ESLint：自定义规则

关于ESLint?

ESLint属于一种QA工具，是一个ECMAScript/JavaScript语法规则和代码风格的检查工具，可以用来保证写出语法正确、风格统一的代码。

ESLint旨在完全**可配置**，它的目标是提供一个插件化的javascript代码检测工具。这意味着您可以关闭每个规则，只能使用基本语法验证，或者混合并匹配捆绑的规则和自定义规则，使ESLint完美的适用于您的项目。

常用配置项

parser

指定 ESLint 使用的语法分析器。ESLint 兼容的语法分析器有：Esprima、Babel-ESLint、@typescript-eslint/parser，ESLint 默认使用 Esprima。

parserOptions

指定语法分析器选项，默认使用的语法分析器支持如下几个选项：`ecmaVersion`、`sourceType`、`ecmaFeatures`。

globals

使用未在当前文件中定义的全局变量时，会命中 `no-undef` 规则，通过 `globals` 配置指定的全局变量无视 `no-undef` 规则。

```
{
  "globals": {
    "var1": "writable",
    "var2": "readonly"
  }
}
```

rules

指定ESLint的规则。规则组成：规则名、错误级别、附加选项

```
"rules": {
  "semi": ["error", "always"],
  "quotes": ["error", "double"]
}
```

`"semi"` 和 `"quotes"` 是规则名，`"error"` 是规则错误级别，`"always"` 和 `"quotes"` 分别是 `semi` 和 `quotes` 各自特有的附加选项。

错误级别有三种：

- `"off"` 或 `0` - 关闭，不校验该规则
- `"warn"` 或 `1` - 警告，不影响 exit code
- `"error"` 或 `2` - 错误，触发该规则时 exit code 为 `1` 关于规则的错误级别，在 ESLint 特性概览 部分已有介绍。

overrides

对于匹配 `overrides.files` 且不匹配 `overrides.excludedFiles` 的 文件，`overrides.rules` 中的规则会覆盖 `rules` 中的同名规则。

```
{
  "rules": {...},
  "overrides": [
    {
      "files": ["*-test.js","*.spec.js"],
      "excludedFiles": "*.test.js",
      "rules": {
        "no-unused-expressions": "off"
      }
    }
  ]
```

```
12 }
```

plugin

插件是第三方定制的规则集合，通常是一些拓展eslint原生js规则之外的规则，例如`eslint-plugin-vue`就是拓展了对vue项目的一些规则。

`plugins`参数用于指定第三方插件，插件名中的`eslint-plugin-`前缀可以省略。使用插件前需要先安装依赖包。

```
1 {
2   "plugins": [
3     "plugin1",
4     "eslint-plugin-vue"
5   ],
6   rules: {
7     'eslint-plugin-react/jsx-boolean-value': 2 //
8   }
9 }
```

需要注意的是，直接引入了插件并不会生效，plugins只是加载了插件，简单来说就是eslint具备解析vue的能力，但是还需要通过rules来开启这个提示能力。

如果我们的插件里面有几十条规则，我们需要手动一条一条的打开？

extends

> *plugins与rules结合是eslint基础能力*，extends可以看做是去集成一个个配置方案的最佳实践。

配置文件可以在已有配置的基础上进行扩展，`extends`用于指定基础配置。支持一个配置包的字符串名字或者多个配置包的字符串名字数组。ESLint支持递归拓展配置，所以拓展配置包里也支持extends

当拓展了配置包的时候，当前配置的rules规则会按照以下的几种情况进行拓展：

- 启用基础配置中没有规则
- 继承基础配置中的规则，改变其错误级别，但不改变其附加选项：
  - 基础配置：`"eqeqeq": ["error", "allow-null"]`
  - 扩展配置：`"eqeqeq": "warn"`
  - 最终有效配置：`"eqeqeq": ["warn", "allow-null"]`
- 覆盖基础配置中的规则:
  - 基础配置：`"quotes": ["error", "single", "avoid-escape"]`
  - 扩展配置：`"quotes": ["error", "single"]`

    - 最终有效配置: `"quotes": ["error", "single"]`

ESLint支持配置包的来源包括：

- ESLint内置规则集的子集： `"eslint:recommended"`
- 共享配置包： `eslint-config-airbnb`
- 插件导出的命名配置："plugin:react/recommended"

plugin和extends的区别

命名

插件的命名是 `eslint-plugin-xxxx`

extends则是 `eslint-config-xxxx`

用途

plugin插件主要是为eslint新增一些检查规则，举个例子： `eslint-plugin-vue` 就会对 `vue` 项目做了一些定制的 `eslint` 规则。

extends拓展就是使用其他根据不同的需求、风格、规范所配置好的规则。在多个项目中就可以通过引用统一的规则来实现快速集成。

Config

extends的config是代表了命名的规范。

plugin的config是一个属性，表示该plugin自定义规则的集合

实现一个插件

手把手实现一个ESLint插件

团队的可共享配置

@digital-rd/eslint-config-conventional