

# methodOverride

August 19, 2026

## methodOverride

**Kind:** Function**Source:** `src/middleware/method-override/index.ts`**Part of:** Middleware

Method Override Middleware for Hono.

`methodOverride` is Hono middleware that changes the request method before route handling. It reads an override method from the request and dispatches the request through the supplied Hono app, allowing clients that can only send `POST` requests to reach routes such as `DELETE` or `PATCH`.

## Signature

```
function methodOverride(options: MethodOverrideOptions): MiddlewareHandler
```

## Parameters

| Name                 | Type                               |
|----------------------|------------------------------------|
| <code>options</code> | <code>MethodOverrideOptions</code> |

**Returns:** `MiddlewareHandler`

## Diagram

```
graph LR
  Client[Client POST request] --> Middleware[methodOverride middleware]
  Middleware --> Override[Read method override]
  Override --> App[Hono app]
  App --> Route[Route matching overridden method]
  Route --> Response[Response]
```

## Usage

```
import { Hono } from 'hono'
import { methodOverride } from 'hono/method-override'

const app = new Hono()

app.use('/posts/*', methodOverride({ app }))

app.delete('/posts/:id', (c) => {
  return c.text(`Deleted post ${c.req.param('id')}`)
})

// Send a POST request to:
// /posts/123?_method=DELETE
export default app
```

## AI Coding Instructions

- Pass the same `Hono` app instance to `methodOverride({ app })`.
- Mount the middleware on paths that need method override handling.
- Keep the target route declared with its actual HTTP method, such as `DELETE` or `PATCH`.
- Send an override value that matches the method expected by the target route.
- Test both the original request method and the overridden route behavior when changing middleware order.