

ShutdownSignal

August 18, 2026

ShutdownSignal

Kind: Enum

Source: `packages/common/enums/shutdown-signal.enum.ts`

Part of: [Common](#)

System signals which shut down a process

`ShutdownSignal` defines the operating-system signals that can trigger graceful process termination. It provides a shared, type-safe set of signal names for shutdown hooks, lifecycle handlers, and process cleanup logic across the application.

Values

- `SIGHUP`
- `SIGINT`
- `SIGQUIT`
- `SIGILL`
- `SIGTRAP`
- `SIGABRT`
- `SIGBUS`
- `SIGFPE`
- `SIGSEGV`
- `SIGUSR2`
- `SIGTERM`

Diagram

```
graph LR
  OS[Operating System] --> Signal[Shutdown Signal]
  Signal --> Enum[ShutdownSignal enum]
```

```
Enum --> Handler[Application Shutdown Handler]
Handler --> Cleanup[Cleanup Resources]
Cleanup --> Exit[Process Exit]
```

Usage

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

function handleShutdown(signal: ShutdownSignal) {
  console.log(`Received ${signal}; closing application resources.`);
}

process.on(ShutdownSignal.SIGTERM, () => {
  handleShutdown(ShutdownSignal.SIGTERM);
});

process.on(ShutdownSignal.SIGINT, () => {
  handleShutdown(ShutdownSignal.SIGINT);
});
```

AI Coding Instructions

- Use `ShutdownSignal` instead of hard-coded signal strings when registering shutdown handlers or lifecycle hooks.
- Register cleanup logic for common termination signals such as `SIGTERM` and `SIGINT`.
- Ensure shutdown handlers close database connections, message consumers, HTTP servers, and other long-lived resources.
- Avoid assuming every signal is available on every operating system; verify platform compatibility when adding signal-specific behavior.
- Keep signal handlers idempotent, since multiple shutdown signals may be received during process termination.

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)

- `NestApplicationContext` — `packages/core/nest-application-context.ts` :40

