Chapter 19: Setting Up React Environment

Section 19.1: Simple React Component

We want to be able to compile below component and render it in our webpage

Filename: src/index.jsx

```
import React from 'react';
import ReactDOM from 'react-dom';

class ToDo extends React.Component {
    render() {
        return (<div>I am working</div>);
    }
}

ReactDOM.render(<ToDo />, document.getElementById('App'));
```

Section 19.2: Install all dependencies

```
# install react and react-dom
$ npm i react react-dom --save

# install webpack for bundling
$ npm i webpack -g

# install babel for module loading, bundling and transpiling
$ npm i babel-core babel-loader --save

# install babel presets for react and es6
$ npm i babel-preset-react babel-preset-es2015 --save
```

Section 19.3: Configure webpack

Create a file `webpack.config.js` in the root of your working directory

Filename: webpack.config.js

```
module.exports = {
    entry: __dirname + "/src/index.jsx",
    devtool: "source-map",
    output: {
        path: __dirname + "/build",
        filename: "bundle.js"
    },
    module: {
        loaders: [
            {test: /\.jsx?$/, exclude: /node_modules/, loader: "babel-loader"}
        ]
    }
}
```

Section 19.4: Configure babel

Create a file `.babelrc` in the root of our working directory

Filename: .babelrc

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

Section 19.5: HTML file to use react component

Setup a simple html file in the root of the project directory

Filename: index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <div id="App"></div>
    <script src="build/bundle.js" charset="utf-8"></script>
  </body>
</html>
```

Section 19.6: Transpile and bundle your component

Using webpack, you can bundle your component:

$ webpack

This will create our output file in `build` directory.

Open the HTML page in a browser to see component in action