ESLINT

**Special doc:** <https://eslint.org/docs/latest/user-guide/getting-started>

**Rules: layout& formatting :** https://eslint.org/docs/latest/rules/

**Use FOR ESLint::** Resolved bugs maintain consistency of code /provide style

**NOTE::** Eslint always works with javascript, not with CSS or HTML or any language except js so we have a .eslintignore file for ignoring \*.css

**NOTE::** Easy to work :: Add extension as eslint (orange)

* add lintel extension in vscode open lintel file by ctrl+p , >lintel search like this
* Click on .eslintrc.json on bottom is set to top and you can also configure here

**Check any particular file for root:** npx eslint ./index.js (for any root structure)

**Check any file from folder like sr>foo.js || src>bar.js :** npx eslint src

**If the root has folder like src>foo.js check only that file:** npx eslint .\src\foo.js

**If warning there is automatically updated :** npx eslint ./index.js --fix

Above one we have to write npx eslint src for cmd or check we set into package.json

* Open package.json
* Write code on the script part
  + "lint": "eslint src",
  + "lint:fix":"eslint --fix", //most of small fix by default
* Use lint command by **npm run lint**
* Avoid writing npm run lint good to use adding extension fo eslint

STEPS FOR FOLLOW ESLINT

* Open vscode
* npm init
* npm i express
* npm init @eslint/config
  + Answer the question
* Package.json
* {
* "name": "typescript-demo",
* "version": "1.0.0",
* "description": "",
* "main": "index.js",
* "scripts": {
* "test": "echo \"Error: no test specified\" && exit 1",
* "lint": "eslint src"
* },
* "author": "",
* "license": "ISC",
* "dependencies": {
* "dotenv": "^16.0.3",
* "express": "^4.18.2"
* },
* "devDependencies": {
* "@typescript-eslint/eslint-plugin": "^5.42.0",
* "eslint": "8.26.0",
* "eslint-config-standard-with-typescript": "23.0.0",
* "eslint-plugin-import": "2.26.0",
* "eslint-plugin-n": "15.4.0",
* "eslint-plugin-promise": "6.1.1",
* "typescript": "4.8.4"
* }
* }
* Index.js
* require('dotenv').config();
* const express = require('express');
* const app = express();
* app.listen(process.env.PORT || 8080, (err) => {
* !err ? console.log('Server listening on Port', process.env.PORT) : console.log('Error in server setup');
* });
* Create .env file
* PORT = 7000
* Create .eslintrc.json or may be created default
* {
* "env": {
* {
* "env": {
* "browser": true, //for document supported document.getElementById
* "node": true,
* "commonjs": true,
* "es2021": true
* },
* "extends": ["standard-with-typescript", "eslint:recommended"],
* "overrides": [],
* "parserOptions": {
* "ecmaVersion": "latest"
* },
* "rules": {
* "semi": ["warn", "always"],
* "quotes": ["warn", "single"],
* "eqeqeq": "error", //==double equqal operator
* "no-undef": "error",
* "space-before-blocks": "warn", //space before block start
* "spaced-comment": "warn", //space comment
* "space-before-function-paren": "warn", //space before bracket start
* "eol-last": ["warn", "never"] //end of line required at end of the page
* }
* }
* {
* "env": {
* "browser": true, //for document supported document.getElementById
* "node": true,
* "commonjs": true,
* "es2021": true
* },
* "extends": ["standard-with-typescript", "eslint:recommended"],
* "overrides": [],
* "parserOptions": {
* "ecmaVersion": "latest"
* },
* "rules": {
* "semi": ["warn", "always"],
* "quotes": ["warn", "single"],
* "eqeqeq": "error", //==double equqal operator
* "no-undef": "error",
* "space-before-blocks": "warn", //space before block start
* "spaced-comment": "warn", //space comment
* "space-before-function-paren": "warn", //space before bracket start
* "eol-last": ["warn", "never"] //end of line required at end of the page
* }
* }
* {
* "env": {
* "browser": true, //for document supported document.getElementById
* "node": true,
* "commonjs": true,
* "es2021": true
* },
* "extends": ["standard-with-typescript", "eslint:recommended"],
* "overrides": [],
* "parserOptions": {
* "ecmaVersion": "latest"
* },
* "rules": {
* "semi": ["warn", "always"],
* "quotes": ["warn", "single"],
* "eqeqeq": "error", //==double equqal operator
* "no-undef": "error",
* "space-before-blocks": "warn", //space before block start
* "spaced-comment": "warn", //space comment
* "space-before-function-paren": "warn", //space before bracket start
* "eol-last": ["warn", "never"] //end of line required at end of the page
* }
* }
* Create .eslintignore
* \*.css
* \*.html

Prettier

**Installation Package** npm i prettier eslint-config-prettier eslint-plugin-prettier

**Uses Of prettier** formatting the code style

**Add plugin of prettier in .eslintrc.json**

**“**extends**”:[**"react-app", "plugin:react/recommended", "eslint:recommended", "plugin:prettier/recommended"**]**

"plugins": ["react", "prettier"],

**Add Prettier configuration** : add .prettierrc file

**Turn on prettier extension in vscode**

**Open vscode> settings> search forma >default formatter set as prettier –code formatter**

|  |
| --- |
| {  "trailingComma": "es5",  "tabWidth": 2,  "semi": true,  "endOfLine": "lf",  "overrides": [  {  "files": "\*.js",  "options": { "parser": "json" }  }  ]  } |

Husky

**Use of husky** control commits for unsatisfactory coding standard

**Installation package** npx husky-init && npm i

Set configuration to your root folder

**/Projects/Trainee/Eslint\_prettier\_husky\_demo/Eslint-Husky-Prettier-Demo$** npx husky set .husky/pre-commit "npm run lint"

Prior commit u are in **npx husky set .husky/pre-commit(file name) “npm run lint”**<(check lint before committing the code)>

Check after the .husky/pre-commit file updated and set as npm run lint

Final page are

* Folder structure

