

logger

August 17, 2026

logger

Kind: Function**Source:** `src/middleware/logger/index.ts`**Part of:** [Middleware](#)

Logger Middleware for Hono.

`logger` creates Hono middleware that writes request and response log entries during request handling. It records the request method and path before execution, then logs the response status and elapsed time after downstream middleware completes.

Signature

```
function logger(fn: PrintFunc): MiddlewareHandler
```

Parameters

Name	Type
<code>fn</code>	<code>PrintFunc</code>

Returns: `MiddlewareHandler`

Diagram

```
graph LR
    Request[Incoming request] --> Logger[logger middleware]
    Logger --> StartLog[Write request log]
    StartLog --> Next[Run downstream handler]
    Next --> ResponseLog[Write response log]
    ResponseLog --> Response[Send response]
```

Usage

```
import { Hono } from 'hono'
import { logger } from 'hono/logger'

const app = new Hono()

app.use('*', logger())

app.get('/', (c) => {
  return c.text('Hello from Hono')
})

export default app
```

AI Coding Instructions

- Register `logger()` with `app.use()` so it runs before the routes or middleware you want to observe.
- Place the logger early in the middleware chain to include downstream handler execution time in its output.
- Keep downstream middleware calling `await next()` so the logger can record the final response status.
- Pass a custom print function to `logger` when logs need to be sent to an application-specific logging destination.

Relationships

- IMPORTS → `getColorEnabledAsync`