## Contributing

\* Pull requests are welcome

\* Please `don't include dist/\* files` on your commits.

## Coding convention

\* eslint: https://eslint.org

\* eslint rule: https://github.com/summernote/summernote/blob/master/.eslintrc

## Build summernote

Summernote uses [`yarn`](https://yarnpkg.com/) as a package manager.

```bash

$ yarn install

# build full version of summernote: dist/summernote.js

$ yarn build

```

At this point, you should now have a `dist/` directory populated with everything you need to use summernote.

## Start local server for developing summernote.

run local server with webpack-dev-server and watch.

```bash

$ yarn dev

# Open a browser on http://localhost:3000.

# If you change source code, automatically reload your page.

```

## Test summernote

run tests with Karma and PhantomJS

```bash

$ yarn test

```

If you want run tests on other browser,

change the values for `browsers` properties in `karma.conf.js`.

```javascript

karma: {

all: {

browsers: ['PhantomJS'],

reporters: ['progress']

}

}

```

Or, pass `--browsers` argument via `yarn test` command.

```bash

$ yarn test -- --browsers Safari,Firefox

```

You can use `Chrome`, `ChromeCanary`, `Firefox`, `Opera`, `Safari`, `PhantomJS` and `IE` beside `PhantomJS`.

Once you run `yarn test`, it will watch all JavaScript file. Therefore karma runs tests every time you change code.

## Test a part of test

If you would like to run some part of your test codes, use the watch mode.

```bash

$ yarn test:watch

```

`karma` will run test and keep waiting other test requests. And then, run `test:grep` in another terminal. Below shows how to run `LinkDialog` related tests only.

```bash

$ yarn test:grep LinkDialog

```

## Prepush Hooks

As part of this repo, we use [Husky](https://github.com/typicode/husky) for git hooks. We leverage the prepush hook to prevent bad commits.

## Document structure

```text

- body container: <div class="note-editable">, <td>, <blockquote>, <ul>

- block node: <div>, <p>, <li>, <h1>, <table>

- void block node: <hr>

- inline node: <span>, <b>, <font>, <a>, ...

- void inline node: <img>

- text node: #text

```

1. A body container has block node, but `<ul>` has only `<li>` nodes.

1. A body container also has inline nodes sometimes. This inline nodes will be wrapped with `<p>` when enter key pressed.

1. A block node only has inline nodes.

1. A inline nodes has another inline nodes

1. `#text` and void inline node doesn't have children.