# 高阶组件

## 6.2以函数为子组件



import React, { Component } from 'react'

const AddProps=(DemoComponent,title)=>{

return class extends Component{

render() {

let newProps={newTitle:title};

return <DemoComponent {...this.props} {...newProps}/>;

}

}

}

export default AddProps;

import React, { Component } from 'react'

import PropTypes from 'prop-types'

class AddUserProp extends Component{

render() {

return this.props.children(this.props.user);

}

}

AddUserProp.propTypes={

children:PropTypes.func.isRequired

};

export default AddUserProp;

<h1>AddProps1\_1</h1>

<AddProps1 user="AddProps1\_1">

{(user)=><div>{user}</div>}

</AddProps1>

<hr />

<h1>AddProps1\_2</h1>

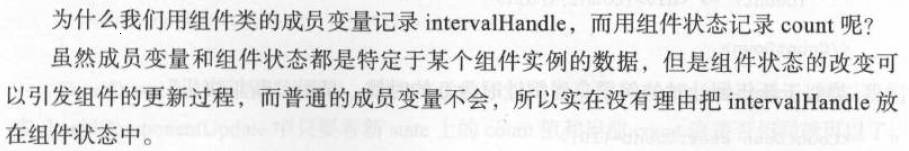
<AddProps1 user="AddProps1\_2">

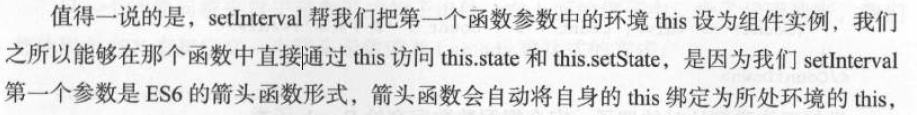
{

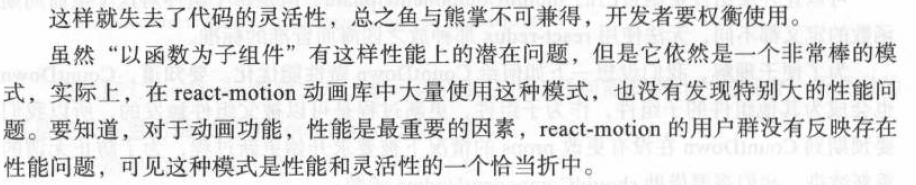
(user)=><RemovePropsDemo {...RemovePropsDemoProps} title={user}></RemovePropsDemo>

}

</AddProps1>

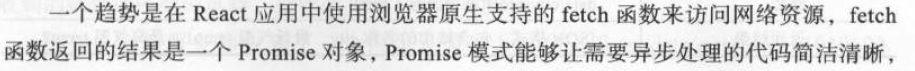






## Redux和服务器通信

### Fetch





import React, { Component } from 'react'

export default class WeatherCom extends Component {

constructor(){

super(...arguments);

this.state={weather:null};

}

componentDidMount = () => {

const apiUrl=`/data/cityinfo/${this.props.cityCode}.html`;

fetch(apiUrl).then((response)=>{

if(response.status!==200){

throw new Error('Fail to get response with status'+response.status);

}

response.json().then((responseJson)=>{

this.setState({weather:responseJson.weatherinfo});

}).catch((error)=>{

this.setState({weather:null});

});

}).catch((error)=>{

this.setState({weather:null});

});

}

render() {

if(!this.state.weather){

return <div>暂无数据</div>;

}

const {city,weather,temp1,temp2}=this.state.weather;

return (

<div>

{city} {weather} 最低气温 {temp1} 最高气温 {temp2}

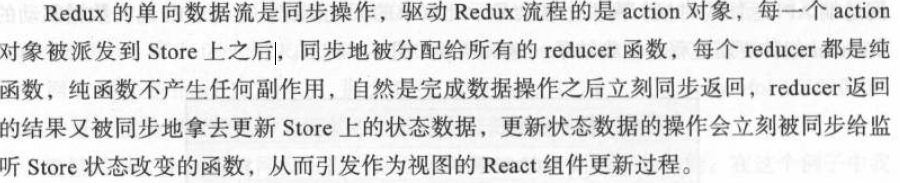
</div>

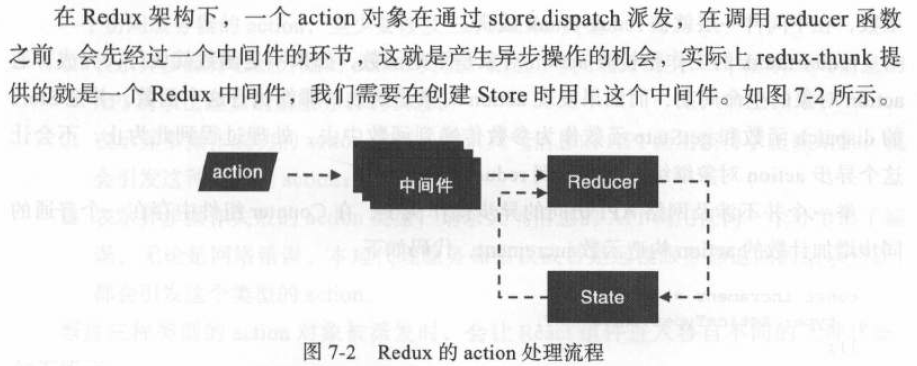
)

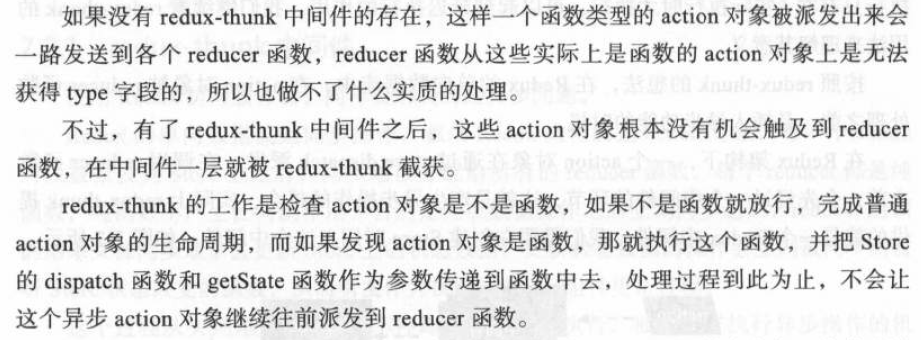
}

}

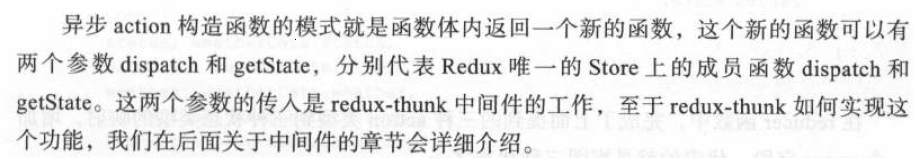
### Redux-thunk 中间件





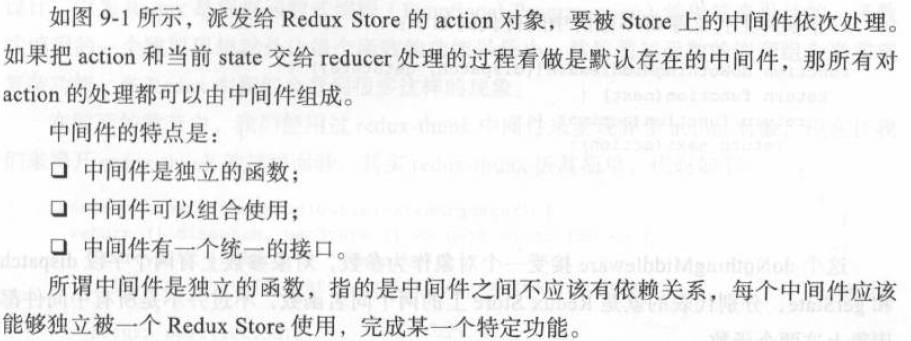


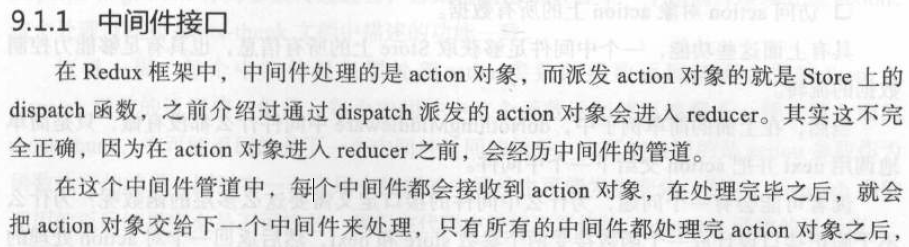
### 异步action构造函数模式

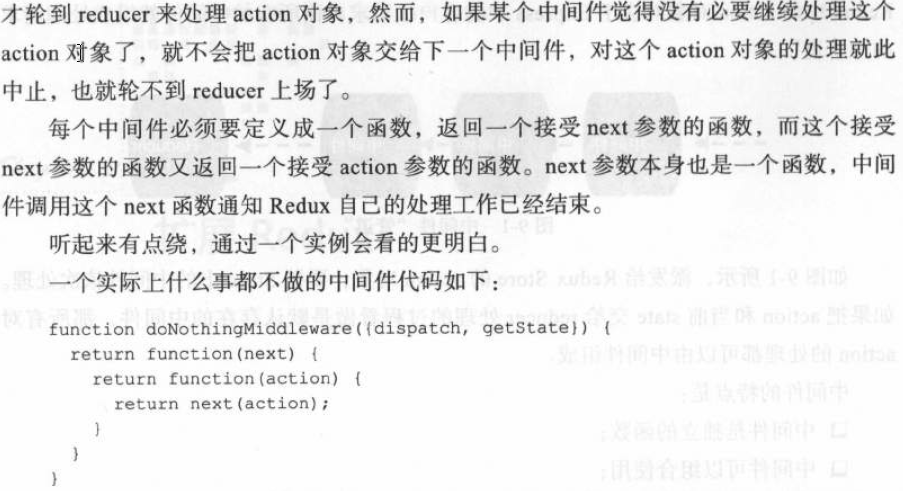


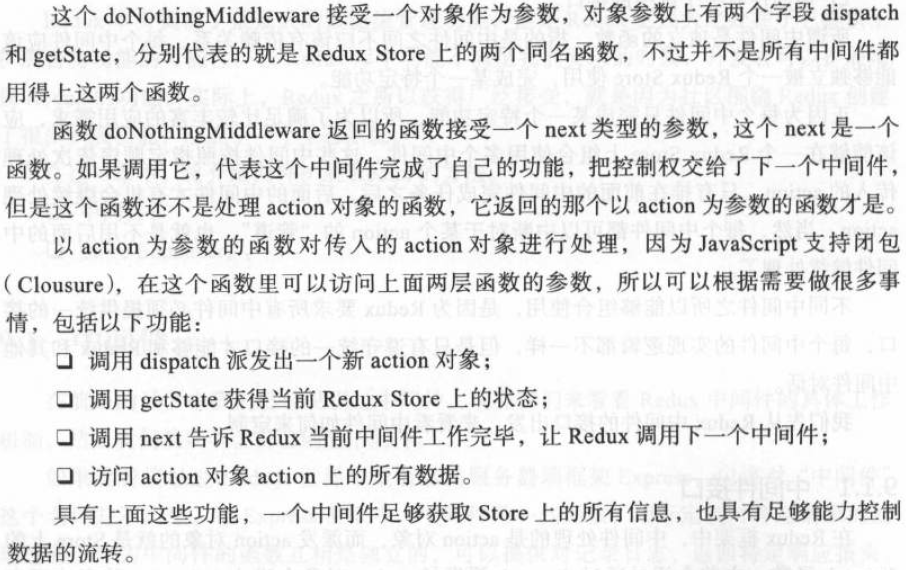
## 扩展redux

### 中间件

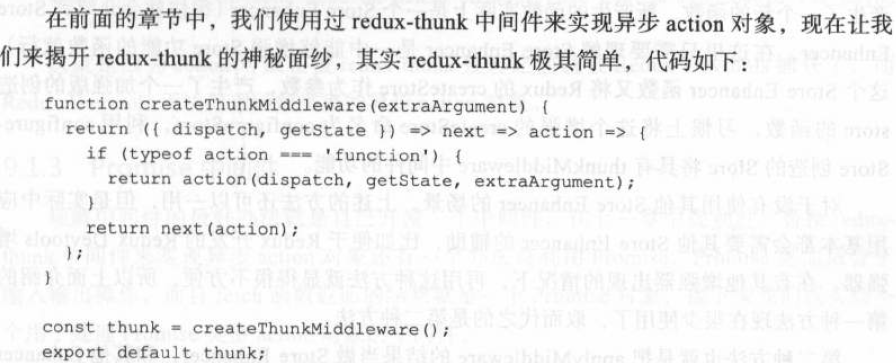






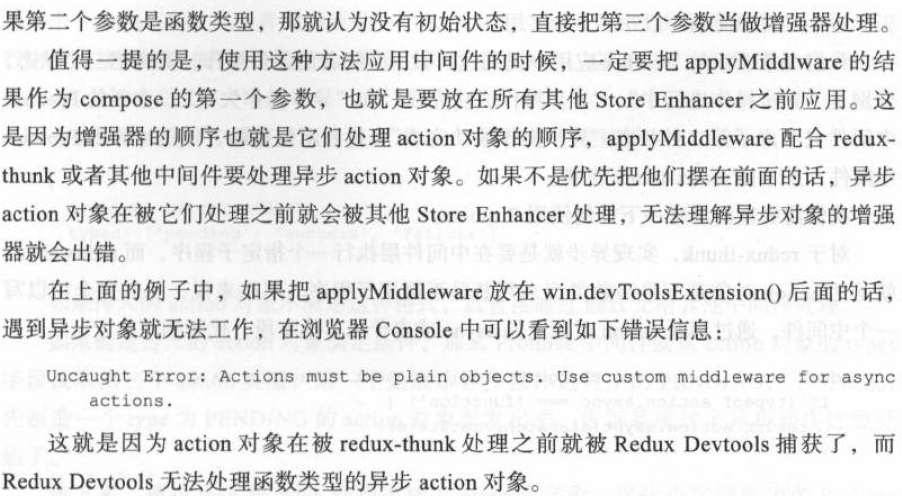


### Redux-thunk 中间件实现



### 使用中间件



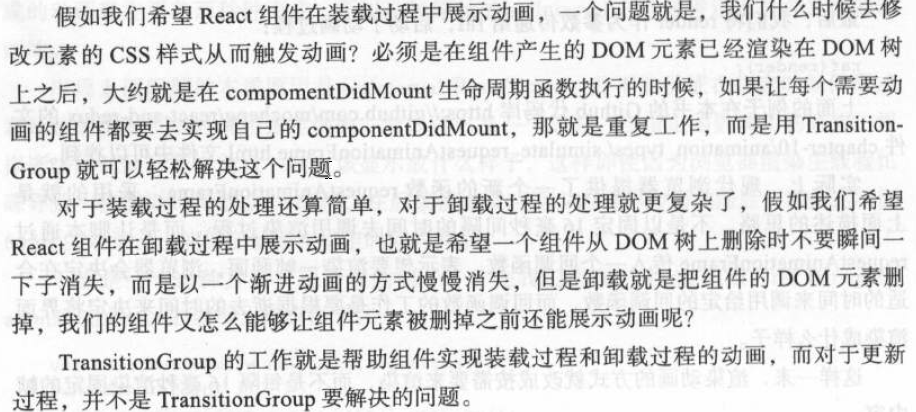


### Promise中间件

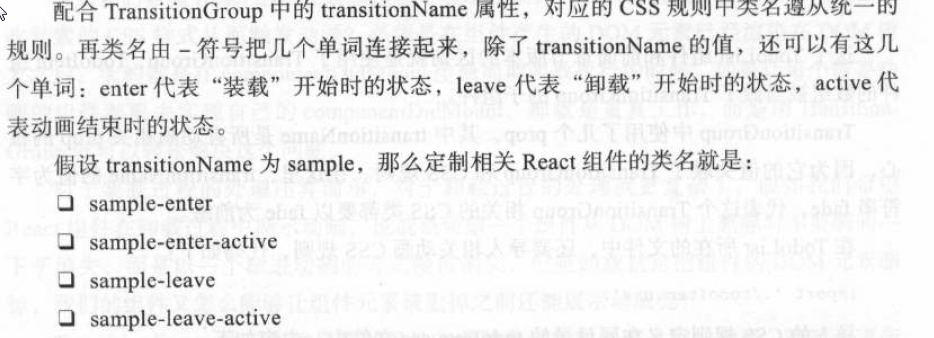


## 动画

### ReactCSSTransitionGroup规则



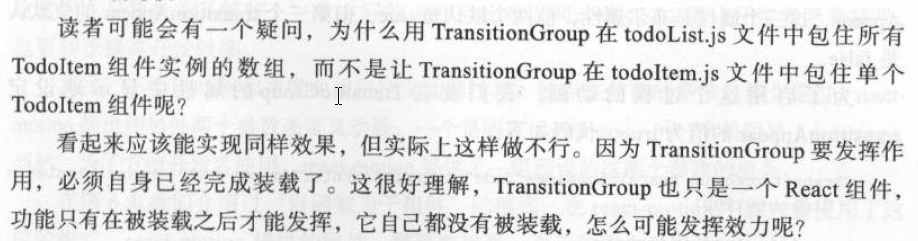
#### 1.类名规则



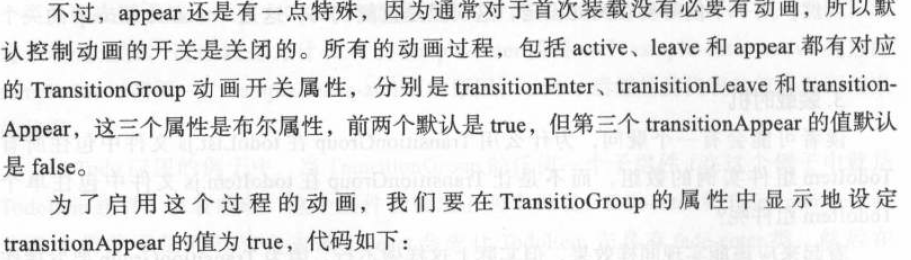
#### 2.动画时间长度

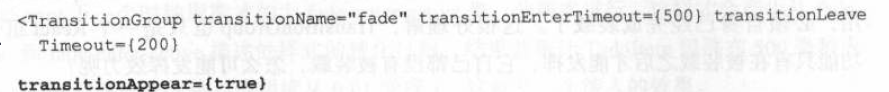


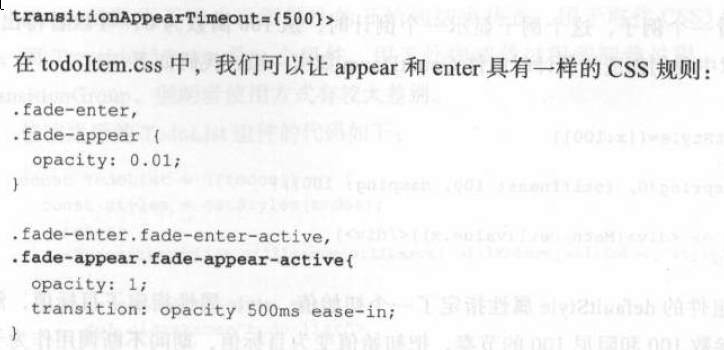
#### 3.装载时机



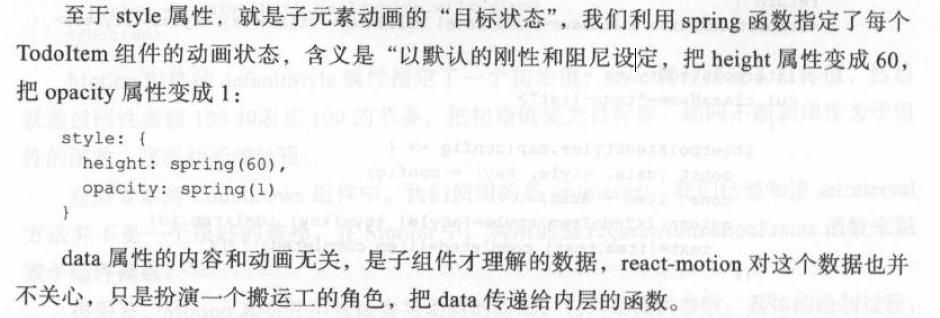
#### 4.首次装载

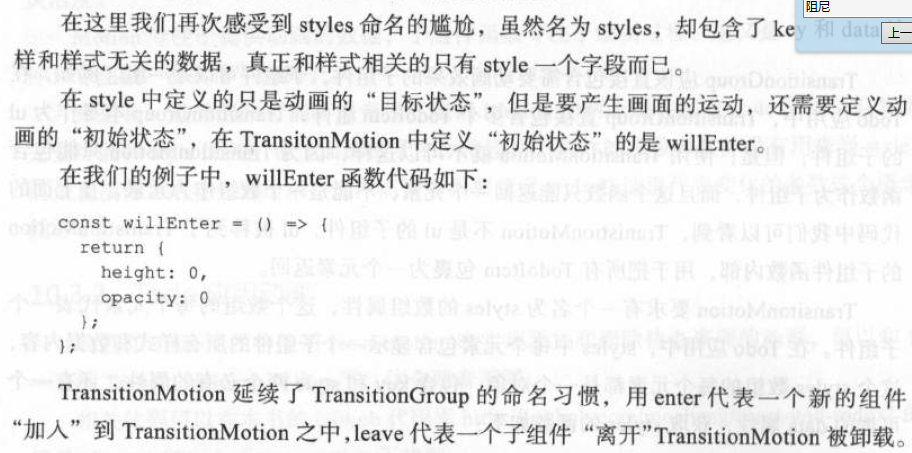


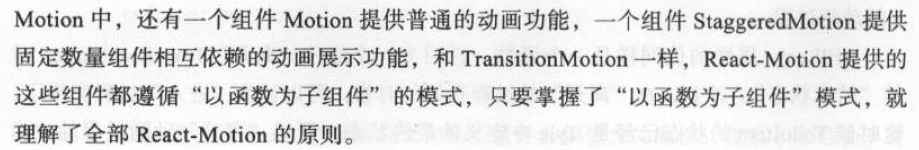




### React-motion的设计原则





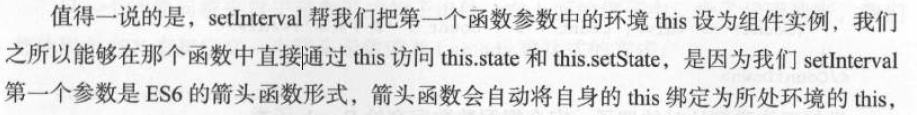


## React同构



# 知识点：

## 1、箭头函数



# 有用的网站

### 中国天气网

<http://www.weather.com.cn/>

## 安装的软件

npm install --save redux-thunk