compose

August 17, 2026

compose

Kind: Function

Source: `src/compose.ts`

Compose middleware functions into a single function based on `koa-compose` package.

`compose` combines middleware functions into one middleware handler using the dispatch pattern from `koa-compose`. Each middleware can run logic before and after `await next()`, allowing request handling, state changes, and cleanup to flow through the chain.

Signature

```
function compose(middleware: [[Function, unknown], unknown][] | [[Function]][], onError: ErrorH
```

Parameters

Name	Type
`middleware`	`[[Function, unknown], unknown][]
`onError`	`ErrorHandler<E>`
`onNotFound`	`NotFoundHandler<E>`

Returns: `((context: Context, next?: Next) => Promise<Context>)`

Diagram

```
graph LR
  A[Composed middleware] --> B[Middleware A]
  B --> C[Middleware B]
  C --> D[Middleware C]
  D --> E[Final next handler]
```

```
  E --> D
  D --> C
  C --> B
  B --> A
```

Usage

```ts
import { compose } from './compose'

type Context = {
  requestId?: string
  logs: string[]
}

const middleware = compose<Context>([
  async (context, next) => {
    context.logs.push('before request')
    await next()
    context.logs.push('after request')
  },
  async (context, next) => {
    context.requestId = crypto.randomUUID()
    await next()
  },
])

const context: Context = { logs: [] }

await middleware(context, async () => {
  context.logs.push(`handling ${context.requestId}`)
})

console.log(context.logs)
```

AI Coding Instructions

- Keep middleware signatures compatible with the composed handler: `(context, next) => Promise<void>`.
- Call and `await next()` when downstream middleware must run before post-processing logic.
- Do not call `next()` more than once from the same middleware invocation.
- Preserve middleware order when adding handlers; earlier middleware wraps later middleware.

- Pass a final `next` handler when composition must continue into code outside the middleware list.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `some` — `src/middleware/combine/index.ts`:38
- `HonoOptions` — `src/hono-base.ts`:46