

CallHandler

August 18, 2026

CallHandler

Kind: Interface

Source: `packages/common/interfaces/features/nest-interceptor.interface.ts`

Part of: [Common](#)

Interface providing access to the response stream.

`CallHandler` represents the downstream execution pipeline in a NestJS interceptor. It provides access to the response stream through `handle()`, allowing interceptors to transform, observe, or replace values returned by route handlers.

Diagram

```
graph LR
  Client[Client Request] --> Interceptor[Nest Interceptor]
  Interceptor --> Handler[Handler]
  Handler --> Controller[Route Handler]
  Controller --> Stream[Stream]
  Stream --> Interceptor
  Interceptor --> Client
```

Usage

```
import {
  CallHandler,
  ExecutionContext,
  Injectable,
  NestInterceptor,
} from '@nestjs/common';
import { Observable, map } from 'rxjs';

@Injectable()
export class ResponseWrapperInterceptor implements NestInterceptor {
  intercept(
```

```

    context: ExecutionContext,
    next: CallHandler,
  ): Observable<{ data: unknown }> {
    return next.handle().pipe(
      map((response) => ({
        data: response,
      })),
    );
  }
}

```

AI Coding Instructions

- Call `next.handle()` to continue execution to the controller or next interceptor and receive an `Observable` response stream.
- Use RxJS operators such as `map`, `tap`, `catchError`, and `timeout` to modify or monitor responses without subscribing manually.
- Return the transformed `Observable`; do not call `subscribe()` inside an interceptor.
- Treat `CallHandler` as the boundary between interceptor logic and the downstream route handler execution.

Used by

32 references from 32 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (32)

- `DataInterceptor` — `integration/graphql-code-first/src/common/interceptors/data.interceptor.ts :10`
- `Interceptor` — `integration/inspector/src/circular-hello/interceptors/logging.interceptor.ts :10`
- `TimeoutInterceptor` — `integration/inspector/src/common/interceptors/timeout.interceptor.ts :10`
- `LoggingInterceptor` — `integration/inspector/src/core/interceptors/logging.interceptor.ts :10`
- `TransformInterceptor` — `integration/inspector/src/core/interceptors/transform.interceptor.ts :14`
- `LoggingInterceptor` — `integration/inspector/src/request-chain/interceptors/logging.interceptor.ts :10`

- `PassthroughInterceptor` — `integration/nest-application/sse/src/app.controller.ts` :28
- `Interceptor` — `integration/scopes/src/circular-hello/interceptors/logging.interceptor.ts` :10

...and 24 more.