

项目说明

王红元 coderwhy



实力IT教育

■ 网易云音乐项目：

□ 网易云音乐团队：

- ✓ 30多个前端（目前）；
- ✓ 经历过数年版本迭代最终完成的一个产品；

■ 我们不可能在上课期间将所有的内容讲完：

- 1.我们安排的课程时长是6节课（至少会讲6节），我会根据大家的接受程度来把握课程的进度；
- 2.最重要的是学习React开发的模式和编程的思想，而不是局限于我上课期间所讲的内容，并且我不会带着大家写很多的布局或者CSS样式；
- 3.在这个项目过程中，我会尽量将之前所学的所有知识都运用起来，但是我们不会为了用某个知识而用某个知识；
- 4.我会将网易云音乐项目的主要页面做完，所以课下，你有充足的时间按照我的代码，完成更多的页面；

- 项目规范：项目中有一些开发规范和代码风格
- 1.文件夹、文件名称统一小写、多个单词以连接符（ - ）连接；
- 2.JavaScript变量名称采用小驼峰标识，常量全部使用大写字母，组件采用大驼峰；
- 3.CSS采用普通CSS和styled-component结合来编写（全局采用普通CSS、局部采用styled-component）；
- 4.整个项目不再使用class组件，统一使用函数式组件，并且全面拥抱Hooks；
- 5.所有的函数式组件，为了避免不必要的渲染，全部使用memo进行包裹；
- 6.组件内部的状态，使用useState、useReducer；业务数据全部放在redux中管理；
- 7.函数组件内部基本按照如下顺序编写代码：
 - 组件内部state管理；
 - redux的hooks代码；
 - 其他组件hooks代码；
 - 其他逻辑代码；
 - 返回JSX代码；

■ 8. redux代码规范如下：

- 每个模块有自己独立的reducer，通过combineReducer进行合并；
- 异步请求代码使用redux-thunk，并且写在actionCreators中；
- redux直接采用redux hooks方式编写，不再使用connect；

■ 9. 网络请求采用axios

- 对axios进行二次封装；
- 所有的模块请求会放到一个请求文件中单独管理；

■ 10. 项目使用AntDesign

- 项目中某些AntDesign中的组件会被拿过来使用；
- 但是多部分组件还是自己进行编写；

■ 其他规范在项目中根据实际情况决定和编写；

为什么看不到react devtools标记

■ <https://github.com/facebook/react-devtools/issues/191>



Venryx commented on 23 Feb 2018



Okay, found a way to keep even that function from getting injected/involved.

I just changed the disabler code to this:

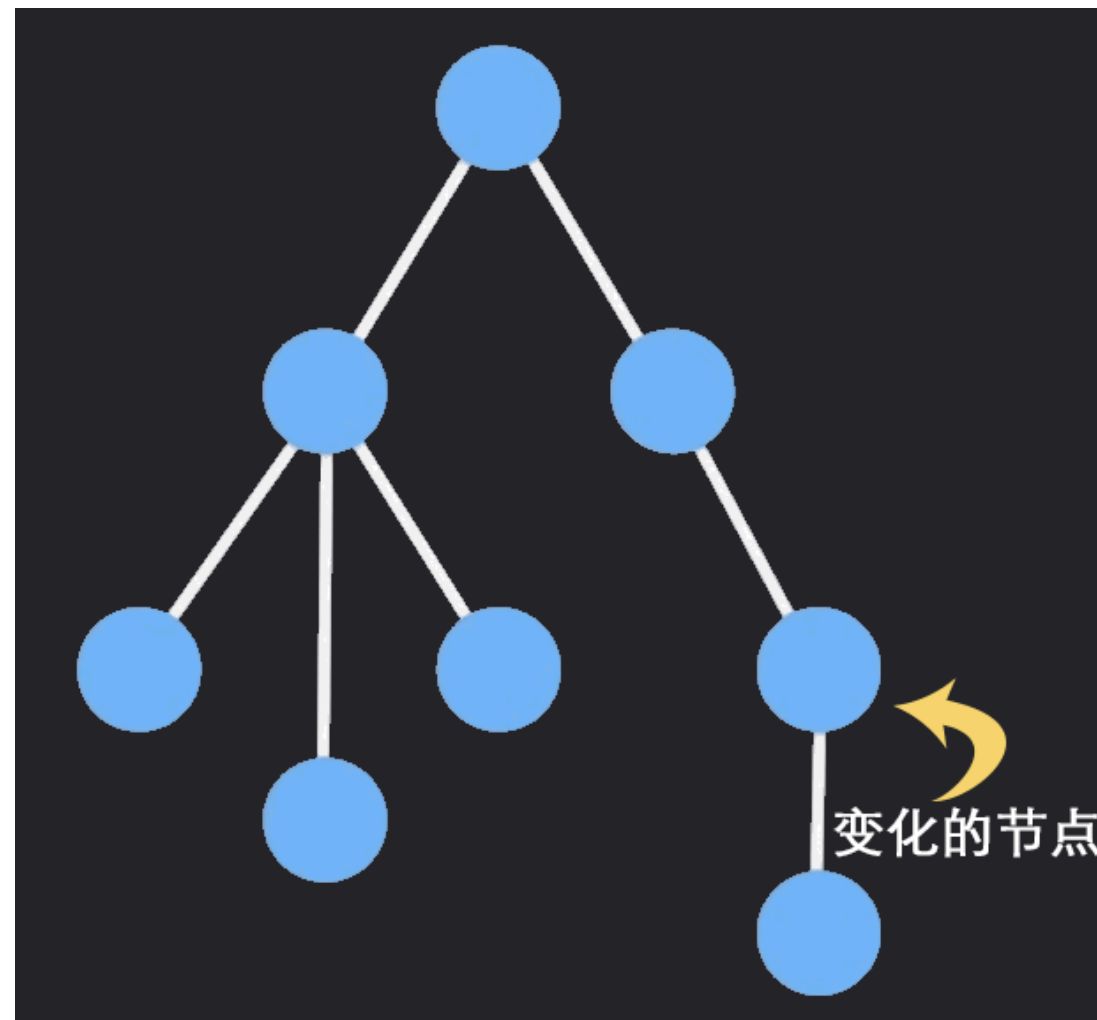
```
// disable react-dev-tools for this project
if (typeof window.__REACT_DEVTOOLS_GLOBAL_HOOK__ === "object") {
  for (let [key, value] of Object.entries(window.__REACT_DEVTOOLS_GLOBAL_HOOK__)) {
    window.__REACT_DEVTOOLS_GLOBAL_HOOK__[key] = typeof value === "function" ? ()=>{} : null;
  }
}
```



数据可变性的问题

- 在React开发中，我们总是会强调数据的不可变性：
 - 无论是类组件中的state，还是redux中管理的state；
 - 事实上在整个JavaScript编码过程中，数据的不可变性都是非常重要的；
- 数据的可变性引发的问题（案例）：
 - 我们明明没有修改obj，只是修改了obj2，但是最终obj也被我们修改掉了；
 - 原因非常简单，对象是引用类型，它们指向同一块内存空间，两个引用都可以任意修改；
- 有没有办法解决上面的问题呢？
 - 进行对象的拷贝即可：Object.assign或扩展运算符
- 这种对象的浅拷贝有没有问题呢？
 - 从代码的角度来说，没有问题，也解决了我们实际开发中一些潜在风险；
 - 从性能的角度来说，有问题，如果对象过于庞大，这种拷贝的方式会带来性能问题以及内存浪费；
- 有人会说，开发中不都是这样做的吗？
 - 从来如此，便是对的吗？

- 为了解决上面的问题，出现了Immutable对象的概念：
 - Immutable对象的特点是只要修改了对象，就会返回一个新的对象，旧的对象不会发生改变；
- 但是这样的方式就不会浪费内存了吗？
 - 为了节约内存，又出现了一个新的算法：Persistent Data Structure（持久化数据结构或一致性数据结构）；
- 当然，我们一听到持久化第一反应应该是数据被保存到本地或者数据库，但是这里并不是这个含义：
 - 用一种数据结构来保存数据；
 - 当数据被修改时，会返回一个对象，但是新的对象会尽可能的利用之前的数据结构而不会对内存造成浪费；
- 如何做到这一点呢？结构共享。





ImmutableJS常见API

- 注意：我这里只是演示了一些API，更多的方式可以参考官网；
- JavaScript和ImmutableJS直接的转换
 - 对象转换成Immutable对象：Map；
 - 数组转换成Immutable数组：List；
 - 深层转换：fromJS；
 - Immutable类型转成：toJS()；
- ImmutableJS的基本操作：
 - 修改数据：set
 - 获取数据：get
- 结合Redux管理数据：
 - 1.使用redux-immutable中的combineReducers；
 - 2.所有的reducer中的数据都转换成Immutable类型的数据；