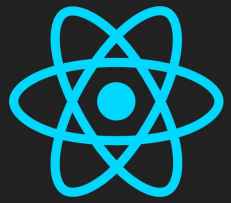


React.js

Lecture 2

Picanteverde
(aka Alejandro Hernández)

https://commons.wikimedia.org/wiki/File:React.js_logo.svg

Tools!

Transform

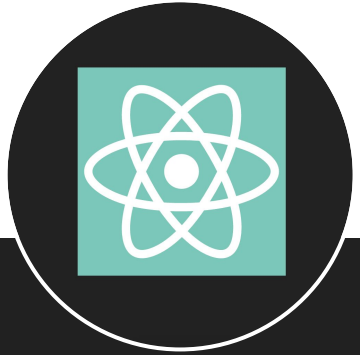
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React dev Tools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State

Transform

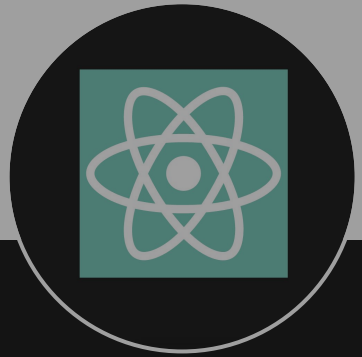
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



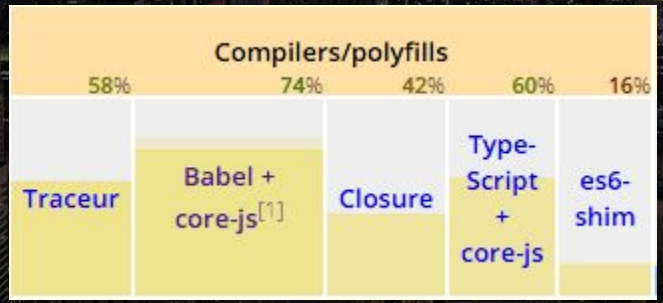
React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State

Compiling to Javascript

https://github.com/jashkenas/coffeescript/wiki/list-of-languages-that-compile-to-js

Is not the only one…
…but it's the best...



…for now.

https://kangax.github.io/compat-table/es6/#babel


BABEL
Installation
$ npm install --save-dev babel-cli
Local not global

Usage

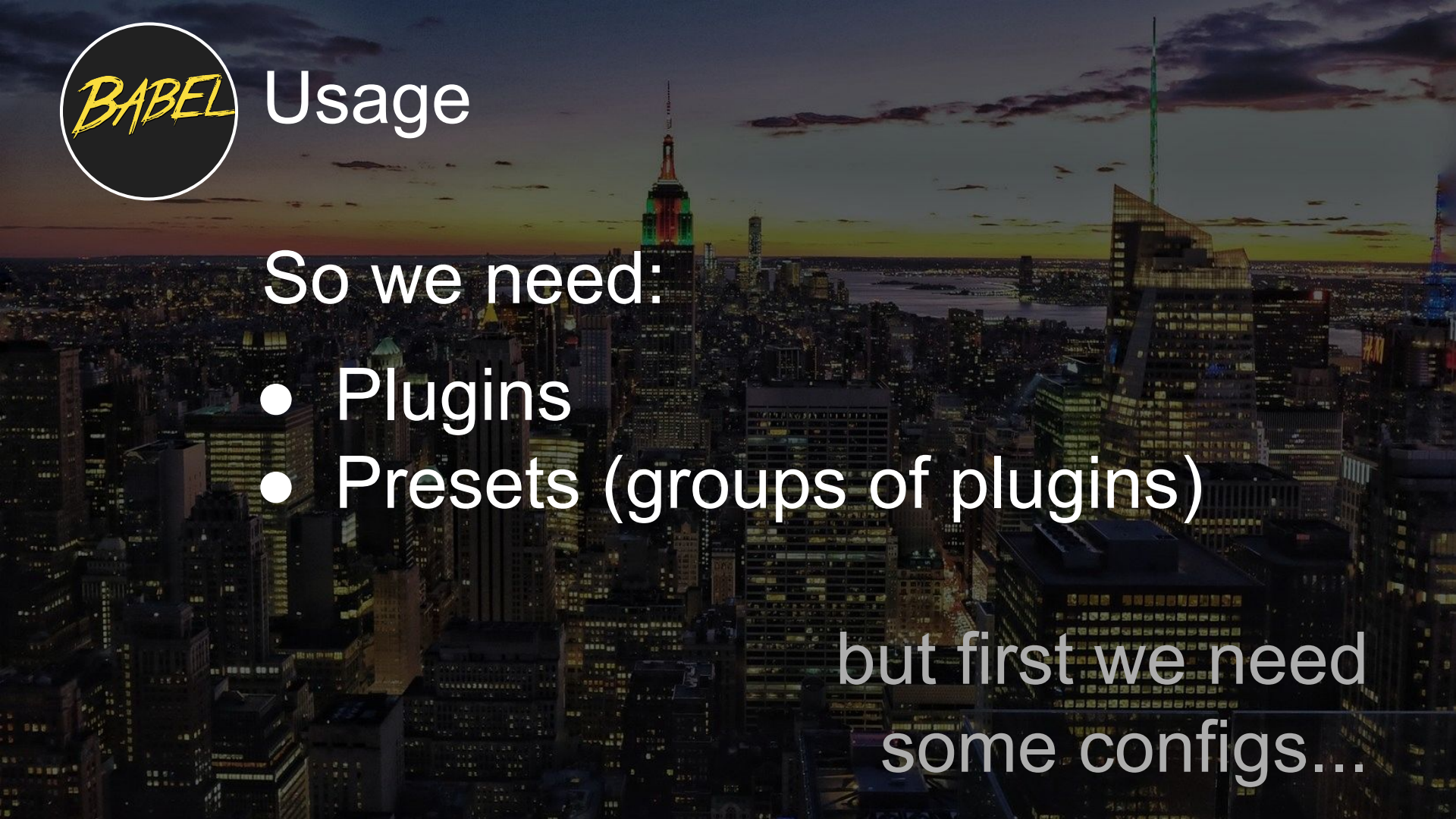
```
$ babel file.js
$ babel file.js --out-file compiled.js
$ babel file.js -o compiled.js
$ babel src --out-dir lib
$ babel src -d lib
```

Usage

But it doesn’t work!!!

./node_modules/.bin/babel

Babel does nothing out of the box!

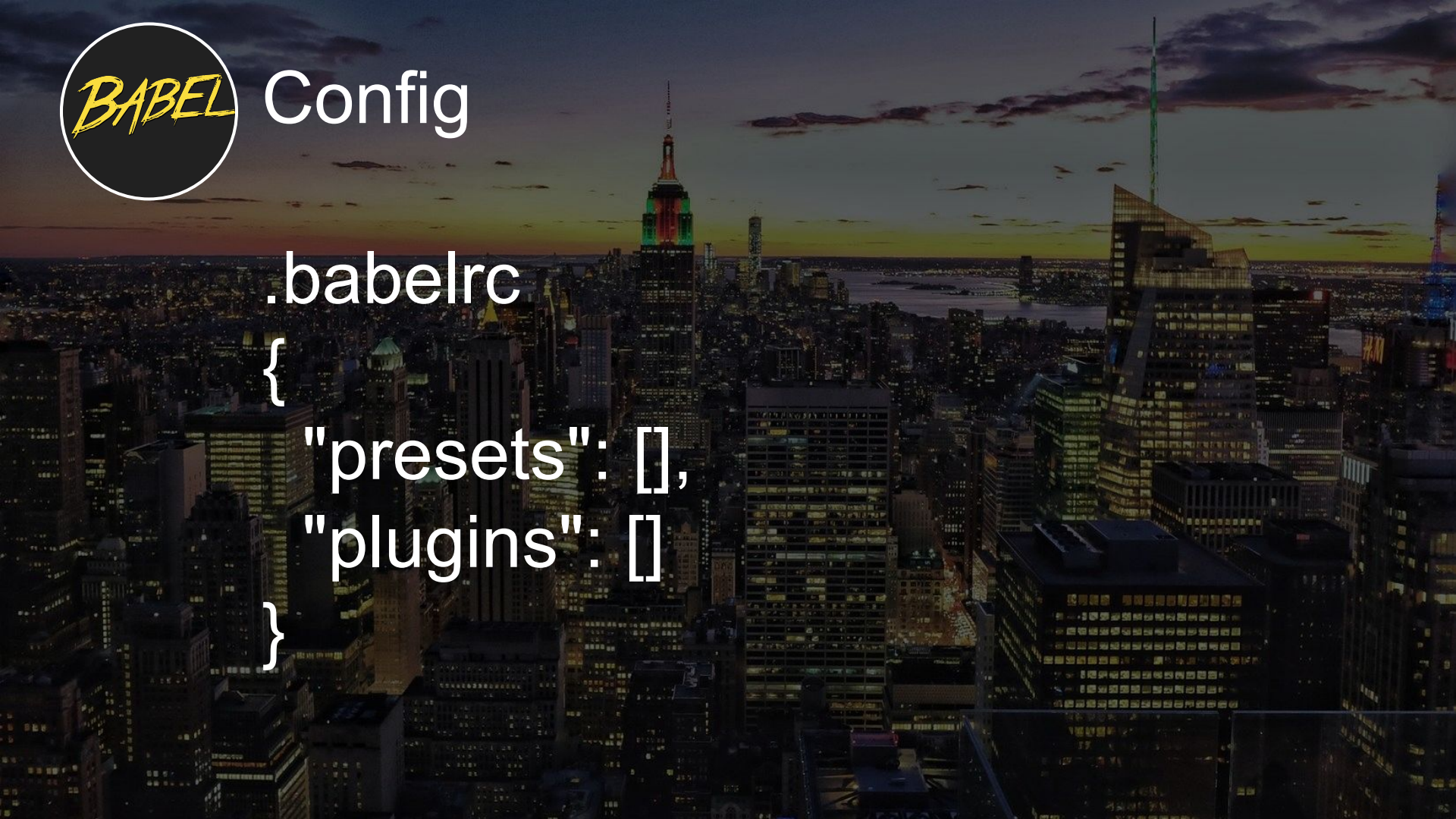


Usage

So we need:

- Plugins
- Presets (groups of plugins)

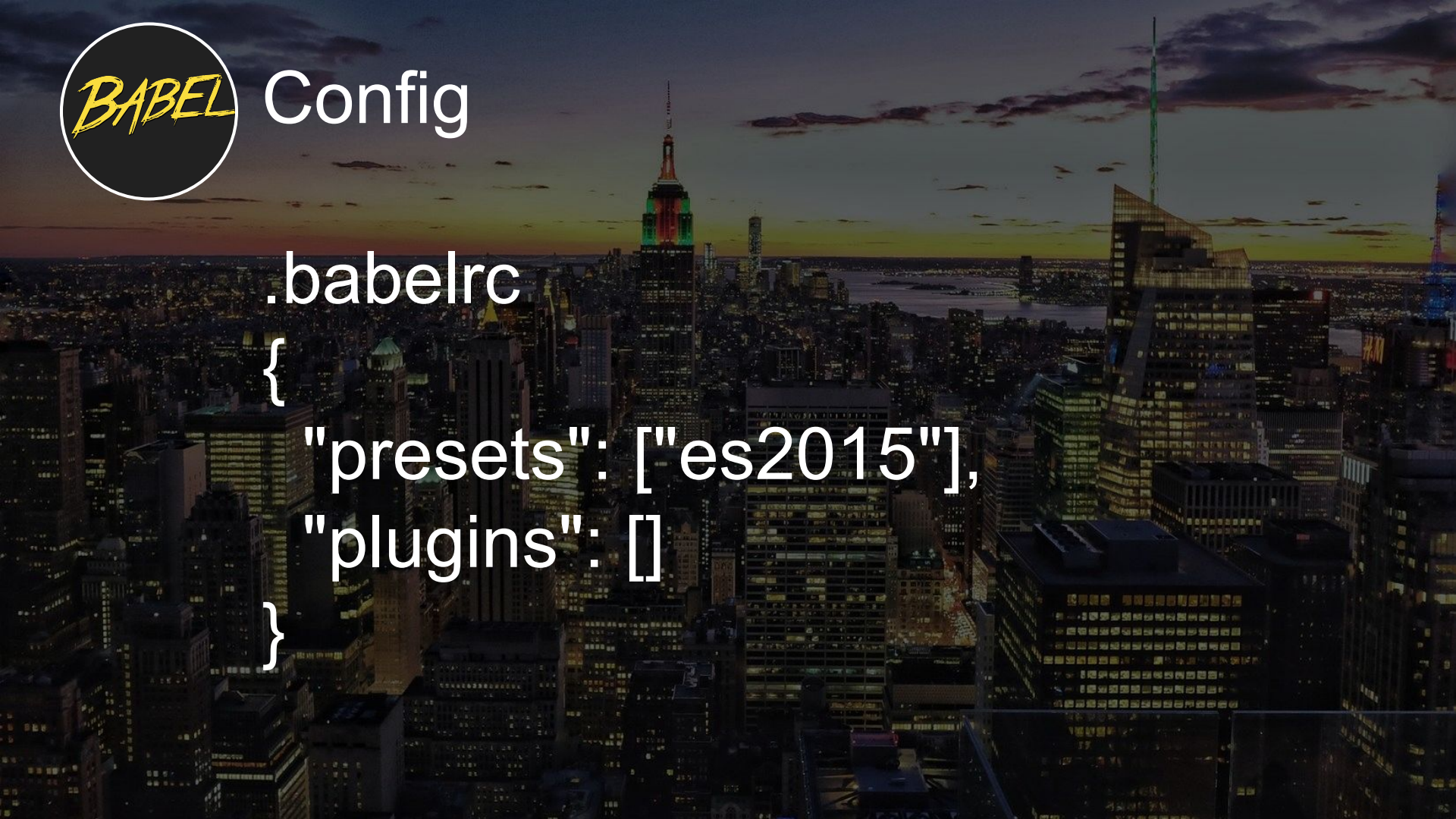
but first we need some configs...


BABEL
Config
.babelrc
{
"presets": [],
"plugins": []
}

Plugins & Presets

Install ES6 I mean ES2015

```
$ npm install --save-dev babel-preset-es2015
```


BABEL
Config
.babelrc
{
"presets": ["es2015"],
"plugins": []
}

Plugins & Presets

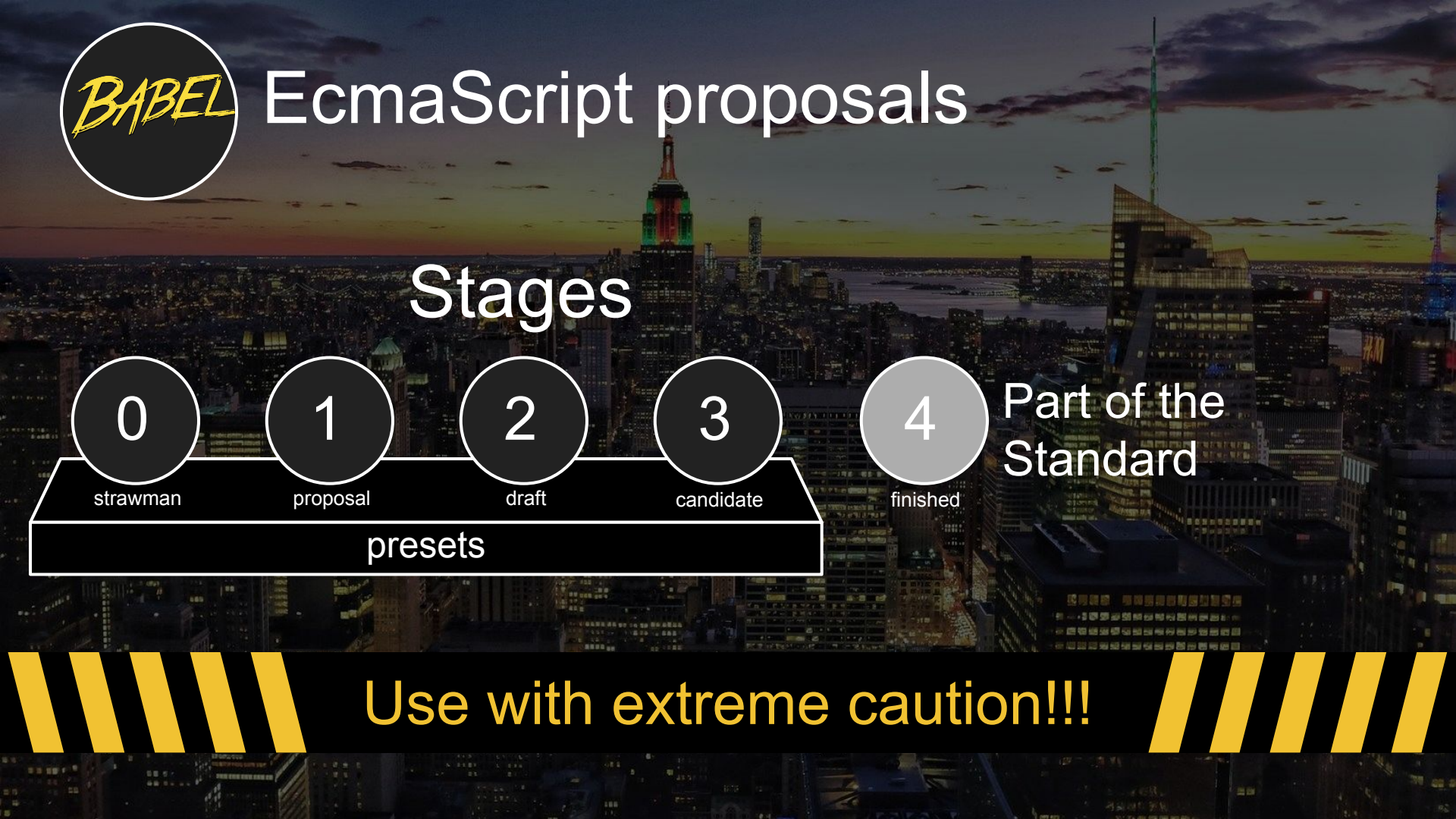
And React.js

$ npm install --save-dev babel-preset-react

Config

.babelrc

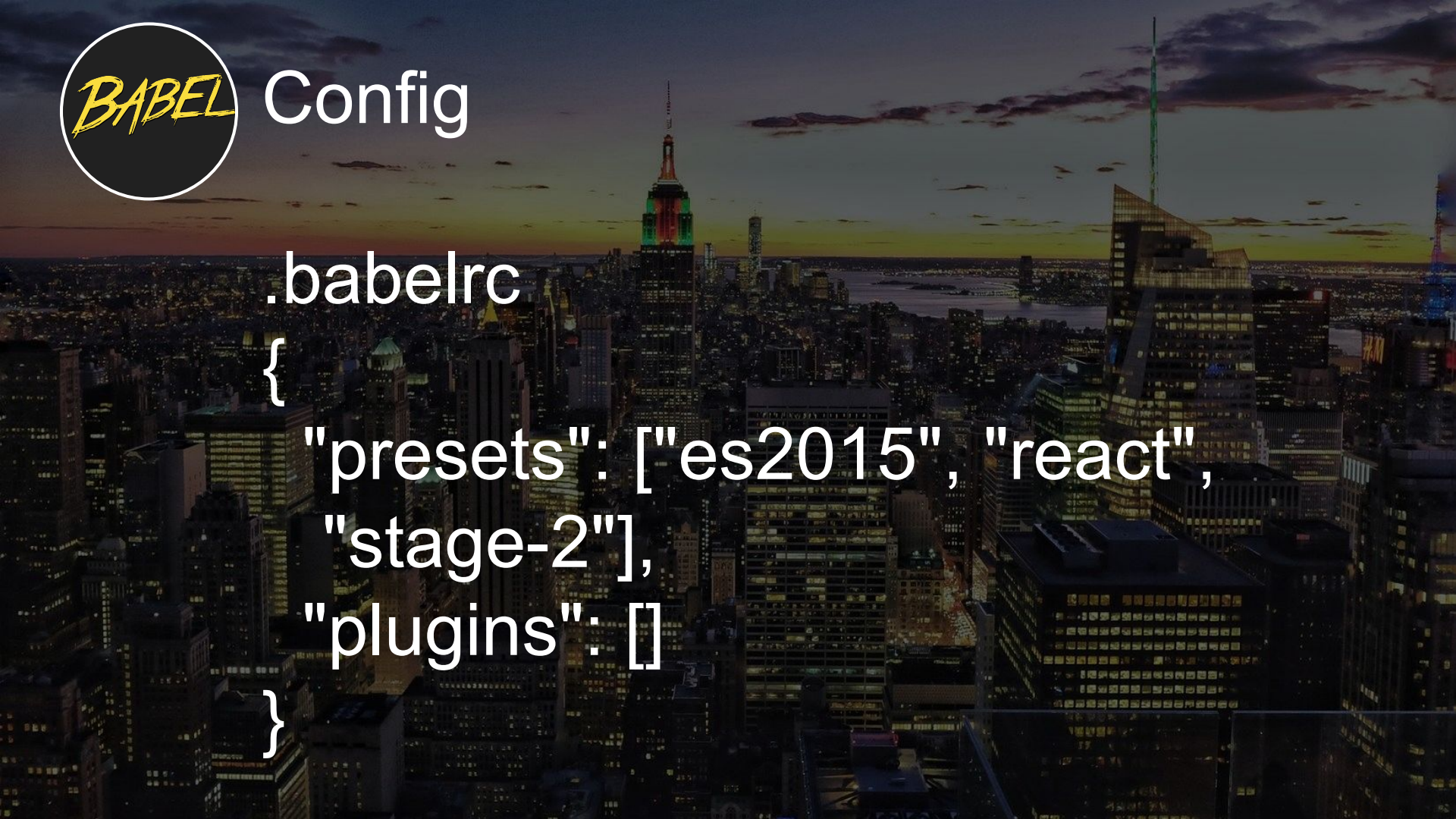
```
{
  "presets": ["es2015", "react"],
  "plugins": []
}
```


BABEL
EcmaScript proposals
Stages
0
1
2
3
4
strawman
proposal
draft
candidate
finished
presets
Part of the Standard
Use with extreme caution!!!

Plugins & Presets

install

```
$ npm install --save-dev babel-preset-stage-2
```


BABEL
Config
.babelrc
{
"presets": ["es2015", "react",
"stage-2"],
"plugins": []
}

EcmaScript proposals

dependencies

EcmaScript proposals

Content

http://babeljs.io/docs/plugins/preset-stage-2/

http://babeljs.io/docs/plugins/transform-object-rest-spread/


BABEL
Executing
new Syntax ⇔ new APIs
Polyfills!!!!
core-js regenerator
babel-polyfill

Installing polyfills

```
$ npm install --save babel-polyfill
```

```
import "babel-polyfill";
```

Runtime

Helper methods (demo classes)

$ npm install --save-dev babel-plugin-transform-runtime

$ npm install --save babel-runtime


BABEL
Runtime
.babelrc
{
"plugins": [
"transform-runtime"
]
}

Cool, ah?

Transform

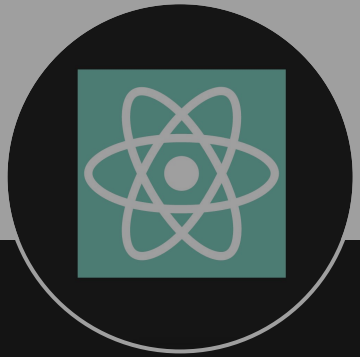
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State

Transform

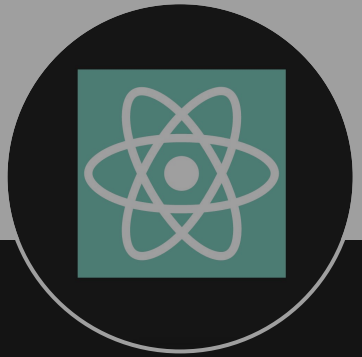
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

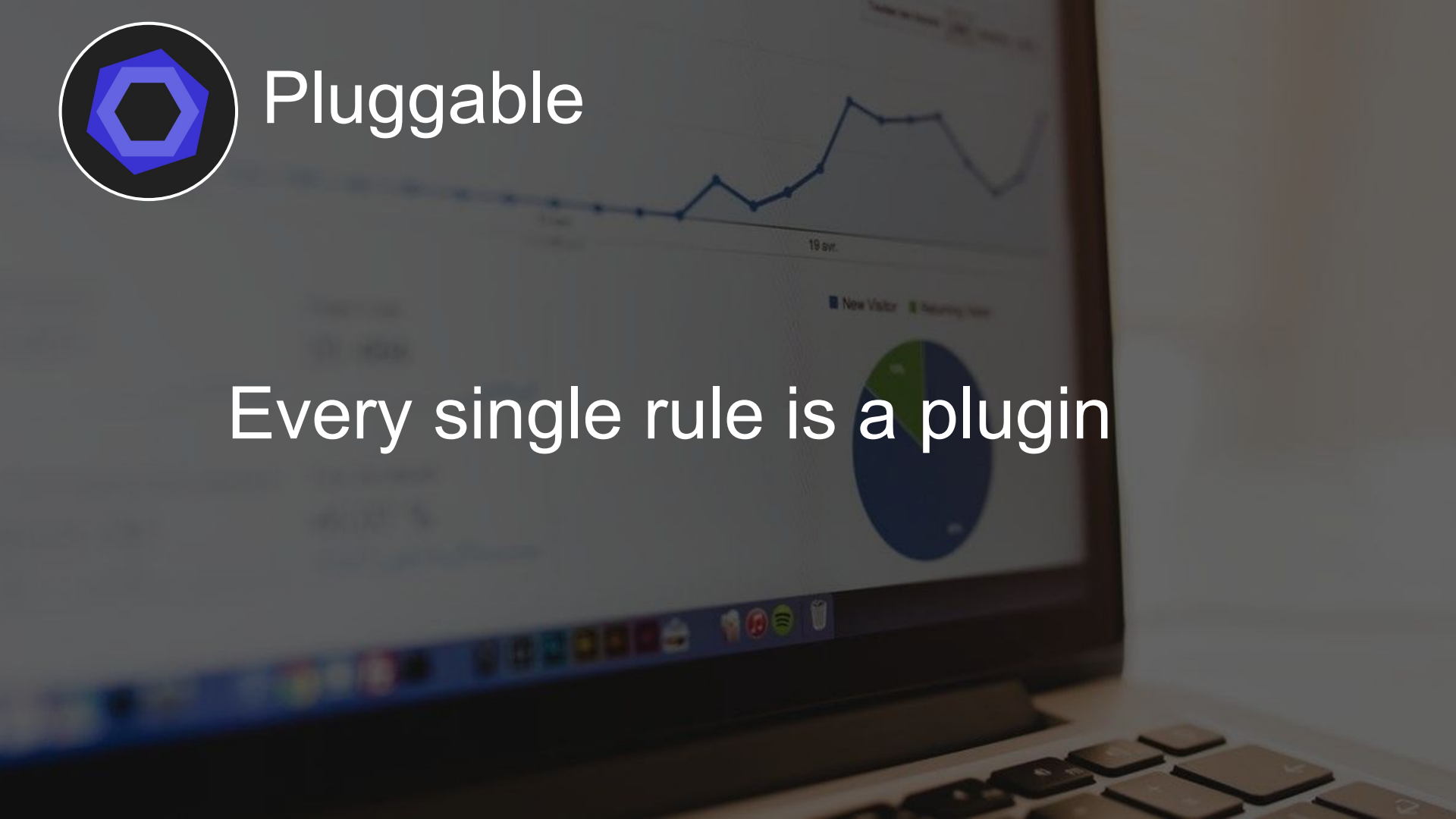
Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State


ESLint
JSLint
JS Hint
JSCS


Pluggable
Every single rule is a plugin

Installation

```
$ npm install -g eslint
```

Yeay! global

Configuring

$ eslint --init

Creates the eslintrc for us

Running

```
$ eslint src
```


ESLint
That’s it…?


ESLint
Well... no.

Configuring

Two ways

comments

- Embed configuration on the file
- Not recommended

```
/*eslint eqeqeq: 0, curly: 2*/
```

files

- JS, JSON, YAML formats
- eslintConfig in package.json
- Extendable
- Cascading and Hierarchy

.eslint.json

Configuring - files

- parserOptions
- parser
- Environment
- Globals
- Rules
- Extend
- Plugins

Configuring - parserOptions

```
"parserOptions": {
  "ecmaVersion": 6, // 3 5 6 7
  "sourceType": "module", // "script" (default)
  "ecmaFeatures": {
    "jsx": true,
    "experimentalObjectRestSpread": true
  }
}
```

Configuring - parser

Espree - Esprima - Babel-ESLint

- Must be an npm module installed locally
- Esprima-compatible interface (parse method)
- Produce Esprima-compatible AST

Configuring - parser

```
$ npm install --save-dev babel-eslint
```

```
{
  "parser": "babel-eslint"
}
```

Configuring - environments

Predefined global variables

- browser
- node
- es6
- amd
- commonjs
- shared-node-browser
- worker
- serviceworker

Configuring - globals

Any other global variable

```
"globals": {
  "var1": true,
  "var2": true
}
```

Configuring - rules

Large set of rules

- "off" or 0
- "warn" or 1
- "error" or 2

```
"rules": {
  "eqeqeq": 0,
  "curly": "error",
  "quotes": [ "error", "double" ]
}
```

http://eslint.org/docs/rules/

Configuring - rules

There are some fixable rules

```
$ eslint src --fix
```

http://eslint.org/docs/rules/

Configuring - cascading & hierarchy

```
your-project
├── .eslintrc.json // "root":true
├── server
│   ├── .eslintrc.json // "env": { "node": true}
│   └── server.js
├── client
│   ├── .eslintrc.json // "env": { "browser": true}
│   └── index.js
└── tests
    ├── .eslintrc.json // "env": { "mocha": true}
    └── tests.js
```

Configuring - extend

```
{
  "extend":
  "./node_modules/coding-style/.eslintrc.json"
}
```

Configuring - plugins

```
$ npm install --save-dev eslint-plugin-react
```

```
{
  "extend": "plugin:react/recommended",
  "plugins": [ "react" ],
}
```

https://github.com/airbnb/javascript/tree/master/packages/eslint-config-airbnb

Good one, right?

Transform

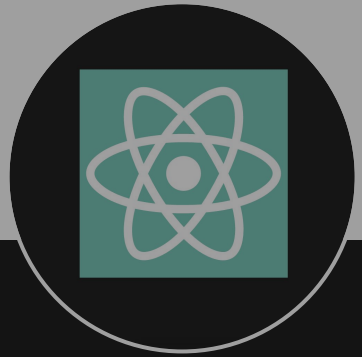
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State

Transform

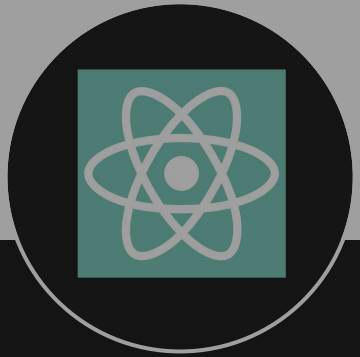
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

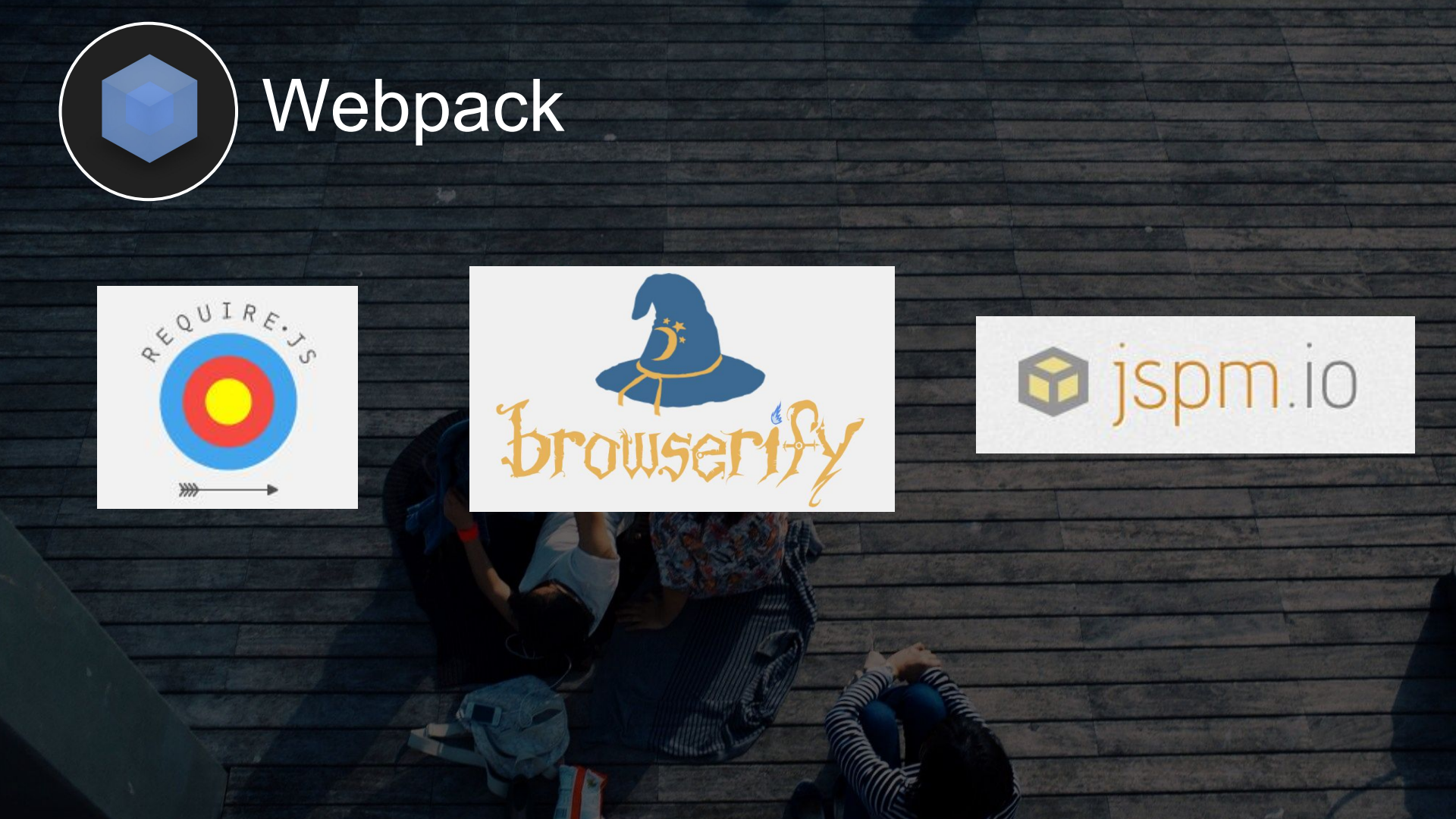
webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State


Webpack
REQUIRE.JS
browserify
jspm.io

Webpack

- AMD commonJS ES6 modules
- Chunks (1req x mod or all-in-1req)
- Why only js? (css, imgs)

Installation

$ npm install -g webpack

global again! but..

$ npm install --save-dev webpack

Usage

```
$ webpack src/index.js public/js/bundle.js
```

but..

Configure with webpack

```
$ npm install --save-dev babel-loader babel-core
```

```
{
    module: {
        loaders: [
            {
                test: /\.js$/,
                loader: 'babel-loader'
            }
        ]
    }
}
```

Usage

$ webpack --progress --watch

devtool eval or source-map

Plugins

$ npm install --save-dev webpack-notifier

```
var WebpackNotifierPlugin = require('webpack-notifier');

plugins: [
      new WebpackNotifierPlugin({alwaysNotify: true})
   ]
```

Finally!

Transform

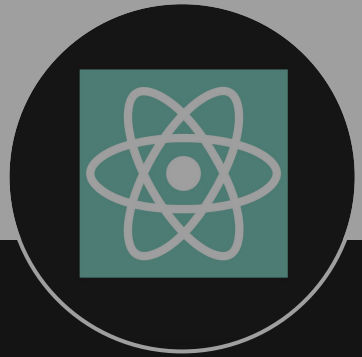
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State

Transform

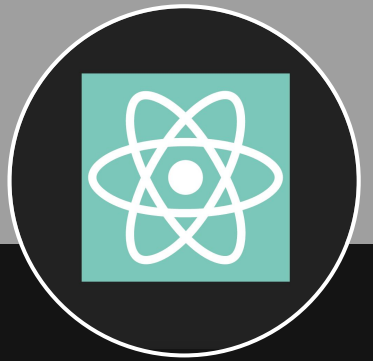
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

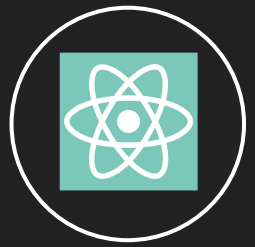
webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React devTools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State



React developer tools

- React elements inspector tab
- Enabled only when react is present
- Shows Elements, props, state, and events

That’s it!

Transform

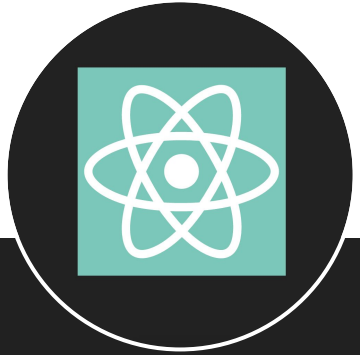
JSX + ES2015 + ES2016 + ES2017 + plugins to ES5

ESLint

Pluggable linting, best practices, programing styles, and guidelines

webpack

Pluggable module bundler, loaders & tools. Automate your build JS, CSS, tmpl, imgs & font



React dev Tools

Allow us to inspect the tree of React Element. Shows properties Event handlers and State