

GoneException

August 17, 2026

GoneException

Kind: Class

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

Part of: [Common](#)

Defines an HTTP exception for *Gone* type errors.

`GoneException` represents an HTTP **410 Gone** error, indicating that a requested resource previously existed but is no longer available. Use it in application services or controllers when a resource has been permanently removed and clients should not retry the same request expecting it to reappear.

Extends: `HttpException`

Diagram

```
graph LR
    Client[Client Request] --> Controller[Controller or Service]
    Controller --> Check{Resource permanently removed?}
    Check -- Yes --> Gone[GoneException]
    Gone --> Response[HTTP 410 Gone Response]
    Check -- No --> Normal[Continue normal handling]
```

Usage

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

async function getArchivedDocument(id: string) {
  const document = await documentRepository.findById(id);

  if (!document || document.isPermanentlyDeleted) {
    throw new GoneException(`Document "${id}" is no longer available`);
  }
}
```

```
return document;  
}
```

AI Coding Instructions

- Throw `GoneException` only when a resource is known to be permanently unavailable; use `NotFoundException` when its existence is unknown.
- Include a clear message or response body that helps clients understand why the resource cannot be retrieved.
- Use this exception in controller or service flows where the framework can translate it into an HTTP 410 response.
- Do not use HTTP 410 for temporary unavailability; prefer appropriate retryable error handling for transient failures.