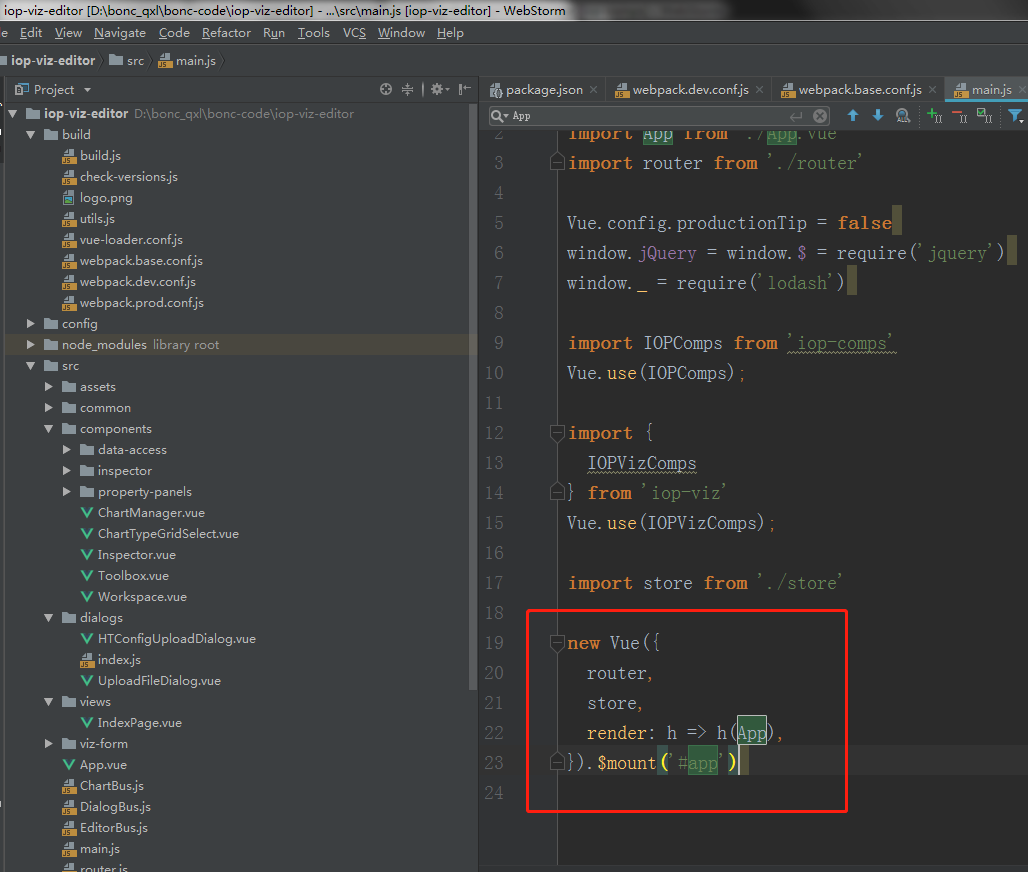
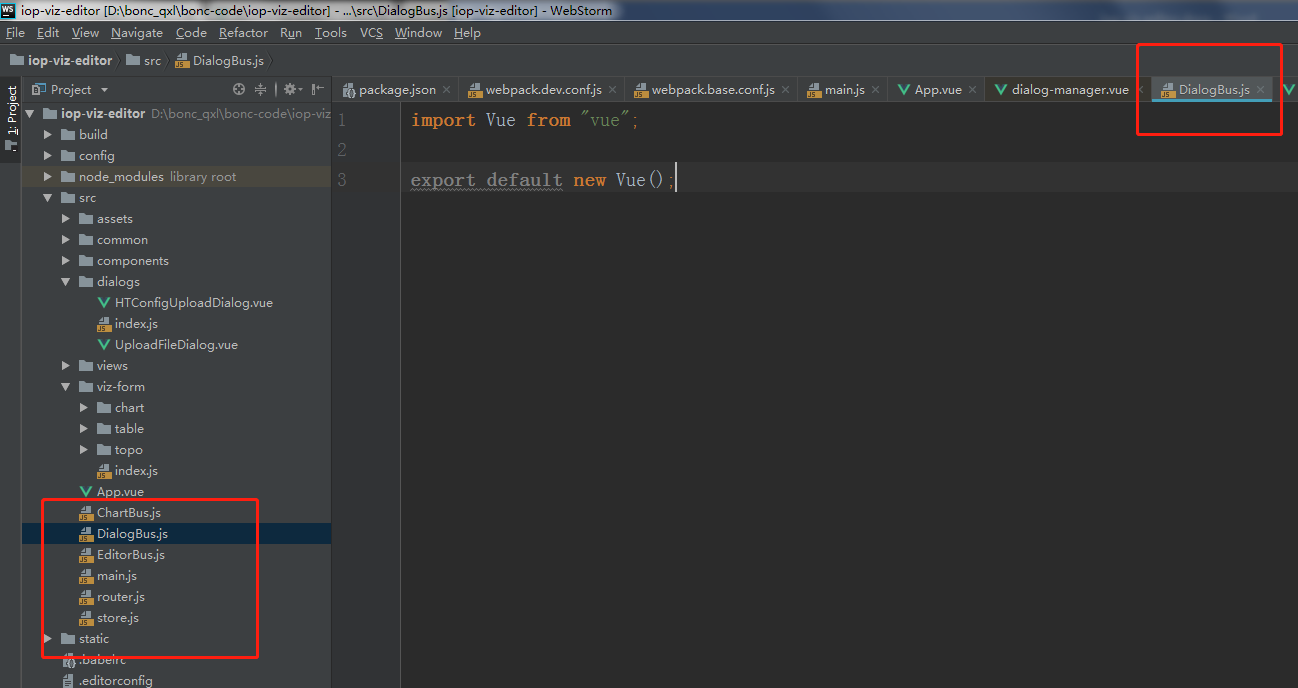
**Iop-viz-editor**

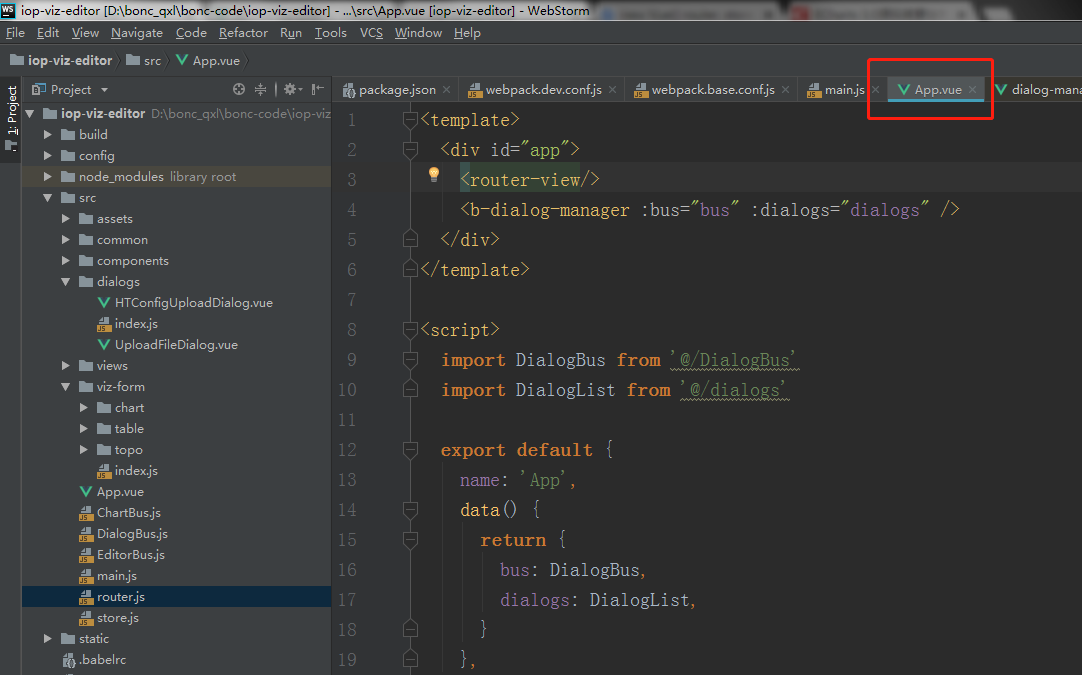
1. 不懂



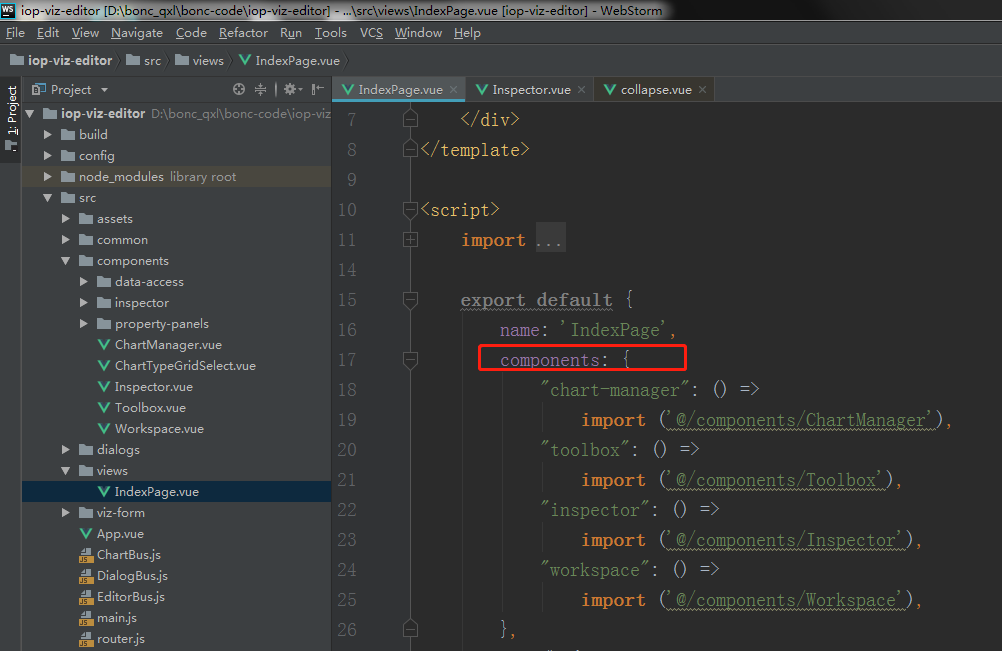
2．各种bus都是这样new 一个vue对象，不知道啥意思



1. 到这就走不下去了，应该怎么找？

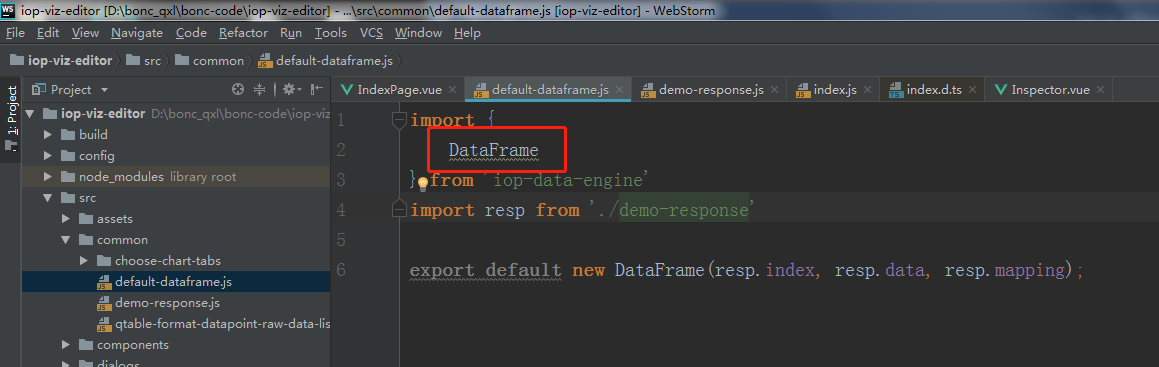


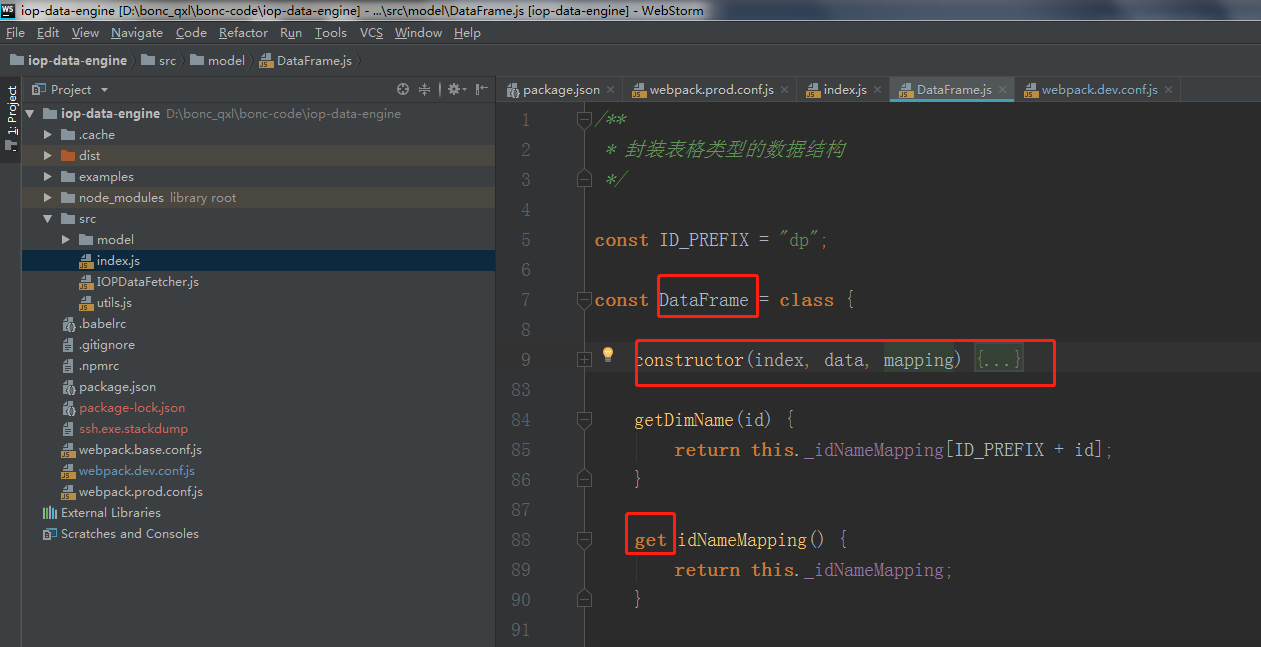
1. components还可以这样写哦：



5.三种类型的图有2种（table和chart）都是从iop-viz里来的，都用了IOPChart

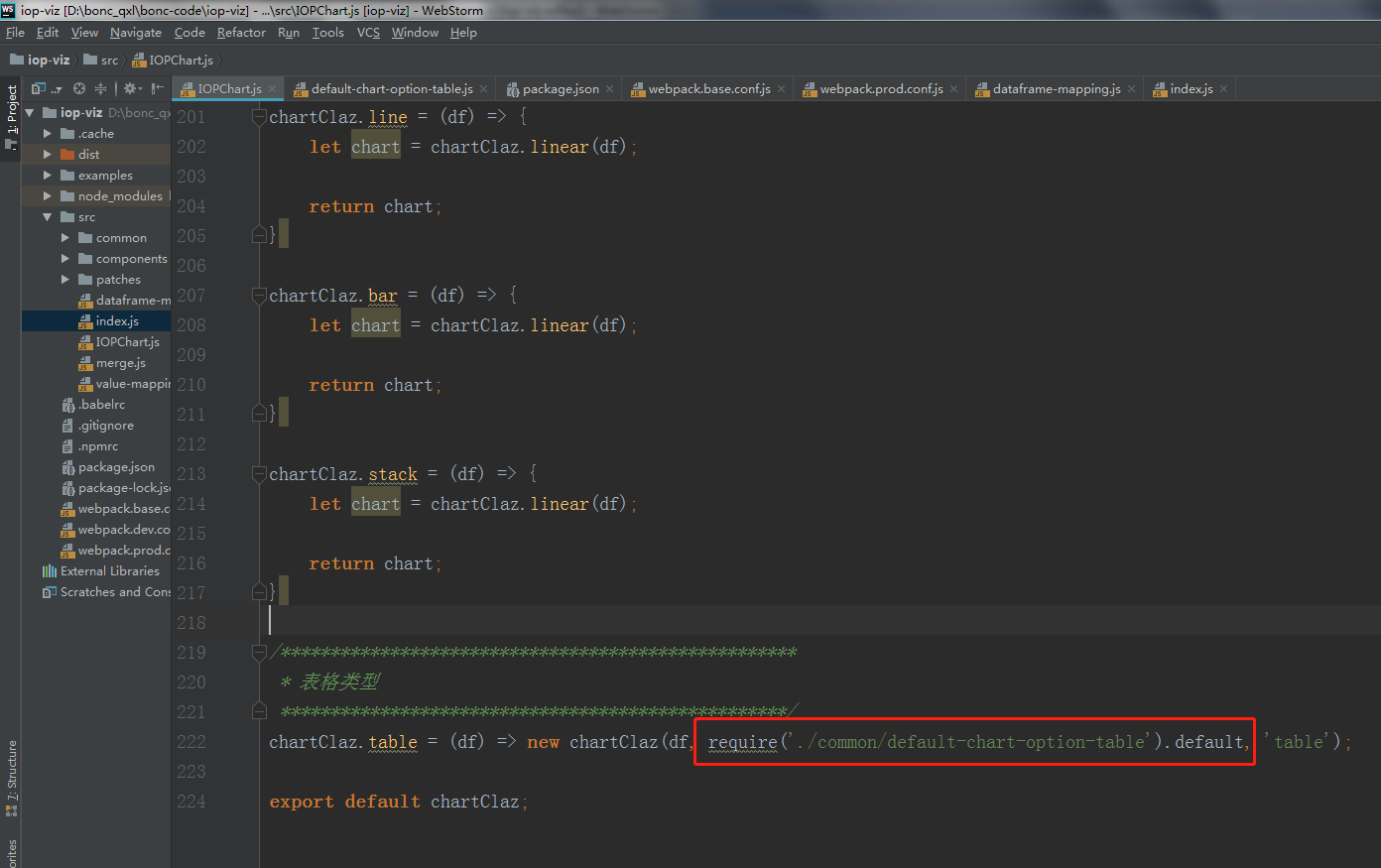
6.关于class,不太懂里面的get等关键字





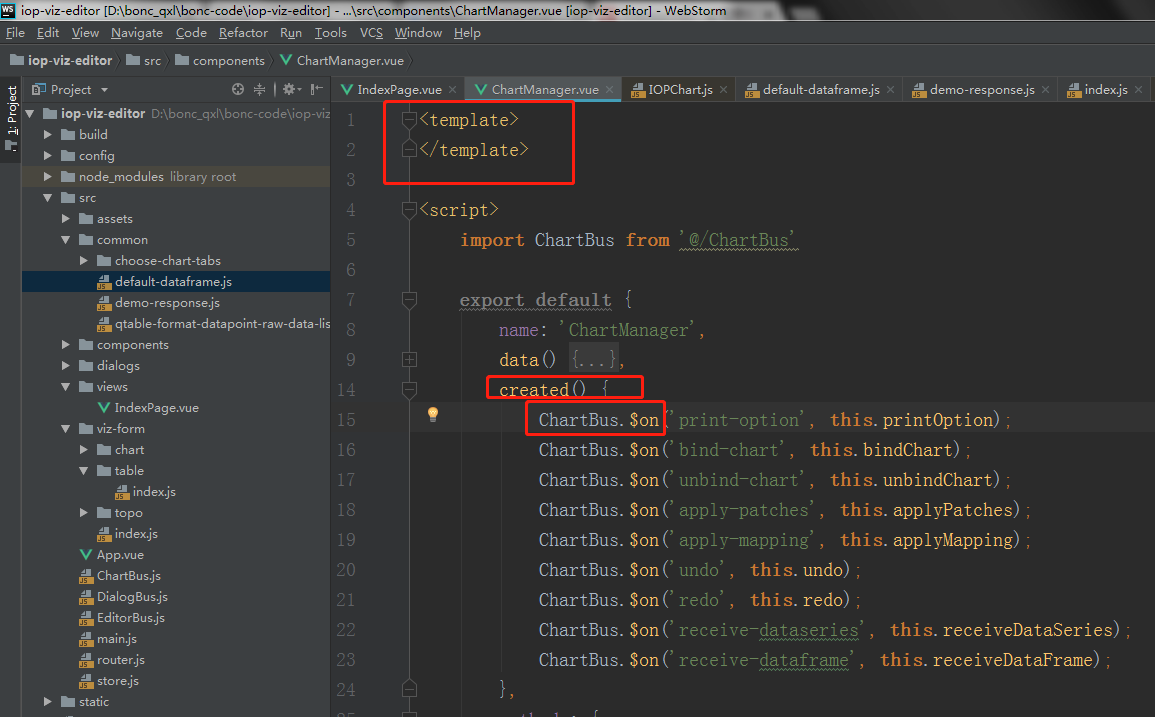
7.iop-data-engine没有细看里面的实现，还有iop-viz里的那个IOPChart的对象

8.这种require.default的写法要注意哟，看看和import的区别是啥

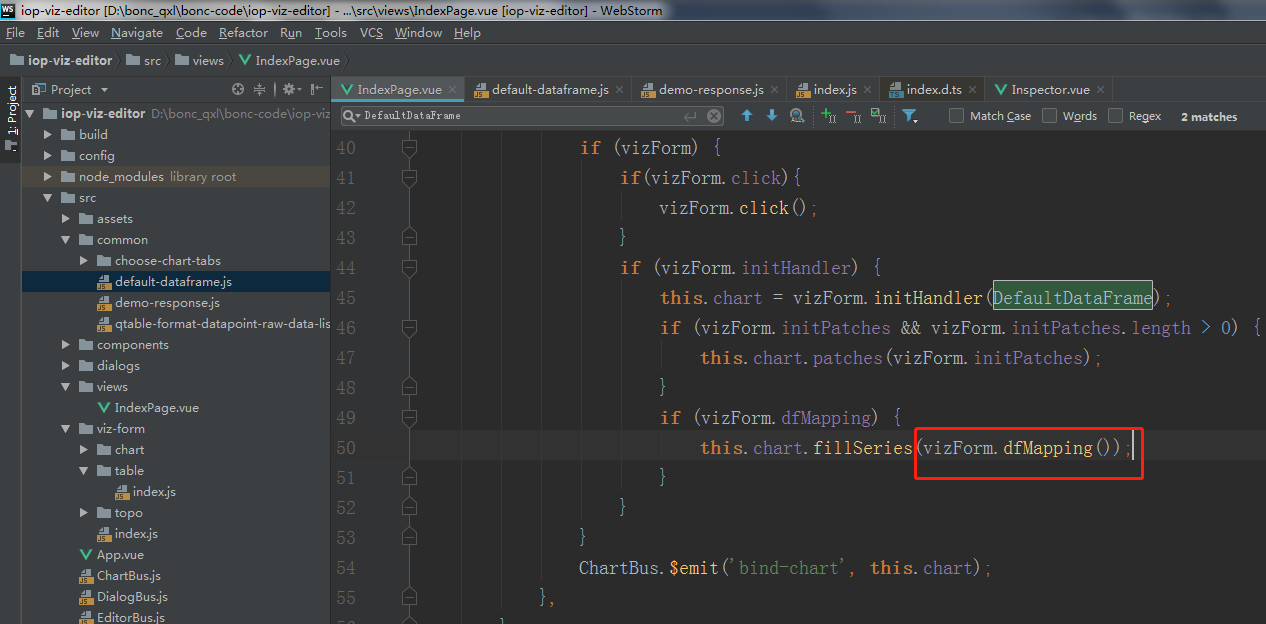


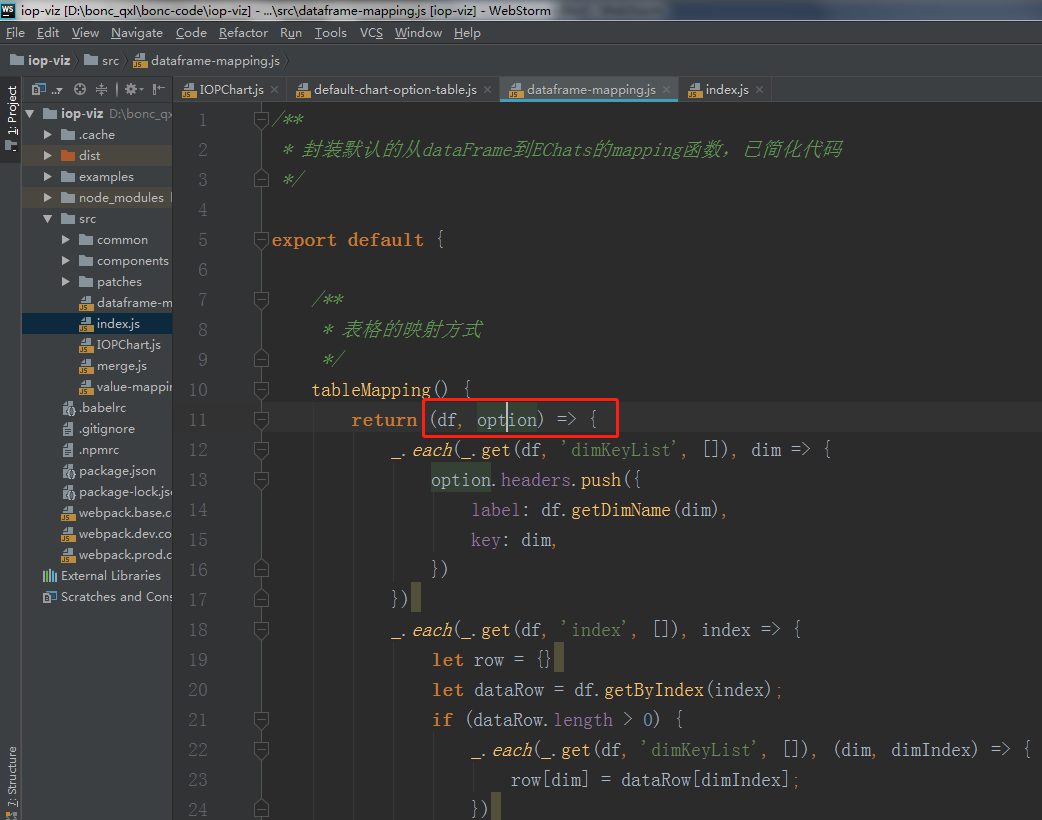
9.为啥有时候用vue,有时候用js,有时候用class?

这里用的原因是为了在created里监听on事件

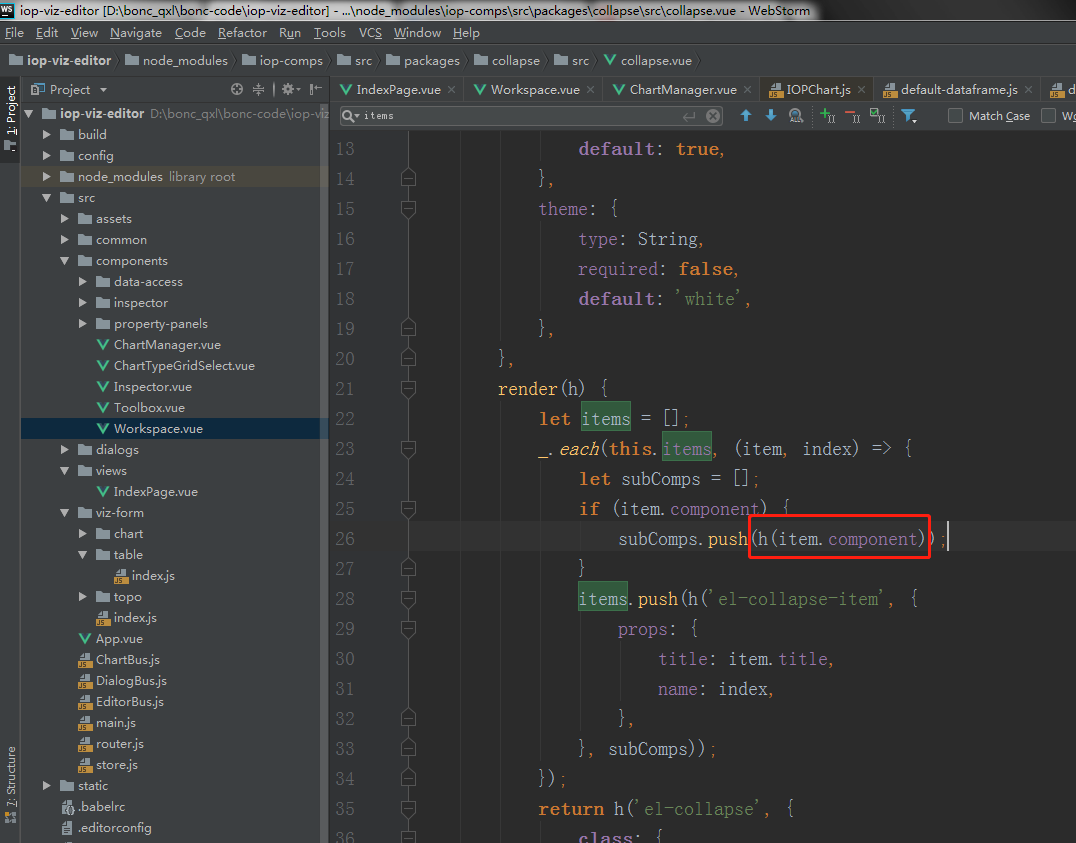


9.这里没有参数，怎么运行的函数？

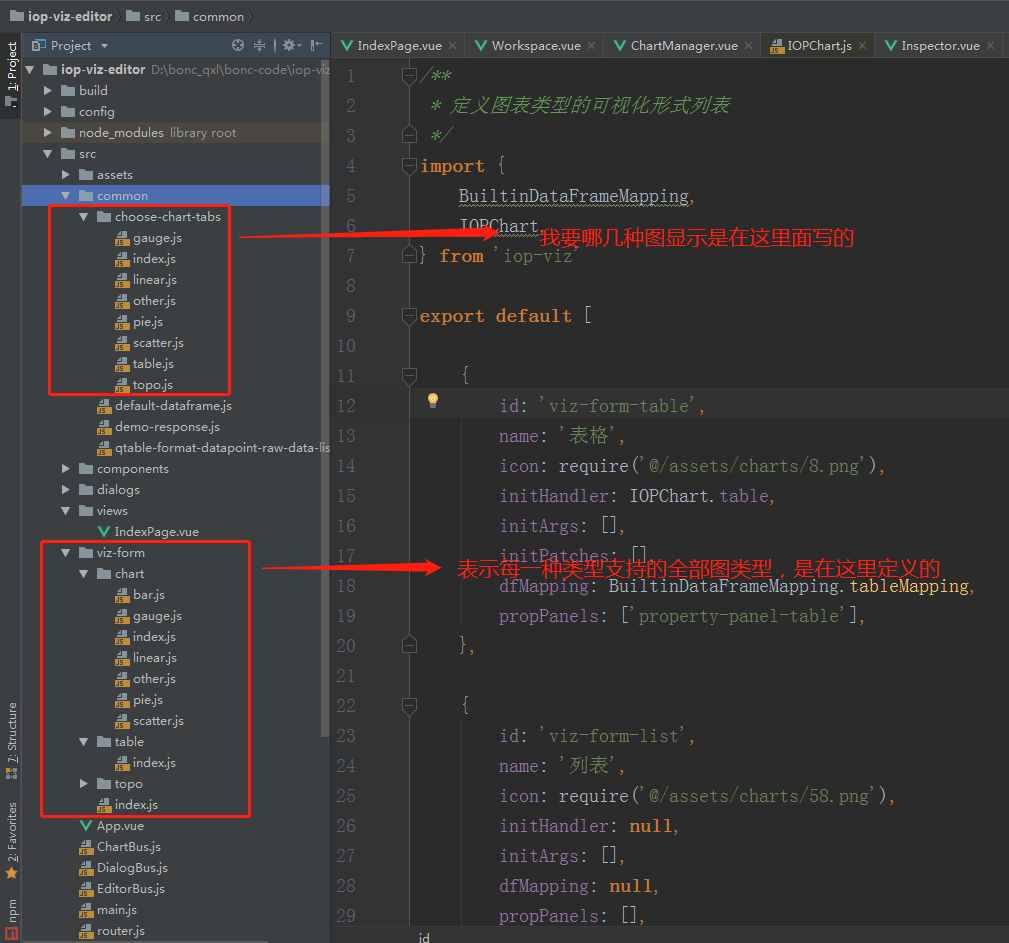


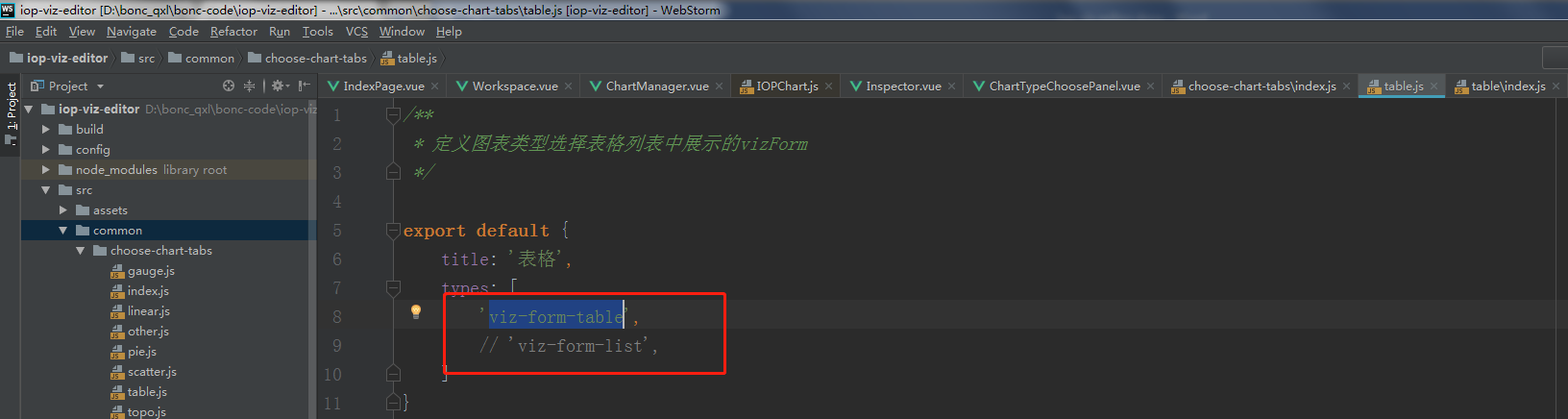


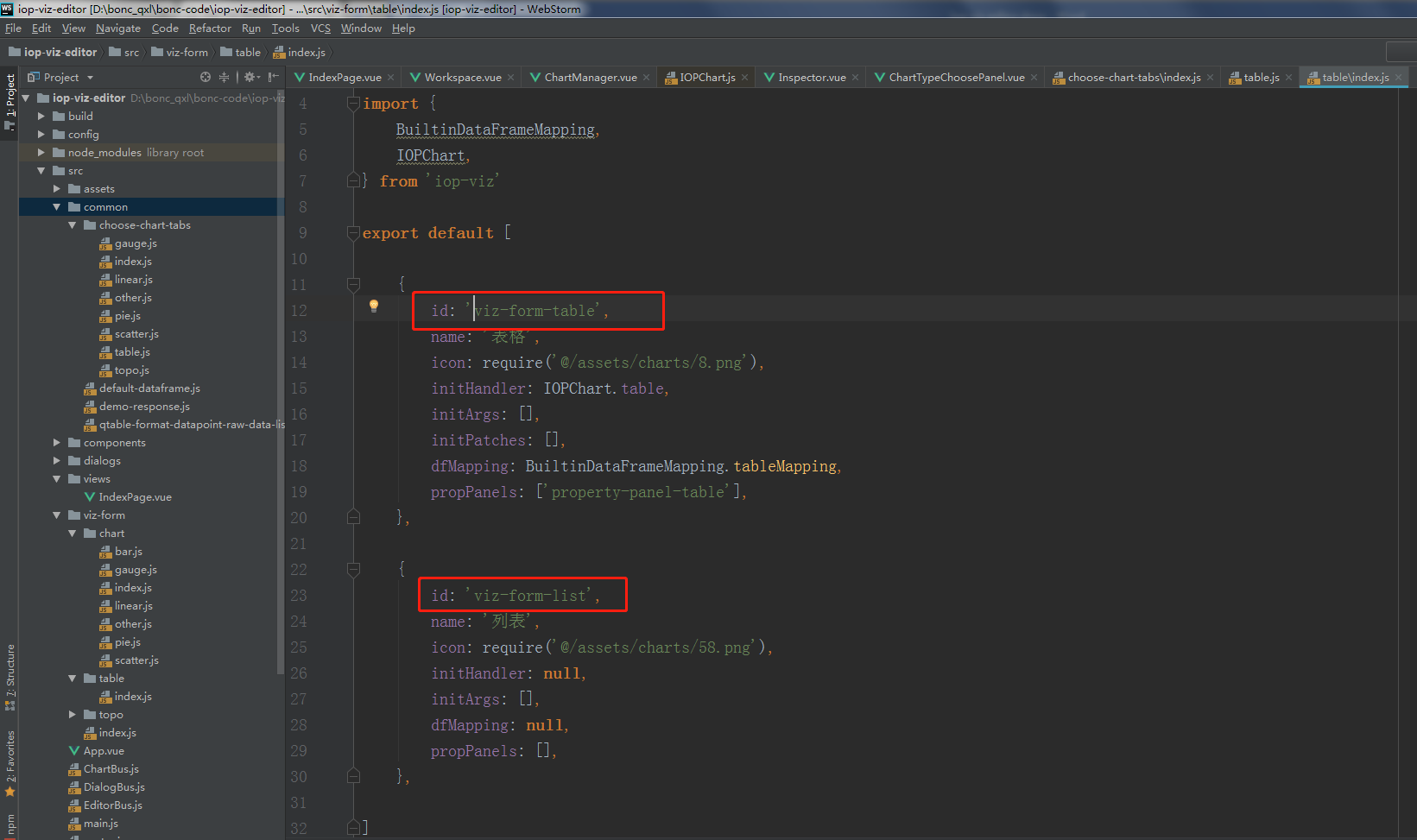
10.这样就可以生成一个组件哦，为啥有的行，有的不行，内部的机制还得好好理解



11.巧妙的设计，把支持的所有类型和要用的类型分离，很好的办法

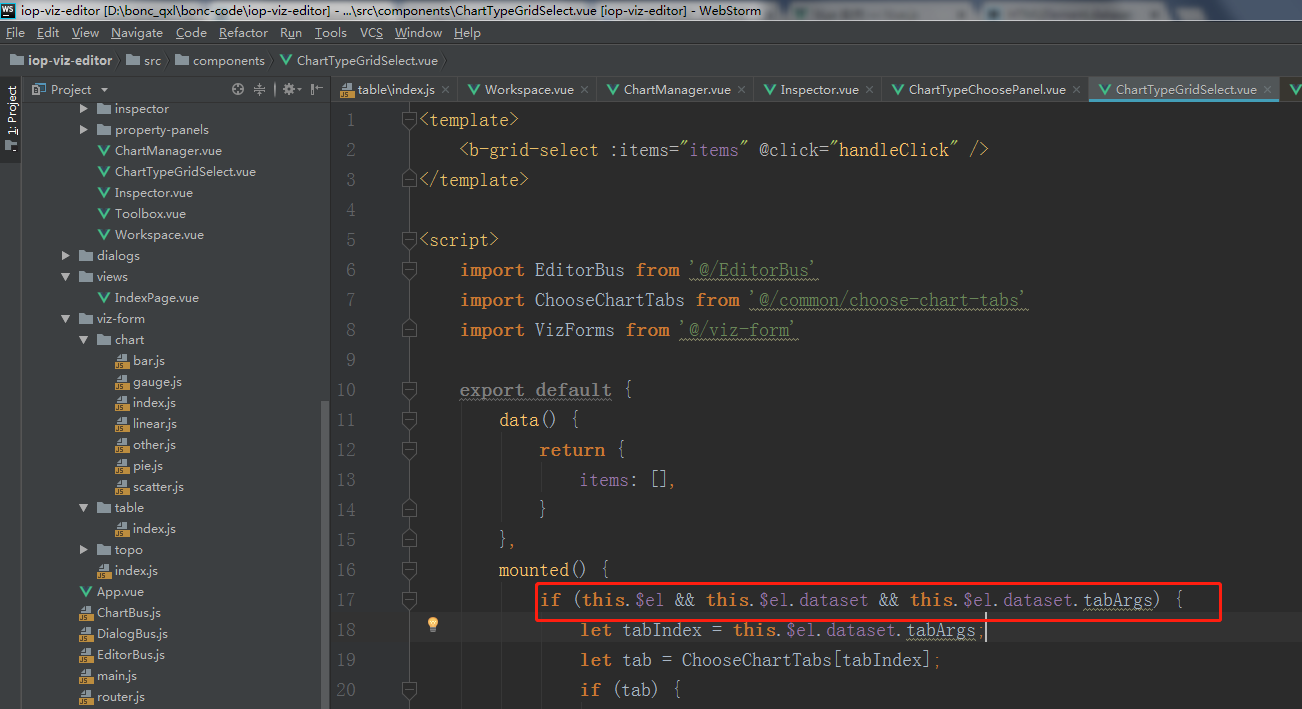




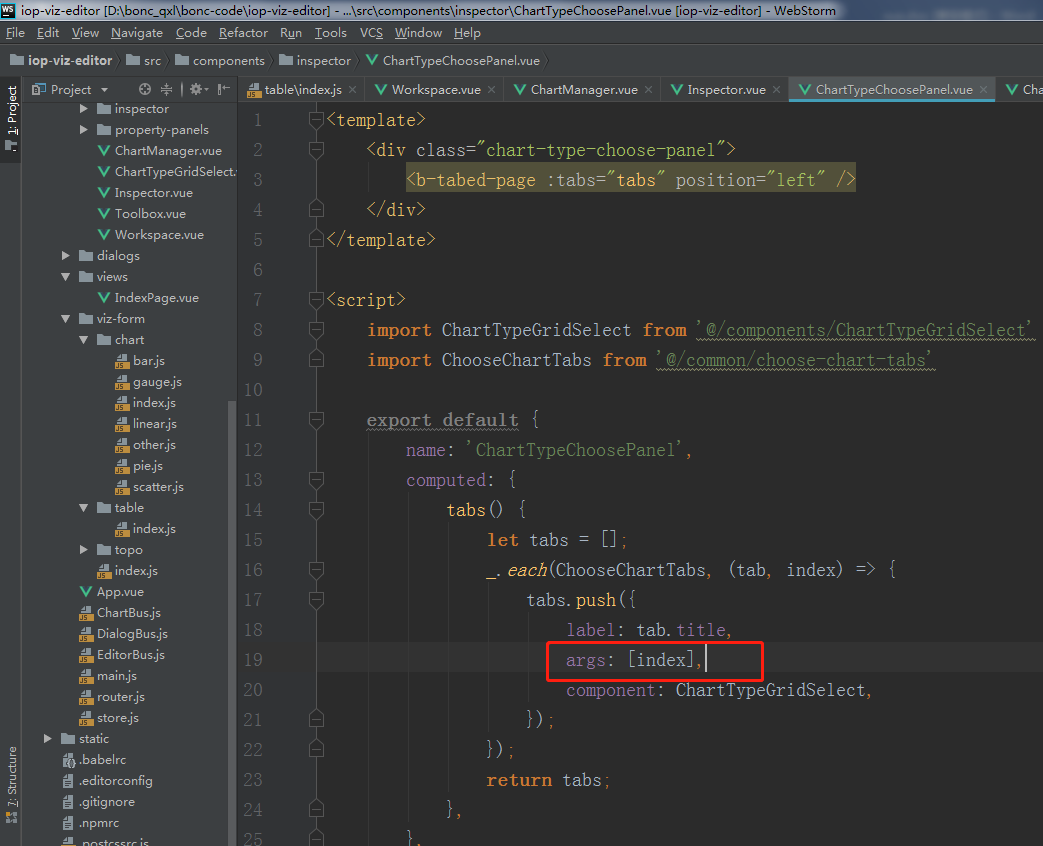


（注意这个是在store.js里面用到的）

12.表示有this.$el，并且有个自定义的属性tab-args

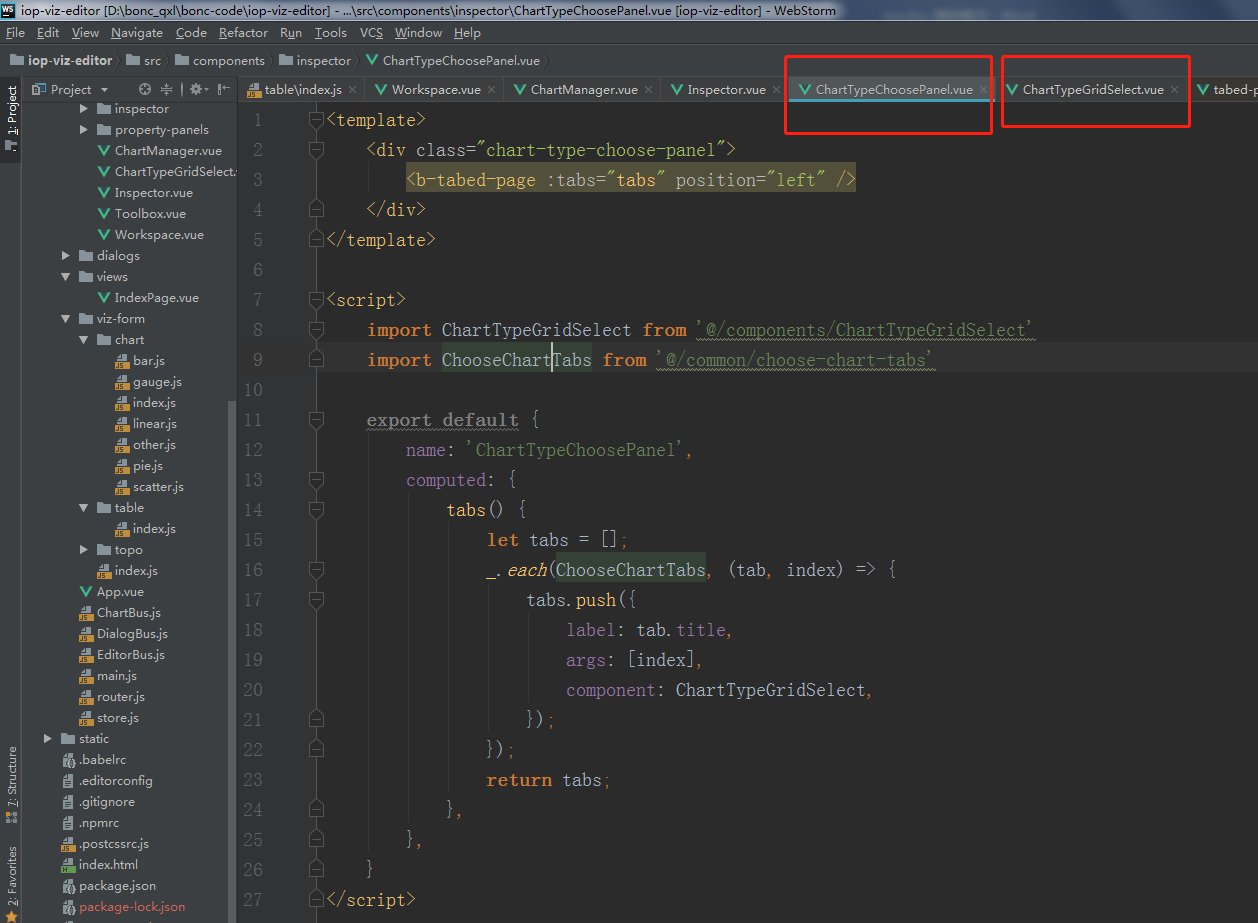


13.这个是什么意思



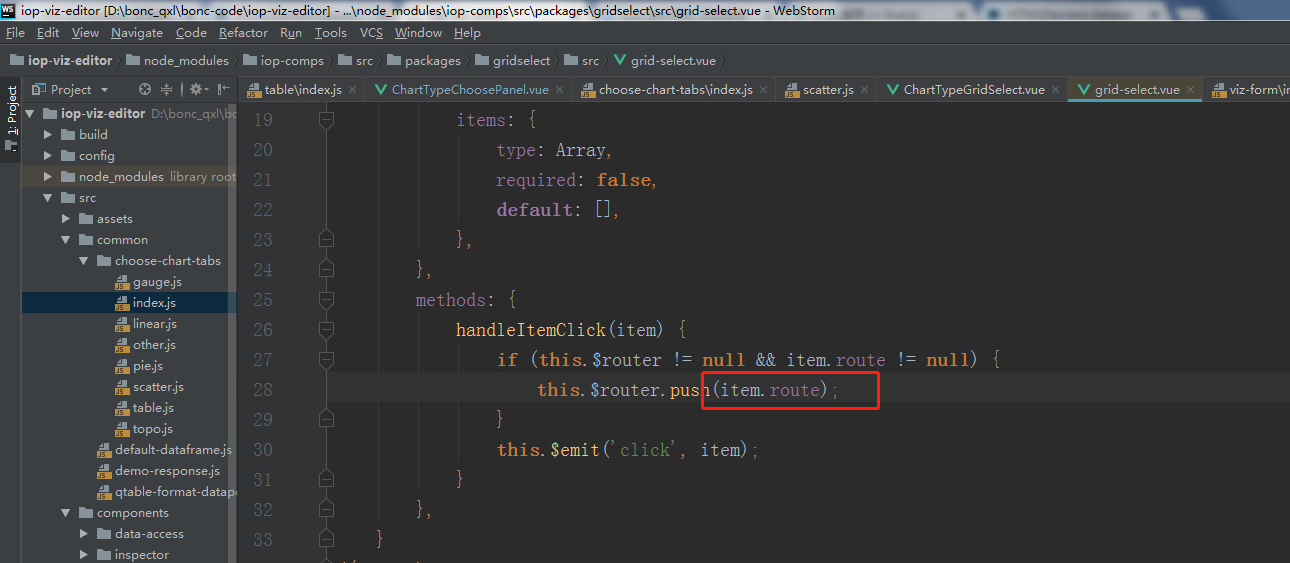
14.这2个文件一定要好好理解

这么麻烦用上了this.$el的原因是不行在引入component的时候给这个component传值，所以才这么麻烦的吧？

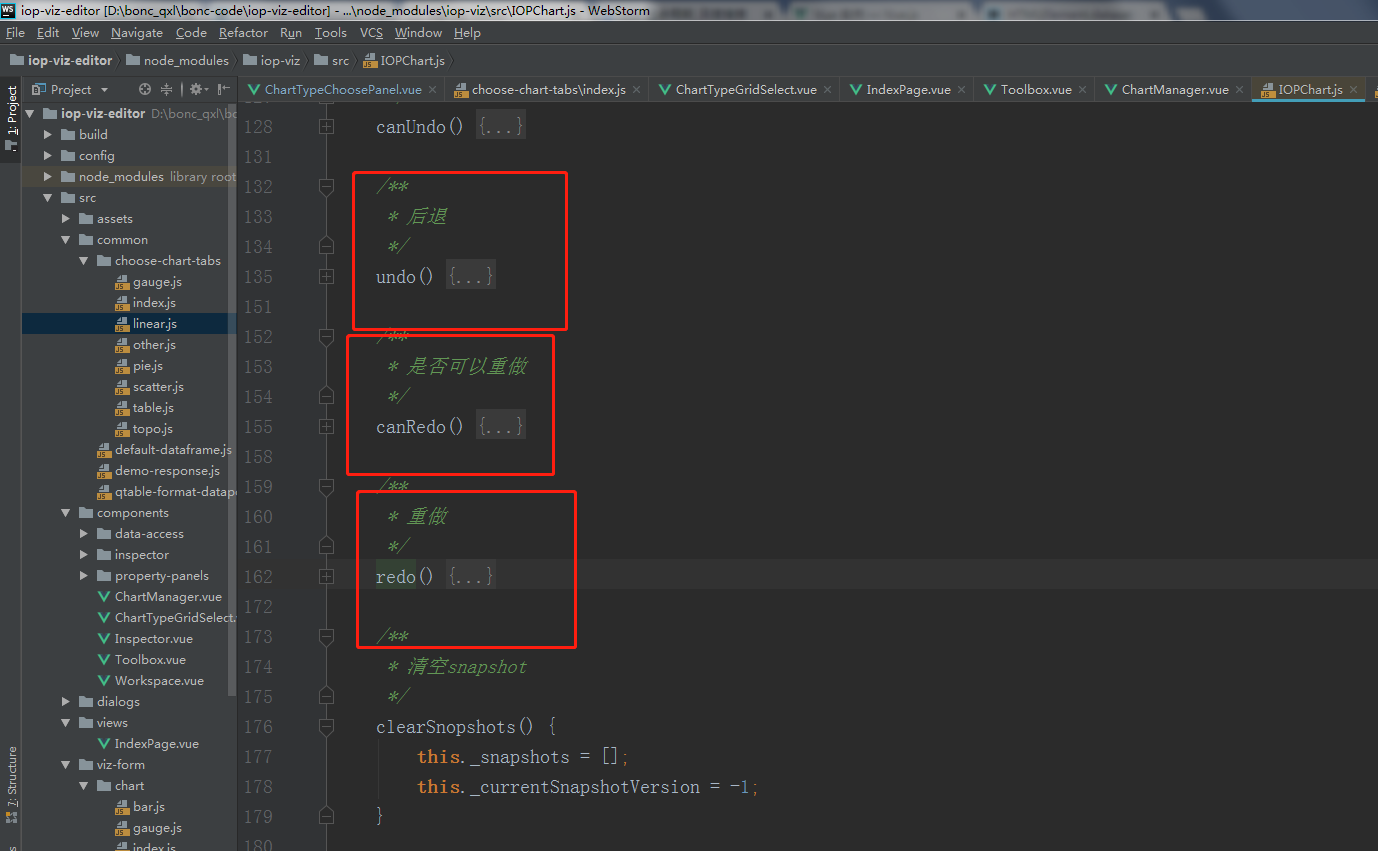


15.这个是啥意思？怎么弄的？

还有，this.$emit是传给了谁了？上一级？得弄清楚

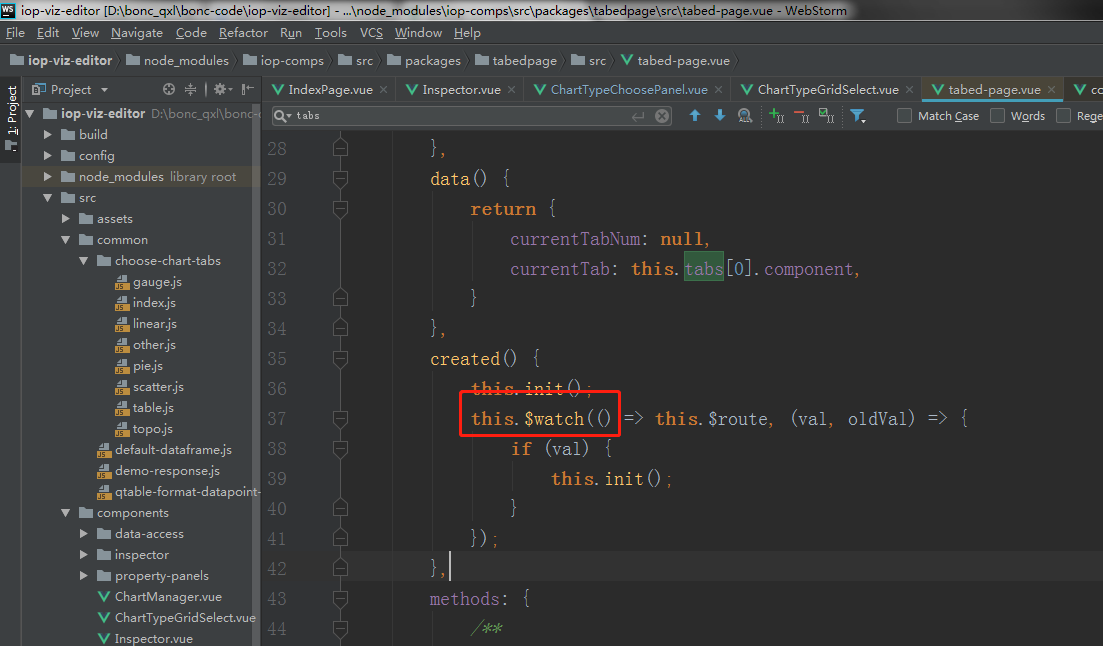


16.仔细看看这些的逻辑



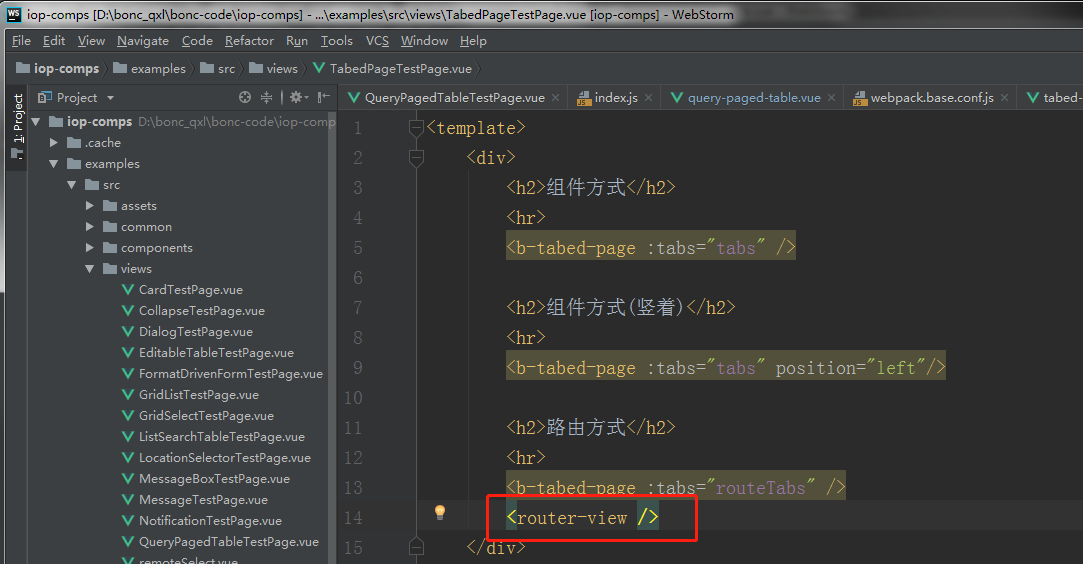
1. 这是什么意思，created和watch的顺序是咋样的？

这是vue实例的方法，自带的，应该和watch那个选项是一个吧？只是位置不同，这个要加上this.

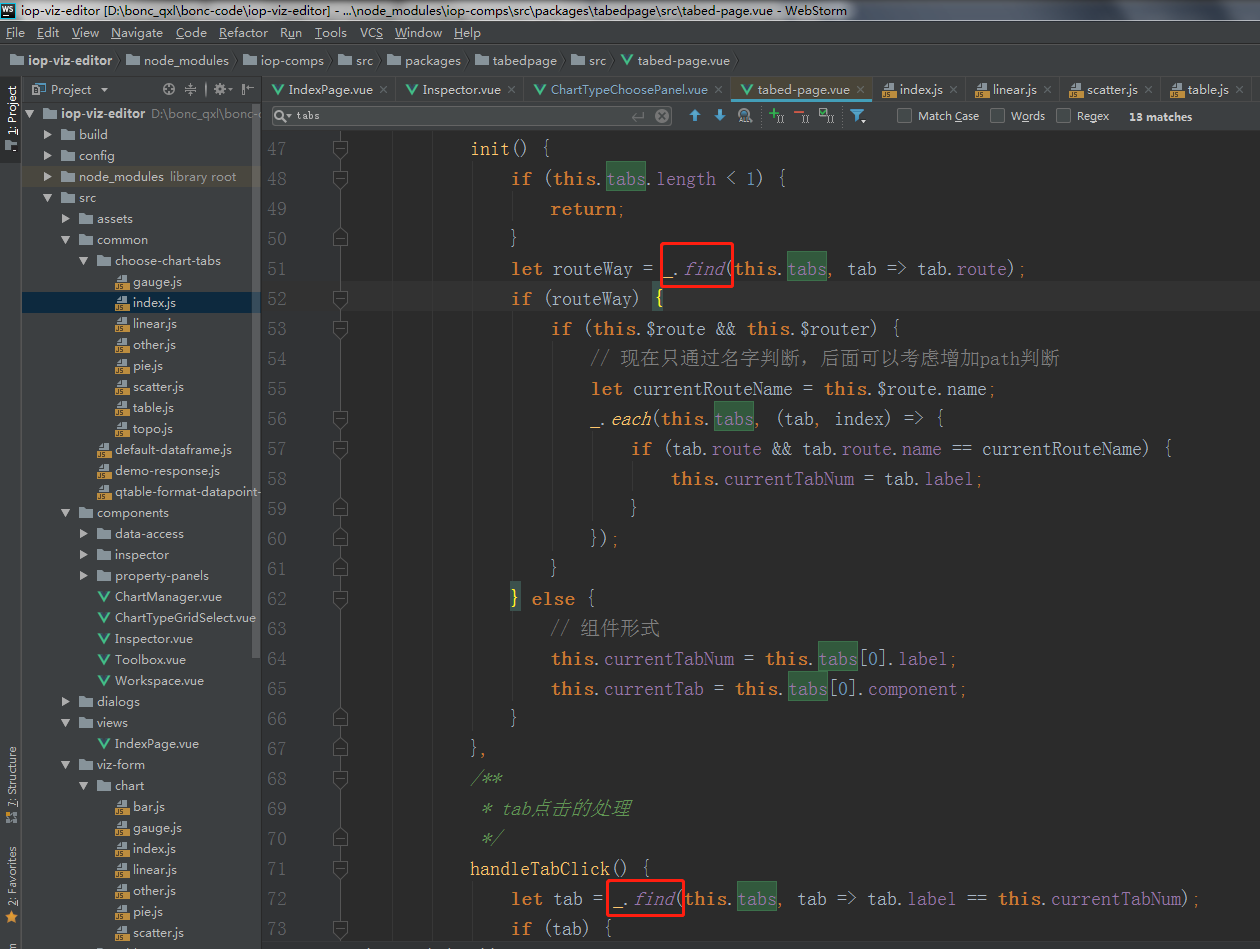




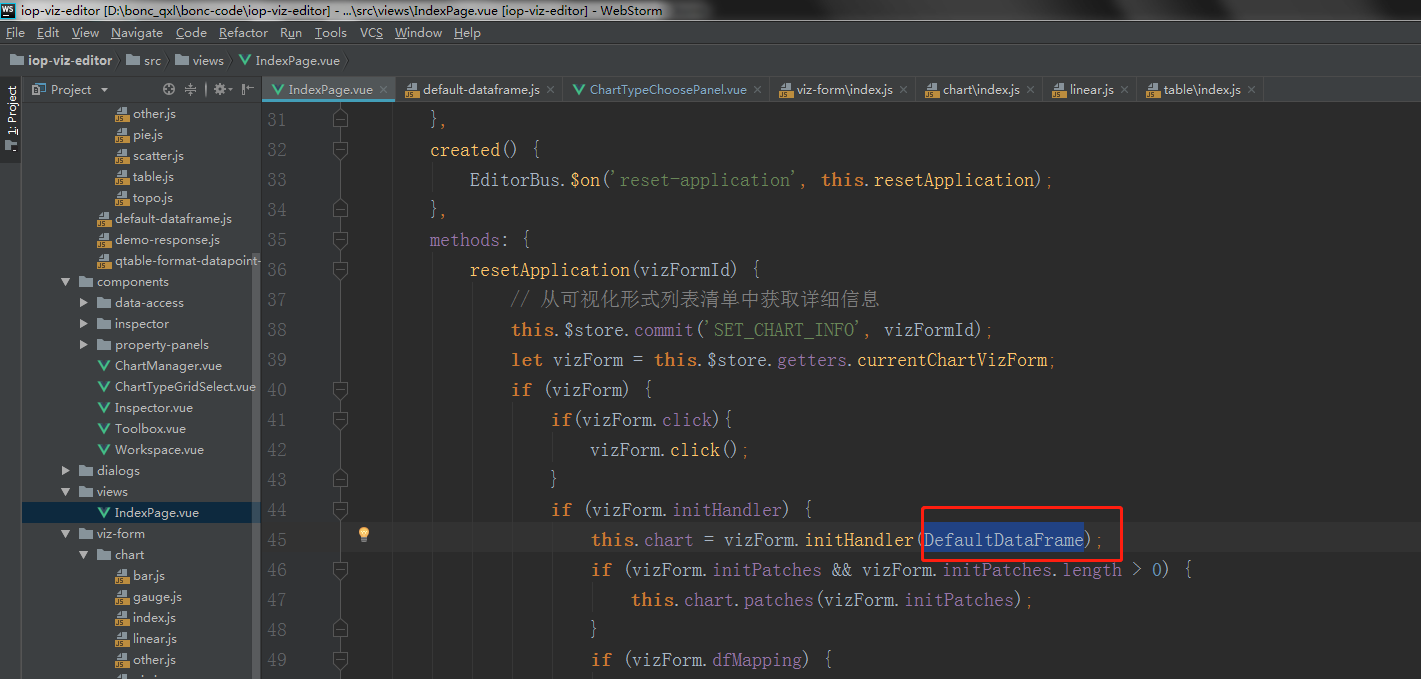
1. 这个是什么意思？

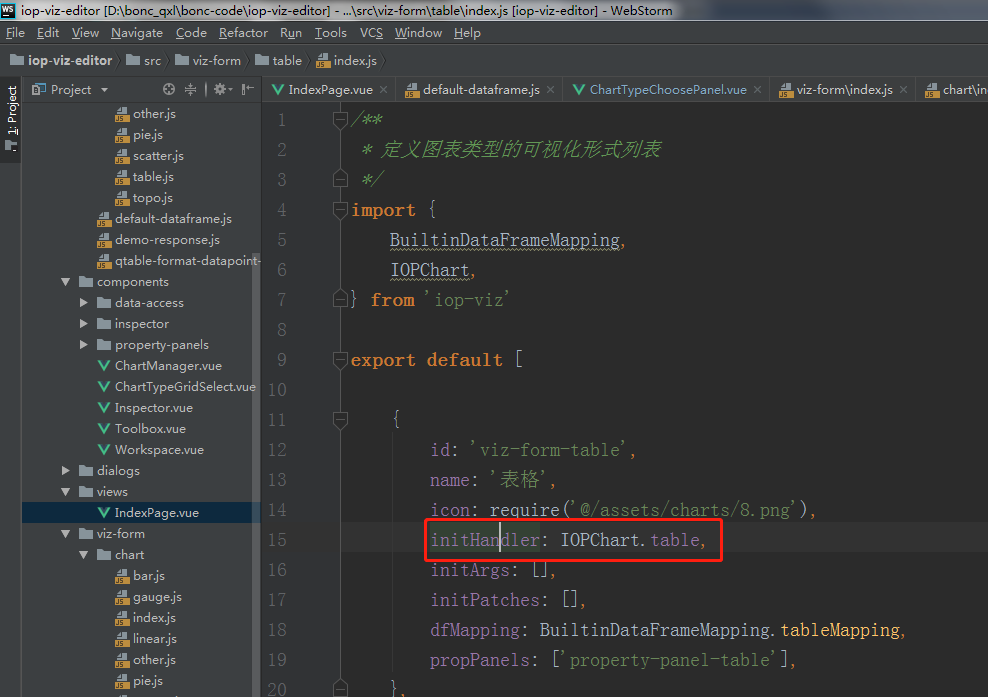


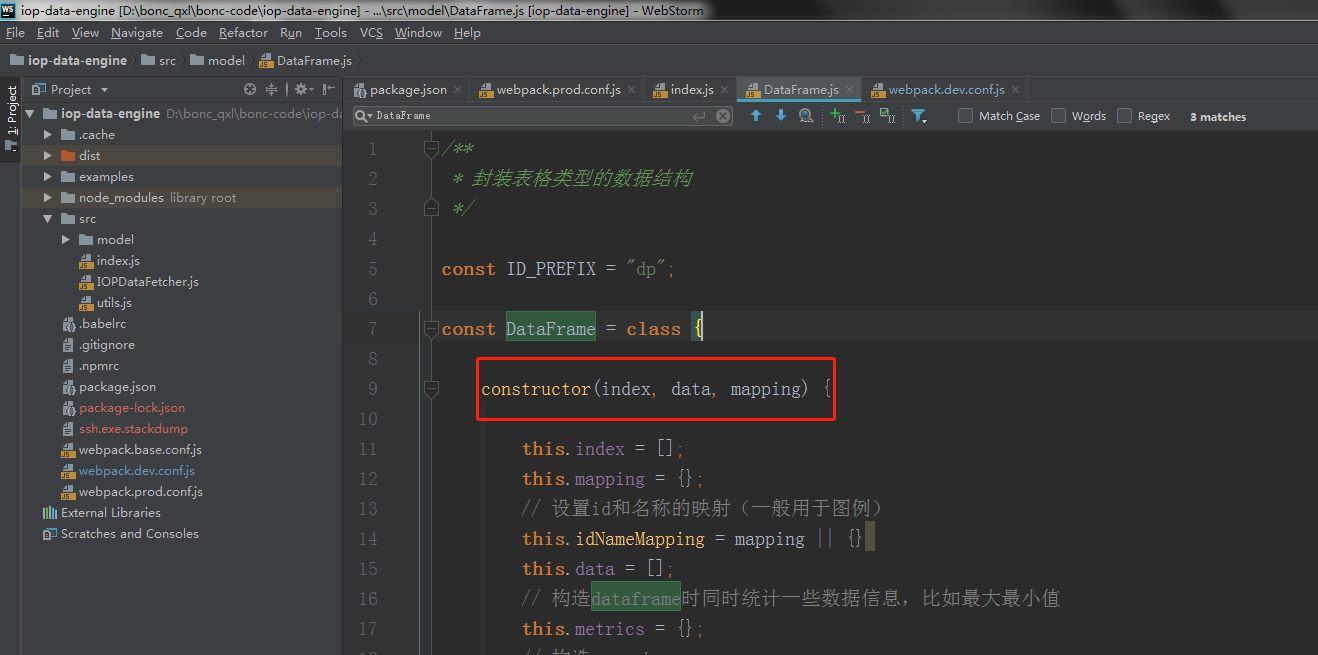
3.注意这个find的用法



1. 这种引用函数的地方也可以加参数的，虽然咋一看不能接收参数，但它实际上是一个函数的引用，所以可以加参数的







1. 不懂这个数据封装

