# **Node.js**

## **Formatting**

### **2 Spaces for indentation**

Use 2 spaces for indenting the codes and never mix tabs and spaces.

### **Newlines**

Use UNIX-style newlines (\n), and a newline character as the last character of a file. Windows-style newlines (\r\n) are forbidden inside any repository.

### **No trailing whitespace**

clean up any trailing whitespace in the JS files before committing

### **Use single quotes**

Right:

var foo = 'bar';

Wrong:

var foo = "bar";

### **Opening braces go on the same line**

Opening braces go on the same line as the statement.

Right:

if (true) {

console.log('winning');

}

Wrong:

if (true)

{

console.log('losing');

}

### **Declare one variable per var statement**

Right:

var keys = ['foo', 'bar'];

var values = [23, 42];

var object = {};

while (keys.length) {

var key = keys.pop();

object[key] = values.pop();

}

Wrong:

var keys = ['foo', 'bar'],

values = [23, 42],

object = {},

key;

while (keys.length) {

key = keys.pop();

object[key] = values.pop();

}

## **Naming Conventions**

### **Use lowerCamelCase for variables, properties and function names**

Right:

var adminUser = db.query('SELECT \* FROM users ...');

Wrong:

var admin\_user = db.query('SELECT \* FROM users ...');

### **Use UpperCamelCase for class names**

Right:

function BankAccount() {

}

Wrong:

function bank\_Account() {

}

### **Use UPPERCASE for Constants**

Right:

var SECOND = 1 \* 1000;

function File() {

}File.FULL\_PERMISSIONS = 0777;

Wrong:

const SECOND = 1 \* 1000;

function File() {

}File.fullPermissions = 0777;

## **Variables**

### **Object / Array creation**

Use trailing commas and put short declarations on a single line. Only quote keys when the interpreter complains:

Right:

var a = ['hello', 'world'];

var b = {

good: 'code',

'is generally': 'pretty',

};

Wrong:

var a = [

'hello', 'world'

];

var b = {"good": 'code'

, is generally: 'pretty'

};

Source: https://github.com/felixge/node-style-guide

# **React**

## **Naming**

****Extensions****: Use .jsx extension for React components.

****Filename****: Use PascalCase for filenames. E.g., ReservationCard.jsx.

****Reference Naming****: Use PascalCase for React components and camelCase for their instances. eslint: [react/jsx-pascal-case](https://github.com/yannickcr/eslint-plugin-react/blob/master/docs/rules/jsx-pascal-case.md)

// bad

import reservationCard from './ReservationCard';

// good

import ReservationCard from './ReservationCard';

// bad

const ReservationItem = <ReservationCard />;

// good

const reservationItem = <ReservationCard />;

****Component Naming****: Use the filename as the component name. For example, ReservationCard.jsx should have a reference name of ReservationCard. However, for root components of a directory, use index.jsx as the filename and use the directory name as the component name:

// bad

import Footer from './Footer/Footer';

// bad

import Footer from './Footer/index';

// good

import Footer from './Footer';

## **Declaration**

Do not use displayName for naming components. Instead, name the component by reference.

// bad

export default React.createClass({ displayName: 'ReservationCard', // stuff goes here

});

// good

export default class ReservationCard extends React.Component {

}

## **Quotes**

Always use double quotes (") for JSX attributes, but single quotes (') for all other JS. eslint: [jsx-quotes](http://eslint.org/docs/rules/jsx-quotes)

Why? Regular HTML attributes also typically use double quotes instead of single, so JSX attributes mirror this convention.

// bad

<Foo bar='bar' />

// good

<Foo bar="bar" />

// bad

<Foo style={{ left: "20px" }} />

// good

<Foo style={{ left: '20px' }} />

## **Spacing**

Always include a single space in the self-closing tag. eslint: [no-multi-spaces](http://eslint.org/docs/rules/no-multi-spaces), [react/jsx-tag-spacing](https://github.com/yannickcr/eslint-plugin-react/blob/master/docs/rules/jsx-tag-spacing.md)

// bad

<Foo/>

// very bad

<Foo />

// bad

<Foo

/>

// good

<Foo />

Do not pad JSX curly braces with spaces. eslint: [react/jsx-curly-spacing](https://github.com/yannickcr/eslint-plugin-react/blob/master/docs/rules/jsx-curly-spacing.md)

// bad

<Foo bar={ baz } />

// good

<Foo bar={baz} />

Source: https://github.com/airbnb/javascript/tree/master/react