

HttpException

August 17, 2026

HttpException

Kind: Class**Source:** `packages/common/exceptions/http.exception.ts`**Part of:** Common

Defines the base Nest HTTP exception, which is handled by the default Exceptions Handler.

`HttpException` is Nest's base error type for representing HTTP failures with a response payload and status code. When thrown from application code, it is processed by Nest's default exceptions handler to produce a structured HTTP error response.

Extends: `IntrinsicException`

Methods

Method	Signature	Returns
<code>initCause</code>	<code>initCause()</code>	<code>void</code>
<code>initMessage</code>	<code>initMessage()</code>	<code>void</code>
<code>initName</code>	<code>initName()</code>	<code>void</code>
<code>getResponse</code>	<code>getResponse()</code>	<code>`string</code>
<code>getStatus</code>	<code>getStatus()</code>	<code>number</code>
<code>createBody</code>	<code>`createBody(nil: null</code>	<code>", message: HttpExceptionBodyMessage, statusCode: number)`</code>
<code>createBody</code>	<code>createBody(message: HttpExceptionBodyMessage, error: string, statusCode: number)</code>	<code>HttpExceptionBody</code>
<code>createBody</code>	<code>createBody(custom: Body)</code>	<code>Body</code>
<code>createBody</code>	<code>`createBody(arg0: null</code>	<code>HttpExceptionBodyMessage</code>
<code>getDescriptionFrom</code>	<code>`getDescriptionFrom(descriptionOrOptions: string</code>	<code>HttpExceptionOptions)`</code>
<code>getHttpExceptionOptionsFrom</code>	<code>`getHttpExceptionOptionsFrom(descriptionOrOptions: string</code>	<code>HttpExceptionOptions)`</code>
<code>extractDescriptionAndOptionsFrom</code>	<code>`extractDescriptionAndOptionsFrom(descriptionOrOptions: string</code>	<code>HttpExceptionOptions)`</code>

Properties

Property	Type
cause	unknown

Where it refuses work

- `HttpException` stops the work with an early return when `!arg0` .
- `HttpException` stops the work with an early return when `isString(arg0) || Array.isArray(arg0) || isNumber(arg0)` .

Diagram

```
graph LR
  A[Controller / Service] -->|throw new HttpException| B[HttpException]
  B --> C[getStatus()]
  B --> D[getResponse()]
  B --> E[Default Exceptions Handler]
  C --> E
  D --> E
  E --> F[HTTP Error Response]
```

Usage

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

@Controller('reports')
export class ReportsController {
  @Get()
  getReport() {
    const reportAvailable = false;

    if (!reportAvailable) {
      throw new HttpException(
        {
          statusCode: HttpStatus.NOT_FOUND,
          message: 'Report not found',
          error: 'Not Found',
        },
        HttpStatus.NOT_FOUND,
      );
    }

    return { id: 'report-123' };
  }
}
```

AI Coding Instructions

- Throw `HttpException` when a request should end with a specific HTTP status and response body.
- Use Nest's `HttpStatus` enum instead of hard-coded numeric status codes.
- Pass either a string or a structured object as the response; `getResponse()` returns the original payload.

- Prefer specialized exceptions such as `NotFoundException` or `BadRequestException` when they match the intended status.
- Use `HttpException.createBody()` when building standardized error response payloads for custom exceptions.

Used by

10 references from 10 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (10)

- `RoutesResolver` — `packages/core/router/routes-resolver.ts` :35
- `ErrorsInterceptor` — `sample/01-cats-app/src/common/interceptors/exception.interceptor.ts` :12
- `ExceptionHandler` — `sample/10-fastify/src/common/interceptors/exception.interceptor.ts` :12
- `ErrorsInterceptor` — `sample/36-hmr-esm/src/common/interceptors/exception.interceptor.ts` :12
- `HttpExceptionFilter` — `integration/inspector/src/common/filters/http-exception.filter.ts` :8
- `BaseExceptionHandler` — `packages/core/exceptions/base-exception-filter.ts` :17
- `ExceptionsHandler` — `packages/core/exceptions/exceptions-handler.ts` :9
- `transformException` — `packages/platform-express/multer/multer/multer.utils.ts` :10

...and 2 more.