

PayloadTooLargeException

August 18, 2026

PayloadTooLargeException

Kind: Class

Source: `packages/common/exceptions/payload-too-large.exception.ts`

Part of: [Common](#)

Defines an HTTP exception for *Payload Too Large* type errors.

`PayloadTooLargeException` represents an HTTP 413 error when a client sends a request body that exceeds the server's accepted size limit. It extends NestJS's HTTP exception system, producing a standardized error response that can be handled by the framework's exception filters.

Extends: `HttpException`

Diagram

```
graph LR
  Client[Client sends request payload] --> Limit{Payload within allowed size?}
  Limit -->|Yes| Handler[Handler[Process request]]
  Limit -->|No| Exception[Exception[PayloadTooLargeException]]
  Exception --> Response[Response[HTTP 413 Payload Too Large response]]
```

Usage

```
import { PayloadTooLargeException } from '@nestjs/common';

function validatePayloadSize(payload: string) {
  const maxBytes = 1024 * 1024; // 1 MB

  if (Buffer.byteLength(payload, 'utf8') > maxBytes) {
    throw new PayloadTooLargeException(
      'Request payload must not exceed 1 MB',
    );
  }
}
```

```
}  
}
```

AI Coding Instructions

- Throw `PayloadTooLargeException` when a request body, uploaded content, or serialized payload exceeds an enforced size limit.
- Prefer a clear client-facing message that explains the limit or how the request can be reduced.
- Do not use this exception for malformed payloads; use a validation-oriented exception such as `BadRequestException` instead.
- Configure body-parser or upload middleware limits alongside application-level checks, since middleware may reject oversized requests before controllers run.
- Preserve NestJS exception patterns by throwing the exception rather than manually constructing HTTP 413 responses.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `transformException` — `packages/platform-express/multer/multer/multer.utils.ts :10`