

methodNotAllowed

August 17, 2026

methodNotAllowed

Kind: Function

Source: `src/middleware/method-not-allowed/index.ts`

Part of: [Middleware](#)

Method Not Allowed Middleware for Hono.

Returns a `405 Method Not Allowed` response with an `Allow` header when the request path matches a registered route but the request method is not supported.

`methodNotAllowed` creates Hono middleware that returns a Method Not Allowed response when a request matches a registered path but uses an unsupported HTTP method. It adds an `Allow` header containing the methods registered for that path.

Signature

```
function methodNotAllowed(options: MethodNotAllowedOptions<E>): MiddlewareHandler<E>
```

Parameters

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

Returns: `MiddlewareHandler<E>`

Diagram

```
graph LR
  Request --> Router
  Router -->|Path and method match| RouteHandler
```

```
Router -->|Path matches, method does not| MethodNotAllowed
MethodNotAllowed --> AllowHeader
AllowHeader --> Response
```

Usage

```
import { Hono } from 'hono'
import { methodNotAllowed } from 'hono/method-not-allowed'

const app = new Hono()

app.use(methodNotAllowed())

app.post('/entries', (c) => {
  return c.json({ created: true })
})

export default app
```

API Coding Instructions

- Register `methodNotAllowed()` before route definitions so it can inspect routing results after downstream handlers run.
- Keep route methods explicit with `app.get`, `app.post`, and related route APIs so the middleware can produce the correct `Allow` header.
- Do not replace the Method Not Allowed response or remove its `Allow` header in later middleware.
- Use this middleware for applications that need to distinguish an unknown path from a known path with an unsupported method.