



WEBINAR

# ESLint and Prettier in 2019

Mark Volkmann, Partner and Principal Software Engineer  
[mark@objectcomputing.com](mailto:mark@objectcomputing.com)

© 2019, Object Computing, Inc. (OCI). All rights reserved. No part of these notes may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior, written permission of Object Computing, Inc. (OCI)

[objectcomputing.com](http://objectcomputing.com)

# ESLint

---



- Popular tool for finding and fixing JavaScript issues
- Soon will be preferred linter for TypeScript
- Can also use with JSX
- Doesn't enforce any rules by default
- Many editor/IDE integrations



## Prettier

---



- Code formatter for many languages
- Opinionated with small set of options
- Many editor/IDE integrations



## package.json

---



- If project doesn't have **package.json** file
  - **npm init**



# Installing

---



- `npm install --save-dev *`

where \* is

`babel-eslint`

`eslint`

`prettier`

`eslint-config-prettier`

`eslint-plugin-html`

`eslint-plugin-import`

`eslint-plugin-prettier`

already installed in  
create-react-app projects



## npm Scripts



- In scripts section of `package.json`

```
"format": "prettier --write src/**/*.{js,scss}",  
"lint": "eslint --cache --fix src/**/*.js",
```

`--cache` option makes it faster  
because it only runs on files that  
have been modified since last run

## ESLint Configuration ... (in `.eslintrc.json`)



```
{  
  "env": {  
    "browser": true,  
    "es6": true,  
    "jest": true,  
    "node": true  
  },  
  "parserOptions": {  
    "ecmaVersion": 9, for ES2018  
    "parser": "babel-eslint",  
    "sourceType": "module"  
  },  
}
```

adds knowledge of browser global variables

adds knowledge of ES6 features; no higher env is available

adds knowledge of Jest global variables

adds knowledge of Node.js global variables

for ES2018

adds support for experimental JS features

to use ES6 imports and exports

ex. public class fields

## ... ESLint Configuration



```
"extends": [  
    "eslint:recommended",  
    "plugin:import/recommended",  
    "plugin:prettier/recommended"  
],  
  
"plugins": [  
    "html",  
    "prettier"  
],  
  
"rules": {  
    "no-console": "off" // allows use of console functions  
}
```

enforces rules with check mark at <https://eslint.org/docs/rules/>

lints ES6 import and export statements

turns off all ESLint rules that conflict with Prettier

lints JavaScript HTML <script> tags

runs Prettier as an ESLint rule so ESLint code formatting rules are not needed

allows use of `console` functions

# Prettier Configuration



- In `.prettierrc`

```
{  
  "bracketSpacing": false,  
  "singleQuote": true  
}
```

- Or in `package.json`

```
prettier: {  
  "bracketSpacing": false,  
  "singleQuote": true  
}
```

{**foo: 1, bar: 2**}  
instead of  
{ **foo: 1, bar: 2** }  
matches typical  
array literal formatting  
[**1, 2**]  
not  
[ **1, 2**  ]

## Running Tools

---



- Run ESLint with `npm run lint`
- Run Prettier with `npm run format`



## React Specifics ...

---



- No additional Prettier configuration needed
- Install more ESLint plugins
  - `npm install --save-dev *`  
where \* is  
`eslint-plugin-jsx-a11y`  
`eslint-plugin-react`



## ... React Specifics



- Update sections of `.eslintrc.json`

```
"parserOptions": {  
    "ecmaFeatures": {  
        "jsx": true  
    }  
},  
"extends": [  
    "plugin:jsx-ally/recommended", to lint JSX code  
    "plugin:react/recommended"  
],  
"plugins": ["jsx-ally", "react", "react-hooks"],  
"settings": {  
    "react": {  
        "version": "detect"  
    }  
}
```

shows things to add;  
keep existing values

enforces accessibility rules on JSX code

enforces React-specific rules

## Vue Specifics ...

---



- No additional Prettier configuration needed
- Install more ESLint plugins
  - `npm install --save-dev *`  
where \* is  
`eslint-plugin-jsx-ally`  
`eslint-plugin-vue`  
`vue-eslint-parser`



## ... Vue Specifics



- Update sections of `.eslintrc.json`

```
"parser": "vue-eslint-parser",
"parserOptions": {
  "ecmaFeatures": {
    "jsx": true
  }
},
"extends": [
  "plugin:jsx-ally/recommended", to lint JSX code
  "plugin:vue/recommended", enforces accessibility rules on JSX code
  "@vue:prettier" enforces Vue-specific rules
],
"plugins": ["jsx-ally", "prettier", "vue"], provided by Vue CLI
```

shows things to add;  
keep existing values

enforces Vue-specific rules

provided by Vue CLI

## Wrap Up

---



- That's it!
- You now know everything needed to configure ESLint and Prettier in projects that
  - use no web framework
  - use React
  - use Vue



## LEARN MORE ABOUT OCI EVENTS AND TRAINING



### Events:

- [objectcomputing.com/events](https://objectcomputing.com/events)

### Training:

- [objectcomputing.com/training](https://objectcomputing.com/training)
- [grailstraining.com](https://grailstraining.com)
- [micronauttraining.com](https://micronauttraining.com)

Or email [info@ocitraining.com](mailto:info@ocitraining.com) to schedule a custom training program for your team online, on site, or in our state-of-the-art, Midwest training lab.



## CONNECT WITH US

- 1+ (314) 579-0066
- @objectcomputing
- objectcomputing.com