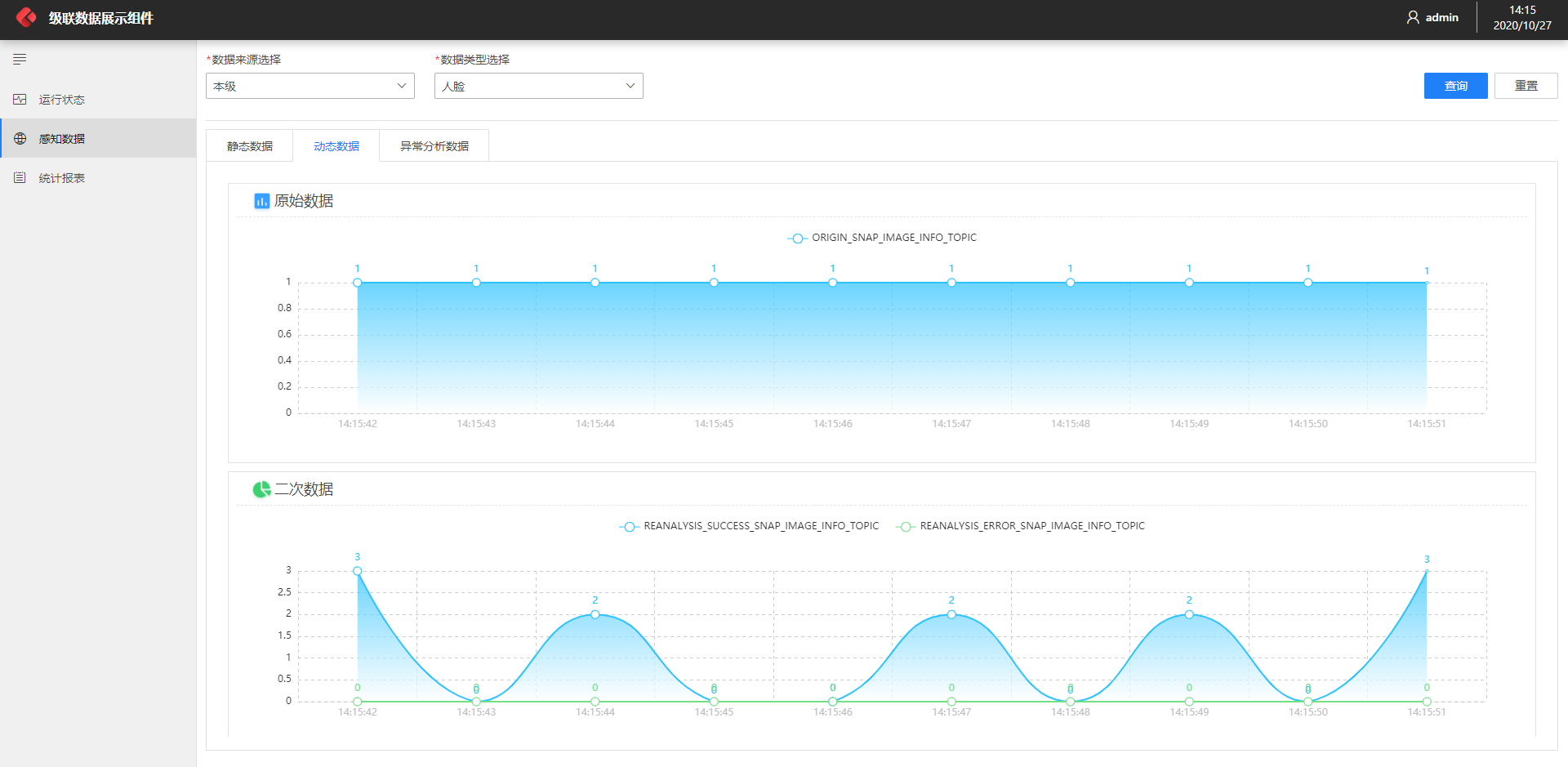
实现echarts图表的动态展示

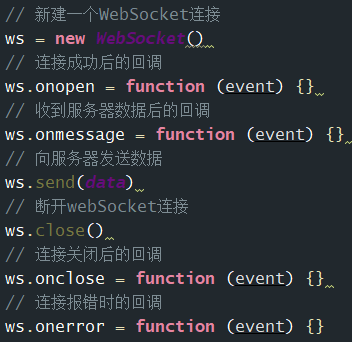
### 实现效果

选择查询条件，点击查询，进行数据的动态展示，一秒更新一次数据，展示形式为echarts的折线图。



### 关键技术

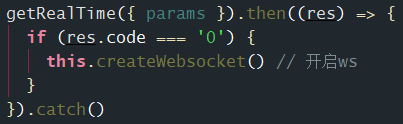
webSocket用法：阮一峰大神<http://www.ruanyifeng.com/blog/2017/05/websocket.html>



### 实现步骤

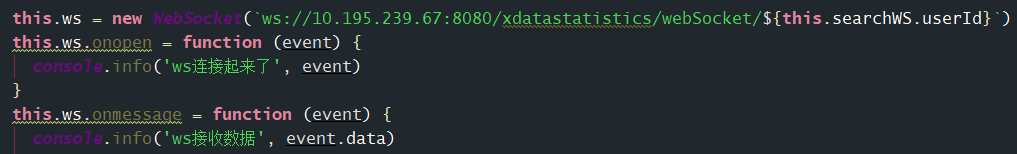
1. 向后端发送查询条件，得到应答后，开启webSocket。

当响应数据为code=0，则前后端同时开启webSocket，也就是createWebSocket()函数。



1. 与后端建立webSocket连接，后端每秒发送一次数据。

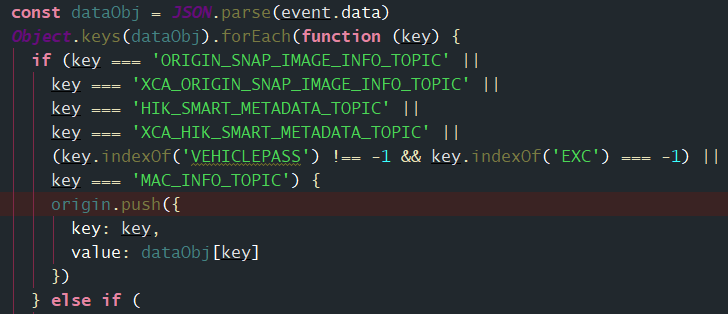
ws连接地址传入userId参数，用于标识不同用户，不同用户开启不同的websocket连接。我们在onmessage回调函数里处理接收到的data数据。



1. 前端将数据解析，按照topic归类为原始、二次、异常数据等。

首先分析报文，根据关键词将报文归类，比如查询的是本级-人脸数据，那么将返回来的数据进行遍历，归类为原始数据、二次分析数据、聚类数据和异常数据，按照key和value的形式拼接，并存放在数组中。

*（origin数组就存放了原始数据。）*



1. 初始化echarts图表，执行渲染函数。(重点)

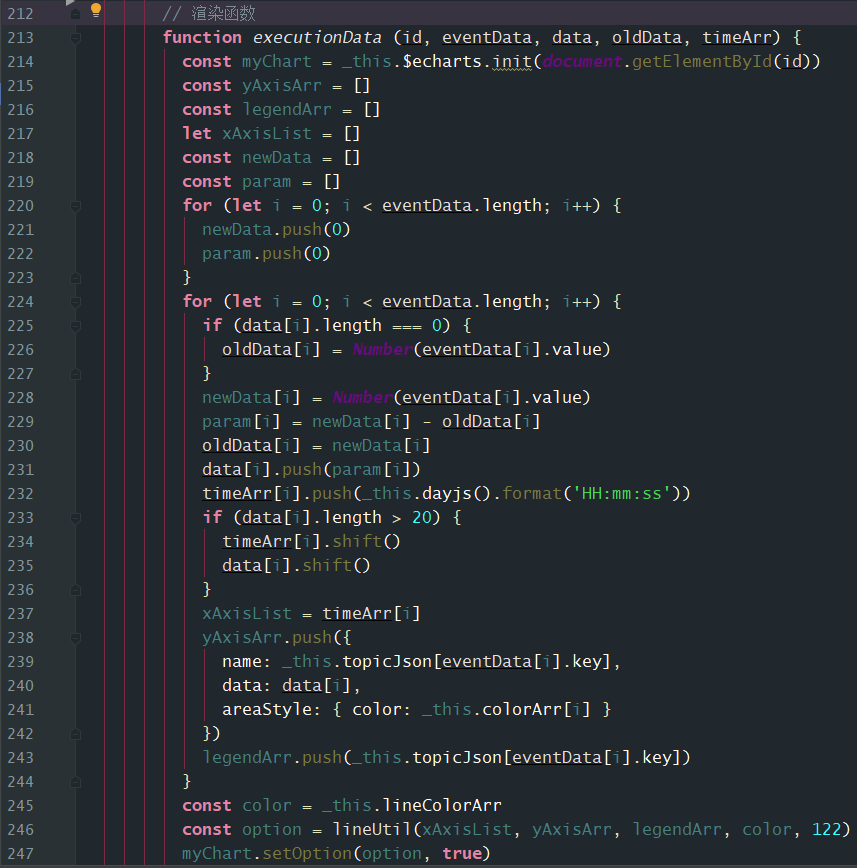
**Id：**要渲染的dom节点

**EventData：**解析归类好的数据

**Data：**用于存放y轴数据（必须在onmessage函数的外部定义，否则每次新进来数据，data就会被清空）

**OldData：**临时变量，存放上一次的数据，用于做差值（同样必须在onmessage函数的外部定义，否则每执行一次onmessage函数，oldData都会被初始化，就无法保存上一次数据了。）

**TimeArr：**用于存放x轴时间数据（必须在onmessage函数的外部定义，存放当前时间。）



220-231行，遍历数组，将本次数据和上次数据做差值param，然后将param放到data数组中，作为y轴的一个数据。为什么这里面都用的数组呢？因为传过来的数据有好几类，每个折线图中的y轴可能不止一条折线，因此需要通过循环，param差值可能有多个，分别push到对应的data中，从而才能在一个图中同时渲染出多条折线。

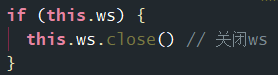
232行，每秒push一个当前时间，直接调用dayjs()方法。

233-236行，判断当data长度等于20，也就是折线图中共有20个数据时，就把第一个数据删除，这样就形成一种效果：数组在每秒添加一个新数据，同时删除第一个旧数据，看起来像数据在一直往前移动，这就是动态数据的原理。

237-247行，将timeArr和data分别赋给x轴和y轴，生成option，chart每秒执行一次setOption方法，将数据渲染到页面。

1. 当查询条件改变或切换页面时，断开webSocket连接。

每次都要先判断是否存在ws连接，存在的话，要先把现有的连接关闭再开启新的ws连接。



同时，在切换页面，实例销毁后也要断开ws连接。

