



浙大城市学院

ZHEJIANG UNIVERSITY CITY COLLEGE

Vue 实现步骤

课程编号： C01098

课程名称： 移动互联网应用开发实践

学 号： 31901049

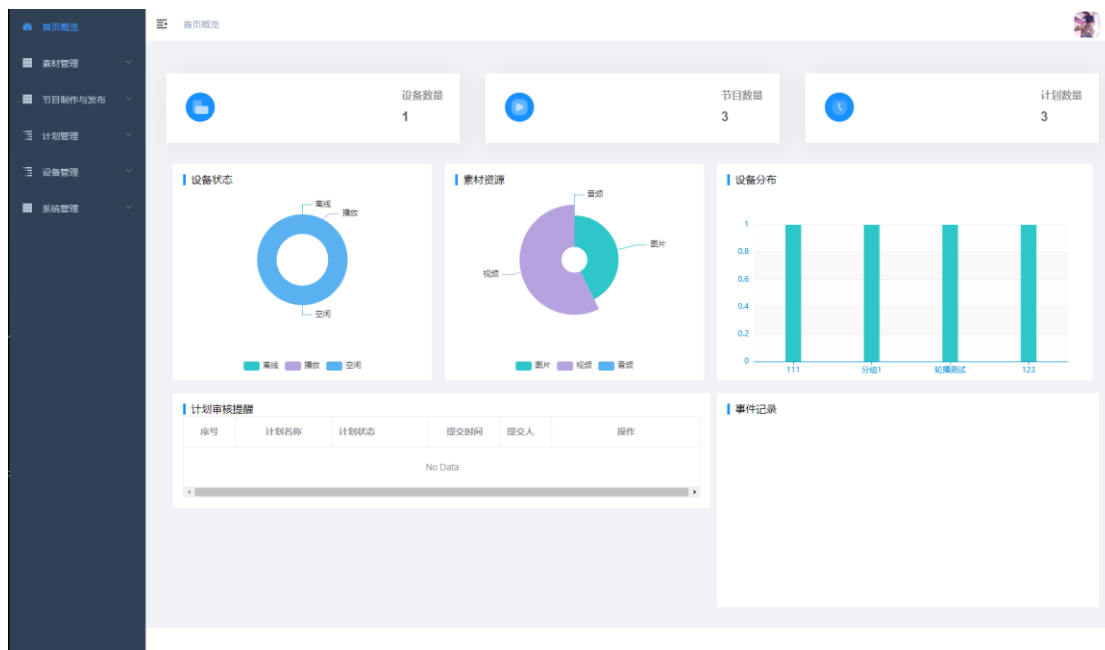
姓 名： 潘振宇

专业班级： 计算 1902

所在学院： 计算学院

报告日期： 2022 年 7 月 7 日

一、echarts:



素材占比饼图代码 setOption

```
this.chart.setOption({
  tooltip: {
    trigger: 'item',
    formatter: '{a} <br/>{b} : {c} '+'MB'+ '({d}%)'
  },
  legend: {
    left: 'center',
    bottom: '10',
    data: ['图片', '视频', '音频']
  },
  series: [
    {
      name: '素材资源',
      type: 'pie',
      roseType: 'radius',
      radius: [20, 85],
      center: ['50%', '38%'],
      data: [
        { value: this.int1, name: '图片' },
        { value: this.int2, name: '视频' },
        { value: this.int3, name: '音频' }
      ],
      animationEasing: 'cubicInOut',
      animationDuration: 2600
    }
  ]
})
```

```
]
})
```

二、设备数量饼图

```
this.chart.setOption({
  tooltip: {
    trigger: 'item',
    formatter: '{a} <br/>{b} : {c} ({d}%)'
  },
  legend: {
    left: 'center',
    bottom: '10',
    data: ['离线', '播放', '空闲']
  },
  series: [
    {
      name: '设备数量',
      type: 'pie',
      roseType: 'radius',
      radius: [40, 70],
      center: ['50%', '38%'],
      data: [
        { value: this.int1, name: '离线' },
        { value: this.int2, name: '播放' },
        { value: this.int3, name: '空闲' }
      ],
      animationEasing: 'cubicInOut',
      animationDuration: 2600
    }
  ]
})
```

每一个都是单独的组件，在首页中 import 引用

```
import PanelGroup from './components/PanelGroup'
import DoughnutChart from './components/DoughnutChart'
import PieChart from './components/PieChart'
import ColumnChart from './components/ColumnChart.vue'
import PlanForm from './components/PlanForm.vue'
import FormRecord from './components/FormRecord.vue'
```

axios 封装（拦截器） 设置响应时长为 **1000ms**，因为有些接口返回参数需要时间比较久

```
1 import axios from 'axios'
2 import { MessageBox, Message } from 'element-ui'
3 import store from '@/store'
4 import { getToken } from '@/utils/auth'
5
6 // create an axios instance
7 const service = axios.create({
8   baseURL: '/api', // url = base url + request url
9   // withCredentials: true, // send cookies when cross-domain requests
10  timeout: 10000 // request timeout
11 })
12
13 // request interceptor
14 service.interceptors.request.use(
15   config => {
16     // do something before request is sent
17     if (store.getters.token) {
18       // let each request carry token
19       // ['X-Token'] is a custom headers key
20       // please modify it according to the actual situation
21       config.headers['X-Token'] = getToken()
22     }
23     return config
24   },
```

三、cookie 存 token

```
请求标头
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: keep-alive
Cookie: Idea-7c7d8a8c=758ffccd-0c42-4886-8485-9bf35b34342a; NMTID=000HhMbOPnu35E5AEUVtt2VnhP1bxkAAAGAhM49pQ; sidebarStatus=1; boetoken=8eefb0b9-f97f-494e-bfa9-a8acc433302; vue_admin_template_token=8eefb0b9-f97f-494e-bfa9-a8acc433302
Host: localhost:9528
Referer: http://localhost:9528/
sec-ch-ua: ".Not(A)Brand";v="99", "Google Chrome";v="103", "Chromium";v="103"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
```

代码实现：设置 token

```
const actions = {
  // user login
  login({ commit }, userInfo) {
    const { username, userpwd } = userInfo
    return new Promise((resolve, reject) => {
      login({ username: username.trim(), userpwd: userpwd }).then(response
=> {
```

```

    const { data } = response
    commit('SET_TOKEN', data)
    setToken(data)
    resolve()
  }).catch(error => {
    reject(error)
  })
})
},

```








四、素材列表

Dashboard / 素材管理 / 素材列表

素材名称:

重置 查询

上传素材 批量删除

<input type="checkbox"/>	缩略图	素材名称	分辨率	大小	作者	更新时间	操作
<input type="checkbox"/>		bg2.jpg	1366*768	38.79KB	morri	2022-07-04 14:04:31	预览 重命名 删除
<input type="checkbox"/>		video1.mp4	无	68.94KB	morri	2022-07-04 14:11:22	预览 重命名 删除
<input type="checkbox"/>		bg3.jpg	720*405	12.47KB	morri	2022-07-04 14:11:48	预览 重命名 删除
<input type="checkbox"/>		lol.png	1920*1080	755.18KB	admin	2022-07-05 09:59:40	预览 重命名 删除
<input type="checkbox"/>		background.jpg	720*497	53.77KB	admin	2022-07-05 14:00:01	预览 重命名 删除
<input type="checkbox"/>		lolm.jpg	2400*1080	954.33KB	admin	2022-07-05 14:01:00	预览 重命名 删除
<input type="checkbox"/>		kobe.mp4	无	16.19MB	admin	2022-07-05 16:10:48	预览 重命名 删除

五、顶部搜索框实现

```

<div class="form-container">
  <el-row>
    <el-col :span="8">
      <el-row>
        <span>素材名称: </span>
        <el-input
          v-model="input1"
          placeholder="请输入素材名称"
        >

```

```

        suffix-icon="el-icon-search"
        size="medium"
        class="input"
      />
    </el-row>
  </el-col>
  <el-col :span="8">
    <el-row type="flex" justify="end">
      <div>
        <el-button plain>重置</el-button>
        <el-button type="primary">查询</el-button>
      </div>
    </el-row>
  </el-col>
</el-row>
</div>

```

六、上传素材按钮和批量删除按钮

```

<div class="btn-container">
  <el-row type="flex" justify="end">
    <el-button type="primary" @click="centerDialogVisible = true">
      >上传素材</el-button>
    >
    <el-button>批量删除</el-button>
  </el-row>
</div>

```

七、表格实现（v-slot）

如果我们想要插入元素到指定的位置需要借助 **slot**, 不然的话将插入失败, 当一个组件中具有多个插槽的时候, 如果没有借助 **slot** 属性会默认把元素插入到没有设置 **name** 属性值的 **slot** 组件中, 当然, 会把元素在每一个没有 **name** 属性的插槽中都插入一遍。

```

<el-table
  :key="tableKey"
  v-loading="listLoading"
  :data="list"
  border

```

```

fit
highlight-current-row
style="width: 100%"
>
<el-table-column type="selection" width="55" />
<el-table-column label="缩略图" prop="murl" align="center"
width="80">
  <template v-slot="{ row }">
    
    
  </template>
</el-table-column>
<el-table-column label="素材名称" align="center">
  <template v-slot="{ row }">
    <span>{{ row.mname }}</span>
  </template>
</el-table-column>
<el-table-column label="分辨率" align="center">
  <template v-slot="{ row }">
    <span>{{ row.dpi }}</span>
  </template>
</el-table-column>
<el-table-column label="大小" align="center">
  <template v-slot="{ row }">
    <span>{{ row.msize }}</span>
  </template>
</el-table-column>
<el-table-column label="作者" align="center">
  <template v-slot="{ row }">
    <span>{{ row.author }}</span>
  </template>
</el-table-column>
<el-table-column label="更新时间" align="center">
  <template v-slot="{ row }">
    <span>{{ row.updatetime }}</span>
  </template>
</el-table-column>
<el-table-column
  label="操作"
  align="center"
  width="230"
  class-name="small-padding fixed-width"
>
  <template v-slot="{ row, $index }">

```

```

        <el-button type="text" size="mini"
@click="handlePreview(row.murl)">
            预览
        </el-button>
        <el-button type="text" size="mini"
@click="handleRename(row.mname)">
            重命名
        </el-button>
        <el-button type="text" size="mini"
@click="handleDelete(row.mname)">
            删除
        </el-button>
    </template>
</el-table-column>
</el-table>

```

引入分页器

```

<pagination
v-show="total>0" :total="total" :page.sync="listQuery.page" :limit.sync
="listQuery.limit"
@pagination="getList"
/>

```

八、上传素材（判断文件后缀名）

```

<div class="list-img">
    <ul>
        <li v-for="(item, index) in imgArr" :key="index">
            
            <a @click="todel(index)"></a>
        </li>
        <li>
            <div class="addImg" v-if="imgArr.length < 3">
                <span class="add">上传素材</span>
                <input
                    type="file"
                    @change="getFile($event)"
                    accept=".jpg,.png,.bmp,.gif,.mp4,.mp3"
                />
            </div>
        </li>
    </ul>

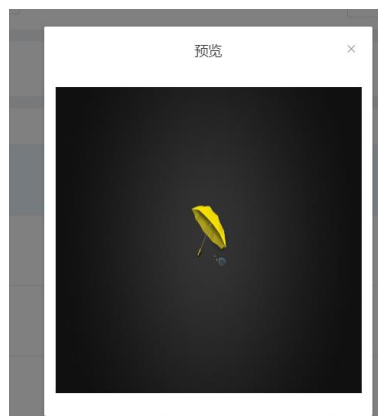
```



```
    </li>
  </ul>
</div>
```

```
getFile(event) {
  var file = event.target.files;
  for (var i = 0; i < file.length; i++) {
    // 上传类型判断
    var imgName = file[i].name;
    var idx = imgName.lastIndexOf(".");
    if (idx != -1) {
      var ext = imgName.substr(idx + 1).toUpperCase();
      ext = ext.toLowerCase();
      if (
        ext != "jpg" &&
        ext != "png" &&
        ext != "bmp" &&
        ext != "gif" &&
        ext != "mp4" &&
        ext != "mp3"
      ) {
      } else {
        this.imgArr.push(file[i]);
      }
    } else {
    }
  }
},
```

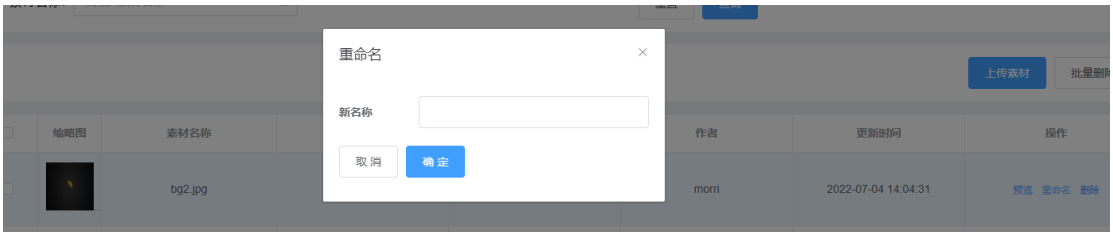
九、预览素材



代码实现（动态绑定：**src**）

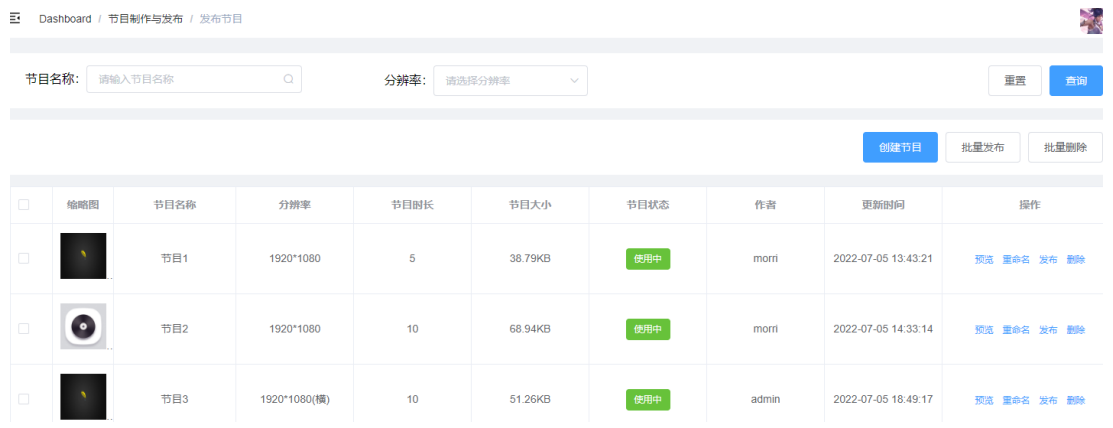
```
<el-dialog title="预览" :visible.sync="dialogVisible" width="25%" center>
  <div class="pic11">
    
  </div>
</el-dialog>
```

十、重命名（需要给后端传入原来的名称和新名称）



```
<el-dialog title="重命名" :visible.sync="dialogVisible1" width="25%">
  <el-form :model="form" label-position="left" label-width="100px">
    <el-form-item label="新名称">
      <el-input v-model="form.new_name"></el-input>
    </el-form-item>
  </el-form>
  <span>
    <el-button @click="dialogVisible1 = false">取 消</el-button>
    <el-button type="primary" @click="rename">确 定</el-button>
  </span>
</el-dialog>
```

十一、节目列表（实现同素材列表）



十二、新建节目

（包括名称，分辨率，（如果是图片）每张图片的时长，选择素材列表）
后端还需要 **token** 和素材种类，按他的要求传入即可，
获取 **token** 代码（先用字符串分割成数组，再从数组取字符串切片）。

```
if (document.cookie.length > 0) {  
    let arr = document.cookie.split("; ");  
    for (let i = 0; i < arr.length; i++) {  
        let index = arr[i].indexOf("=");  
        if (arr[i].substring(0, index) === "boetoken") {  
            this.token = arr[i].substring(index + 1);  
            break;  
        }  
    }  
}
```

设置了一个 **imgArr[]** 数组，为了能一次传多张素材（**forEach**）

```
this.imgArr.forEach((file) => {  
    fd.append("file", file);  
    uploadExample(fd, this.token).then((response) => {  
        console.log(response);  
        if (response.rspCode == "200") {  
            this.$message({  
                message: "上传成功",  
                type: "success",  
            });  
            this.centerDialogVisible = false;  
        } else {  
            this.$message({  
                message: "上传失败",  
                type: "error",  
            });  
        }  
        this.imgArr = [];  
    });  
});
```

新建节目

×







名称

分辨率

请选择dpi

图片时长

5

<input type="checkbox"/>	缩略图	素材名称	分辨率	素材大小	素材种类
<input type="checkbox"/>		bg2.jpg	1366*768	38.79KB	图片
<input type="checkbox"/>	FAILED	video1.mp4	无	68.94KB	视频
<input type="checkbox"/>		bg3.jpg	720*405	12.47KB	图片
<input type="checkbox"/>		lol.png	1920*1080	755.18KB	图片
<input type="checkbox"/>		background.jpg	720*497	53.77KB	图片
<input type="checkbox"/>		lolm.jpg	2400*1080	954.33KB	图片
<input type="checkbox"/>	FAILED	kobe.mp4	无	16.19MB	视频
<input type="checkbox"/>	FAILED	douyin.mp4	无	4.47MB	视频
<input type="checkbox"/>		grape.jpg	600*400	16.88KB	图片

取消

创建

十三、公告列表（实现同理）

Dashboard / 节目制作与发布 / 发布公告

公告状态: 请选择公告状态							重置	清除
							新增公告	批量删除
<input type="checkbox"/>	公告内容	发布时间	公告状态	作者	创建时间	操作		
<input type="checkbox"/>	hello world	2022-07-03 22:51:23	发布成功	zdf	2022-06-29 15:12:41	详情 编辑 发布 删除		
<input type="checkbox"/>	helloworld	2022-07-03 15:21:00	发布成功	moni	2022-07-03 23:07:02	详情 编辑 发布 删除		
<input type="checkbox"/>	helloworld	2022-07-03 22:06:15	发布成功	moni	2022-07-03 23:07:37	详情 编辑 发布 删除		

十四、创建公告（传入内容和时间即可，同样的办法获取 **token** 来知道作者）



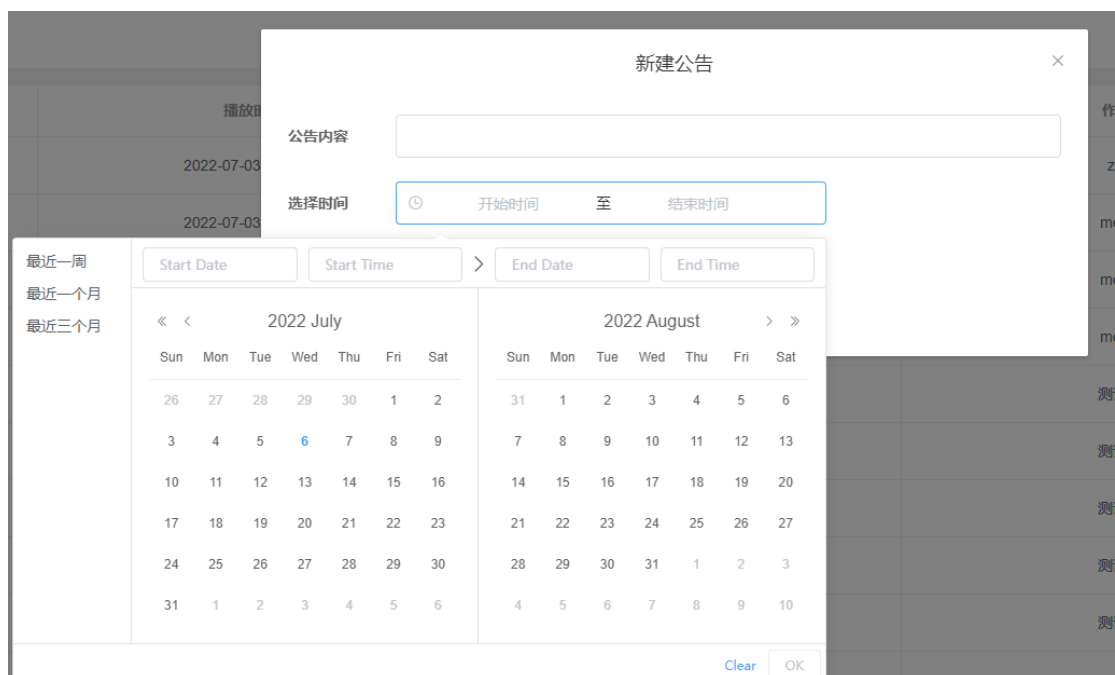
时间选择器（**element-ui** 的组件，包装了一下）

这里遇到了时间格式转换的问题，用了 **moment** 转换成 **yy-mm-dd HH:mm:ss** 格式

先 **npm install moment**

再 **import moment from 'moment';**

```
this.form.start_time=moment(this.value1[0]).format("YYYY-MM-DD  
HH:mm:ss");  
this.form.finish_time=moment(this.value1[1]).format("YYYY-MM-DD  
HH:mm:ss")
```



十五、计划列表（同理）

Dashboard / 计划管理

计划名称:

计划状态:

请选择

重置

筛选

新增计划

批量删除

<input type="checkbox"/>	缩略图	计划名称	计划状态	播放模式	播放日期	作者	审核人	更新时间	操作
<input type="checkbox"/>		演示计划1	已发布	按时段播放	2022-07-05-2022-07-05	admin	admin	2022-07-05 16:13:04	详情 修改 删除
<input type="checkbox"/>		演示计划之重复2	已发布	按时段播放	2022-07-05-2022-07-05	admin	admin	2022-07-05 16:23:40	详情 修改 删除
<input type="checkbox"/>		录屏视频计划1	已发布	按时段播放	2022-07-05-2022-07-05	admin	admin	2022-07-05 16:31:24	详情 修改 删除

十六、发布计划

这边选项绑定值 **0** 或者 **1**。后端需要两个时间，所以弄了一个日期选择器和时间选择器。

Dashboard / 计划管理

发布计划

* 计划名称

播放策略

☐ 替换 ☐ 插播

播放模式

☐ 时段播放 ☐ 持续播放（打开此功能后会使得播放同一计划的设备画面保持统一）

多屏同步

☐ 打开 ☐ 关闭

* 播放日期

开始日期

至

结束日期

添加时间段

添加节目

取消

保存

下一步

设置播放时间段

设置播放时间段

* 循环时间段

08:40:00

至

09:40:00

* 循环周期

每天

每天

每周

每月

确认

取消

十七、添加节目（选素材，这边获取一个列表，绑定 list 为素材 list）

添加节目

节目名称:

分辨率:

请选择

重置

查询

<input type="checkbox"/>	缩略图	节目名称	分辨率	节目时长	节目大小
<input type="checkbox"/>		节目1	1920*1080	5	38.79KB
<input type="checkbox"/>	FAILED	节目2	1920*1080	10	68.94KB
<input type="checkbox"/>		节目3	1920*1080(竖)	10	51.26KB
<input type="checkbox"/>		节目4	1920*1080(竖)	10	767.65KB
<input type="checkbox"/>		节目5	1920*1080(竖)	5	53.77KB
<input type="checkbox"/>		演示1	1920*1080(竖)	10	1.67MB
<input type="checkbox"/>	FAILED	演示2	1920*1080(竖)	5	16.19MB

返回

保存

十八、设备列表（列表同理）

Dashboard / 设备管理 / 设备列表

设备名称:

所属机构:

请选择

所属分组:

请选择

MAC地址:

当前计划:

请选择

设备状态:

请选择

重置

查询

设备名称	所属机构	所属分组	MAC地址	设备状态	固件升级	操作
设备1	福建罗永浩测试	测试分组1	A00000000000	在线	有新版本	<div><div>详情</div><div>删除</div><div>重置</div><div>禁用</div></div>

Total 1

10/page

Go to: 1

十九、设备详情（对话框内一个 el-tabs，里面三个 pane）

```
<el-dialog title="设备详情" :visible.sync="dialogFormVisible">
  <el-tabs v-model="activeName" type="card">
```

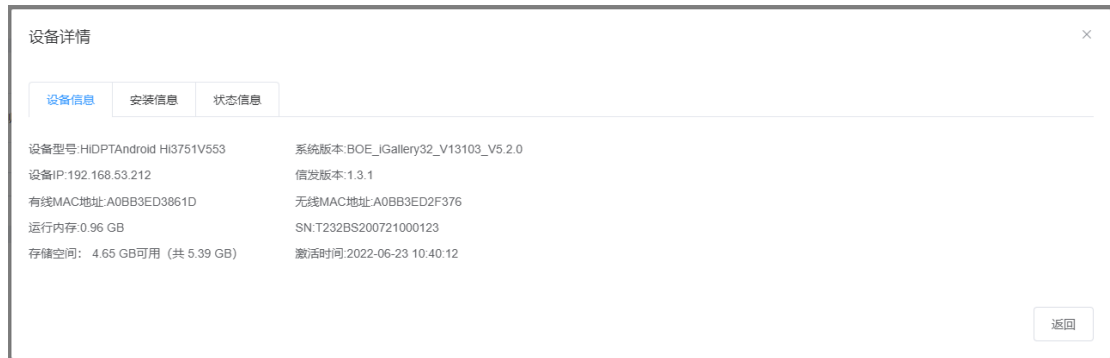
```
<el-tab-pane label="设备信息" name="first">
  <div class="container">
    <div class="e-left">
      <p>设备型号:HiDPTAndroid Hi3751V553</p>
      <p>设备 IP:192.168.53.212</p>
      <p>有线 MAC 地址:A0BB3ED3861D</p>
      <p>运行内存:0.96 GB</p>
      <p>存储空间: 4.65 GB 可用 (共 5.39 GB) </p>
    </div>
    <div class="e-right">
      <p>系统版本:BOE_iGallery32_V13103_V5.2.0</p>
      <p>信发版本:1.3.1</p>
      <p>无线 MAC 地址:A0BB3ED2F376</p>
      <p>SN:T232BS200721000123</p>
      <p>激活时间:2022-06-23 10:40:12</p>
    </div>
  </div>
</el-tab-pane>
<el-tab-pane label="安装信息" name="second">
  <div class="container">
    <div class="e-left">
      <p>设备名称: 测试误删</p>
      <p>所属分组: 11111</p>
      <p>所属机构: 城院罗老师测试</p>
      <p>安装位置: 中国浙江省杭州市拱墅区上塘街道东教路</p>
    </div>
    <div class="e-right">
      <p>注册时间: 2022-06-27 14:31:14</p>
      <p>分辨率: 1920x1080</p>
      <p>屏显方式: 横屏</p>
    </div>
  </div>
</el-tab-pane>
<el-tab-pane label="状态信息" name="third">
  <div class="container">
    <div>
      <p>设备状态: 离线</p>
      <p>当前计划: --</p>
      <p>设备运行时长: 9 小时 52 分钟</p>
      <p>最后心跳时间: 2022-06-27 16:01:20</p>
    </div>
  </div>
</el-tab-pane>
</el-tabs>
```



```

<div slot="footer" class="dialog-footer">
  <el-button @click="dialogFormVisible = false"> 返回 </el-button>
</div>
</el-dialog>

```



二十、设备控制

（写按钮和调节滑块来实现）

系统升级按钮需要从数据库取数据来判断是否有新版本，有新版本按钮才能按；

否则就是 **disable**。

```

<el-tab-pane label="系统升级">
  <button v-if="`${list[0].d_update}` == '已是最新' disabled">
    检查更新
  </button>
  <button v-else @click="checkupdate">检查更新</button>
</el-tab-pane>

```

远程控制

×

当前设备：测试误删

屏幕截图

定时开关机

重启系统

音量控制

亮度控制

系统升级

开始截屏

返回

当前设备：测试误删

屏幕截图

定时开关机

重启系统

音量控制

亮度控制

系统升级

🕒 起始时间

🕒 结束时间

发送定时任务

返回

当前设备：测试误删

屏幕截图

定时开关机

重启系统

音量控制

亮度控制

系统升级

重启

返回

远程控制

×

当前设备：测试误删

屏幕截图

定时开关机

重启系统

音量控制

亮度控制

系统升级

音量:20

确定

返回

远程控制

×

当前设备：测试误删

屏幕截图

定时开关机

重启系统

音量控制

亮度控制

系统升级

亮度:30

确定

返回

远程控制

当前设备: 测试吴刚

屏幕截图

定时开关机

重启系统

音量控制

亮度控制

系统升级

检查更新

返回

二十一、分组管理（列表）

Dashboard / 设备管理 / 分组管理

分组名称: 请输入分组名称

所属机构: 请选择

重置

新增

新增分组

<input type="checkbox"/>	分组ID	分组名称	所属机构	描述	操作
<input type="checkbox"/>	2	测试分组1	城院罗老师测试	11	<div>详情</div> <div>编辑</div> <div>删除</div>
<input type="checkbox"/>	3	测试分组2	城院罗老师测试		<div>详情</div> <div>编辑</div> <div>删除</div>

Total 2

10/page

1

Go to: 1

对话框里面套对话框

分组详情

分组名称: 测试分组1

所属机构: 城院罗老师测试

描述: 11

设备选择

设备名称	MAC地址	所属分组	所属机构	设备状态
设备1	A0B83E0F376	测试分组1	城院罗老师测试	<div>在线</div>

返回

新建就是把绑定的值改成input，就可以输入了（编辑也是，只不过编辑是带原数据的）。

编辑分组

分组名称:

所属机构: 请选择

描述:

选择设备

<input type="checkbox"/>	设备名称	MAC地址	所属分组	分辨率	所属机构	当前计划	设备状态
<input type="checkbox"/>	888	888	群	1920*1080		-	<div>在线</div>

Total 2

10/page

1

Go to: 1

取消

保存

二十二、用户列表（状态采用 tag 不同状态 更显眼）

Dashboard / 系统管理 / 用户管理

用户名: 请输入用户名

所属机构: 请选择机构名称

重置

新增

新增用户

用户名	所属机构	所属角色	状态	真实姓名	手机号	邮箱	注册时间	操作
测试1	城院罗老师测试	罗老师	<div>禁用</div>	胡德胡德	13806	1880@qq.com	2022-06-28 22:18:37	<div>禁用</div> <div>启用</div> <div>删除</div>
测试2	城院罗老师测试	罗老师	<div>已禁用</div>	mmmm	999	999@qq.com	2022-06-27 08:57:12	<div>禁用</div> <div>启用</div> <div>删除</div>
zsf	城院罗老师测试	罗老师	<div>启用</div>	胡德	88000000	12222@qq.com	2022-06-27 10:36:57	<div>禁用</div> <div>启用</div> <div>删除</div>
测试3	城院罗老师测试	罗老师	<div>禁用</div>	mmmm	12226	11@qq.com	2022-06-30 09:39:18	<div>禁用</div> <div>启用</div> <div>删除</div>
aaa	城院罗老师测试	罗老师	<div>已禁用</div>	胡德胡德	111111	111111@qq.com	2022-06-28 22:18:46	<div>禁用</div> <div>启用</div> <div>删除</div>
admin	城院罗老师测试	罗老师	<div>启用</div>	胡德	138888	123@qq.com	2022-06-30 09:38:54	<div>禁用</div> <div>启用</div> <div>删除</div>
pyy	城院罗老师测试	罗老师	<div>启用</div>	潘后平	1836881325	786313762@qq.com	2022-07-06 00:27:09	<div>禁用</div> <div>启用</div> <div>删除</div>
mom	城院罗老师测试	罗老师	<div>启用</div>	张万菊	111	123@qq.com	2022-07-01 11:23:39	<div>禁用</div> <div>启用</div> <div>删除</div>

二十三、编辑用户（带原数据，而且有密码格式判断）

编辑用户

* 账户名

admin

* 密码

请输入密码

所属机构

城院罗老师测试

所属角色

罗老师

账号状态

启用

真实名字

哈哈

* 邮箱

123@qq.com

* 手机号

138666

取消

确定

编辑和新建采用同一个对话框

```
textMap: {
  update: "编辑用户",
  create: "新建用户",
},
```

```
<el-dialog :title="textMap[dialogStatus]" :visible.sync="dialogFormVisible">
  <el-form
    ref="dataForm"
    :model="form"
    :rules="rules"
    label-position="left"
    label-width="100px"
  >
    <el-form-item label="账户名" prop="username">
      <el-input v-model="form.username"> </el-input>
    </el-form-item>
    <el-form-item label="密码" prop="userpwd">
      <el-input
        placeholder="请输入密码"
        v-model="form.userpwd"
        show-password
      ></el-input>
    </el-form-item>
    <el-form-item label="所属机构" prop="organ">
      <el-select v-model="form.organ" placeholder="请选择所属机构">
```

```

        <el-option
          label="城院罗老师测试"
          value="城院罗老师测试"
        ></el-option>
      </el-select>
    </el-form-item>
    <el-form-item label="所属角色" prop="userrole">
      <el-select v-model="form.userrole" placeholder="请选择所属角色">
        <el-option label="罗老师" value="罗老师"></el-option>
      </el-select>
    </el-form-item>
    <el-form-item label="账号状态" prop="ustatus">
      <el-select v-model="form.ustatus" placeholder="请选择所属角色">
        <el-option label="启用" value="启用"></el-option>
        <el-option label="停用" value="停用"></el-option>
      </el-select>
    </el-form-item>
    <el-form-item label="真实名字" prop="realname">
      <el-input v-model="form.realname"></el-input>
    </el-form-item>
    <el-form-item label="邮箱" prop="useremail">
      <el-input v-model="form.useremail"></el-input>
    </el-form-item>
    <el-form-item label="手机号" prop="phone">
      <el-input v-model="form.phone"></el-input>
    </el-form-item>
  </el-form>
  <div slot="footer" class="dialog-footer">
    <el-button @click="dialogFormVisible = false">取 消</el-button>
    <el-button
      type="primary"
      @click="dialogStatus === 'create' ? createData() : updateData()"
    >确 定</el-button>
  >
</div>
</el-dialog>

```

通过按不同的按钮修改 `textMap` 的值，从而显示不同的内容。