# Hướng dẫn cài ứng dụng sails, reactjs với webpack:

* Tạo mới Project sails: sails new <ten project>
* Chỉnh sửa trong config : alter

## 1.1: Cài webpack, sử dụng Reactjs trong prj:

- Tạo 1 <div id=”layout”> để chứa các Component

- npm install grunt-babel babel-preset-es2015 babel-preset-react path react react-dom babel-loader babel-core

- Tạo 1 folder App chứa file index, bao gồm các Component: ReactDOM.render(…) -> bundle file này

- tạo file webpack.config.js

- npm install webpack –save-dev-g

- Run webpack –w and sails lift => done

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WEBPACK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### Các thư viện sử dụng cho Reactjs

          - thư viên react, react-dom để có thể sử dụng react trong project

          - thư viện redux để sử dụng các thư viện trong redux vào project

          - thư viện react-redux để quản lý state trong reactjs bởi redux

          - thư viện react-router để sử dụng router react trong project

          - webpack ....

#### 1.2 Cài đặt webpack và cấu hình file webpack.config.js

###### 1.2.1 công dụng của webpack

                 - Webpack là công cụ để ta đóng gói các file js, css .. vào static asset để sử dụng trong ứng dụng, có thể đọc thêm tại <https://webpack.github.io/> .

                - Công dụng của nó:

                     +Bundle các file JS thành 1 file JS

                     +Bundle ra các file JS chính từ nhiều file JS

                     +Chia file JS và load khi cần

                     +Nhúng HTML vào trong file JS

                     +Nhúng CSS vào trong file JS

                     +Nhúng file hình Base64 vào trong file JS

                - Cài đặt Webpack trên toàn cầu:

                     +npm install webpack -g

###### 1.2.2 Cấu hình file webpack.config.js trong project

                -  **entry** **:** tên của file hoặc một mảng những file mà chúng ta muốn include. Trong project đang dùng 1 file duy nhất là index.js.

                - **output :** là một object chứa những thiết lập output. Trong project chỉ ra filename là bundle.js là file mà chúng ta muốn Webpack build ra.

                +  thuộc tính **path** là đường dẫn thư mục gốc.

**module :**sẽ khai báo những thứ mà webpack cần sử dụng để nó có thể hiểu được các đoạn mã jsx hay html trong file mà nó nén vào file bundle.js

                 - cần các các gói tương ứng trong project :

**npm install babel-core babel-loader babel-preset-es2015 babel-preset-react webpack --save-dev**

                 -**resolve :**chỉ rõ các file cần được xử lý ..

                         + thuộc tính **extensions** : tự động giải quyết các tiện ích mở rộng nhất định này.

                        + thuộc tính **alias** : phục vụ cho việc requerie import các file dễ dàng hơn.

                         xem chi tiết tại <https://webpack.js.org/configuration/>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## 1.2: Cài Redux trong Project:

- npm install redux redux-thunk react-redux redux-devtools-extension –save

- Trong folder app tạo các folder Reducer, Action và Components, tạo file index và file Store.js

- Nội dụng file Store.js

var redux = require('redux');

var reducer = require('reducer');

import thunk from 'redux-thunk';

//var store = redux.createStore(reducer);

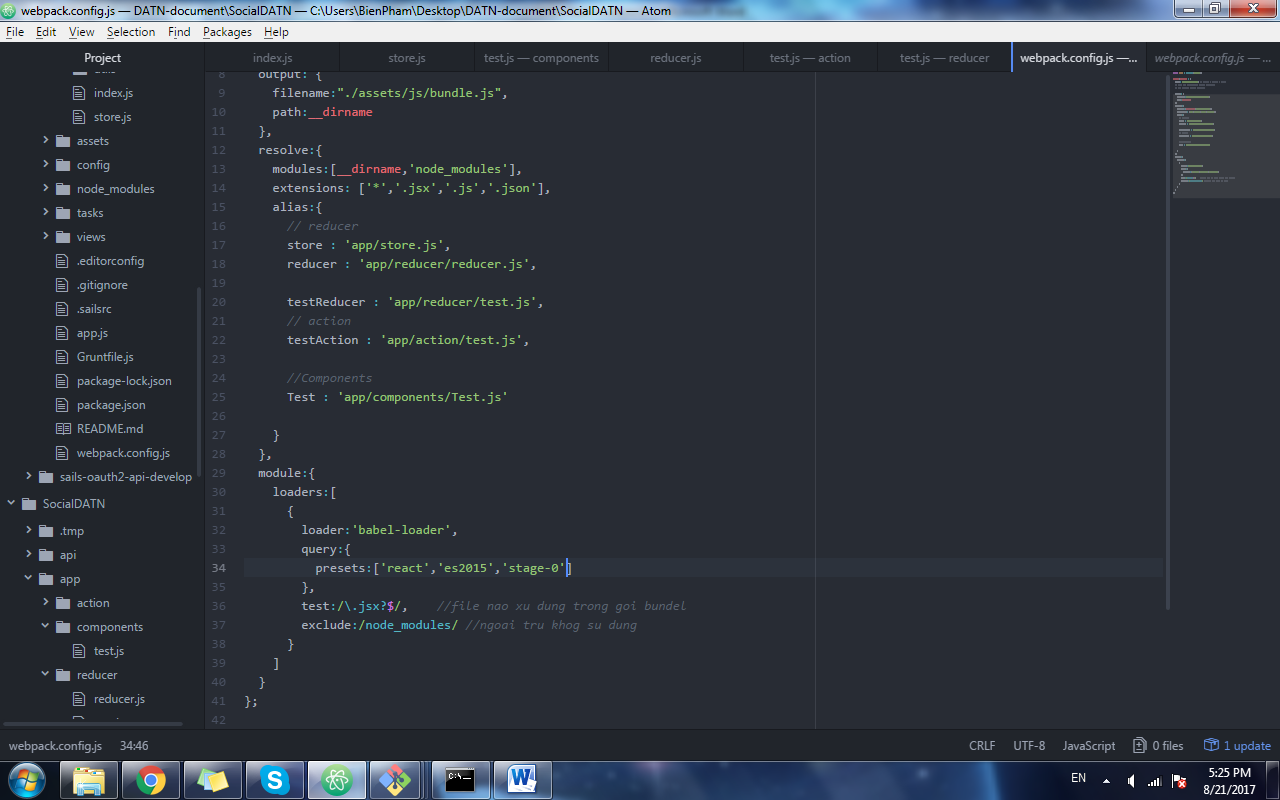
var store = redux.createStore(reducer,redux.compose(redux.applyMiddleware(thunk),

window.devToolsExtension? window.devToolsExtension(): f => f

));

module.exports = store;

* Trong folder Reducer tạo file reducer.js để combine các Reducer con lại để cho vào trong store.js
* Để sử dụng toán tử … , thêm state-0 vào trong file webpack.config như dưới



* Để sử dụng Redux trong các Component, trong file index Component khai báo sử dụng Router và Provider, npm install react-router-dom –save

