

Convention for eventmap selectors (prefix-eventmap-selectors)

When you are setting up event maps in your JS files, you need to 'select' the element in the template that the event attaches to. Rather than using the same CSS class names that are used to style the elements, it's better practice to use classnames that are specifically added for those event maps. A reasonable convention is a class starting with `js-` to indicate it is used by the JavaScript. - [source](#)

This rule enforces that convention.

Rule Details

This rule aims to ensure all classes with attached event listeners have the same prefix, so they are distinguishable from classes used for styling.

Options

This rule takes two arguments:

- the prefix css classes must use, defaults to `js-`.
- the mode in which to run the rule, can be one of the following options
 - `relaxed` (default): events can be assigned through any selectors, but class selectors must be prefixed
 - `strict` : events can only be assigned to prefixed class selectors

relaxed

Examples of **incorrect** code for the default `"relaxed"` mode:

```
/*eslint prefix-eventmap-selectors: [2, "js-", "relaxed"]*/

Template.foo.events({
  'click .foo': function () {}
})
```

Examples of **correct** code for the default `"relaxed"` mode:

```
/*eslint prefix-eventmap-selectors: [2, "js-", "relaxed"]*/

Template.foo.events({
  'click .js-foo': function () {},
  'blur .js-bar': function () {},
  'click #foo': function () {},
  'click [data-foo="bar"]': function () {},
  'click input': function () {},
  'click': function () {},
})
```

strict

Examples of **incorrect** code for the `"strict"` mode:

```
/*eslint prefix-eventmap-selectors: [2, "js-", "strict"]*/

Template.foo.events({
  'click .foo': function () {},
  'click #foo': function () {},
  'click input': function () {},
  'click': function () {},
  'click [data-foo="bar"]': function () {},
})
```

Examples of **correct** code for the default `"relaxed"` mode:

```
/*eslint prefix-eventmap-selectors: [2, "js-", "strict"]*/

Template.foo.events({
  'click .js-foo': function () {}
})
```

When Not To Use It

This rule can be disabled if you are not using Blaze.

Possible Improvements

- forbid nested selectors `.js-foo .bar`, `.js-foo.bar`, `.js-foo#bar`, `#bar.js-foo`, `.js-foo + .bar`
- enable switching on/off errors for selection by attribute, nesting, plain (no selector), ..

Further Reading

- <http://guide.meteor.com/blaze.html#js-selectors-for-events>