重构说明

重构前瞻

可能遇到的问题：
1. 没有接口文档，对于接口的参数传递不确定是否会因为代码结构的改变而发生问题

2. 改造范围：
理想的改造范围主要为右图涉及范围，但是可能会波及到运算符表达式配置那块儿，之前运算符和表达式区域改动的不多，所以配置运算符和表达式可能会存在业务流程上的问题

3.预计重构的时间：2~3周（可能会因为不熟悉接口以及部分业务逻辑而花费一些时间）

重构原因

不宜维护

组件通信

watch结合bus的通信方式
1. 繁琐混乱的通信方式：$bus
 $bus的书写方式决定了它只适用于较小的项目且嵌套层级不要过多，简言之不易在当前项目中使用。混乱的通信方式产生出冗余代码-网状结构
 a. 代码逻辑性极具下降，可阅读性变低
 b. 对于每一个事件的发起点都需要另外一个事件来处理
 c. 你将很难查找到每一个事件是从哪里触发，满篇都是业务逻辑
 d.频繁大量的使用不宜维护
2. watch监控数据改变来重新绘制区域内容也会占用大量内存

代码规范

1. 通篇行内样式，每个组件都是，即使是相同的样式，也会在代码里面重复多次，降低代码的可阅读性
2. 冗余的data变量，没有注释
3. 行内标签嵌套块标签，且没有代码缩进

重构方案

组件通信

不使用bus。举例：如上图所示左右两侧想要实现即使通信，vuex是最好的选择，不管是父子传值也好，兄弟传值也好，能够高效的实现组件之间的数据共享和传输，最关键的是存储在vuex中的数据都是响应式的，能够实时的保持数据和页面的同步，这样就可以避免大量多次去重复修改数据

代码规范

1. 最基本的尽量避免书写行内样式
2. 不必大量书写data变量，对于局部代码-声明局部变量即可

重构结果

1. 代码的可阅读性的提升
2. 减少冗余代码，优化代码结构
总结：降低维护成本