ESLINT

Introduction

Sreekanth M. E.

Freelance Trainer & Consultant

http://www.SreekanthME.com

Static code analysis is the analysis of source code that is performed without actually executing programs.

Analysis performed on executing programs is known as dynamic analysis.

Linting is a kind of static code analysis that analyzes code for potential errors.

Linting will run through source code to find:

- Formatting discrepancy

- Non-adherence to coding standards and conventions

- Pinpointing possible logical errors in the program

Without linting tools, a language like JS that is not compiled will end up showing issues only at execution time.

This will affect developer productivity.

ESLINT is written using Node.js to provide a fast runtime environment and easy installation via npm.

npm install -g eslint

Run below command to setup ESLint for a project:

eslint --init

eslint --init command accepts user inputs before generating a config file named .eslintrc in the local folder.

It provides options to either choose a standard style guide (like Google or Airbnb guides) or answer a detailed questionnaire.

The .eslintrc file can be edited to add/remove linting configurations including rules.

Every ESLint rule:

- Is standalone

- Can be turned off or on

- Can be set to a warning or error individually

The error level of the rule can be one of these values:

- **"off" or 0:** Turn the rule off

- **"warn" or 1:** Turn the rule on as a warning

- **"error" or 2:** Turn the rule on as an error

ESLint can be integrated with JS source code editors by installing ESLint extensions or plugins.

IMPORTANT NOTE:

When setting up ESLint for VS-Code, it is important to install ESLint in the local folder. Otherwise, it does not work.

The Airbnb guide package needs ESLint to be available locally.

END OF CHAPTER

APPENDIX