logging in angular project using NGX-LOGGER

TABLE OF CONTENTS

Introduction 3

Why is Logging important? 3

NGX LOGGER 3

Implementation 3

Step 1: Installation 3

Step 2: Configuration 4

Step 3: Logging 4

Configuration options 5

Logging in server 7

Environment based logging 8

Testing logging implemented application 9

Conclusion 9

Janani Priya T

# Introduction

This documentation provides guidance for implementing logging in Angular projects using NGX-LOGGER, procedure for installation, configuration options, procedure for implementing console logging and server-side logging in angular project and procedure to test a logging implemented application.

## Why is logging important?

Logging is important in any application as it makes debugging easier and facilitates monitoring in production environment. In development environment it provides context about the project execution which can be analyzed to accelerate the process of debugging. In production environment, effective logging aids in monitoring an application, diagnosing, and fixing issues. It becomes a source of execution details which can be referred to diagnose in case of a crash.

Logging in angular applications can be done with the help of ngx-logger.

## NGX LOGGER

NGX Logger is a npm package that enables angular logging in your application. It provides features such as

* Console logging
* Level based logging
* Logging in server
* Configure logging based on environment

# implementation

## Step 1: Installation

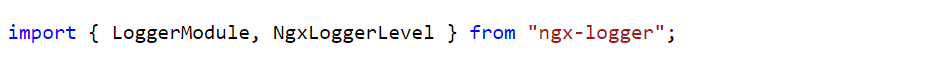
To add **ngx-logger** to your application, open your application, in the terminal run the below command to install ngx-logger.

A picture containing letter

Description automatically generated

## Step 2: Configuration

Once ngx-logger is installed we need to import the modules



next import **LoggerModule.forRoot()** in your application module by passing logging configuration to initialize the logger.

Text, letter

Description automatically generated

This completes the setup, now you can use logging in your application

## Step 3: Logging

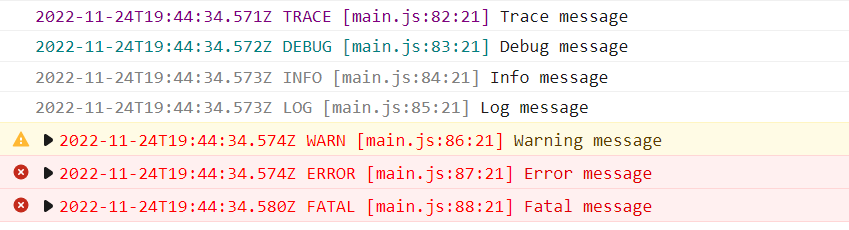
To perform logging import ngx-logger locally and inject it as a dependency in constructor to get ngx-logger instance.

Ngx-logger provides logging methods for each log level (discussed in next section). Implement logging as per requirements such us entry to a method or successful completion of an action, etc.

Graphical user interface, text, application

Description automatically generated

The following output can be seen in developer console.



# Configuration options

NGX LOGGER provides various config options for both console and server logging. The following are few of the config options:

**level: NgxLoggerLevel;**

log messages of this level or higher will only be written.

NgxLoggerLevel provides the following logging levels: TRACE, DEBUG, INFO, LOG, WARN, ERROR, FATAL and OFF

**timestampFormat?: string;**

format for timestamp in log message. Can be any format accepted by Angular Date Pipe.

**disableConsoleLogging?: boolean;**

If set to true, disables console logging

**colorScheme?: NGXLoggerColorScheme;**

can define color scheme for different log levels

NGXLoggerColorScheme type is an array of 7 colors for 7 log levels.

**disableFileDetails?: boolean;**

If set to true, the console log will exclude file details such as filename, line number and column number.

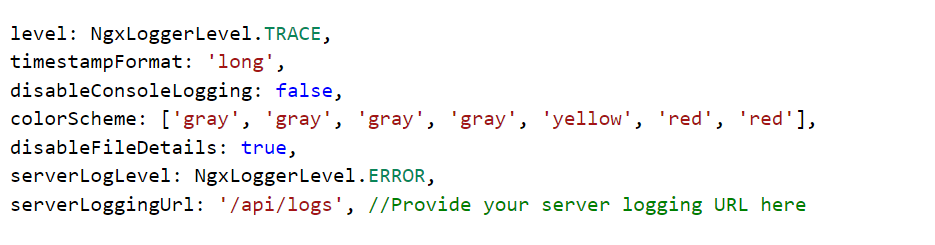
**serverLogLevel?: NgxLoggerLevel;**

log messages of this level or higher will only be sent to server.

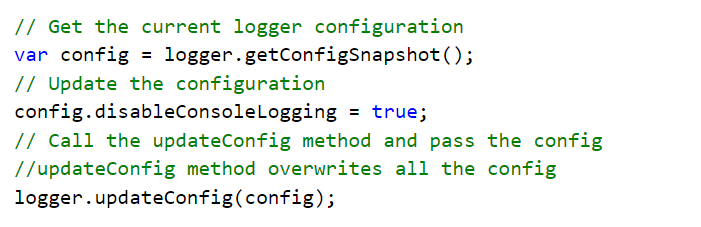
**serverLoggingUrl?: string;**

defines the URL used to send log to server.

An example of configuration options is shown below.



You can either specify the logging options in the forRoot() method call or in the updateConfig() method.



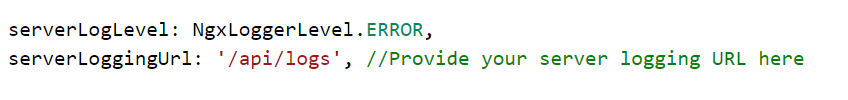
# Logging in server

Logging in server is done by sending the log message as POST request to the server. The server must define a POST method in the URL specified to handle the log messages received.

Import HttpClientModule in the main module.



Set the serverLogLevel and serverLoggingURL config options.



Logging method sends a post request to the requested URL. An example of the log details sent is shown below

Text

Description automatically generated

Server side implementation differs based on the platform used. An example of C# implementation is given below

Text

Description automatically generated

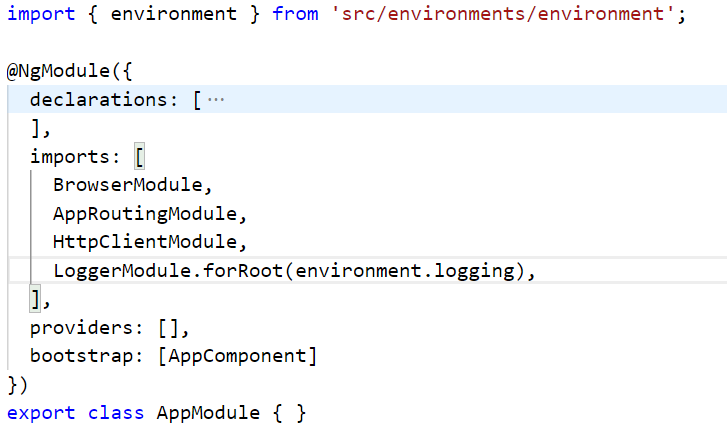
# environment based logging

Logging can be configured based on the environment. Add the logging configuration to the environment.ts file.

Text

Description automatically generated with low confidence

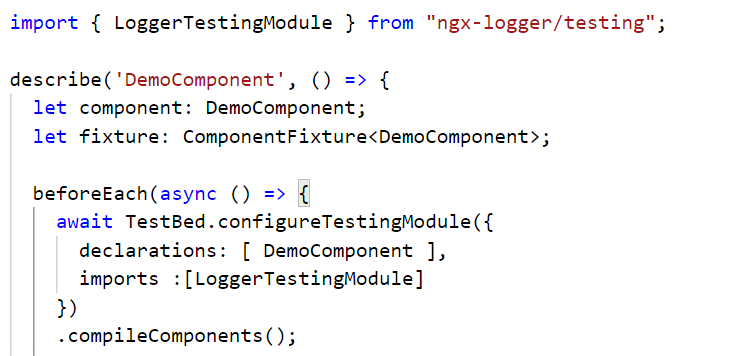
Change the import of logger module as such.



This will configure the logging based on the environment. The environment.ts file will be replaced based on the environment, for example, in production environment it will be replaced by environment.prod.ts file.

# testing logging implemented application

To test the NGX-LOGGER implemented application, import LoggerTestingModule in spec.ts file where the corresponding ts file is injected with NGX-LOGGER.



# Conclusion

NGX-LOGGER provides an easy implementation of logging in angular projects. The documentation has provided guidance for implementing NGX-LOGGER and has discussed features such as server-side logging, configuration and testing logging implemented application.