



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  
`eslint`  
`prettier`  
`eslint-config-prettier`  
`eslint-plugin-html`  
`eslint-plugin-import`  
`eslint-plugin-prettier`



## npm Scripts

---



- In scripts section of `package.json`

```
"format": "prettier --write src/**/*.js",  
"lint": "eslint src/** --ext .js",
```



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



```
{  
  "env": {  
    "browser": true,  
    "es6": true,  
    "jest": true,  
    "node": true  
  },  
  
  "parserOptions": {  
    "ecmaVersion": 9, for ES2018  
    "sourceType": "module" to use ES6 imports and exports  
  },
```

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



## ... 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  
}
```

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

- In `package.json`

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

## 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", enforces accessibility rules on JSX code  
    "plugin:react/recommended" enforces React-specific rules  
],  
"plugins": ["jsx-ally", "react"],  
"settings": {  
    "react": {  
        "version": "detect"  
    }  
}
```

## 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 Specifics



- Update sections of `.eslintrc.json`

```
"parserOptions": {  
    "ecmaFeatures": {  
        "jsx": true  
    }  
},  
"extends": [  
    "plugin:jsx-ally/recommended", to lint JSX code  
    "plugin:vue/essential", enforces accessibility rules on JSX code  
    "@vue/prettier", enforces Vue-specific rules  
],  
"plugins": ["jsx-ally"],  
"settings": {  
    "react": {  
        "version": "detect"  
    }  
}
```

A callout box points from the "jsx" key in the "parserOptions" section to the text "to lint JSX code". Another callout box points from the "plugin:jsx-ally/recommended" entry in the "extends" array to the text "enforces accessibility rules on JSX code". A third callout box points from the "@vue/prettier" entry in the "extends" array to the text "enforces Vue-specific rules". A fourth callout box points from the "jsx-ally" entry in the "plugins" array to the text "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.



OBJECT  
COMPUTING

## CONNECT WITH US

---

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