## the three dots{...this.props}

an spread attribute, It clones our props

eg:(NavLink.js)

// modules/NavLink.js

import React from 'react'

import { Link } from 'react-router'

export default React.createClass({

render() {

return <Link {...this.props} activeClassName="active"/>

}

})

In this example above,we custom a NavLink.js.Every NavLink component transfer all their props to the Inside Link component,and every Link component inside keep a activeClassName property.

## Shared Router

|  |  |
| --- | --- |
| Index.js | Server.js |
| Routes.js | |
| Components | |

## Server-side Webpack

1. webpack.server.config.js

**var *fs*** = *require*(**'fs'**);  
**var *path*** = *require*(**'path'**);  
  
***module***.exports = {  
 **entry**: ***path***.resolve(\_\_dirname, **'server.js'**),  
 **output**: {  
 **filename**: **'server.bundle.js'** },  
 **target**: **'node'**,  
 **externals**: ***fs***.readdirSync(***path***.resolve(\_\_dirname, **'node\_modules'**))  
 .concat([**'react-dom/server'**, **'react/addons'**])  
 .reduce(**function** (ext, mod) {  
 ext[mod] = **'commonjs '** + mod;  
 **return** ext;  
 }, {}),  
 **node**: {  
 **\_\_filename**: **true**,  
 **\_\_dirname**: **true** },  
 **module**: {  
 **loaders**: [  
 {  
 **test**: /\.js$/,  
 **exclude**: /node\_modules/,  
 **loader**: **'babel-loader?presets[]=es2015&presets[]=react'** }  
 ]  
 }  
}

1. 注意事项
   1. path.resolve(\_\_dirname):转换相对路径为绝对路径；
   2. ext[mod]='commonjs '+mod,注意在commonjs后面有一个空格；
   3. \_\_filename:true,才能使\_\_filename有值；
   4. require('fs')中的fs模块貌似没有安装？

## Server-side Render