

isLogLevelEnabled

August 18, 2026

isLogLevelEnabled

Kind: Function

Source: `packages/common/services/utils/is-log-level-enabled.util.ts`

Part of: [Common](#)

Checks if target level is enabled.

`isLogLevelEnabled` determines whether a target log level should be emitted based on the configured logger levels. It supports direct level matches and level-threshold behavior, allowing logging components to consistently filter messages before they are written.

Signature

```
function isLogLevelEnabled(targetLevel: LogLevel, logLevels: LogLevel[] | undefined): boolean
```

Parameters

Name	Type
<code>targetLevel</code>	<code>LogLevel</code>
<code>logLevels</code>	<code>LogLevel[]</code>

Returns: `boolean`

Diagram

```
graph LR
  A["Target log level"] --> C["isLogLevelEnabled"]
  B["Configured log levels"] --> C
  C --> D["Configuration exists?"]
  D -- No --> E["Return false"]
```

```
D -- Yes --> F{Level explicitly enabled or allowed by threshold?}
F -- Yes --> G[Return true]
F -- No --> H[Return false]
```

Usage

```
import { isLogLevelEnabled } from '@nestjsjs/common/services/utils/is-log-level-enabled.util';

const enabledLevels = ['debug'];

if (isLogLevelEnabled('log', enabledLevels)) {
  console.log('Application started');
}

if (isLogLevelEnabled('verbose', enabledLevels)) {
  console.log('Detailed diagnostic output');
}
```

AI Coding Instructions

- Pass the requested log level as the first argument and the configured `LogLevel[]` collection as the second argument.
- Treat an empty or undefined configured level list as logging disabled; do not assume a default level is enabled.
- Reuse this utility before formatting or writing log messages to avoid unnecessary logging work.
- Preserve the established log-level ordering when changing supported levels, since threshold evaluation depends on that order.