

Getting Started with ESLint

ESLint is a tool for identifying and reporting on patterns found in ECMAScript/JavaScript code, with the goal of making code more consistent and avoiding bugs. In many ways, it is similar to JSLint and JSHint with a few exceptions:

ESLint 是在 ECMAScript/JavaScript 代码中识别和报告模式匹配的工具，它的目标是保证代码的一致性和避免错误。在许多方面，它和 JSLint、JSHint 相似，除了少数的例外：

- ESLint uses [Espree](#) for JavaScript parsing.
- ESLint 使用 [Espree](#) 解析 JavaScript。
- ESLint uses an AST to evaluate patterns in code.
- ESLint 使用 AST 去分析代码中的模式
- ESLint is completely pluggable, every single rule is a plugin and you can add more at runtime.
- ESLint 是完全插件化的。每一个规则都是一个插件并且你可以在运行时添加更多的规则。

Getting Started Tutorial



Why ESLint @0:00, Installing and using ESLint @2:20. [Full ESLint Course at Pluralsight](#)

Why ESLint @0:00, Installing and using ESLint @2:20. [去 Pluralsight 查看全部 ESLint 课程](#)

Note that, for some reason, not everyone can see the vedio on youtube, so the translator use a vedio screenshot here instead of a vedio.

注：由于某些未知原因，你可能看不到 [youtube](#) 的视频，所以只放了[视频截图](#)。

Installation and Usage

Prerequisites: [Node.js](#) ([>=6.14](#)), [npm](#) version 3+.

先决条件：[Node.js](#) ([>=6.14](#)), [npm](#) version 3+。

You can install ESLint using npm:

你可以使用 npm 安装 ESLint：

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

You should then set up a configuration file:

紧接着你应该设置一个配置文件：

```
$ ./node_modules/.bin/eslint --init
```

After that, you can run ESLint on any file or directory like this:

之后，你可以在任何文件或目录上运行ESLint如下：

```
$ ./node_modules/.bin/eslint yourfile.js
```

It is also possible to install ESLint globally rather than locally (using `npm install eslint --global`). However, any plugins or shareable configs that you use must be installed locally in either case.

也可以在全局而不是本地安装 ESLint (使用 `npm install eslint --global`)。但是，你使用的任何插件或可共享配置都必须安装在本地。

Configuration

Note: If you are coming from a version before 1.0.0 please see the [migration guide](#).

注意：如果你之前使用的版本低于 1.0.0，请查看 [迁移指南](#)。

After running `eslint --init`, you'll have a `.eslintrc` file in your directory. In it, you'll see some rules configured like this:

运行 `eslint --init` 之后，`.eslintrc` 文件会在你的文件夹中自动创建。你可以在 `.eslintrc` 文件中看到许多像这样的规则：

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

The names `"semi"` and `"quotes"` are the names of rules in ESLint. The first value is the error level of the rule and can be one of these values:

`"semi"` 和 `"quotes"` 是 ESLint 中 [规则](#) 的名称。第一个值是错误级别，可以使下面的值之一：

- `"off"` or `0` - turn the rule off
- `"off"` or `0` - 关闭规则
- `"warn"` or `1` - turn the rule on as a warning (doesn't affect exit code)
- `"warn"` or `1` - 将规则视为一个警告（不会影响退出码）
- `"error"` or `2` - turn the rule on as an error (exit code will be 1)
- `"error"` or `2` - 将规则视为一个错误（退出码为1）

The three error levels allow you fine-grained control over how ESLint applies rules (for more configuration options and details, see the [configuration docs](#)).

这三个错误级别可以让你细粒度的控制 ESLint 是如何应用规则（更多关于配置选项和细节的问题，请查看[配置文件](#)）

Your `.eslintrc` configuration file will also include the line:

你的 `.eslintrc` 配置文件可以包含下面的一行：

```
"extends": "eslint:recommended"
```

Because of this line, all of the rules marked “[✔](#)” on the [rules page](#) will be turned on. Alternatively, you can use configurations that others have created by searching for “eslint-config” on [npmjs.com](#). ESLint will not lint your code unless you extend from a shared configuration or explicitly turn rules on in your configuration.

由于这行，所有在 [规则页面](#) 被标记为“[✔](#)”的规则将会默认开启。另外，你可以在 [npmjs.com](#) 搜索“eslint-config”使用别人创建好的配置。只有在你的配置文件中扩展了一个可分享的配置或者明确开启一个规则，ESLint 才会去校验你的代码。

Next Steps

- Learn about [advanced configuration](#) of ESLint.
- 学习 ESLint 的[高级配置](#)。
- Get familiar with the [command line options](#).
- 熟悉 [命令行选项](#)。
- Explore [ESLint integrations](#) into other tools like editors, build systems, and more.
- 探索将 [ESLint 集成](#) 到其它工具，比如 [编辑器](#)，[构建工具](#)等等。
- Can't find just the right rule? Make your own [custom rule](#).
- 没有找到适合的规则？创建[自定义规则](#)。
- Make ESLint even better by [contributing](#).
- 通过 [贡献代码](#) 让 ESLint 变得更好。