

# confusing-browser-globals

A curated list of browser globals that commonly cause confusion and are not recommended to use without an explicit `window.` qualifier.

## Motivation

Some global variables in browser are likely to be used by people without the intent of using them as globals, such as `status`, `name`, `event`, etc.

For example:

```
handleClick() { // missing `event` argument
  this.setState({
    text: event.target.value // uses the `event` global: oops!
  });
}
```

This package exports a list of globals that are often used by mistake. You can feed this list to a static analysis tool like ESLint to prevent their usage without an explicit `window.` qualifier.

## Installation

```
npm install --save confusing-browser-globals
```

## Usage

If you use Create React App, you don't need to configure anything, as this rule is already included in the default `eslint-config-react-app` preset.

If you maintain your own ESLint configuration, you can do this:

```
const restrictedGlobals = require('confusing-browser-globals');

module.exports = {
  rules: {
    'no-restricted-globals': ['error'].concat(restrictedGlobals),
  },
};
```

## License

MIT