发现问题所在：就是大力的那个接口有问题。

中间件是什么：一个代理，别人帮着做事情，

Express 是一个自身功能极简，完全是由路由和中间件构成一个的 web 开发框架：从本质上来说，一个 Express 应用就是在调用各种中间件。

中间件的配置：

config.local.js:

middleware: ['setTraceid', 'auth', 'userInfo', 'errorHandler']

async/await

Node.js 7.6已经支持async/await了,现在社区上的版本是6.11.3

Promise.all()把多个promise对象包装成一个新的promise实例，这样写的好处是在用then方法的时候可以统一用一个变量来表示几个promise实例。

异步方法：用promise写法和async写法的区别是：

1. async函数会隐式地返回一个promise 对象，该promise的reosolve值就是函数return的值
2. await关键字只能用在async定义的函数内
3. 我们不需要写.then，
4. 不需要写匿名函数处理Promise的resolve值，
5. 也不需要定义多余的data变量，
6. 还避免了嵌套代码。
7. 可以同时处理同步和异步错误,之前的写法同步和异步要分开处理。
8. 需要获取数据，然后根据返回数据决定是直接返回，还是继续获取更多的数据。

使用Promise 有多层嵌套，如果使用async await则能够像写同步代码一样写异步代码，非常简洁和易读。

1. 调试非常方便，在promise的then中没有办法设置断点，但是使用await 可以轻易地设置断点。

为什么使用：async/await https://blog.fundebug.com/2017/04/04/nodejs-async-await/

It allows you to change the behavior of, or add behavior to methods and functions (including constructors)

before(this, 'selectAncestorBtn', () => {

trackEvent({

e\_c: uaActionCategory.ANCESTOR,

e\_a: uaActionName.CLICK,

e\_n: '检测结果',

});

});

后面的函数在中间函数的前面执行

只有在组件的render方法被调用时，ref才会被调用，组件才会返回ref。如果你在调用this.refs.xx时render方法还没被调用，那么你得到的是undefined。

Legend：传说

Piechart：饼形图

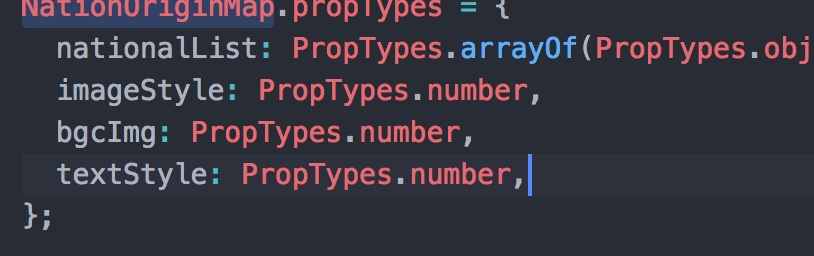
RN 的文字阴影：

Require的图片是数字？后台怎么知道？

文字阴影？

TODO:运算符优先级？

逻辑与逻辑或在比较运算符和赋值运算符之间？

样式为什么是number类型

中间件是干嘛的？什么时候会执行？