ConsoleLoggerOptions

August 18, 2026

ConsoleLoggerOptions

Kind: Interface

Source: `packages/common/services/console-logger.service.ts`

Part of: Common

`ConsoleLoggerOptions` configures how the console logger formats and emits application log messages. It controls enabled log levels, timestamp and context metadata, JSON/color output, object inspection limits, and whether output is forced through the native console.

Properties

Property	Type
`logLevels`	`LogLevel[]`
`timestamp`	`boolean`
`prefix`	`string`
`json`	`boolean`
`colors`	`boolean`
`context`	`string`
`forceConsole`	`boolean`
`compact`	`boolean
`maxArrayLength`	`number`
`maxStringLength`	`number`
`sorted`	`boolean
`depth`	`number`
`showHidden`	`boolean`

Property	Type
`breakLength`	`number`

Diagram

```
graph LR
  A[ConsoleLoggerOptions] --> B[Filtering]
  A --> C[Formatting]
  A --> D[Output Behavior]
  A --> E[Inspection Limits]

  B --> B1[logLevels: LogLevel[]]

  C --> C1[timestamp]
  C --> C2[prefix]
  C --> C3[context]
  C --> C4[json]
  C --> C5[colors]
  C --> C6[compact]

  D --> D1[forceConsole]

  E --> E1[maxArrayLength]
  E --> E2[maxStringLength]
```

Usage

```
import type { ConsoleLoggerOptions } from '@nestjs/common';

const loggerOptions: ConsoleLoggerOptions = {
  logLevels: ['log', 'error', 'warn', 'debug'],
  timestamp: true,
  prefix: 'api',
  context: 'Bootstrap',
  json: false,
  colors: true,
  forceConsole: false,
  compact: true,
  maxArrayLength: 20,
  maxStringLength: 200,
};

// Pass the options to the application's console logger configuration.
console.log(loggerOptions);
```

AI Coding Instructions

- Use `logLevels` to limit emitted messages; include `error` and `warn` in production configurations unless intentionally suppressing them.
- Enable `json` for structured logging pipelines; avoid combining it with human-focused formatting expectations such as colorized terminal output.
- Set `context` and `prefix` consistently so log entries can be traced back to their module, service, or application.
- Use `maxArrayLength`, `maxStringLength`, and `compact` to prevent large objects or payloads from producing excessive console output.
- Set `forceConsole` only when native console output is required instead of the framework's configured logging transport.