

HttpExceptionOptions

August 18, 2026

HttpExceptionOptions

Kind: Interface

Source: `packages/common/exceptions/http.exception.ts`

Part of: `Common`

`HttpExceptionOptions` defines optional metadata that can be attached to an HTTP exception. It provides a human-readable `description` and an underlying `cause` value, helping applications preserve diagnostic context when errors are created and handled.

Properties

Property	Type
<code>cause</code>	<code>unknown</code>
<code>description</code>	<code>string</code>

Diagram

```
graph LR
  A[Application error] --> B[HttpExceptionOptions]
  B --> C[cause: unknown]
  B --> D[description: string]
  B --> E[HttpException]
  E --> F[Exception handler / HTTP response]
```

Usage

```
import { HttpException, HttpStatus } from '@nestjs/common';

function findUser(userId: string) {
  try {
```

```
// Example operation that may fail
throw new Error(`User ${userId} was not found`);
} catch (error) {
  throw new HttpException(
    'Unable to retrieve user',
    HttpStatus.NOT_FOUND,
    {
      description: 'The requested user does not exist or cannot be accessed.',
      cause: error,
    },
  );
}
```

AI Coding Instructions

- Pass `HttpExceptionOptions` as the third argument when constructing an `HttpException`.
- Use `description` for stable, developer-facing context; do not rely on it as the client response message.
- Preserve the original thrown value in `cause` to support logging, tracing, and error debugging.
- Treat `cause` as `unknown`; narrow its type before accessing properties such as `message` or `stack`.
- Avoid placing sensitive data, credentials, or internal implementation details in exception descriptions.