

MisdirectedException

August 18, 2026

MisdirectedException

Kind: Class

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

Part of: [Common](#)

Defines an HTTP exception for *Misdirected* type errors.

`MisdirectedException` represents an HTTP `421 Misdirected Request` error, indicating that a request was sent to a server or connection that cannot serve the requested resource. Use it when request routing, host validation, or connection-level handling determines that the request should be handled by a different server or endpoint.

Extends: `HttpException`

Diagram

```
graph LR
  A[Incoming HTTP Request] --> B{Request targets correct server?}
  B -- Yes --> C[Continue request handling]
  B -- No --> D[Throw MisdirectedException]
  D --> E[HTTP 421 Misdirected Request Response]
```

Usage

```
import { Controller, Get, Headers } from '@nestjs/common';
import { MisdirectedException } from '@nestjs/common';

@Controller('api')
export class ApiController {
  @Get()
  getData(@Headers('host') host?: string) {
    if (host !== 'api.example.com') {
      throw new MisdirectedException(
        'This request was sent to the wrong host.',
      );
    }
  }
}
```

```
    );  
  }  
  
  return { message: 'Request handled successfully.' };  
}  
}
```

AI Coding Instructions

- Throw `MisdirectedException` only for HTTP `421` scenarios, such as invalid host routing or requests received by the wrong server connection.
- Prefer a clear, client-safe error message that explains why the request cannot be handled by the current target.
- Do not use this exception for missing routes, authorization failures, or generic proxy errors; use the corresponding HTTP exception instead.
- Ensure reverse-proxy, gateway, and host-header validation logic consistently identifies the expected target before throwing this exception.