**Настройка ESLint и Prettier**

**Шаг 1: Установка ESLint и Prettier**

npm install eslint prettier --save-dev

**Шаг 2: Инициализация ESLint**

Запустите команду для инициализации конфигурации ESLint. Это создаст базовый файл конфигурации .eslintrc.

npx eslint --init

При запуске этой команды вам будет предложено ответить на несколько вопросов о вашем проекте

• Какой тип проекта вы используете (например, JavaScript, React)?

• Используете ли вы TypeScript?

• Какой стиль кода вы предпочитаете (например, Airbnb, Standard)?

• Используете ли вы модульный импорт?

• Где вы хотите хранить конфигурацию (в отдельном файле или в package.json)?

**Шаг 3: Настройка файла .eslintrc.js**

module.exports = {

  env: { // Окружения

    browser: true, // Определяет глобальные переменные браузера

    es6: true, // Использование последних возможностей js

node: true, // Указывает, что код может выполняться в среде Node.js, что позволяет использовать глобальные переменные Node, такие как process и module.

Jest: true, // Указывает, что тесты будут выполняться с использованием Jest, позволяя использовать глобальные переменные Jest

  },

  extends: [ // Расширения

    'eslint:recommended', // Рекомендуемые правила ESLint

    'plugin:react/recommended', // Рекомендуемые правила для React

    'plugin:prettier/recommended' // Интеграция Prettier с ESLint

  ],

  parserOptions: { // Опции парсера

    ecmaVersion: 12, // Версия ECMAScript для синтаксического анализа

    sourceType: 'module', // Использование модульного синтаксиса

  },

  rules: { // Правила

    'prettier/prettier': 'error', // Отображение ошибок Prettier как ошибок ESLint

'no-unused-vars': 'warn', // Устанавливает правило для предупреждения о неиспользуемых переменных

    'no-console': 'warn', // Предупреждение при использовании console.log

    'quotes': ['error', 'single'], // Использование одинарных кавычек

  },

};

**Шаг 4: Установка плагинов для интеграции ESLint и Prettier**

Чтобы ESLint и Prettier работали вместе без конфликтов, установите дополнительные пакеты:

npm install eslint-config-prettier eslint-plugin-prettier --save-dev

• eslint-config-prettier: Отключает все правила ESLint, которые могут конфликтовать с Prettier.

• eslint-plugin-prettier: Запускает Prettier как правило ESLint.

**Шаг 5: Настройка Prettier**

Создайте файл .prettierrc для конфигурации Prettier:

{

  "semi": true, // Использование точек с запятой в конце строк

  "singleQuote": true, // Использование одинарных кавычек вместо двойных

  "trailingComma": "es5", // Добавление запятых в конце последнего элемента в списках ES5 (например, объектов)

  "printWidth": 80, // Максимальная длина строки

  "tabWidth": 2, // Размер отступа в пробелах

  "useTabs": false // Использование пробелов вместо табуляций

}

**Шаг 6: Игнорирование файлов**

**Создайте файл .prettierignore**, чтобы указать файлы и директории, которые Prettier должен игнорировать:

node\_modules/

build/

dist/

**Также создайте файл .eslintignore**, если хотите игнорировать файлы для ESLint:

node\_modules/

build/

dist/

**Шаг 7: Запуск ESLint и Prettier**

Теперь вы можете запускать ESLint и Prettier для проверки и форматирования вашего кода.

Для проверки кода с помощью ESLint:

npx eslint . --ext .js,.jsx

Для форматирования кода с помощью Prettier:

npx prettier --write .

**Файл .editorconfig**

root = true // указывает, что это корневой файл конфигурации, и другие файлы `.editorconfig` в родительских директориях не будут учитываться.

[\*] // задает настройки для всех файлов в проекте.

indent\_style = space //устанавливает стиль отступов на пробелы.

indent\_size = 2 //задает размер отступа в 2 пробела

end\_of\_line = lf //устанавливает символ конца строки на LF (Line Feed), что соответствует Unix-подобным системам.

charset = utf-8 // устанавливает кодировку файлов на UTF-8.

trim\_trailing\_whitespace = true // удаляет пробелы в конце строк.

insert\_final\_newline = true //добавляет пустую строку в конце файла.

[\*.md] // — задает специальные настройки для файлов с расширением `.md

trim\_trailing\_whitespace = false // для файлов Markdown отключает удаление пробелов в конце строк, что может быть полезно для сохранения форматирования.