

HTTPException

August 17, 2026

HTTPException

Kind: Class**Source:** [src/http-exception.ts](#)

`HTTPException` must be used when a fatal error such as authentication failure occurs.

`HTTPException` represents a fatal HTTP failure, such as an authentication error. Throw it when request handling must stop and an existing `Response` should be returned to the caller.

Extends: `Error`

Methods

Method	Signature	Returns
<code>getResponse</code>	<code>getResponse()</code>	<code>Response</code>

Properties

Property	Type
<code>res</code>	<code>Response</code>
<code>status</code>	<code>ContentfulStatusCode</code>

Diagram

```
graph LR
  A[Request handler] --> B{Fatal HTTP error?}
  B -- Yes --> C[Create Response]
  C --> D[Throw HTTPException]
  D --> E[getResponse]
```

```
E --> F[Return Response]
B -- No --> G[Continue handling]
```

Usage

```
import { HTTPException } from './http-exception'

function requireAuthentication(token?: string): void {
  if (!token) {
    const response = new Response('Unauthorized', {
      status: 401,
      headers: {
        'content-type': 'text/plain',
      },
    })

    throw new HTTPException(response)
  }
}

async function handleRequest(request: Request): Promise<Response> {
  try {
    requireAuthentication(request.headers.get('authorization') ?? undefined)

    return new Response('OK')
  } catch (error) {
    if (error instanceof HTTPException) {
      return error.getResponse()
    }

    throw error
  }
}
```

AI Coding Instructions

- Throw `HTTPException` only for fatal request failures that already have a `Response` to return.
- Build the response status, headers, and body before creating the exception.
- Catch `HTTPException` at the request-handling boundary and return `getResponse()`.
- Do not replace unrelated runtime errors with `HTTPException`; allow unexpected errors to follow normal error handling.

Used by

12 references from 12 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (12)

- `handler` — `src/middleware/basic-auth/index.ts` :118
- `bearerAuth` — `src/middleware/bearer-auth/index.ts` :104
- `handler` — `src/middleware/body-limit/index.ts` :72
- `csrf` — `src/middleware/csrf/index.ts` :94
- `IPRestrictionRule` — `src/middleware/ip-restriction/index.ts` :38
- `headerName` — `src/middleware/jwk/jwk.ts` :81
- `headerName` — `src/middleware/jwt/jwt.ts` :81
- `HTTPExceptionFunction` — `src/middleware/timeout/index.ts` :10

...and 4 more.