**vds5-react-react bootstrap.胡永钢.V1.00**

（2016/8/24）

目录

[1 安装方法 3](#_Toc459834631)

[1.1 Node & Npm环境 3](#_Toc459834632)

[1.2 预编译模式安装 3](#_Toc459834633)

[1.3 配置文件说明 5](#_Toc459834634)

[2 react-bootstrap常用控件 11](#_Toc459834635)

[2.1 ButtonToolbar & Button 11](#_Toc459834636)

[2.2 FormGroup & FormControl & ControlLabel & HelpBlock 12](#_Toc459834637)

[2.3 Checkbox & Radio 13](#_Toc459834638)

[3 编译代码 14](#_Toc459834639)

[4 访问项目 15](#_Toc459834640)

# 安装方法

## Node & Npm环境

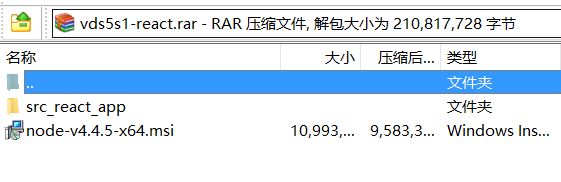
Node安装包路径: \\vds5s1-react\node-v4.4.5-x64.msi, NPM（node package manager），称为node包管理器，Node安装包自带NPM工具。

本实例使用Node版本为v4.4.5,NPM版本为2.15.5。

参考资料<http://www.tuicool.com/articles/VB7nYn>

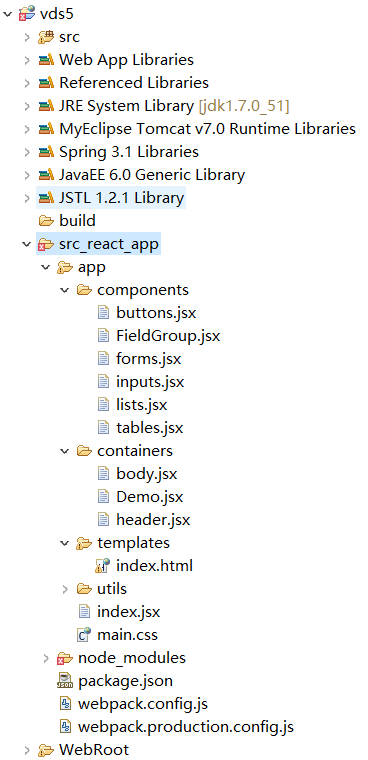
## 预编译模式安装

第一步、解压vds5s1-react.rar 如下：



第二步、将vds5s1-react.rarsrc\_react\_app文件夹copy到工作空间项目根目录下，如下图：

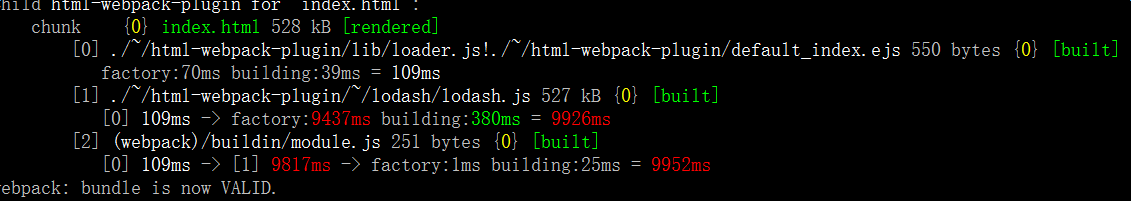
(copy过程中可能会因为node\_modules文件夹里面文件路径过长报错，建议先将vds5s1-react.rar复制到项目路径在解压)



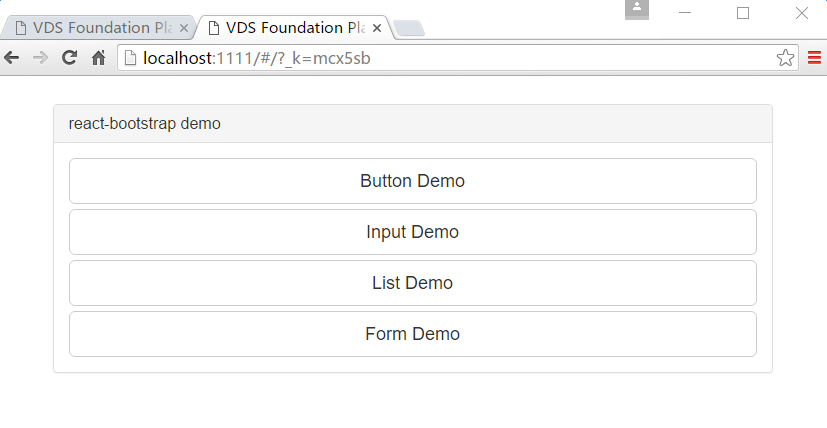
第三步、在cmd控制台进入src\_react\_app目录执行npm run dev运行项目



第四步、看到如下图片出现的信息则node服务端开启成功，



浏览器访问<http://localhost:1111/> ，将会看到下面图片内容，表示运行成功。



## 配置文件说明

#### package.json

|  |
| --- |
| {  "name": "vds5s1",  "version": "1.0.0",  "description": "VDS Foundation Platform",  "main": "index.js",  "scripts": {  "dev": "webpack-dev-server --progress --profile --colors --hot",  "build": "set NODE\_ENV=production&&webpack -p --process --colors --profile --config webpack.production.config.js"  },  "author": "B.D.",  "license": "ISC",  "devDependencies": {  "babel-core": "^6.13.2",  "babel-loader": "^6.2.5",  "babel-plugin-react-transform": "^2.0.2",  "babel-preset-es2015": "^6.13.2",  "babel-preset-react": "^6.11.1",  "babel-preset-react-hmre": "^1.1.1",  "css-loader": "^0.23.1",  "extract-text-webpack-plugin": "^1.0.1",  "file-loader": "^0.9.0",  "html-webpack-plugin": "^2.22.0",  "imports-loader": "^0.6.5",  "jquery": "^3.1.0",  "jshint": "^2.9.2",  "jshint-loader": "^0.8.3",  "moment": "^2.14.1",  "node-sass": "^3.8.0",  "react-transform-catch-errors": "^1.0.2",  "react-transform-hmr": "^1.0.4",  "redbox-react": "^1.3.0",  "sass-loader": "^4.0.0",  "style-loader": "^0.13.1",  "url-loader": "^0.5.7",  "webpack": "^1.13.1",  "webpack-dev-server": "^1.14.1"  },  "dependencies": {  "bootstrap": "^3.3.5",  "react": "^15.3.0",  "react-bootstrap": "^0.30.2",  "react-dom": "^15.3.0",  "react-router": "^2.7.0"  }  } |

重要字段说明：

name: 项目名

version: 版本

description: 项目描述

scripts: 命令定义

(例子：npm run dev调用就是"webpack-dev-server --progress --profile --colors --hot)

参考资料: <http://my.oschina.net/u/1992917/blog/523046>

#### webpack.config.js（开发模式npm run dev调用）

|  |
| --- |
| const path = require('path');  const webpack = require('webpack');  const HtmlwebpackPlugin = require('html-webpack-plugin');  const ROOT\_PATH = path.resolve(\_\_dirname);  const APP\_PATH = path.resolve(ROOT\_PATH, 'app');  const BUILD\_PATH = path.resolve(ROOT\_PATH, 'build');  const config = {  entry: {  vds5: path.resolve(APP\_PATH, 'index.jsx')  },  output: {  path: BUILD\_PATH,  filename: '[name].bundle.js'  },  devtool: 'eval-source-map',  devServer: {  port: 1111,  historyApiFallback: true,  hot: true,  inline: true,  progress: true  },  module: {  loaders: [{  test: /\.jsx?$/,  loader: 'babel',  include: APP\_PATH  }, {  test: /\.css$/,  loaders: ['style', 'css']  }, {  test: /\.scss$/,  loaders: ['style', 'css', 'sass']  }, {  test: /\.svg$/,  loader: 'url?limit=65000&mimetype=image/svg+xml&name=assets/fonts/[name].[ext]'  }, {  test: /\.woff$/,  loader: 'url?limit=65000&mimetype=application/font-woff&name=assets/fonts/[name].[ext]'  }, {  test: /\.woff2$/,  loader: 'url?limit=65000&mimetype=application/font-woff2&name=assets/fonts/[name].[ext]'  }, {  test: /\.[ot]tf$/,  loader: 'url?limit=65000&mimetype=application/octet-stream&name=assets/fonts/[name].[ext]'  }, {  test: /\.eot$/,  loader: 'url?limit=65000&mimetype=application/vnd.ms-fontobject&name=assets/fonts/[name].[ext]'  }, ]  },  resolve: {  /\*自动补全\*/  extensions: ['', '.js', '.jsx']  },  plugins: [  new webpack.optimize.UglifyJsPlugin({  minimize: true  }),  new HtmlwebpackPlugin({  title: 'VDS Foundation Platform',  template: './app/templates/index.html',  filename: 'index.html',  inject: 'body'  })  ]  };  module.exports = config |

关键点说明：

entry：entry 当中是 js 文件引用的入口, 一般每个 entry 的属性对应一个文件(这个例子的入口是 …/src\_react\_src/app/index.jsx)

output: 生成的文件怎样访问的问题, 主要 path 和 filename 两个属性(这个例子的输出编译好的路径是…/src\_react\_src/build/vds5.bundle.js)

|  |
| --- |
| new HtmlwebpackPlugin({  title: 'VDS Foundation Platform',  template: './app/templates/index.html',  filename: 'index.html',  inject: 'body'  }) |

title: 用做于生成文件(模板路径是 src\_react\_app/app/templates/index.html)的title。

filename：生成的文件名叫做index.html。

inject: js插入在body标签。

#### webpack.production.config.js（生产模式npm run build调用）

|  |
| --- |
| const path = require('path');  const webpack = require('webpack');  const HtmlwebpackPlugin = require('html-webpack-plugin');  const ExtractTextPlugin = require("extract-text-webpack-plugin");  const env = process.env.NODE\_ENV;  const ROOT\_PATH = path.resolve(\_\_dirname);  const APP\_PATH = path.resolve(ROOT\_PATH, 'app');  const BUILD\_PATH = path.resolve(ROOT\_PATH, '../WebRoot/vdsapp');  const config = {  devtool: 'cheap-module-source-map',  entry: {  vds5: path.resolve(APP\_PATH, 'index.jsx'),  //vendors: ['']  },  output: {  path: BUILD\_PATH,  filename: 'assets/[name].bundle.min.js'  },  module: {  loaders: [{  test: /\.jsx?$/,  loader: 'babel',  include: APP\_PATH  }, {  test: /\.css$/,  loader: ExtractTextPlugin.extract("style-loader", "css-loader")  }, {  test: /\.scss$/,  loader: ExtractTextPlugin.extract('style-loader', 'css!sass')  }, {  test: /\.svg$/,  loader: 'url?limit=65000&mimetype=image/svg+xml&name=assets/fonts/[name].[ext]'  }, {  test: /\.woff$/,  loader: 'url?limit=65000&mimetype=application/font-woff&name=assets/fonts/[name].[ext]'  }, {  test: /\.woff2$/,  loader: 'url?limit=65000&mimetype=application/font-woff2&name=assets/fonts/[name].[ext]'  }, {  test: /\.[ot]tf$/,  loader: 'url?limit=65000&mimetype=application/octet-stream&name=assets/fonts/[name].[ext]'  }, {  test: /\.eot$/,  loader: 'url?limit=65000&mimetype=application/vnd.ms-fontobject&name=assets/fonts/[name].[ext]'  }, ]  },  resolve: {  extensions: ['', '.js', '.jsx']  },  plugins: [  new webpack.optimize.UglifyJsPlugin({  minimize: true,  compress: {  warnings: false  }  }),  /\*new webpack.optimize.CommonsChunkPlugin({  name: 'vendors',  chunks: ['vendors'],  filename: 'assets/vendors.js',  minChunks: Infinity  }),\*/  new webpack.DefinePlugin({  'process.env.NODE\_ENV': JSON.stringify(env)  }),  new ExtractTextPlugin("assets/[name]-style.min.css"),  new HtmlwebpackPlugin({  title: 'VDS Foundation Platform',  template: './app/templates/index.html',  filename: 'index.html',  inject: 'body'  })  ]  };  module.exports = config |

关键配置说明：

|  |
| --- |
| const BUILD\_PATH = path.resolve(ROOT\_PATH, '../WebRoot/vdsapp'); |

生产模式编译后的文件放在java web项目的WebRoot/vdsapp/文件夹

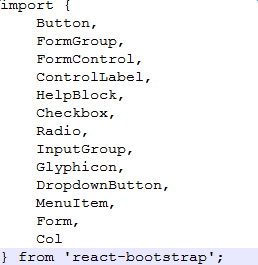
|  |
| --- |
| filename: 'assets/[name].bundle.min.js' |

|  |
| --- |
| new ExtractTextPlugin("assets/[name]-style.min.css"), |

编译后的js,css等静态资源文件放在WebRoot/vdsapp/assets文件夹，name为定义entry入口定义的key（例如本例子为vds5，js文件名为vds5.bundle.min.js，css文件名为vds5-style.min.css）

参考资料：<http://www.tuicool.com/articles/uQfmqie>

# react-bootstrap常用控件

使用到的控件的需要导入：

（有且不仅上面列出这些，不import的组件就使用会报错）。

参考资料：<https://react-bootstrap.github.io/components.html>

## ButtonToolbar & Button

一般ButtonToolbar和Button结合使用

例子：

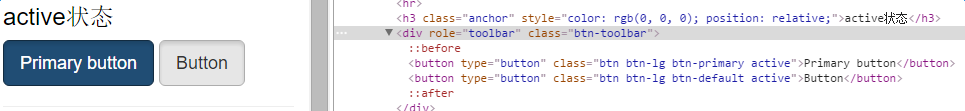
<ButtonToolbar>

<Button bsStyle="primary" bsSize="large" active>Primary button</Button>

<Button bsSize="large" active>Button</Button>

</ButtonToolbar>

渲染效果：



<Button />主要属性：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 解释 |
| active | boolean | False | 是否激活状态 |
| block | boolean | False | 是否为block块 |
| bsSize | "lg", "large", "sm","small", "xs", "xsmall" |  | 按钮尺寸 |
| bsStyle | "success", "warning","danger",  "info", "default","primary", "link" | 'default' | 按钮样式类型 |
| disabled | boolean | False | 是否禁用 |
| href | String |  | 链接 |
| onClick | Function |  | 点击事件 |
| type | 'button', 'reset','submit' |  | 按钮事件类型 |

## FormGroup & FormControl & ControlLabel & HelpBlock

FormGrouph可以和FormControl, ControlLabel, HelpBlock结合使用

例子：

<FormGroup controlId="formValidationSuccess1" validationState="success">

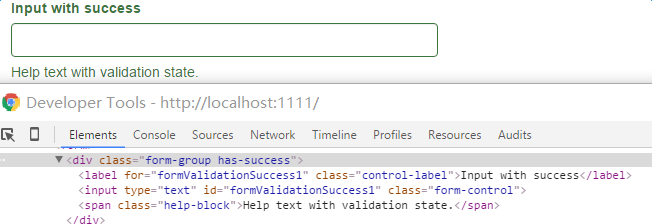
<ControlLabel>Input with success</ControlLabel>

<FormControl type="text" />

<HelpBlock>Help text with validation state.</HelpBlock>

</FormGroup>

渲染效果：



< FormGroup />主要参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 解释 |
| controlId | string |  | 设置里面<FormGroup>  标签的id |
| validationState | 'success','warning', 'error' |  | 验证的状态 |
| bsSize | "lg", "large", "sm","small", "xs", "xsmall" |  | 按钮尺寸 |

< FormControl/>主要参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 解释 |
| id | string |  | 如果这个属性为空，则取<FormGroup />的controlId属性 |
| placeholder | String |  | 输入框默认提示值 |
| componentClass | elementType | 'input' | 默认渲染成input输入框 |
| type | 'text','email', 'password ' |  | Input输入框类型 |

< ControlLabel />与< HelpBlock />主要参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 解释 |
| bsClass | string | < ControlLabel /> : 'control-label'  < HelpBlock /> : 'help-block' | css class |

## Checkbox & Radio

Checkbox和Radio可以和FormGroup组合使用

例子：

<FormGroup validationState="success">

<Checkbox inline>

Checkbox

</Checkbox>

{' '}

<Checkbox inline>

with

</Checkbox>

{' '}

<Checkbox inline>

success

</Checkbox>

</FormGroup>

渲染效果：

<Checkbox />主要属性：

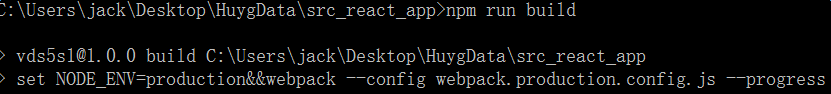
|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 解释 |
| bsClass | string | 'checkbox' | css class |
| disabled | boolean | false | 输入框默认提示值 |
| inline | boolean | false | 默认渲染成input输入框 |
| validationState | 'success','warning', 'error' |  | 验证的状态（inline 属性设置此属性无效） |

< Radio />主要属性：

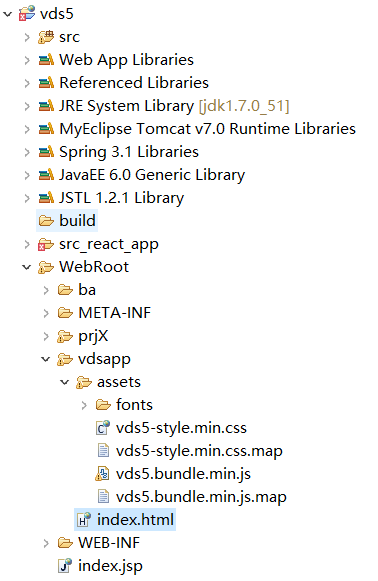
|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 解释 |
| bsClass | string | 'radio' | css class |
| disabled | boolean | false | 输入框默认提示值 |
| inline | boolean | false | 默认渲染成input输入框 |
| validationState | 'success','warning', 'error' |  | 验证的状态（inline 属性设置此属性无效） |

# 编译代码

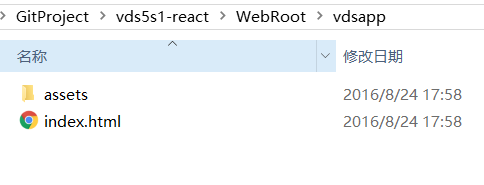
cmd控制台执行npm run build：



执行成功后将在WebRoot生成vdsapp文件夹：



js,css及其它资源文件保存在dsapp/assets文件夹



# 访问项目

浏览器访问<http://localhost:8080/vds5s1/vdsapp/>

