

compress

August 17, 2026

compress

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

Compress Middleware for Hono.

`compress` creates Hono middleware that compresses eligible HTTP responses based on the client's `Accept-Encoding` header. Apply it to routes or the application to reduce response body size while preserving the response flow.

Signature

```
function compress(options: CompressionOptions): MiddlewareHandler
```

Parameters

Name	Type
<code>options</code>	<code>CompressionOptions</code>

Returns: `MiddlewareHandler`

Diagram

```
graph LR
  A[Client Request] --> B[Hono App]
  B --> C[Route Handler]
  C --> D[Response]
  D --> E["Accept-Encoding supports compression?"]
  E -->|Yes| F[compress middleware]
```

```
E -->|No| G[Uncompressed Response]
F --> H[Compressed Response]
```

Usage

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

const app = new Hono()

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

app.get('/data', (c) => {
  return c.json({
    message: 'This response can be compressed when supported by the client.',
  })
})

export default app
```

AI Coding Instructions

- Register `compress()` before route handlers when responses from those routes should be compressed.
- Keep response headers and bodies valid before middleware returns the final response.
- Test requests with different `Accept-Encoding` headers when changing compression behavior.
- Apply the middleware to a route pattern when only part of the application should return compressed responses.