# 一、 环境准备：

1、安装redux

2、安装react-redux

# 二、<Provider> 组件、

用于向组件传递state数据

使用方法：

1. 在项目的index.js中引入Provider：

import {Provider} from 'react-redux'

1. 引入store:

import store from './store'

3、定义render函数并定复制给变量：

const App=(

<Provider store={store}> //绑定store

<TodoList></TodoList> //包裹组件

</Provider>

)

Provider组件绑定store,并将组件包裹起来，这一步可以使TodoList组件和下面的子组件都能通过props访问到store的state数据

4、将App render内容与reactDom 绑定:

ReactDOM.render(App, document.getElementById('root'));

# 三、mapStateToProps(state)、

这个方法用于将Store的state数据通过props传递到组件，组件可以通过this.props访问到store的数据：

const mapStateToProps=(state)=>{ //参数state

return {

inputValue:state.inputValue,

list:state.list

}

}

在组建中定义以上方法后、在方法中，参数state表示store传递过来的state数据,return返回的数据可以在组件里面访问：

Store reducer存储的值：

const deafultState={ //初始化state

inputValue:'123213', //这里就是要存储的数据

list:[],

}

组件中访问：

<input type="text" value={this.props.inputValue}/>

页面展示：

C:\Users\ADMINI~1\AppData\Local\Temp\1557284357(1).png

# 四、mapDispatchToProps(dispatch)

这个方法用于将action发送到store，reducer将数据修改后由于已经和页面绑定数据，所以现在reducer中修改数据之后、页面的数据就会随之改变、使用步骤：

1. 定义mapDispatchToProps（）方法：

const mapDispatchToProps = (dispatch) => {

return {

}

}

1. 由于这个方法是向store传递action和数据，所以需要在组件的事件中绑定数据：

<input value={this.props.inputValue} onChange={this.props.handelInputChange} />

// handelInputChange是当inut输入内容的时候执行的函数

<button onClick={this.props.handelClick}>提交</button>

// handelClick是当点击提交按钮的时候执行的函数

mapDispatchToProps()方法是将store.dispath()方法传递到props,所以可以将事件函数绑定到mapDispatchToProps()里面的方法中：

const mapDispatchToProps = (dispatch) => {

return {

handelInputChange(e) {

const action = {type: "change\_input",value: e.target.value //将输入的数据传到store

}

dispatch(action)

},

handelClick() {

const action = {type: 'add\_item'}

dispatch(action)

}

}

}

export default connect(mapStateToProps, mapDispatchToProps)(TodoList)

在向store 发起dispatch()后,store通过reducer 对dispatch()的action进行监听，如果reducer中监听的action和组建中的disptch的action相同，则对数据进行修改

**mapStateToProps(state)和mapDispatchToProps()方法都是从store向组件传递数据，前者是传递普通属性，mapDispatchToProps()是传递函数属性**

# 五、connect()方法