

bearerAuth

August 18, 2026

bearerAuth

Kind: Function

Source: `src/middleware/bearer-auth/index.ts`

Part of: [Middleware](#)

Bearer Auth Middleware for Hono.

`bearerAuth` is Hono middleware that validates the Bearer token from an incoming request's `Authorization` header. Apply it to routes or an app to reject requests with missing or invalid tokens before the route handler runs.

Signature

```
function bearerAuth(options: BearerAuthOptions<E>): MiddlewareHandler<E>
```

Parameters

Name	Type
<code>options</code>	<code>BearerAuthOptions<E></code>

Returns: `MiddlewareHandler<E>`

Diagram

```
graph LR
    Request[Incoming request] --> Middleware[bearerAuth middleware]
    Middleware --> Header[Authorization header]
    Header --> Token[Bearer token validation]
    Token -->|Valid| Handler[Route handler]
    Token -->|Missing or invalid| Unauthorized[Unauthorized response]
```

Usage

```
import { Hono } from 'hono'
import { bearerAuth } from 'hono/bearer-auth'

const app = new Hono()

app.use(
  '/api/*',
  bearerAuth({
    token: 'my-secret-token',
  })
)

app.get('/api/profile', (c) => {
  return c.json({ message: 'Authenticated request' })
})

export default app
```

AI Coding Instructions

- Apply `bearerAuth` before handlers that require authenticated requests.
- Pass the expected token with the `token` option, or provide token verification logic when authentication rules need custom validation.
- Send credentials through the `Authorization: Bearer <token>` request header.
- Scope middleware to protected route prefixes when public routes must remain accessible.

Relationships

- IMPORTS → `HTTPException`
- IMPORTS → `timingSafeEqual`