Modeling低代码

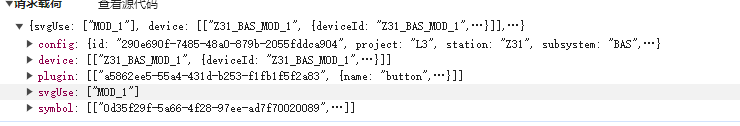
这是我自己想做的一个项目，主要通过低代码形式来完成建模工具流程图的搭建以及渲染

页面主要看src/views/configurePlugin里的，其他页面可以不看

这套代码的主要流程是从原版建模工具中获取项目，车站，子系统，图元数据，然后用这些数据搭成全新的流程图，业务流程如下：

1. 选择项目打开，目前后端还有问题，需要在老版建模工具里先打开该项目，
2. 选择流程图/新建流程图
3. 如果是新建流程图的话会自动给一个默认的底图，底图可以点击左侧的组态层进行更换，也支持自己上传svg以及dwg文件生成底图
4. 图元拖入之后可以配置图元三维数据，设备图层，以及点位配置数据，还支持点击事件配置
5. 点击预览可以跳转到对应流程图的页面查看生成后的流程图

数据结构：



Config, 主要包括流程图的项目，车站，子系统，流程图ID，背景图设置等数据

Device, 包括当前流程图中使用的设备id，以及设备对应的图元id，单个设备可以对应多个图元

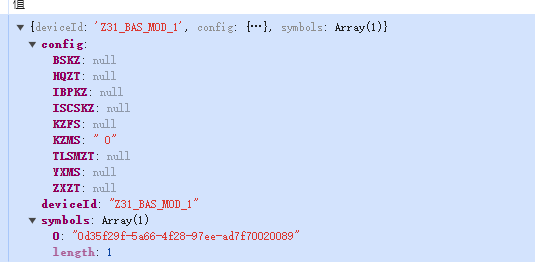
Plugin, 使用的插件数据

SvgUse, 使用的svg数据

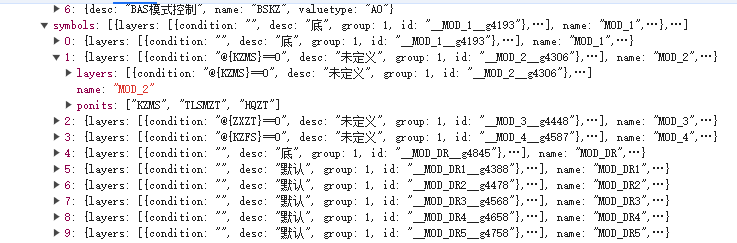
Symbol, 图元数据

工作流程：

整体流程上对vue进行了仿写，以数据驱动为核心，图元拖入流程图后仿照vue的observer写了一版依赖收集代码，比如说通过设备点位数据生成设备对象



选择了数据劫持方案对config上的点位进行劫持



[图元数据的layers上面存在condition为’@{zxzt}=0’，那么通过正则获取zxzt这个字符，就会在该图元的config数据上对zxzt这个进行订阅，本质上是套订阅发布模型，每当zxzt数据变化时都会触发订阅回调函数，实现页面实时更新，代码可见src\views\configure\configureWrap.vue的observer函数以及watchSymbolConfig函数](mailto:图元数据的layers上面存在condition为’@{zxzt}=0’，那么通过正则获取zxzt这个字符，就会在该图元的config数据上对zxzt这个进行订阅，每当zxzt数据变化时都会触发订阅回调函数，实现页面更新，代码可见src/views/configure/configureWrap.vue的observer函数以及watchSymbolConfig函数)

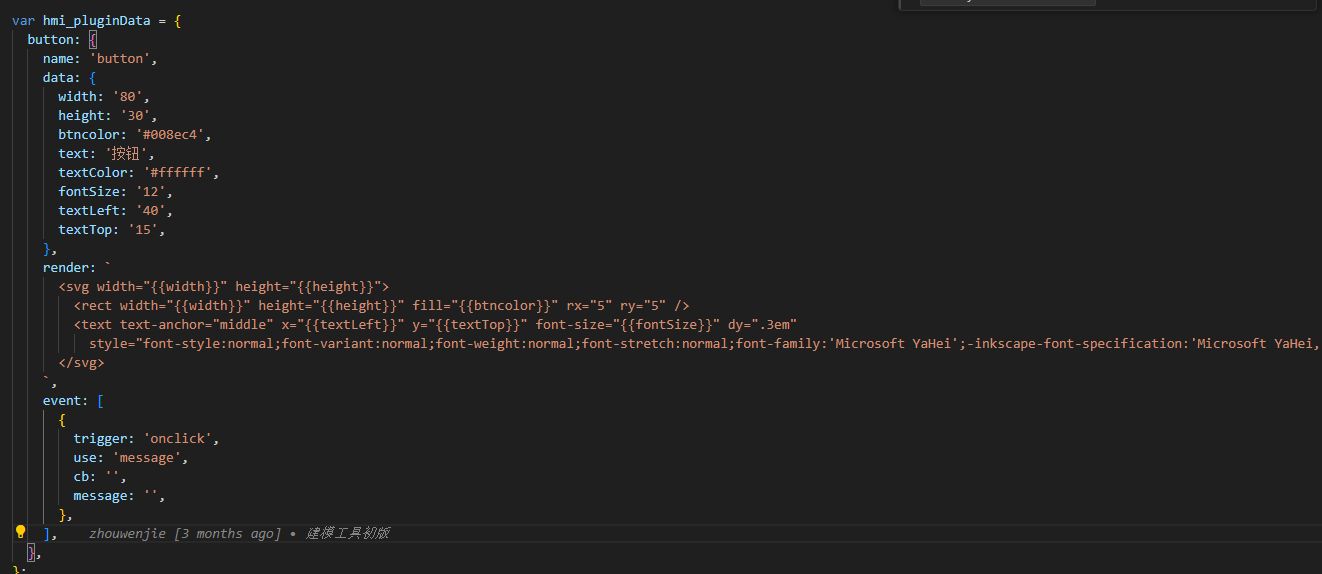




与vue相比，移除了nextTick，直接将回调写入了watch中，而且因为直接作用于层级，流程图svg显隐方案是变更元素的visibility属性，并没有元素变化，因此diff算法也可以不需要了（当然可以进行扩展）

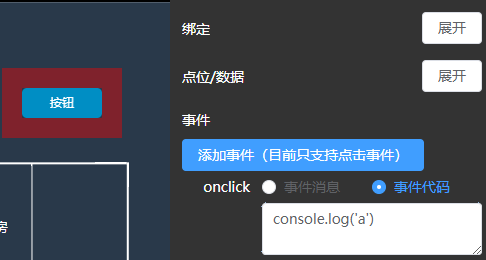
插件功能（未完成）：

比如一个按钮插件，他的数据如图

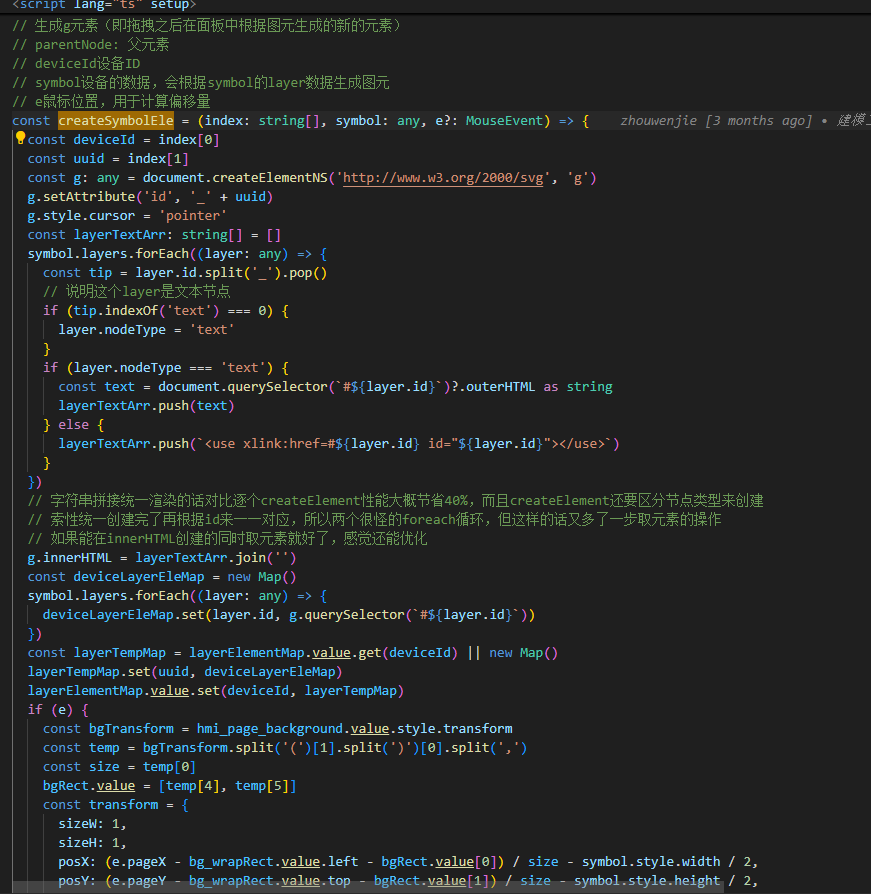


模仿了vue的render函数，同样对数据进行了劫持，每次数据变更之后会直接更新整个插件，event则是该插件的事件，use可以为’cb’或’message’，cb则是可以在编辑器上配置插件的点击事件，message则是采用冒泡消息的形式触发事件

例如：



页面渲染的话，是自己写的渲染函数，并没有参考vue的函数，原因在于在多图元的情况下，vue对于每一个元素都会用document.createElement，量太大的话渲染速度完全不行，考虑到现实情况，因此直接采用字符串数组形式进行渲染，具体可以看代码createSymbolEle方法



全局变量：

新增的功能，预期是做成UE4的蓝图功能的，现在实验功能是可以在图元上绑定变量控制显隐，其他图元也绑定变量的话那么该变量将会共享，具体实现在src\utils\binding.ts中

数据整体使用indexedDB存储

其实这东西现在看着是svg编辑器，插件功能本身就是为自定义组件服务的，中间的svg随时都能改成html模板，其实只差全局变量以及全局函数，设备数据采用层级结构，现在是设备：[设备ID，[图元id，图元id]]，也支持向多级结构拓展，可以按需求是否写diff或者nextTick，vue针对特有业务其实也有不足，有些模块还是要自己实现

渲染sdk：

位置public\js\sdk.js，实现与项目中差不多，但是移除了编辑逻辑

具体使用可以参考src\views\configure\ceshi.vue