

16-10 首页交互——上传图片

Vuex 的 state 增加两个变量，用来记录上传任务和下载任务

```
const store = createStore({
  state: {
    username: 'zhangsan',
    user: null,
    token: null,
    uploadList: [],
    downloadList: []
  },

```

actions 中对它们初始化

```
actions: {
  initList({
    state
  }) {
    if (state.user) {
      let d = uni.getStorageSync('download_' + state.user.id)
      let u = uni.getStorageSync('upload_' + state.user.id)

      state.downloadList = d ? JSON.parse(d) : []
      state.uploadList = u ? JSON.parse(d) : []
    }
  },
  logout({
    state
  }) {
    $H.post('/logout', {}, {

```

actions 中再增加两个方法，具体作用看注释

```

index.js
25 |     }
26 |   },
27 |   // 创建一个上传任务
28 |   createUploadJob({
29 |     state
30 |   }, obj) {
31 |     // 添加到上传队列的最前面
32 |     state.uploadList.unshift(obj)
33 |     // 异步设置本地存储，记录键值对为：上传人和上传内容
34 |     uni.setStorage({
35 |       key: 'upload_' + state.user.id,
36 |       data: JSON.stringify(state.uploadList)
37 |     })
38 |   },
39 |   // 更新上传任务进度
40 |   updateUploadJob({
41 |     state
42 |   }, obj) {
43 |     // 在上传队列中查找该用户的上传任务
44 |     let i = state.uploadList.findIndex(item => item.key === obj.key) {
45 |       // 存在
46 |       if (i !== -1) {
47 |         // 更新 progress 属性的值和上传状态的值
48 |         state.uploadList[i].progress = obj.progress
49 |         state.uploadList[i].status = obj.status
50 |         // 异步更新本地存储
51 |         uni.setStorage({
52 |           key: 'upload_' + state.user.id,
53 |           data: JSON.stringify(state.uploadList)
54 |         })
55 |       }
56 |     }
57 |   },
58 |   logout({

```

useIndex.js 增加三个方法

1、上传图片的 case 分支

使用 uniapp 的 API 来访问相册或拍照，最多 9 张，需要循环调用自己封装好的 upload 方法（异步调用，效率还不错），传递需要上传的文件和类型（当前为 image）

```

314 // 处理添加操作条的各种事件
315 const handleAddEvent = (item) => {
316   addPopup.value.close()
317   switch (item.name) {
318     case '新建文件夹':
319       newDirDialogRef.value.showPopup()
320       break;
321     case '上传图片':
322       // 选择图片，限制9张
323       uni.chooseImage({
324         count: 9,
325         success: (res) => {
326           // 选择图片成功，循环异步调用上传接口
327           res.tempFiles.forEach(item => {
328             upload(item, 'image')
329           })
330         }
331       })
332     }
333   }
334 }

```

2、生成一个唯一 ID

```

// 生成唯一 id
const genId = (length) => {
  return Number(Math.random().toString().substr(3, length) + Date.now().toString(36))
}

```

3、上传

```

// 文件上传
const upload = (file, type) => {
  // 上传文件的类型
  let t = type
  // 上传的key, 用来区分每个文件
  const key = genId(8)
  // 构建上传文件对象, 文件名、类型、大小、唯一key、进度、状态、创建时间
  let obj = {
    name: file.name,
    type: t,
    key,
    progress: 0,
    status: true,
    created_time: new Date().getTime()
  }
  // 创建上传任务, 通过 dispatch 分发给 Vuex 的 actions, 异步上传调度, 主要是实现上传进度的回调
  store.dispatch('createUploadJob', obj)
  // 上传, 查询参数为当前位置所在的目录id, body参数为文件路径
  $H.upload('/upload?file_id=' + file_id.value, {
    filePath: file.path
  }, p => {
    // 更新上传进度
    store.dispatch('updateUploadJob', {
      status: true,
      process: p,
      key
    })
  }).then(res => {
    // 上传成功, 重新请求数据
    getList()
  })
}

```

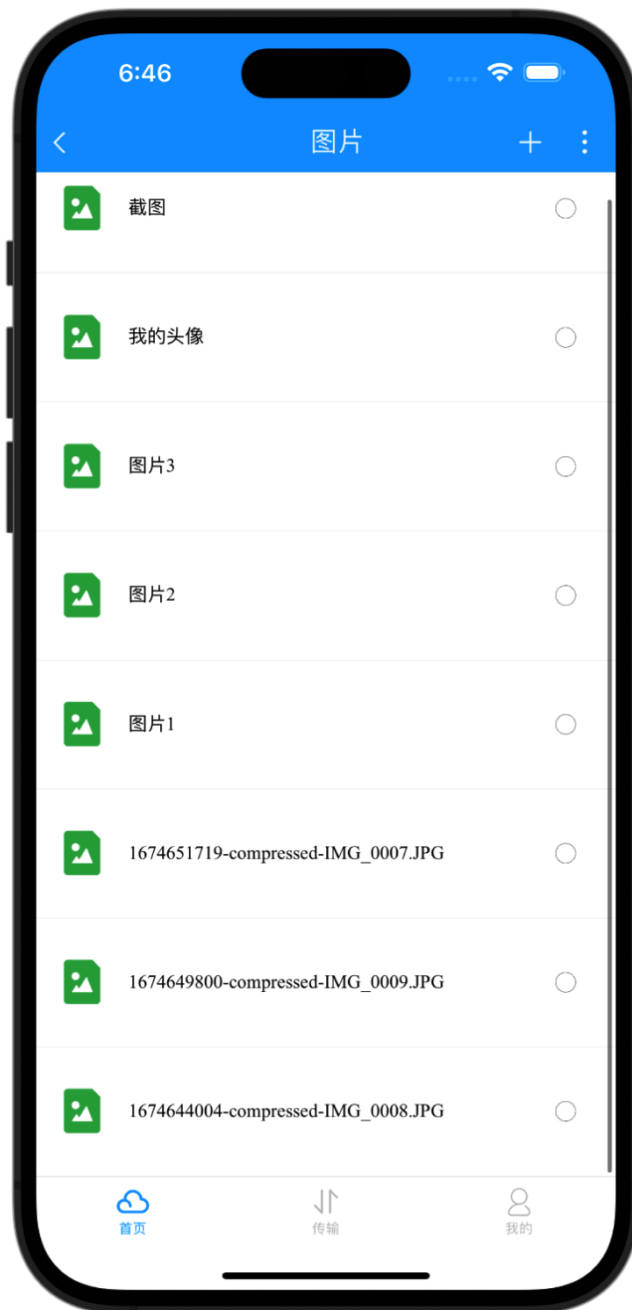
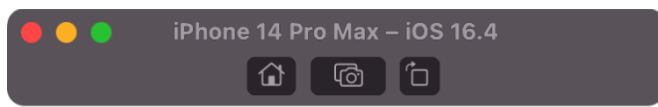
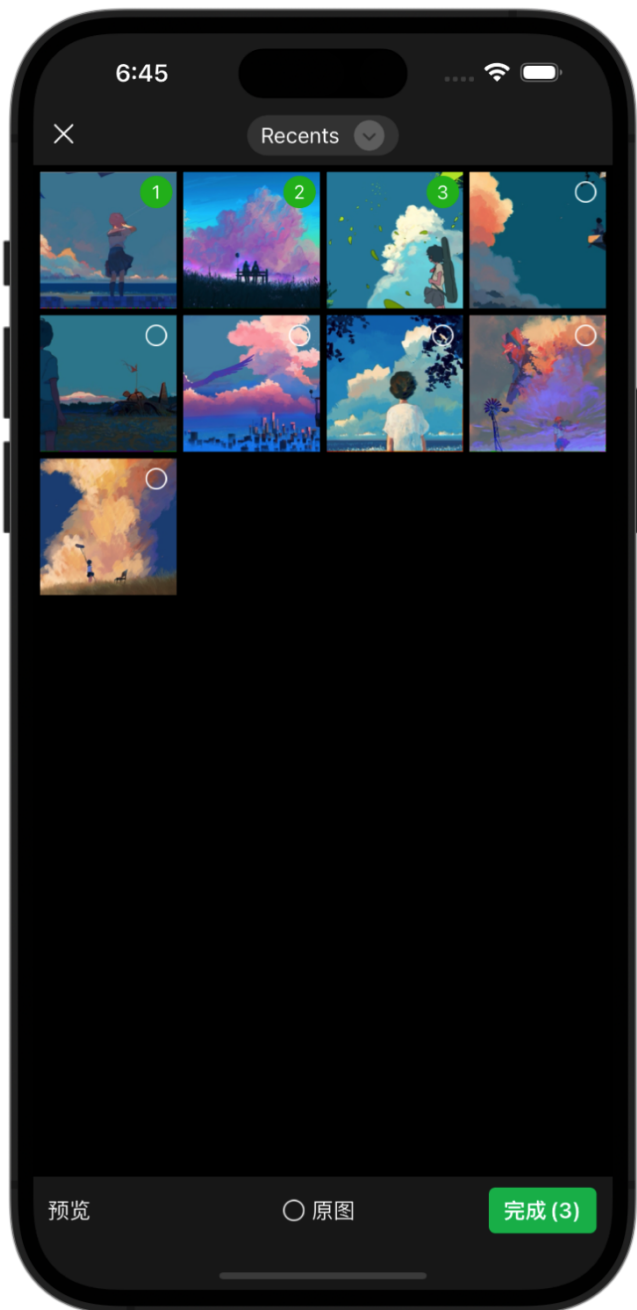
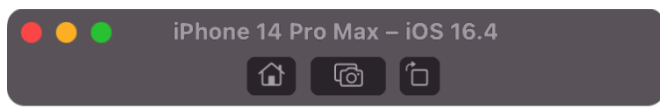
如果文件名太长, 可以把 f-list 组件中显示文件名的 text 组件加个样式, 让它超出长度就显示 ...

```

f-list.vue
56
57 <template>
58   <view>
59     <!-- 左中右 flex -->
60     <view class="p-3 flex align-center border-bottom border-light-secondary" hover-class="bg-light" @click="emits('click')">
61       <!-- 左侧文件图标 -->
62       <text class="iconfont" :class="iconClass" style="font-size: 60rpx;"></text>
63
64       <!-- 中间文件名和创建时间 -->
65       <view class="flex flex-column ml-3" style="line-height: 1.2;">
66         <text class="font-sm text-ellipsis" style="width: 500rpx;">{{ item.name }}</text>
67         <text class="font-sm text-muted">{{ item.create_time }}</text>
68       </view>
69
70       <!-- 右侧, 如果 showRight 属性为true, 则显示右侧的默认插槽 -->
71       <view v-if="props.showRight" class="ml-auto">
72         <slot>
73           <view class="flex align-center justify-center" style="width: 70rpx;height: 70rpx;" @click.stop="onSelect">
74             <text v-if="!props.item.checked" style="height: 25rpx;width: 25rpx;" class="rounded-circle border border-dark"></text>
75             <text v-else class="iconfont icon-xuanze-yixuan text-primary" style="font-size: 40rpx;"></text>
76           </view>
77         </slot>
78       </view>
79     </view>
80
81     <!-- bottom 具名插槽 -->
82     <slot name="bottom"></slot>
83   </view>
84 </template>

```

测试图片上传（支持多图）



```
commit 197af89934f4a94cb168a5cc1e80c43c993cc343 (HEAD -> main)
Author: mxqu <moqi1977@gmail.com>
Date: Mon Jul 10 18:47:25 2023 +0800
```

图片上传