



ELSEWHERE

cloneRawRequest

Kind: Function

VERSION
5

UPDATED
12 August 2026

DOCUMENT
reference-elsewhere/function-clonerawrequest

CONTENTS

1	Signature	2	4	Usage	3
2	Parameters	2	5	AI Coding Instructions	3
3	Diagram	2	6	Used by	3

Kind: Function

Source: `src/request.ts`

Clones a HonoRequest's underlying raw Request object.

This utility handles both consumed and unconsumed request bodies:

- If the request body hasn't been consumed, it uses the native `clone()` method
- If the request body has been consumed, it reconstructs a new Request using cached body data

This is particularly useful when you need to:

- Process the same request body multiple times
- Pass requests to external services after validation

`cloneRawRequest` creates a new native `Request` from a `HonoRequest` without losing access to its body. It calls `Request.clone()` when the body is unconsumed and rebuilds the request from Hono's cached body data when the body has already been read.

1 Signature

```
async function cloneRawRequest(req: HonoRequest): Promise<Request>
```

2 Parameters

Name	Type
<code>req</code>	<code>HonoRequest</code>

Returns: `Promise<Request>`

3 Diagram

```
graph LR
  A[HonoRequest] --> B{Raw body consumed?}
  B -->|No| C[raw.clone()]
  B -->|Yes| D[Read cached body data]
  D --> E[Create new Request]
  C --> F[Cloned Request]
  E --> F
```

4 Usage

```
import { Hono } from 'hono'
import { cloneRawRequest } from './request'

const app = new Hono()

app.post('/validate-and-forward', async (c) => {
  const payload = await c.req.json()

  if (!payload.email) {
    return c.json({ error: 'email is required' }, 400)
  }

  const requestToForward = cloneRawRequest(c.req)

  const response = await fetch('https://api.example.com/submit', {
    method: requestToForward.method,
    headers: requestToForward.headers,
    body: requestToForward.body,
    duplex: 'half',
  })

  return new Response(response.body, response)
})
```

5 AI Coding Instructions

- Pass the `HonoRequest` instance to `cloneRawRequest`, not only its `raw` property, so cached body data remains available.
- Use this function before forwarding a request when middleware or validation may have already read the body.
- Treat the returned value as a native `Request` and pass it to APIs such as `fetch`.
- Do not assume `request.raw.clone()` works after the request body has been consumed; use `cloneRawRequest` for that case.

6 Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `handler` — `src/middleware/cache/index.ts`:310