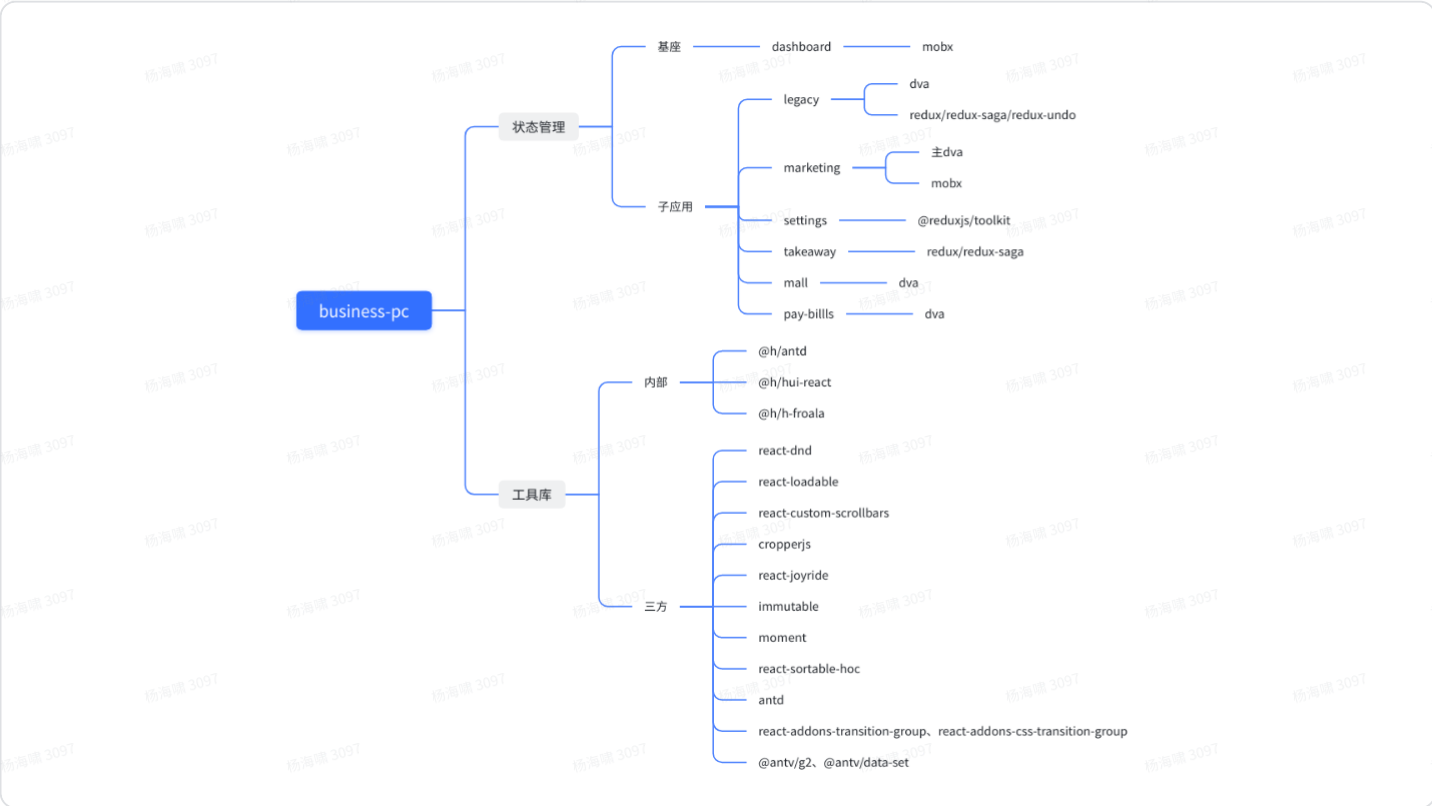


CRM 升级 React 18 方案

CRM 技术栈



升级难点

React 版本带来问题

- v17
 - 去除事件池

在React17之前，如果使用异步的方式来获取事件e对象，会发现合成事件对象被销毁，如下：

```
1 function App() {
2   const handleClick = (e: React.MouseEvent) => {
3     console.log('直接打印e', e.target) // <button>React事件池</button>
4     // v17以下在异步方法拿不到事件e，必须先调用 e.persist()
5     // e.persist()
6
7     // 异步方式获取事件e
8     setTimeout(() => {
9       console.log('setTimeout打印e', e.target) // null
10    })
11  }
```

```

11   }
12   return (
13     <div className="App">
14       <button onClick={handleClick}>React事件池</button>
15     </div>
16   )
17 }
18

```

直接打印e <button>React事件池</button>

App.tsx:3

Warning: This synthetic event is reused for performance reasons. If you're seeing this, you're accessing the property 'target' on a released/nullified synthetic event. This is set to null. If you must keep the original synthetic event around, use event.persist(). See <https://fb.me/react-event-pooling> for more information.

setTimeout打印e null

@稀土掘金技术社区 App.tsx:8

如果你需要在事件处理函数运行之后获取事件对象的属性，你需要调用 `e.persist()`，它会将当前的合成事件从事件池中删除，并允许保留对事件的引用。

• v18

• `setState` 同步/异步，是否对页面有影响

- 这是 React 此次版本中最大的破坏性更新，并且**无法向下兼容**。
- React 中的批处理简单来说就是将多个状态更新合并为一次重新渲染。在 18 之前，`Promise`、`setTimeout` 以及原生事件中是不会对其进行批处理的

<https://github.com/reactwg/react-18/discussions/21>

• 替换 `unsafe` 的类组件生命周期

• 关于 React 组件的返回值

- 对于 React 17，如果组件在 `render` 中返回了 `undefined`，React 会在运行时抛出一个错误。需要改为返回 `null`
- 对于 React 18 的 Suspense fallback 会出现 `undefined` 而不报错从而导致出现不一致。

路由方面问题

- mobx-react-router、react-redux-router、dva 里面的 router 方法要统一到 react-router-dom
- 在 react-dom 的v6以上版本中，一些操作的 dom 方法被弃用，如废弃findDOMNode、withRouter方法

现有子应用间跳转问题？

三方库

- 状态管理库
 - dva 不再维护。不支持 17 以上。需要迁移重构

@reduxjs/toolkit	<p>写法比较类似</p> <ul style="list-style-type: none"> Dva effects 对应 asyncThunk，相当于异步处理的表现 yield 改成 async put 简单使用可以直接对应 reducers 错误处理可以对应 extraReducers 中处理 state 对象基本可以直接复用
mobx	<ul style="list-style-type: none"> Dva effects 对应 mobx actions 错误处理也在 actions 里进行处理 state 要手动拆分绑定
redux/redux-saga	<ul style="list-style-type: none"> 更适用类组件，使用 dva 的子应用基本都是函数式组件开发了。不如 @reduxjs/toolkit 迁移方便
recoil	<ul style="list-style-type: none"> 写法差异太大，需要学习成本
react-query	<ul style="list-style-type: none"> 写法差异太大，工作量应该和迁移 mobx 差不多

重构的工作量？需要重构的子应用 marketing、mall、legacy、pay-bills

- react-redux 版本统一
 - 已知 react-redux 8.0 以上版本基于 React 18 新 API `useSyncExternalStore` 重新实现了。其他版本与 react 18 兼容性暂时未知
- 拖拽库
 - react-sortable-hoc 已不再维护，
 - 见官方 [readme](#)，在 React 的未来版本中，由于 `findDOMNode` 方法将被弃用。开发工作都转到 **@dnd-kit**，官方推荐 react-sortable-hoc 迁移到 **@dnd-kit**
 - react-dnd 与 react 18 不兼容，见官方 Issue，目前不知道修复了没
 - [Incompatibility with REACT 18 · Issue #3593 · react-dnd/react-dnd](#)

这里先升级 React，升级后如果有问题考虑迁移到 **@dnd-kit** (没问题就不迁移)，不过工作量可能稍大。

- 动画库
 - react-addons-transition-group/react-addons-css-transition-group 16 年 github 最后更新。兼容性未知

package.json

- 优化点。一些没用的依赖，没有移除。
 - Legacy

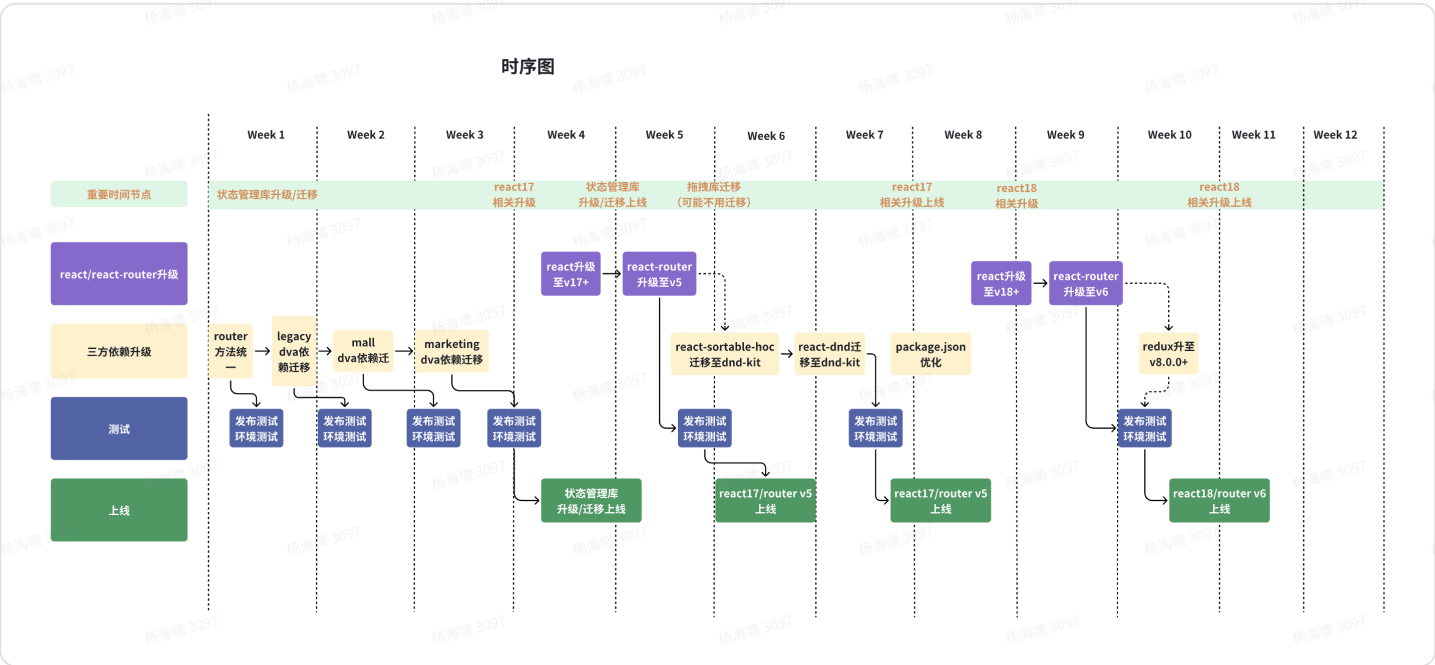
- raven-js
- isomorphic-fetch
- react-color
- Mall
 - reselect
- Marketing
 - reselect
 - react-redux
- Settings
 - reselect
- engines 统一更新一下 node 版本方便开发

升级方案

当前V16.14.0。dll注入导致所有子应用 React 版本锁死？

- 先升级后创建的子应用
- 渐进式升级。先升至17，react-router-dom v5。再升级到18，react-router-dom v6升级

时间线/上线计划



任务具体拆分

- CRM 状态库升级/迁移
 - router方法统一至react-router

- react-redux升至8.x版本
- dva技术栈迁移/魔改dva
 - legacy
 - marketing
 - mall
 - pay-bills
- 子应用 React 17 升级
 - 替换unsafe、deprecated生命周期
 - 去除事件池
 - 组件返回值
 - setState合并更新影响
- 子应用 react-router v5 升级
 - legacy
 - pay-bills
 - takeaway
- 子应用 React 18 升级
 - mall
 - marketing
 - settings
 - legacy
 - pay-bills
 - takeaway
 - dashboard
- 子应用 react-router v6 升级
 - mall
 - marketing
 - settings
 - legacy
 - pay-bills
 - takeaway
 - dashboard

- package.json 优化