

HTTPResponseError

August 17, 2026

HTTPResponseError

Kind: Interface**Source:** [src/types.ts](#)

`HTTPResponseError` describes an error object that exposes a `Response` through `getResponse()`. Error-handling code can detect this interface and return the associated response instead of creating a new error response.

Properties

Property	Type
<code>getResponse</code>	<code>() => Response</code>

Diagram

```
graph LR
  Error[Thrown error] --> Check{Has getResponse?}
  Check -->|Yes| HTTPResponseError[HTTPResponseError]
  HTTPResponseError --> Response[getResponse returns Response]
  Check -->|No| Fallback[Default error handling]
```

Usage

```
interface HTTPResponseError {
  getResponse: () => Response
}

const isHTTPResponseError = (error: unknown): error is HTTPResponseError => {
  return (
    typeof error === 'object' &&
    error !== null &&
    'getResponse' in error &&
    typeof error.getResponse === 'function'
  )
}
```

```

    )
  }

  try {
    throw {
      getResponse: () => new Response('Access denied'),
    }
  } catch (error) {
    if (isHTTPResponseError(error)) {
      return error.getResponse()
    }

    return new Response('Unexpected error')
  }
}

```

AI Coding Instructions

- Implement `getResponse` as a function that returns a `Response` instance.
- Check unknown thrown values before calling `getResponse`.
- Return the response from `getResponse` directly when handling this error type.
- Keep response construction inside the error when the error defines its own HTTP output.

How it works

`HTTPResponseError` is an exported TypeScript interface for an `Error` that also has a `getResponse()` method returning a web `Response`. It has no constructor or runtime implementation in `src/types.ts`; it only describes this structural shape.

[src/types.ts:113-115](#)

An `ErrorHandler` receives either `Error` or `HTTPResponseError` plus a `Context`, and must return a `Response` or a promise of one. [src/types.ts:116-119](#)