

NotFoundResponse

August 17, 2026

NotFoundResponse

Kind: Interface

Source: `src/types.ts`

You can extend this interface to define a custom `c.notFound()` Response type.

`NotFoundResponse` is a type extension point for the response returned by `c.notFound()`. Extend it through module augmentation to describe the JSON body your application returns for missing routes or resources.

Diagram

```
graph LR
  A[Module augmentation] --> B[NotFoundResponse]
  B --> C[c.notFound()]
  C --> D[Typed not-found response]
```

Usage

```
import { Hono } from 'hono'

declare module 'hono' {
  interface NotFoundResponse {
    error: 'NOT_FOUND'
    resource: string
  }
}

const app = new Hono()

app.notFound((c) => {
  return c.json(
    {
      error: 'NOT_FOUND',
      resource: c.req.path,
```

```
    },  
    404  
  )  
})  
  
app.get('/posts/:id', (c) => {  
  const post = null  
  
  if (!post) {  
    return c.notFound()  
  }  
  
  return c.json(post)  
})
```

AI Coding Instructions

- Extend `NotFoundResponse` with module augmentation instead of replacing the interface.
- Keep fields declared on `NotFoundResponse` aligned with the object returned by the configured not-found handler.
- Return `c.notFound()` from routes that cannot locate a requested resource.
- Define a shared not-found handler with `app.notFound()` when the application needs a consistent response body.