

Booco Development Guide

Style Guide

Booco code style guide is based on [Airbnb JavaScript Style Guide](#). There are additions and changes depending on environment.

General additions

The following additions/changes were made to the original guide:

- Disallow a space before function parenthesis (`space-before-function-paren: ["error", "never"]`)
- Enforce a maximum line length 120 symbols (`max-len: ["error", 120]`)
- Allow the unary operators (++)/-- for loops (`no-plusplus: ["error", {"allowForLoopAfterthoughts": true}]`)
- Allow (but not require) trailing commas for multiline (`comma-dangle: ["error", "only-multiline"]`)
- React component static properties should be positioned inside class body (`react/static-property-placement: ["error", "property assignment", {defaultProps: "static public field", propTypes: "static public field"}]`).
- Requires function expressions to have a name, if the name cannot be assigned automatically in an ES6 environment (`func-names: ["error", "as-needed"]`)

In addition there following changes in case of use of **Meteor**:

- ``no-underscore-dangle: ["error", { "allow": ["_id", "_str"]}]`
- `import/no-unresolved: [2, { ignore: ["^meteor/"] }]`

Additions and changes for Iridium scripts

Iridium uses ECMAScript 3 specification so many changes are required.

parserOptions should be set:

- `ecmaVersion: 3`
- `sourceType: "script"`

The following rules must be disabled:

- `no-var`
- `prefer-template`
- `prefer-arrow-callback`
- `prefer-destructuring`
- `object-shorthand`

Global variables:

- `IR: readonly`

- `_Debug`: readonly
- `_Error`: readonly
- `_Log`: readonly
- `_DEBUGGER`: readonly

Package managers

Recommended package manager is yarn v1.21.x. For Iridium projects (not modules, but projects) and Meteor - NPM should be used.

Initialization of new projects

1. Install the following apps (if not installed)

- [Node JS](#) (Usually LTS). Includes **NPM**.
- [Git](#)
- If required: [yarn](#). Recommended v1.21.x.

2. Project initialization:

- `git init` - initializing git repository
- `yarn init` (or `npm init`) - initializing package.json
- `yarn add eslint -D` (or `npm install eslint --save-dev`)

3. Eslint initial configuration:

- `node ./node_modules/eslint/bin/eslint.js --init`
 - "How would you like to use ESLint?" - "To check syntax, find problems, and enforce code style"
 - "What type of modules does your project use?" - depending on the environment (for Iridium - "none of these")
 - "Which framework does your project use?" - depending on the environment (for Iridium - "none of these")
 - "Does your project use TypeScript?" - No
 - "Where does your code run?" - Node
 - "How would you like to define a style for your project?" - "Use a particular style guide" - "Airbnb"
 - "What format do you want your config file to be in?" - YAML
 - Downgrade and install dependencies if necessary.
- Replace `.eslint.yml` with proper file.