Context<E, P, I>`

Diagram

```
graph LR
  Request[Incoming request] --> Renderer[JSX renderer middleware]
  Renderer --> Component[JSX component]
  Component --> Hook[useRequestContext]
  Hook --> Context[Hono Context]
```

Usage

```
import { Hono } from 'hono'
import { jsxRenderer, useRequestContext } from 'hono/jsx-renderer'

const app = new Hono()
```

```

const RequestInfo = () => {
  const c = useRequestContext()

  return (
    <main>
      <h1>{c.req.method} {c.req.path}</h1>
      <p>Request ID: {c.req.header('x-request-id') ?? 'not provided'}</p>
    </main>
  )
}

app.get(
  '*',
  jsxRenderer(({ children }) => (
    <html>
      <body>{children}</body>
    </html>
  ))
)

app.get('/', (c) => c.render(<RequestInfo />))

```

AI Coding Instructions

- Call `useRequestContext` only while rendering a JSX component under `jsxRenderer`.
- Read request values through the returned Hono context, such as `c.req.path`, `c.req.param()`, and `c.req.header()`.
- Do not pass the request context through component props when the component can access it with this hook.
- Keep request-dependent rendering inside components that run during the current request.