

contextStorage

August 18, 2026

contextStorage

Kind: Function

Source: `src/middleware/context-storage/index.ts`

Part of: [Middleware](#)

Context Storage Middleware for Hono.

`contextStorage` creates middleware that stores the active Hono `Context` for the lifetime of a request. Code running within that request can access the same context through `getContext()` without passing `c` through each function call.

Signature

```
function contextStorage(): MiddlewareHandler
```

Returns: `MiddlewareHandler`

Diagram

```
graph LR
  Request[Incoming request] --> Middleware[contextStorage middleware]
  Middleware --> Handler[Route handler]
  Handler --> Service[Application code]
  Service --> GetContext[getContext()]
  GetContext --> Context[Active Hono Context]
```

Usage

```
import { Hono } from 'hono'
import { contextStorage, getContext } from 'hono/context-storage'

const app = new Hono()
```

```

app.use(contextStorage())

function getRequestPath() {
  return getContext().req.path
}

app.get('/status', (c) => {
  return c.json({
    path: getRequestPath(),
  })
})

export default app

```

AI Coding Instructions

- Register `contextStorage()` before routes or middleware that call `getContext()` .
- Call `getContext()` only while handling an active request; do not retain the returned context for later work.
- Prefer the route handler's `c` parameter when it is already available; use context storage for code paths where passing `c` is impractical.
- Keep request-specific values on the Hono context so concurrent requests remain isolated.