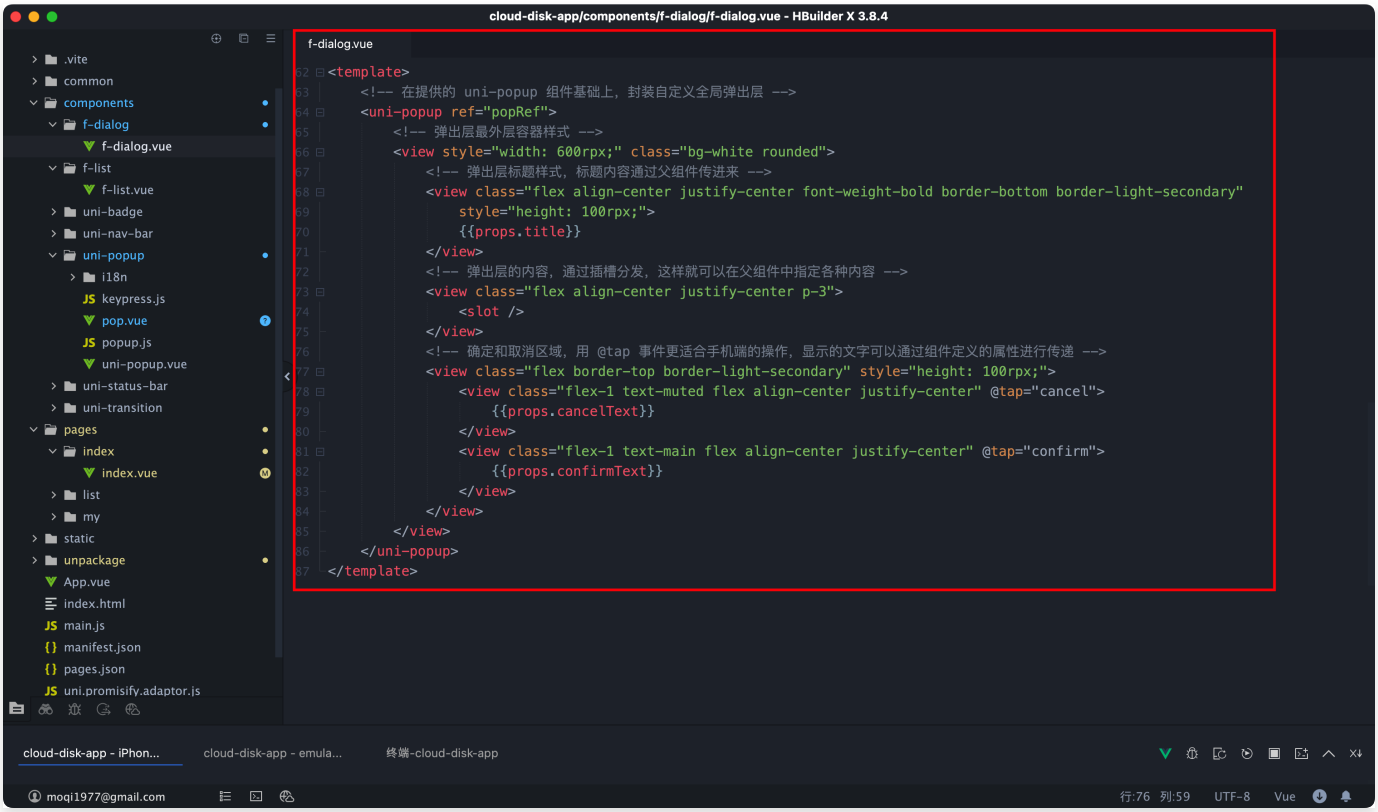
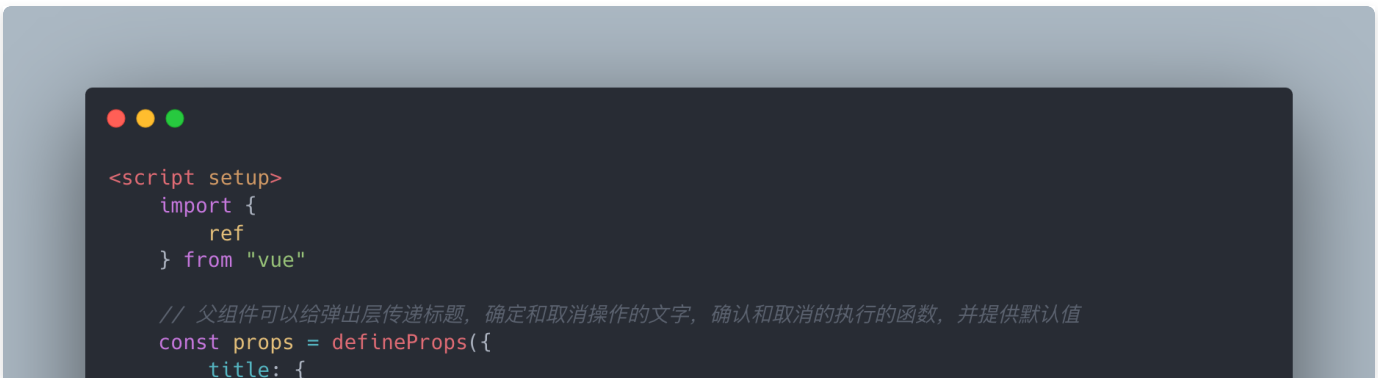
7. 重命名和删除功能

删除功能实现

我们先来封装自己的全局弹出层组件 **f-dialog.vue**，按照 easycom 规范建立组件文件，编写模版部分代码



然后编写其脚本部分代码，主要做法详见代码注释，如下：



```
            type: String,
            default: '提示'
        },
        cancelText: {
            type: String,
            default: '取消'
        },
        confirmText: {
            type: String,
            default: '确定'
        },
        onConfirm: {
            type: Function,
            default: null
        },
        onCancel: {
            type: Function,
            default: null
        }
    })

    // 根据 template 中的 ref 获取弹出层元素
    const popRef = ref(null)

    // 打开弹出层
    const showPopup = () => {
        popRef.value.open()
    };

    // 关闭弹出层
    const hidePopup = () => {
        popRef.value.close()
    };

    // 点击确认执行的函数
    const confirm = () => {
        // 这里的判断起到容错作用，因为如果 props.onConfirm 不是函数，就不能执行了
        if (typeof props.onConfirm === 'function') {
            // 执行属性中定义的，由父组件传入的确认操作函数
            props.onConfirm();
        }
        // 关闭弹出层
        hidePopup();
    };

    const cancel = () => {
        if (typeof props.onCancel === 'function') {
            // 执行属性中定义的，由父组件传入的取消操作函数
            props.onCancel();
        }
        hidePopup();
    };

    // 向父组件暴露弹出层的打开和关闭方法，省去了 emit
    defineExpose({
        showPopup,
        hidePopup
    })
</script>
```

在 index 页面需要执行删除功能的时候，我们希望弹出对话框进行确认

- 先在 index 页面的 template 部分使用 f-dialog 组件，因为后面还有重命名等其他对话框，所以用 ref 命名区分清楚，并给删除对话框的确认和取消操作指定对应的处理函数。

```
* index.vue
<template>
    <view>
        <!-- 选中个数为0 -->
        <uni-nav-bar v-if="checkedList.length === 0"> ...
        <!-- 选中个数不为0 -->
        <uni-nav-bar v-else> ...
        <!-- 搜索框 -->
        <view class="px-3 py-2"> ...
        <!-- 自定义列表组件 -->
        <f-list v-for="(item, index) in list" :key="index" :item="item" :index="index"
            @my-select="handleSelect(index)"></f-list>
        <!-- 底部操作条 -->
        <!-- 选中元素个数大于0，才会出现操作条 -->
        <view v-if="checkedList.length > 0"> ...
        <!-- 删除对话框 -->
        <f-dialog ref="deleteDialogRef" :onConfirm="handleDeleteConfirm" :onCancel="handleCancel">是否删除选中的文件？</f-dialog>
    </view>
</template>
```

- 为底部操作条的每个 item 绑定处理事件，**添加红框标注代码**

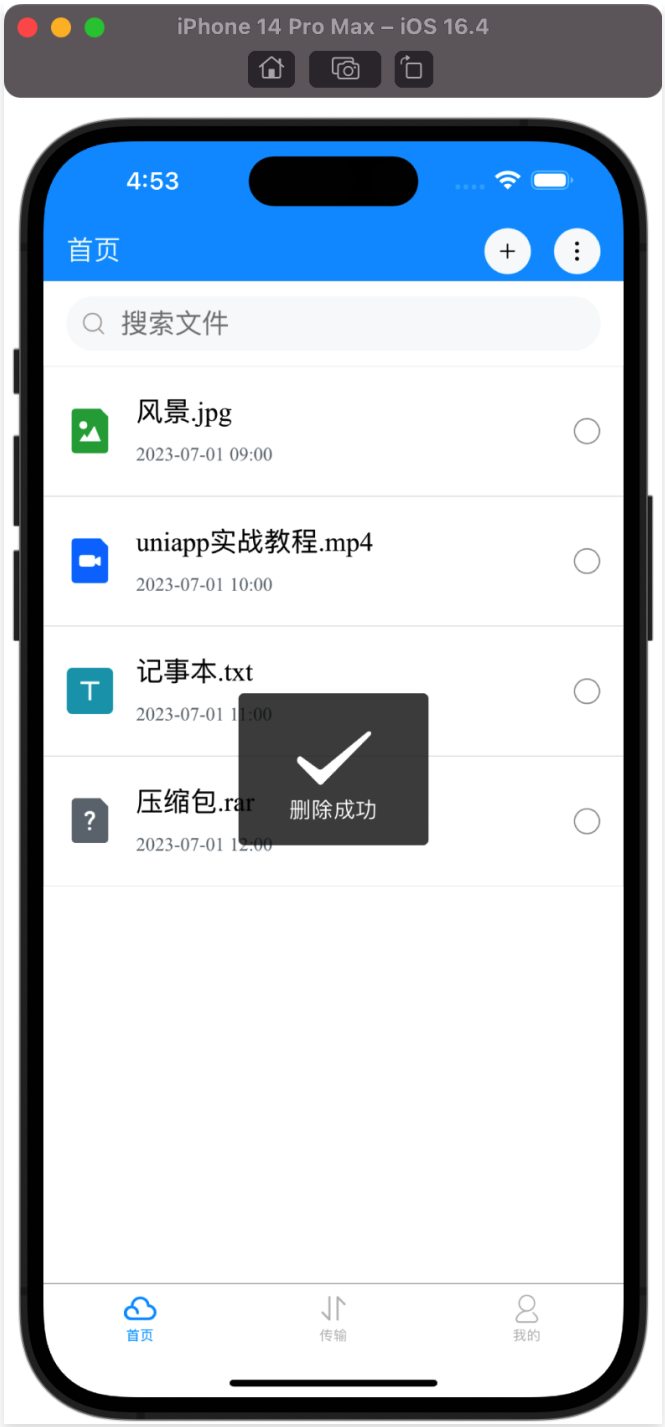
```
<!-- 底部操作条 -->
<!-- 选中元素个数大于0，才会出现操作条 -->
<view v-if="checkedList.length > 0">
    <!-- 设置操作条容器样式：高度、颜色，固定在底部，flex布局，垂直方向为拉升方式 -->
    <view style="height: 115rpx;" class="flex align-stretch bg-main text-white fixed-bottom">
        <!-- 根据操作菜单元素个数等分容器，要么四等分，要么二等分，行高的修改可以让图标和文字之间的距离变得合理，点击变色通过 :hover-class 实现 -->
        <view class="flex-1 flex flex-column align-center justify-center" style="line-height: 1.5;"
            v-for="(item, index) in actions" :key="index" hover-class="bg-hover-primary"
            @click="handleBottomEvent(item)">
            <text class="iconfont" :class="item.icon"></text>
            {{ item.name }}
        </view>
    </view>
</view>
```

- 在 script 部分添加具体的处理事件 handleBottomEvent，根据 item 进行判断，执行不同的操作

* index.vue

```vue
const actions = computed(() => { ··· }

// 获得删除对话框元素
const deleteDialogRef = ref(null)

// 底部操作条处理（根据传入的 item 进行判断，执行不同的操作）
const handleBottomEvent = (item) => {
    switch (item.name) {
        case '删除':
            deleteDialogRef.value.showPopup()
            break;
        default:
            break;
    }
}

const handleDeleteConfirm = () => {
    //对 list 进行过滤，留下未被选中的元素（选中的即被删除）
    list.value = list.value.filter(item => !item.checked);
    uni.showToast({
        title: '删除成功',
        icon: 'success'
    });
}

const handleCancel = () => {
    console.log('取消');
    // 执行取消后的逻辑
}
</script>
```

效果




```
commit 9f1f5fdb40d0a99aeac58d202c703b31c4287972 (HEAD -> main)
Author: mqxu <moqi1977@gmail.com>
Date:   Sun Jul 2 16:58:41 2023 +0800

    弹出层组件封装 + 删除功能
```

重命名功能实现

先在页面中增加一个重命名对话框组件，通过 ref 的值来和删除对话框进行区分，并且为它中间的 slot 插槽（输入框）使用 v-model 绑定重命名的值，注意红框部分代码：

index.vue

```
        <uni-nav-bar v-else> ...

        <!-- 搜索框 -->
        <view class="px-3 py-2"> ...

        <!-- 自定义列表组件 -->
        <f-list v-for="(item, index) in list" :key="index" :item="item" :index="index"
            @my-select="handleSelect(index)"></f-list>

        <!-- 底部操作条 -->
        <!-- 选中元素个数大于0，才会出现操作条 -->
        <view v-if="checkedList.length > 0"> ...

        <!-- 删除对话框 -->
        <f-dialog ref="deleteDialogRef" :onConfirm="handleDeleteConfirm" :onCancel="handleCancel">是否删除选中的文件？</f-dialog>

        <!-- 重命名对话框 -->
        <f-dialog ref="renameDialogRef" :onConfirm="handleRenameConfirm" :onCancel="handleCancel">
            <input type="text" v-model="renameValue" class="flex-1 bg-light rounded px-2" style="height: 95rpx;"
                placeholder="重命名">
        </f-dialog>

    </view>
</template>
```

然后到底部操作条事件中增加“重命名”的 case 分支

index.vue

```
const actions = computed(() => { ...

// 获得删除对话框元素
const deleteDialogRef = ref(null)

// 获得重命名对话框元素
const renameDialogRef = ref(null)

// 底部操作条处理（根据传入的 item 进行判断，执行不同的操作）
const handleBottomEvent = (item) => {
    switch (item.name) {
        case '删除':
            deleteDialogRef.value.showPopup()
            break;
        case '重命名':
            // 重命名只能对单个文件进行，checkedList数组中只有一个选中的元素，checkedList.value[0]
            renameValue.value = checkedList.value[0].name
            renameDialogRef.value.showPopup()
            break;
        default:
            break;
    }
}
```

定义重命名输入框的 v-model 变量，并对重命名的确认操作编写处理函数

index.vue

```
                break;
            }
        }

        const handleDeleteConfirm = () => { ...

        const handleCancel = () => { ...

        // 重命名相关
        const renameValue = ref('')

        const handleRenameConfirm = () => {
            if (renameValue.value === '') {
                return uni.showToast({
                    title: '文件名不能为空',
                    icon: 'none'
                })
            }
            // 更新该元素的 name 值
            checkedList.value[0].name = renameValue.value
            renameDialogRef.value.hidePopup()
        }
    </script>
```

效果


```
commit 8dd3610e4da0d07b1d9d1b0db8ff400e314b28a2 (HEAD -> main)
Author: mqxu <moqi1977@gmail.com>
Date:   Sun Jul 2 17:24:49 2023 +0800

    重命名功能实现
```