9. 图片预览和视频播放功能

图片预览功能

- 首先我们需要给 f-list 组件添加点击事件，通过 emits 回传给父组件

f-list.vue

```
53
54 <template>
55     <!-- 左中右三部分 -->
56     <view class="p-3 flex align-center border bottom-border" @click="emits('click')">
57         <!-- 左侧：用计算属性得出动态样式，显示不同类型的文件图标 -->
58         <text class="iconfont" :class="iconClass" style="font-size:60rpx"></text>
59
60         <!-- 中间：渲染由父组件传入的对象中的名称和时间属性 -->
61         <view class="flex flex-column ml-3" style="line-height: 1.2;">
62             <text class="font-md">{{props.item.name}}</text>
63             <text class="font-sm text-muted mt-2">{{props.item.create_time}}</text>
64         </view>
65
66         <!-- 右侧：根据传入的对象中的checked属性，进行条件渲染 -->
67         <view class="ml-auto flex align-center justify-center" @click="onSelect">
68             <!-- 未选中，画一个灰色的圆圈 -->
69             <text v-if="!props.item.checked" style="width: 30rpx;height: 30rpx;border:1px solid #999;"
70                 class="rounded-circle"></text>
71             <!-- 选中 ，用字体图标 -->
72             <text v-else class="iconfont icon-xuanze-yixuan text-main" style="font-size: 20px;"></text>
73         </view>
74     </view>
75 </template>
```

- 父组件 index.vue 接收到之后，调用 doEvent(item) 方法来处理，根据点击的 item 元素是什么类型文件，进行具体处理

```
282
283         <!-- 自定义列表组件 -->
284         <f-list v-for="(item, index) in list"
285                 :key="index"
286                 :item="item"
287                 :index="index"
288                 @click="doEvent(item)"
289                 @my-select="handleSelect(index)">
290         </f-list>
291
```

- doEvent 事件处理函数，**对于 item 的类型是图片的，使用预览功能**（因为列表中可能有多个图片文件，所以要过滤出所有图片文件）后续需要做视频播放，只要添加 case 分支即可。

index.vue

```
216
217     // 列表点击事件处理(图片预览、视频播放等)
218     const doEvent = (item) => {
219         console.log(item);
220         switch (item.type) {
221             case 'image':
222                 //从 list 中过滤得到所有类型为 image 的文件
223                 let images = list.value.filter(item => {
224                     return item.type === 'image';
225                 });
226                 // 预览图片
227                 uni.previewImage({
228                     current: item.url,
229                     urls: images.map(item => item.url)
230                 });
231                 break;
232             default:
233                 break;
234         }
235     }
236 </script>
```

注意：运行前，我们需要去修改下 list 的数据，让图片类型的文件具有可访问的 url 属性值，如下

index.vue

```vue

    const list = ref([
        {
            type: 'dir',
            name: '我的笔记',
            create_time: '2023-07-01 08:00',
            checked: false
        },
        {
            type: 'image',
            name: '风景.jpg',
            create_time: '2023-07-01 09:00',
            url: 'https://i2.100024.xyz/2023/01/26/3kq106.webp',
            checked: false,
            download: 100
        },
        {
            type: 'image',
            name: '壁纸.jpg',
            create_time: '2023-07-01 09:00',
            url: 'https://i2.100024.xyz/2023/01/27/u9qa4p.webp',
            checked: false,
            download: 90
        },
        {
            type: 'image',
            name: '头像.jpg',
            create_time: '2023-07-01 09:00',
            url: 'https://i2.100024.xyz/2023/01/26/3e727b.webp',
            checked: false,
            download: 80
```

效果


```
commit 8682060863c8ae2a6daad1b74a28e1c6cb6f437d (HEAD -> main)
Author: mqxu <moqi1977@gmail.com>
Date:   Sun Jul 2 18:25:08 2023 +0800

    图片预览功能
```

视频播放功能实现

- 先给 list 数组中的视频文件数据添加可访问的 data 属性

```
{
    type: 'video',
    name: '可爱的海豚.mp4',
    data: 'https://niit-soft.oss-cn-hangzhou.aliyuncs.com/video/3-1.mp4',
    create_time: '2023-07-01 10:00',
    checked: false
},
```

- 新建 video.vue 页面，代码如下

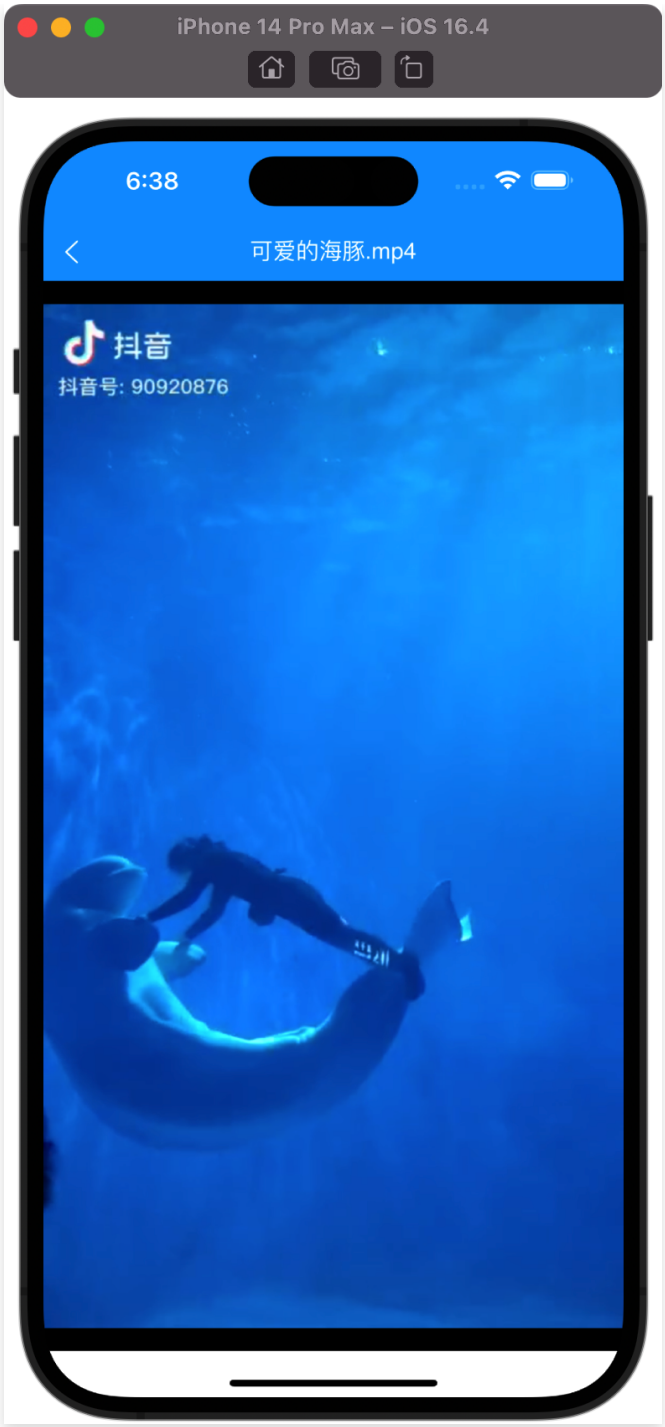
```vue
<script setup>
    import {
        onLoad
    } from '@dcloudio/uni-app'
    import {
        ref
    } from 'vue'
    const url = ref('')

    // 通过 e 对象在两个页面之间传参
    onLoad((e) => {
        //非法地址的处理
        if (!e.url) {
            uni.showToast({
                title: '视频地址非法',
                icon: 'none'
            });
            return uni.navigateBack({
                delta: 1
            });
        }
        url.value = e.url;
        if (e.title) {
            // 把视频文件名作为当前页面的导航标题
            uni.setNavigationBarTitle({
                title: e.title
            });
        }
    })

    // 返回上一层路由
    const back = () => {
        uni.navigateBack({
            delta: 1
        });
    }
</script>

<template>
    <view style="height: 100vh;">
        <!-- 全屏自动播放视频 -->
        <video :src="url" controls autoplay style="width: 750rpx;height: 100vh;" @ended="back"></video>
    </view>
</template>
```

- 处理点击事件，添加一个 case 分支，判断如果类型为 video 的文件，则跳转到 video.vue 页面，**并把视频的地址和名称带过去**

index.vue

```
217
218     // 列表点击事件处理(图片预览、视频播放等)
219     const doEvent = (item) => {
220         console.log(item);
221         switch (item.type) {
222             case 'image':
223                 //从 list 中过滤得到所有类型为 image 的文件
224                 let images = list.value.filter(item => {
225                     return item.type === 'image';
226                 });
227                 // 预览图片
228                 uni.previewImage({
229                     current: item.url,
230                     urls: images.map(item => item.url)
231                 });
232                 break;
233             case 'video':
234                 uni.navigateTo({
235                     url: '../video/video?url=' + item.data + '&title=' + item.name
236                 })
237                 break;
238             default:
239                 break;
240         }
241     }
242 </script>
```

效果，点击视频文件，跳转到 video 页面，并直接全屏播放



注意：加入图片预览和视频播放功能后，我们会发现一个问题：图片和视频类型的文件数据，右侧的圆形选择框，本来点击可以实现选中或取消选中功能，但是因为这个元素是外层整个 view 的子元素，而外层点击之后是实现图片预览和视频播放功能。其他类型的文件不受影响。

如何解决呢?

答案就是阻止事件冒泡。如图，给内层的选择框加上 .stop 事件修饰符，用于阻止事件冒泡，这样就各自独立啦。

```
<view class="flex align-center justify-center" style="width: 70rpx;height: 70rpx;" @click.stop="onSelect">
    <!-- 未选中，画一个灰色的圆圈 -->
    <text v-if="!props.item.checked" style="width: 30rpx;height: 30rpx;border:1px solid #999;"
        class="rounded-circle"></text>
    <!-- 选中 ，用字体图标 -->
    <text v-else class="iconfont icon-xuanze-yixuan text-main" style="font-size: 20px;"></text>
```

阻止事件冒泡

即可解决问题。

```
commit 6583fa06e03c3bace80ddace5fa053ef8f771f68 (HEAD -> main)
Author: mqxu <moqi1977@gmail.com>
Date:   Sun Jul 2 18:39:33 2023 +0800

    视频播放功能实现
```