


主页 (/) / Blogs (/category/3/blogs) / React/React Native 的ES5 ES6写法对照表  (/topic/15.rss)

React/React Native 的ES5 ES6写法对照表



(/user/tdzl2003) **tdzl2003** (/user/tdzl2003) 大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/1) 

很多React/React Native的初学者都被ES6的问题迷惑：各路大神都建议我们直接学习ES6的语法(`class Foo extends React.Component`), 然而网上搜到的很多教程都是ES5版本的, 所以很多人在学习的时候连照猫画虎都不知道怎么做。今天在此整理了一些ES5和ES6的写法对照表, 希望大家以后读到ES5的代码, 也能通过对照, 在ES6里实现相同的功能。

模块

引用

在ES5里, 如果使用CommonJS标准, 引入React包基本通过require进行, 代码类似这样:

```
//ES5
var React = require("react");
var {
  Component,
  PropTypes
} = React; //引用React抽象组件

var ReactNative = require("react-native");
var {
  Image,
  Text,
} = ReactNative; //引用具体的React Native组件
```

在ES6里, import写法更为标准

```
//ES6
import React, {
  Component,
  PropTypes,
} from 'react';
import {
  Image,
  Text
} from 'react-native'
```

导出单个类

在ES5里, 要导出一个类给别的模块用, 一般通过module.exports来导出

```
//ES5
var MyComponent = React.createClass({
  ...
});
module.exports = MyComponent;
```

在ES6里, 通常用export default来实现相同的功能:

```
//ES6
export default class MyComponent extends Component{
  ...
}
```

引用的时候也类似:

```
//ES5
var MyComponent = require('./MyComponent');

//ES6
import MyComponent from './MyComponent';
```

注意导入和导出的写法必须配套, 不能混用!

定义组件

在ES5里，通常通过React.createClass来定义一个组件类，像这样：

```
//ES5
var Photo = React.createClass({
  render: function() {
    return (
      <Image source={this.props.source} />
    );
  },
});
```

在ES6里，我们通过定义一个继承自React.Component的class来定义一个组件类，像这样：

```
//ES6
class Photo extends React.Component {
  render() {
    return (
      <Image source={this.props.source} />
    );
  }
}
```

给组件定义方法

从上面的例子里可以看到，给组件定义方法不再用 名字: function() 的写法，而是直接用 名字()，在方法的最后也不能有逗号了。

```
//ES5
var Photo = React.createClass({
  componentWillMount: function(){

  },
  render: function() {
    return (
      <Image source={this.props.source} />
    );
  },
});

//ES6
class Photo extends React.Component {
  componentWillMount() {

  }
  render() {
    return (
      <Image source={this.props.source} />
    );
  }
}
```

定义组件的属性类型和默认属性

在ES5里，属性类型和默认属性分别通过propTypes成员和getDefaultProps方法来实现

```
var Video = React.createClass({
  getDefaultProps: function() {
    return {
      autoPlay: false,
      maxLoops: 10,
    };
  },
  propTypes: {
    autoPlay: React.PropTypes.bool.isRequired,
    maxLoops: React.PropTypes.number.isRequired,
    posterFrameSrc: React.PropTypes.string.isRequired,
    videoSrc: React.PropTypes.string.isRequired,
  },
  render: function() {
    return (
      <View />
    );
  },
});
```

在ES6里，可以统一使用static成员来实现

```
//ES6
class Video extends React.Component {
  static defaultProps = {
    autoPlay: false,
    maxLoops: 10,
  }; // 注意这里有分号
  static propTypes = {
    autoPlay: React.PropTypes.bool.isRequired,
    maxLoops: React.PropTypes.number.isRequired,
    posterFrameSrc: React.PropTypes.string.isRequired,
    videoSrc: React.PropTypes.string.isRequired,
  }; // 注意这里有分号
  render() {
    return (
      <View />
    );
  } // 注意这里既没有分号也没有逗号
}
```

也有人这么写，虽然不推荐，但读到代码的时候你应当能明白它的意思：

```
//ES6
class Video extends React.Component {
  render() {
    return (
      <View />
    );
  }
}
Video.defaultProps = {
  autoPlay: false,
  maxLoops: 10,
};
Video.propTypes = {
  autoPlay: React.PropTypes.bool.isRequired,
  maxLoops: React.PropTypes.number.isRequired,
  posterFrameSrc: React.PropTypes.string.isRequired,
  videoSrc: React.PropTypes.string.isRequired,
};
```

注意: 对React开发者而言，static成员在IE10及之前版本不能被继承，而在IE11和其它浏览器上可以，这有时候会带来一些问题。React Native开发者可以不用担心这个问题。

初始化STATE

ES5下情况类似，

```
//ES5
var Video = React.createClass({
  getInitialState: function() {
    return {
      loopsRemaining: this.props.maxLoops,
    };
  },
});
```

ES6下，有两种写法：

```
//ES6
class Video extends React.Component {
  state = {
    loopsRemaining: this.props.maxLoops,
  }
}
```

不过我们推荐更易理解的在构造函数中初始化（这样你还可以根据需要做一些计算）：

```
//ES6
class Video extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      loopsRemaining: this.props.maxLoops,
    };
  }
}
```

把方法作为回调提供

很多习惯于ES6的用户反而不理解在ES5下可以这么做：

```
//ES5
var PostInfo = React.createClass({
  handleOptionsButtonClick: function(e) {
    // Here, 'this' refers to the component instance.
    this.setState({showOptionsModal: true});
  },
  render: function(){
    return (
      <TouchableHighlight onPress={this.handleOptionsButtonClick}>
        <Text>{this.props.label}</Text>
      </TouchableHighlight>
    )
  },
});
```

在ES5下，React.createClass会把所有的方法都bind一遍，这样可以提交到任意的地方作为回调函数，而this不会变化。但官方现在逐步认为这反而是不标准、不易理解的。在ES6下，你需要通过bind来绑定this引用，或者使用箭头函数（它会绑定当前scope的this引用）来调用。

```
//ES6
class PostInfo extends React.Component
{
  handleOptionsButtonClick(e){
    this.setState({showOptionsModal: true});
  }
  render(){
    return (
      <TouchableHighlight
        onPress={this.handleOptionsButtonClick.bind(this)}
        onPress={e=>this.handleOptionsButtonClick(e)}
      >
        <Text>{this.props.label}</Text>
      </TouchableHighlight>
    )
  },
}
```

箭头函数实际上是在这里定义了一个临时的函数，箭头函数的箭头 => 之前是一个空括号、单个的参数名、或用括号括起的多个参数名，而箭头之后可以是一个表达式（它的返回值），或者用花括号括起的函数体（需要自行通过return来返回值，否则返回的是undefined）。

```
// 箭头函数的例子
()=>1
v=>v+1
(a,b)>=>a+b
()=>{
  alert("foo");
}
e=>{
  if (e == 0){
    return 0;
  }
  return 1000/e;
}
```

需要注意的是，不论是bind还是箭头函数，每次被执行都返回的是一个新的函数引用，因此如果你还需要函数的引用去做一些别的事情（譬如卸载监听器），那么你必须保存这个引用。

```
// 错误的做法
class PauseMenu extends React.Component{
  componentWillMount(){
    AppStateIOS.addEventListener('change', this.onAppPaused.bind(this));
  }
  componentWillUnmount(){
    AppStateIOS.removeEventListener('change', this.onAppPaused.bind(this));
  }
  onAppPaused(event){
  }
}
```

```
// 正确的做法
class PauseMenu extends React.Component{
  constructor(props){
    super(props);
    this._onAppPaused = this.onAppPaused.bind(this);
  }
  componentWillMount(){
    AppStateIOS.addEventListener('change', this._onAppPaused);
  }
  componentWillUnmount(){
    AppStateIOS.removeEventListener('change', this._onAppPaused);
  }
  onAppPaused(event){
  }
}
```

从这个帖子 (<http://www.tuicool.com/articles/Rj6RFnm>)中我们还学习到一种新的做法:

```
// 正确的做法
class PauseMenu extends React.Component{
  componentWillMount(){
    AppStateIOS.addEventListener('change', this.onAppPaused);
  }
  componentWillUnmount(){
    AppStateIOS.removeEventListener('change', this.onAppPaused);
  }
  onAppPaused = (event) => {
    //把方法直接作为一个arrow function的属性来定义，初始化的时候就绑定好了this指针
  }
}
```

Mixins

在ES5下，我们经常使用mixin来为我们的类添加一些新的方法，譬如PureRenderMixin

```
var PureRenderMixin = require('react-addons-pure-render-mixin');
React.createClass({
  mixins: [PureRenderMixin],

  render: function() {
    return <div className={this.props.className}>foo</div>;
  }
});
```

然而现在官方已经不再打算在ES6里继续推行Mixin，他们说：Mixins Are Dead. Long Live Composition (https://medium.com/@dan_abramov/mixins-are-dead-long-live-h-order-components-94a0d2f9e750)。

尽管如果要继续使用mixin，还是有一些第三方的方案可以用，譬如这个方案 (<https://github.com/brigand/react-mixin>)

不过官方推荐，对于库编写者而言，应当尽快放弃Mixin的编写方式，上文中提到Sebastian Markbåge (<https://gist.github.com/sebmarkbage/ef0bf1f338a7182b6775>)的码推荐了一种新的编码方式:

```
//Enhance.js
import { Component } from "React";

export var Enhance = ComposedComponent => class extends Component {
  constructor() {
    this.state = { data: null };
  }
  componentDidMount() {
    this.setState({ data: 'Hello' });
  }
  render() {
    return <ComposedComponent {...this.props} data={this.state.data} />;
  }
};
```

```
//HigherOrderComponent.js
import { Enhance } from './Enhance';

class MyComponent {
  render() {
    if (!this.data) return <div>Waiting...</div>;
    return <div>{this.data}</div>;
  }
}

export default Enhance(MyComponent); // Enhanced component
```

用一个“增强函数”，来某个类增加一些方法，并且返回一个新类，这无疑能实现mixin所实现的大部分需求。

ES6+带来的其它好处

解构&属性延展

结合使用ES6+的解构和属性延展，我们给孩子传递一批属性更为方便了。这个例子把className以外的所有属性传递给div标签：

```
class AutoloadingPostsGrid extends React.Component {
  render() {
    var {
      className,
      ...others, // contains all properties of this.props except for className
    } = this.props;
    return (
      <div className={className}>
        <PostsGrid {...others} />
        <button onClick={this.handleLoadMoreClick}>Load more</button>
      </div>
    );
  }
}
```

下面这种写法，则是传递所有属性的同时，用覆盖新的className值：

```
<div {...this.props} className="override">
  ...
</div>
```

这个例子则相反，如果属性中没有包含className，则提供默认的值，而如果属性中已经包含了，则使用属性中的值

```
<div className="base" {...this.props}>
  ...
</div>
```

回复 引用 ^ 0 v

⋮

29 198.6k

帖子 浏览

回复 ▾

关注 👁

标记为未读 📖

排

L

laughing (/user/laughing)


大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/2)

(/user/laughing)

写的很不错，正考虑用es6语法写react

回复 引用 ^ 0 v

⋮



(/user/sunnylqm)sunnylqm (/user/sunnylqm)

ADMINISTRATORS (/groups/administrators)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/3)

@laughing

@laughing (http://bbs.reactnative.cn/user/laughing)

不用考虑了，从0.18开始，rn的新项目默认模板已经全面转向ES6了

回复 引用 ^ 2 v

⋮

L

Lesdesel (/user/lesdesel)


大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/4)

(/user/lesdesel)

平贤。写了6,7个js文件，好像有的是ES5，请问有没有官方的关于这个的对照表，会不会更完整一些。

回复 引用 ^ 0 v

⋮



(/user/sunnylqm)

●

sunnylqm (/user/sunnylqm)

ADMINISTRATORS (/groups/administrators)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/5)

🔗

@Lesdesel

@Lesdesel (<http://bbs.reactnative.cn/user/lesdesel>) 没有比这个帖子更完整的

回复 引用

⤴ ⤵

⋮

A

●

a497823463 (/user/a497823463)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/6)

(/user/a497823463)

在Sublime 上有想应得插件吗？现在的插件都是ES5的，好别扭。

回复 引用

⤴ ⤵

⋮

C

●

chinesels (/user/chinesels)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/7)

(/user/chinesels)

贵写的通俗易懂

回复 引用

⤴ ⤵

⋮

J

●

jflame (/user/jflame)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/8)

(/user/jflame)

静态属性和属性延展是 ES7 提案。
虽然通过Babel 都可以用，但最好能明确这一点。
虽然不一定大，毕竟也是有一定风险，而且最好不要进一步加深新人对 js 语法的混淆

回复 引用

⤴ ⤵

⋮

S

●

sherlock221b (/user/sherlock221b)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/9)

(/user/sherlock221b)

很棒！结合 <http://reactjs.cn/react/docs/reusable-components.html> (<http://reactjs.cn/react/docs/reusable-components.html>) 这篇文档 基本react 基本用法掌握清楚了👍

个

回复 引用

⤴ ⤵

⋮

A

●

adispring (/user/adispring)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/10)

(/user/adispring)

这里有一篇外国入写的React on ES6: <https://babeljs.io/blog/2015/06/07/react-on-es6-plus> (<https://babeljs.io/blog/2015/06/07/react-on-es6-plus>)

回复 引用

⤴ ⤵

⋮

H

●

huaganshang (/user/huaganshang)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/11)


(/user/huaganshang)

@tdzl2003 (<http://bbs.reactnative.cn/user/tdzl2003>) mark，感谢；

回复 引用

⤴ ⤵

⋮



(/user/hainuo)

●

hainuo (/user/hainuo)

大约1年之前 (/topic/15/react-react-native-的es5-es6写法对照表/12)

这个帖子太有帮助了 很清楚

海诺博客 (<https://blog.hainuo.info>)

回复 引用

⤴ ⤵

⋮

W

●

wesley (/user/wesley)

12月之前 (/topic/15/react-react-native-的es5-es6写法对照表/13)

(/user/wesley)

.+1: .+1: .+1: .+1: .+1: .+1: .+1:

回复 引用

⤴ ⤵

⋮

L

●

luax (/user/luax)

11月之前 (/topic/15/react-react-native-的es5-es6写法对照表/14)

(/user/luax)

写得很好,楼主加油.

回复 引用

⤴ ⤵

⋮

<http://bbs.reactnative.cn/topic/15/react-react-native-的es5-es6写法对照表>

第 7 页 (共 10 页)

Y

yukaaaaang (/user/yukaaaaang)

10月之前 (/topic/15/react-react-native-的es5-es6写法对照表/15)

(/user/yukaaaaang)

不错，对于新手来说很有意义，建议标识精华帖。赞

回复

引用

^0v

⋮

P

phpsmarter (/user/phpsmarter)

10月之前 (/topic/15/react-react-native-的es5-es6写法对照表/16)

(/user/phpsmarter)


我总补充一下，前面看到楼主这个帖子，刚好又遇到es5的编译不通过，以为两者不能混用。这几天才发现es5写的组件也可用通过import 导入到es6写的js文件中。旧的组件不用重新写成es6的语法。

回复

引用

^0v

⋮



(/user/sunnylqm)

sunnylqm (/user/sunnylqm)

ADMINISTRATORS (/groups/administrators)

10月之前 (/topic/15/react-react-native-的es5-es6写法对照表/17)

@phpsmarter

@phpsmarter (http://bbs.reactnative.cn/user/phpsmarter)

只要正确export default即可import

最主要的区别是class的声明方式及其带来的附加影响

回复

引用

^0v

⋮

C

cssjidi (/user/cssjidi)

10月之前 (/topic/15/react-react-native-的es5-es6写法对照表/18)

(/user/cssjidi)

正好需要，非常好的文章

回复

引用

^0v

⋮

S

sandheart (/user/sandheart)

9月之前 (/topic/15/react-react-native-的es5-es6写法对照表/19)

(/user/sandheart)

在React里ES5和ES6的混合写法是大坑，这个帖子对新手很实用

回复

引用

^0v

⋮

J

Jay-Y (/user/jay-y)

9月之前 (/topic/15/react-react-native-的es5-es6写法对照表/20)

(/user/jay-y)

请问这些对照在官方react native文档有说明吗，如果有，能否分享下！谢谢！

回复

引用

^0v

⋮



(/user/sunnylqm)

sunnylqm (/user/sunnylqm)

ADMINISTRATORS (/groups/administrators)

9月之前 (/topic/15/react-react-native-的es5-es6写法对照表/21)

@Jay-Y

@Jay-Y (http://bbs.reactnative.cn/user/jay-y)

没有

回复

引用

^0v

⋮

K

kaenry (/user/kaenry)

9月之前 (/topic/15/react-react-native-的es5-es6写法对照表/22)

(/user/kaenry)

mark先，以后对照

回复

引用

^0v

⋮

Y

ypzhou (/user/ypzhou)

8月之前 (/topic/15/react-react-native-的es5-es6写法对照表/23)

(/user/ypzhou)

```
// 正确的做法
class PauseMenu extends React.Component{
  componentWillMount(){
    AppStateIOS.addEventListener('change', this.onAppPaused);
  }
  componentDidUnmount(){
    AppStateIOS.removeEventListener('change', this.onAppPaused);
  }
  onAppPaused = (event) => {
    //把方法直接作为一个arrow (//xn-arrow-fg1hyjke76i380f7scnwi9z0aom6a) function的属性来定义，初始化的时候就绑定好了this指针
  }
}
```


回复

引用

^0v

⋮

按这个写法报错啊 箭头函数换成es5的就可以， 不知道怎么解决。现在都是在constructor里从新绑定一次

 (/user/sunnylqm)sunnylqm (/user/sunnylqm) ADMINISTRATORS (/groups/administrators) 8月之前 (/topic/15/react-react-native-的es5-es6写法对照表/24) @ypzhou

@ypzhou (http://bbs.reactnative.cn/user/ypzhou) 报什么错

回复 引用 ^ 0 v

⋮

Y (/user/ypzhou) ypzhou (/user/ypzhou) 8月之前 (/topic/15/react-react-native-的es5-es6写法对照表/25) @sunnylqm

@sunnylqm (http://bbs.reactnative.cn/user/sunnylqm)

```
129
130
131 class PauseMenu extends React.Component{
132   componentWillMount(){
133     window.addEventListener('scroll', this.onAppPaused);
134   }
135   componentDidUnmount(){
136     window.removeEventListener('scroll', this.onAppPaused);
137   }
138   onAppPaused = (event) => {
139     //把方法直接作为一个arrow function的属性来定义，初始化的时候就
140   }
141 }
```

cal npm

Hash: ae5f7a9d0425be603cdf
Version: webpack 1.13.1
Time: 327ms
chunk <0> app.js <app> 1.35 MB
 + 318 hidden modules


ERROR in ./components/tag/index.jsx
Module build failed: SyntaxError: E:/react-sf/app/components/tag/index.jsx: Unex
pected token <138:16>
136 | window.removeEventListener('scroll', this.onAppPaused);
137 | }
> 138 | onAppPaused = (event) => {
 | ^
139 | //把方法直接作为一个arrow function的属性来定义，初始化的时候就
 | 定好了this指针

(uploads/files/1468478415315-562da4cf8415dbf5.png)

就是这个错误，在render里使用箭头函数就不会报错

回复 引用 ^ 0 v

⋮

 (/user/sunnylqm)sunnylqm (/user/sunnylqm) ADMINISTRATORS (/groups/administrators) 8月之前 (/topic/15/react-react-native-的es5-es6写法对照表/26) @ypzhou

@ypzhou (http://bbs.reactnative.cn/user/ypzhou) web上默认的babel配置不支持es7 class properties

回复 引用 ^ 0 v

⋮

Y (/user/ypzhou) ypzhou (/user/ypzhou) 8月之前 (/topic/15/react-react-native-的es5-es6写法对照表/27)


@sunnylqm (http://bbs.reactnative.cn/user/sunnylqm) 说：

不

那要怎么解决啊？这里的箭头函数用起来很方便的

回复 引用 ^ 0 v


⋮

 (/user/sunnylqm)sunnylqm (/user/sunnylqm) ADMINISTRATORS (/groups/administrators) 8月之前 (/topic/15/react-react-native-的es5-es6写法对照表/28) @ypzhou

@ypzhou (http://bbs.reactnative.cn/user/ypzhou) 自己了解一下babel的配置吧，我三两句话讲不清楚

回复 引用 ^ 0 v

⋮

Y  **ypzhou (/user/ypzhou)** 8日之前 (/topic/15/react-react-native-的es5-es6写法对照表/29)
(/user/ypzhou)
解决了问题? 加入
npm install babel-preset-stage-0 --save-dev //支持ES7
这个配置就可以了

回复 引用 ^ 0 v
⋮

29
帖子

198.6k
浏览

回复 ▾

关注 👁

标记为未读 📧

排