



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС

Занятие 2

Настройка проекта



План занятия

1. ESLint
2. Prettier
3. Pre-Hooks Git
4. Hasky и lint-stage

ESLint - это линтер для языка программирования JavaScript, написанный на Node.js

ESLint полезен для JavaScript, т.к. будучи интерпретируемым языком JavaScript, не имеет этапа компиляции и многие ошибки могут быть обнаружены только во время выполнения



ESLint

- Тут какой-то текст

ESLint решает задачи:

- найти существующие ошибки в коде;
- избежать глупых ошибок;
- избежать бесконечные циклы в условиях цикла `for`;
- убедиться, что все методы `getter` возвращают что-то;
- не разрешить выражения `console.log` (и аналогичные);
- проверить наличие дубликатов `cases` в `switch`
- проверить недоступный код;
- проверить правильность `JSDoc`;

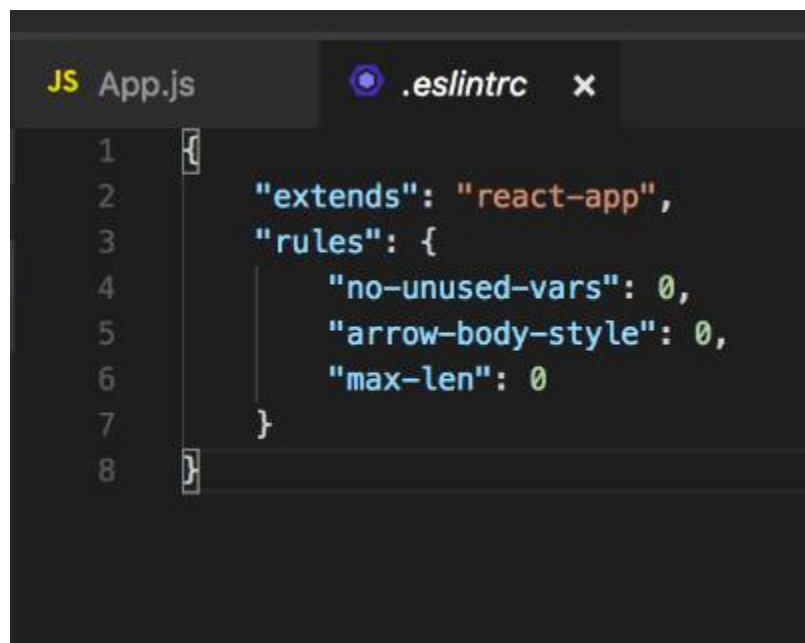
Правила ESLint - <https://eslint.org/docs/latest/rules/>



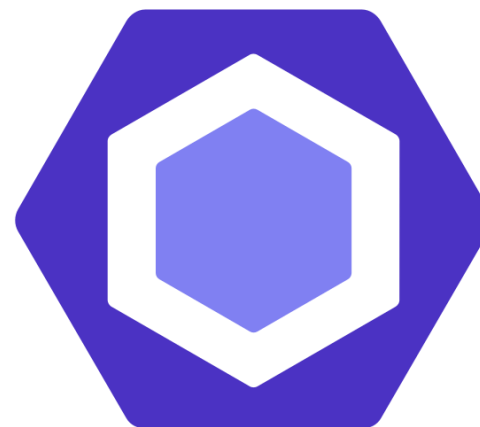
ESLint

• Тут какой-то текст

Файл конфигурации `.eslintrc` может быть глобальным или локальным только для проекта



```
JS App.js  .eslintrc x
1  {
2    "extends": "react-app",
3    "rules": {
4      "no-unused-vars": 0,
5      "arrow-body-style": 0,
6      "max-len": 0
7    }
8  }
```



ESLint

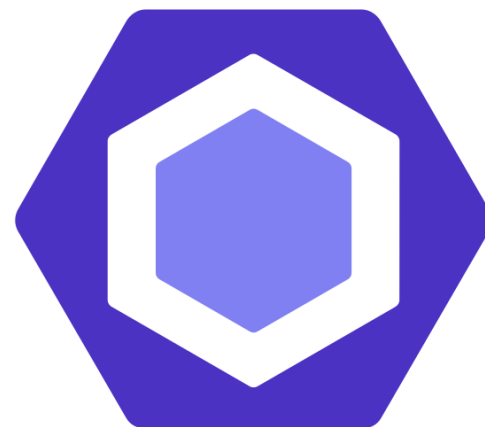
- Тут какой-то текст

Установка ESLint глобально

```
npm install -g eslint
```

```
# создать файл конфигурации  
eslint --init
```

```
# запуск на проверку определенного файла  
eslint file.js
```



ESLint

- Тут какой-то текст

Установка ESLint локально

```
npm install eslint --save-dev
```

```
# создать файл конфигурации  
eslint --init
```

```
# запуск на проверку определенного файла  
eslint file.js
```



ESLint

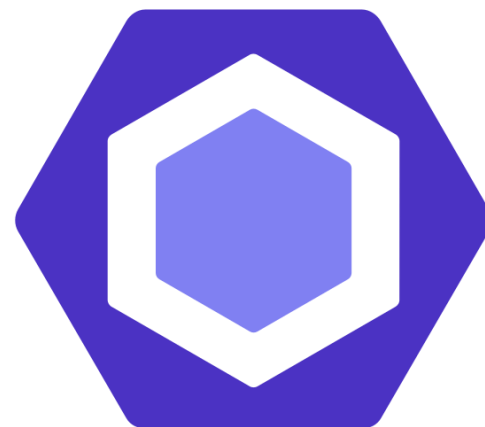
- Тут какой-то текст

Установка стилей Airbnb

```
npm install eslint-config-airbnb --save-dev
```

После установки нужно добавить в .eslintrc

```
{  
  "extends": "airbnb"  
}
```



ESLint

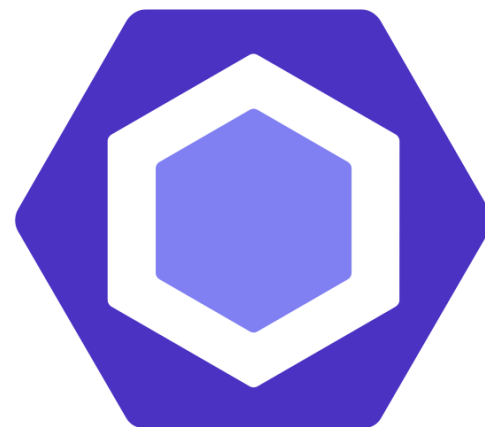
- Тут какой-то текст

Установка стилей для React

```
npm install eslint-plugin-react --save-dev
```

После установки нужно добавить в .eslintrc

```
{  
  "extends": "airbnb",  
  "plugins": ["react"],  
  "parserOptions": {  
    "ecmaFeatures": {  
      "jsx": true  
    }  
  }  
}
```

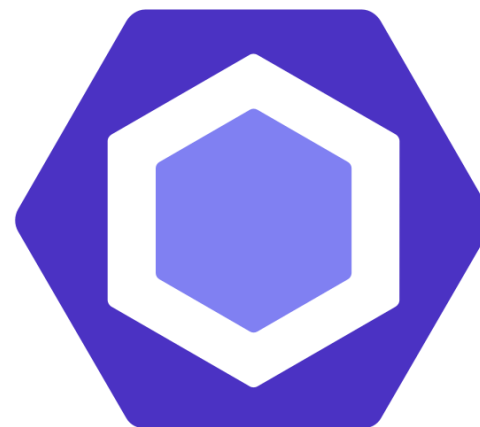


ESLint

• Тут какой-то текст

.eslintrc

```
{  
  "parserOptions": {  
    "ecmaVersion": 6  
  }  
}
```



ESLint

- Тут какой-то текст

Другие правила и их конфигурация:

<https://eslint.org/docs/user-guide/configuring>



ESLint

- Тут какой-то текст

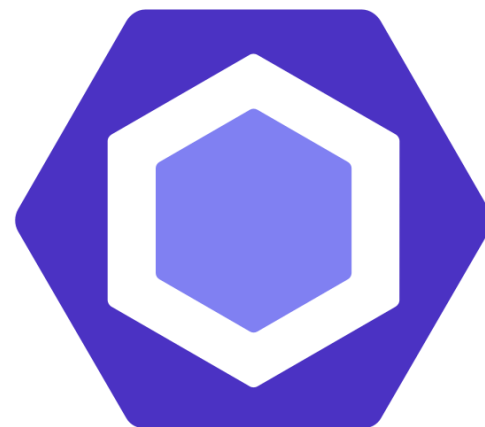
Отключение правил где это необходимо

отключить правила в конкретном месте:

```
/* eslint-disable */  
alert('testing')  
/* eslint-enable */
```

или по другому

```
alert('testing') // eslint-disable-line
```



ESLint

- Тут какой-то текст

Отключить одно или несколько конкретных правил для нескольких строк:

```
/* eslint-disable no-alert, no-console */  
alert('testing')  
console.log  
/* eslint-enable no-alert, no-console */
```

или для одной строки

```
alert("testing") // eslint-disable-line no alert, quotes, semi
```



ESLint

- Тут какой-то текст

Это помощник по части оформления кода

Можно писать с пробелами перед именем свойства, кавычками, запятыми в последней строке.

Prettier настроенный на сохранение или на pre-commit hook — «перетрясет» ваши файлы и оформит их в соответствии с настройками, которых у него минимум

Меньше настроек - меньше конфигураций



Prettier

- Тут какой-то текст

Еще раз про установку в связке

```
npm install eslint-config-prettier eslint-plugin-prettier prettier lint-staged husky --save-dev
```

- husky — упрощает работу с git hooks (то есть, тот самый момент, когда вы собираетесь делать коммит «пре-коммит» легко настроить с помощью этой «собаки»)
- lint-staged — пакет, который позволяет вам сделать обработку командой из терминала только тех файлов, которые собираются улететь в коммит



Prettier

• Тут какой-то текст

.eslintrc

```
{  
  "extends": [  
    "react-app",  
    "prettier"  
  ],  
  "rules": {  
    "jsx-quotes": [  
      1,  
      "prefer-double"  
    ]  
  },  
  "plugins": [  
    "prettier"  
  ]  
}
```



Prettier

• Тут какой-то текст

Конфиг, который «наследует» стандартные правила из *react-app* и *prettier* (это глобальные конфиги, один встроен в *create-react-app*, второй в пакета [eslint-config-prettier](#))

Последняя опция в конфиге — использование плагинов

Используется плагин *prettier* (пакет [eslint-plugin-prettier](#)), с настройкой `autoFixOnSave` (автоматически исправляй на «сохранение») в редакторе и в момент сохранения код будет преобразоваться



Prettier

- Тут какой-то текст

Настройка Visual Studio Code

```
"editor.formatOnPaste": false,  
"editor.formatOnSave": true,  
"[javascript]": {  
  "editor.formatOnSave": true,  
},  
"[html]": {  
  "editor.formatOnSave": false,  
},  
"[json]": {  
  "editor.formatOnSave": false,  
},  
"eslint.autoFixOnSave": true,  
"eslint.alwaysShowStatus": true,
```

плюс установка plugin-ов: ESLint и Prettier



Prettier

• Тут какой-то текст

Настройка Prettier

`.prettierrc`

```
{  
  "useTabs": true, // использовать табы  
  "printWidth": 80, // длина строки - 80  
  "tabWidth": 2, // длина "таба" - 2 пробела  
  "singleQuote": true, // использовать одинарные кавычки  
  "trailingComma": "es5", // запятая в последней строке  
  "jsxBracketSameLine": false, // закрывающийся jsx в этой же строке  
  "parser": "flow", // парсер - flow  
  "semi": false // точка с запятой  
}
```



Prettier

• Тут какой-то текст

Pre-hook git

Сейчас будет отрабатывать eslint

Так же в момент сохранения, код будет преобразовываться с помощью prettier

Однако, нам еще не хватает настройки пре-коммит хука.

package.json

```
"scripts": {  
  "precommit": "lint-staged"  
},  
"lint-staged": {  
  "*.js, jsx": [  
    "prettier --write",  
    "git add"  
  ]  
}
```



- Тут какой-то текст



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС

Спасибо за внимание

