

TestingLogger

August 17, 2026

TestingLogger

Kind: Class

Source: `packages/testing/services/testing-logger.service.ts`

Part of: `Testing`

`TestingLogger` is a test-focused logger implementation that provides the standard logging methods used throughout the application: `log` , `warn` , `debug` , `verbose` , and `error` . It allows tests to capture, inspect, or safely handle log output without depending on the production logging infrastructure.

Extends: `ConsoleLogger`

Methods

Method	Signature	Returns
<code>log</code>	<code>log(message: string)</code>	<code>void</code>
<code>warn</code>	<code>warn(message: string)</code>	<code>void</code>
<code>debug</code>	<code>debug(message: string)</code>	<code>void</code>
<code>verbose</code>	<code>verbose(message: string)</code>	<code>void</code>
<code>error</code>	<code>error(message: string, optionalParams: any[])</code>	<code>void</code>

Diagram



Warn --> Assertions
Debug --> Assertions
Verbose --> Assertions
Error --> Assertions

Usage

```
import { TestingLogger } from '@your-package/testing';

describe('ExampleService', () => {
  it('logs an expected message', () => {
    const logger = new TestingLogger();

    logger.log('Processing started');
    logger.warn('Optional configuration is missing');
    logger.debug('Request payload received');
    logger.verbose('Additional diagnostic details');
    logger.error('Processing failed');

    // Use the testing logger with the service under test.
    // const service = new ExampleService(logger);
  });
});
```

API Coding Instructions

- Use `TestingLogger` when unit or integration tests need a logger dependency instead of the production logger.
- Call the method that matches the intended log level: `log` for normal messages, `warn` for recoverable issues, and `error` for failures.
- Keep test logging deterministic; avoid relying on timestamps, environment-specific values, or external output destinations.
- When adding logger-compatible services, preserve support for all standard methods exposed by `TestingLogger`.
- Prefer assertions against expected logging behavior when logs represent important operational or error-handling outcomes.

How it works

`TestingLogger`

`TestingLogger` is a public class that extends Nest's `ConsoleLogger`. Its constructor initializes the inherited logger with the context string `"Testing"`.
[packages/testing/services/testing-logger.service.ts:3-9]

Relationships

- IMPORTS → `ConsoleLogger`