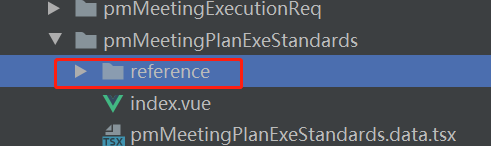
# 弹窗使用说明

## 主子表使用弹窗

1. 主子表主表弹窗
2. 添加组件包



1. 在页面上方tempalte中引入弹窗组件

<template>  
 <DefUser @register="registerDefUserModal" @success="handleDefUserSelect" ref="defUserRef" />  
</template>

1. import弹窗页面

import DefUser from '../pmProjectStakeholder/reference/defUser.vue';

import { useModal } from '/@/components/Modal';

1. components中添加组件

components: {  
 DefUser,  
},

1. setup（）

const defUserRef= ref<any>(null);  
const [registerDefUserModal, { openModal:openDefUserModal }] = useModal();

方法：

function defUserReference(){  
 visible.value=true;  
 var data={"record":{"id":id.value}};  
 //console.log('record:'+JSON.stringify(data));  
 openDefUserModal(true, data );  
}

在editFormSchema中传入defUserReference getFieldsValue,

const [registerForm, { setFieldsValue, resetFields,getFieldsValue,

updateSchema, validate }] = useForm({  
 labelWidth: 100,  
 schemas: editFormSchema(type,defUserReference),  
 showActionButtonGroup: false,  
 baseColProps: { span: 24 },  
 actionColOptions: {  
 span: 23,  
 },  
});

在return中添加 registerDefUserModal，defUserReference handleDefUserSelect

return { tvisible,handleDefUserSelect,defUserRef,registerDefUserModal,defUserReference};  
},

方法选中确认后回填数据

function handleDefUserSelect(){  
 debugger;  
 //*TODO:根据业务需求，自己回填字段* if(defUserRef.value.selectRowList.length>0){  
 const params = getFieldsValue();  
 //console.log('params :'+JSON.stringify(params));  
 params.userId=defUserRef.value.selectRowList[0].userId;  
 params.userId=defUserRef.value.selectRowList[0].realName;  
 params.userId=defUserRef.value.selectRowList[0].orgIdList[0];  
 params.phone=defUserRef.value.selectRowList[0].defUser.mobile;  
 params.name=defUserRef.value.selectRowList[0].defUser.username;  
 params.userId=defUserRef.value.selectRowList[0].echoMap.orgIdList[0];  
 setFieldsValue(params);  
 visible.value=false;  
 }  
}

1. 在data.tsx的editFormSchema中添加defUserReference

// 编辑页字段  
export const editFormSchema = (\_type: Ref<ActionEnum>,defUserReference): FormSchema[] => {

1. 在需要弹窗的字段上修改如下

{  
label: t('learnpm.pm.pmProjectMeetingPlanning.meetingType'),  
field: 'meetingType',  
 component: 'InputSearch',  
 ifShow: ({ values }) => {  
 return true;  
 },  
 componentProps: ({ formActionType }) => {  
 return {  
 enterButton: '选择',  
 onSearch: (value: string) => {  
 defUserReference();  
 return value;  
 },  
 };  
 },  
 /\*componentProps: {  
 api: query,  
 labelField: 'meetingType',  
 valueField: 'meetingType',  
 showSearch: true,  
 filterOption: (input: string, option: any) => {  
 alert(option)  
 return option.label.toUpperCase().indexOf(input.toUpperCase()) >= 0;  
 },  
 },\*/  
colProps: { span: 12 },  
 },

1. 主子表子表弹窗

1.大致和主表一致，区别不用再data.txs传方法。直接在vxe-grid 添加点价事件

<vxe-grid ref="xGrid" v-bind="gridOptions" v-on="gridEvents" @cell-click="handles">

2.子表弹窗方法如下

const index = ref(-1);  
const row = ref<any>(null);  
const boolean = ref(false);  
  
function handles(ev) {  
 row.value = ev.visibleData[ev.$rowIndex];  
 index.value = ev.$rowIndex;  
 ***console***.log(ev, 'eevvv');  
 if (ev.cell.classList.contains('col--actived') && ev.\_columnIndex === 2) {  
 var data = { record: { id: id.value } };  
 ***console***.log('handle');  
 boolean.value = true  
 openDefUserModal(true, data);  
 }  
}  
  
function remove() {  
 xGrid.value.remove(row.value);  
 boolean.value = false;  
}

3.选中的事件和主表一致

# 单表和主表弹窗一样