React+Webpack 实现组件的按需加载

利用Webpack实现工程的按需加载，这里使用的是import()，这也是比较推荐的一种方式。(注import方法需要webpack2及以上版本)

使用Webpack的import()实现按需加载，首先需要安装如下插件

C:\Users\E570\AppData\Local\Packages\Microsoft.Office.OneNote_8wekyb3d8bbwe\TempState\msohtmlclip\clip_image001.png

然后在webpack的配置文件配置：



以下是实现动态加载的类AsyncLoader,使用时传入需动态加载的组件路径，例：<**AsyncLoader path=**{path}/>

|  |
| --- |
| **import React from 'react'**;  **export default class** AsyncLoader **extends** React.*Component*{  **static** *propTypes*={  **path**:React.**PropTypes**.**string**.**isRequired**,  **loading**:React.**PropTypes**.**element**,  }  **static** *defaultProps*={  **path**:**''**,  **loading**:<**p**>Loading...</**p**>,  **error**:<**p**>Error</**p**>  };  constructor(props){  **super**(props);  **this**.**state**={  **module**:**null**  };  }  componentWillMount(){  **this**.load(**this**.**props**);  }  componentWillReceiveProps(nextProps){  **if**(nextProps.**path**!==**this**.**props**.**path**  ||nextProps.**error**!==**this**.**props**.**error**  ||nextProps.**loading**!==**this**.**props**.**loading**){  **this**.load(nextProps);  }  }  load(props){  **this**.setState({**module**:props.**loading**});  **import**(**`src/routes/**${props.**path**}**/index.jsx`**).then(  (m)=>{  **let**Module=m.**default**?m.**default**:m;  **this**.setState({**module**:<**Module**/>});  }  ).catch(()=>{  **this**.setState({**module**:props.**error**});  })  }  render(){  **return this**.**state**.**module**;  }  } |

其核心为load方法的。

在import()中，将模块的名字作为参数并返回一个Promoise对象，

即import(name) ->Promise

注：需要注意的是import()函数里面的参数不能是一个完全变量，使用完全变量时会报错。我们需要让Webpack知晓应该预先加载那些内容，import里面的参数必须是一个常量或是使用模板字符串，如上例所示。

参阅文档：

<https://www.cnblogs.com/skylor/p/7008756.html>