**在Windows下安装和配置ReactJS**

1. **安装Node.js和npm**

* 从 <https://nodejs.org/> 下载Node安装包
* 运行安装包文件，注意勾选全部组件，包括Add to Path
* 安装完成之后，输入node –v及npm –v确认安装成功

1. **创建项目**

* 输入mkdir *hello*创建项目，注意不能与项目所依赖的程序包重名，并且应该小写
* 输入cd *hello*进入项目根目录
* 输入mkdir app创建app子目录
* 输入mkdir app/components在app子目录下创建components子目录
* 新建app/index.html文件，添加内容如下：

<!DOCTYPE html>

<html>

<head>

<meta charset= "UTF-8">

<title>...</title>

</head>

<body>

<div id= "app"></div>

</body>

</html>

* 新建app/index.js文件，添加内容如下：

var React = require('react');

var ReactDOM = require('react-dom');

var App = require('./components/App');

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

* 新建app/components/App.js文件，添加内容如下：

var React = require('react');

var App = () => <div>*...*</div>;

module.exports = App;

* 输入npm init初始化项目，按照提示创建package.json
* 编辑package.json文件scripts属性如下：

"scripts": {

"build": "webpack",

"start": "webpack-dev-server"

},

1. **安装react/babel/webpack程序包**

* 输入npm install –-save react安装react程序包
* 输入npm install --save react-dom安装reactDOM程序包
* 上述指令可缩写成npm i –S {react,react-dom}
* 输入npm install --save-dev babel-core babel-loader babel-preset-react安装babelCore、babelLoader和babelPresetReact程序包至开发环境
* 上述指令可缩写成npm i –D babel-{core,loader,preset-react}
* 新建.babelrc文件，添加内容如下：

{ presets: ['react'] }

* 输入npm install --save-dev webpack webpack-dev-server html-webpack-plugin安装webpack、webpackDevServer和htmlWebpackPlugin程序包至开发环境
* 上述指令可缩写成npm i –D {webpack,webpack-dev-server,html-webpack-plugin}
* 新建webpack.config.js文件，添加内容如下：

var HTMLWebpackPlugin = require('html-webpack-plugin');

var HTMLWebpackPluginConfig = new HTMLWebpackPlugin({

template: \_\_dirname + '/*app*/index.html',

filename: 'index.html',

inject: 'body'

});

module.exports = {

entry: \_\_dirname + '/app/index.js',

module: {

loaders: [{

test: /\.js$/,

exclude: /node\_modules/,

loader: 'babel-loader'

}]

},

output: {

filename: 'transformed.js',

path: \_\_dirname + '/build'

},

plugins: [HTMLWebpackPluginConfig]

};

1. **调试**

* 输入npm run build创建项目子目录build
* 确认build子目录下包含build/index.html和build/transformed.js文件
* 输入npm run start启动服务器
* 在浏览器地址栏输入 <http://localhost:8080> 查看效果
* 更新app/components/App.js组件内容后刷新浏览器查看效果