

山东大学

SHANDONG UNIVERSITY



家教管理模块开发文档 V1.2

课程名称：项目实训

姓名：_____

学号：_____

专业班级：21软件工程1班

授课教师：张亚峰

2024 年 7 月 11 日

目录

家教管理模块开发文档

- 一、概述
- 二、功能列表
- 三、技术栈
- 四、目录结构
- 五、关系图
- 六、代码详情
- 七、组件说明
- 八、主要方法
- 九、API说明
- 十、使用指南
- 十一、注意事项
- 十二、总结

家教管理模块开发文档

一、概述

该组件使用了Vue.js框架，结合ant-design-vue组件库，确保界面的美观性和一致性。通过该组件，管理员可以轻松地管理教师的基本信息，包括教师的姓名、分类、标签、封面图片、简介、价格、手机号、年龄、性别、所在地区和状态等。

二、功能列表

- 教师信息展示：在表格中显示教师的基本信息，包括姓名、价格、性别、年龄、地区、简介和状态等。
- 搜索功能：通过关键词搜索教师信息，支持模糊搜索，快速找到所需的教师。
- 新增教师：通过弹窗表单，管理员可以新增教师信息，填写相关字段并提交。
- 编辑教师：通过表格中的编辑按钮，管理员可以打开弹窗表单，编辑现有教师的信息。
- 删除教师：支持单个删除和批量删除操作，确保教师信息的及时更新。
- 图片上传：支持上传教师封面图片，并预览上传的图片。

（一）详细功能描述

- 教师信息展示：
 - 使用ant-design-vue的a-table组件，实现教师信息的展示。
 - 表格中包括教师的序号、姓名、价格、性别、年龄、地区、简介和状态等信息。
 - 提供编辑和删除操作按钮。
- 搜索功能：
 - 使用输入框输入关键词，实时更新搜索结果。
 - 通过listApi接口获取符合搜索条件的教师数据，并更新表格展示。
- 新增教师：
 - 点击新增按钮，弹出包含各项教师信息字段的表单。

- 表单字段包括教师的姓名、分类、标签、封面、简介、价格、手机号、年龄、性别、所在地区和状态等。
 - 支持封面图片的上传和预览。
- 编辑教师：
 - 点击表格中的编辑按钮，弹出包含现有教师信息的表单。
 - 可以编辑教师的各项信息，并保存修改。
- 删除教师：
 - 支持单个删除，通过表格中的删除按钮删除指定教师。
 - 支持批量删除，通过勾选表格中的多条记录，并点击批量删除按钮进行删除。
- 图片上传：
 - 使用Upload组件实现封面图片的上传。
 - 支持图片预览功能，方便管理员查看上传的图片。

三、技术栈

- Vue 3
- TypeScript
- Ant Design Vue

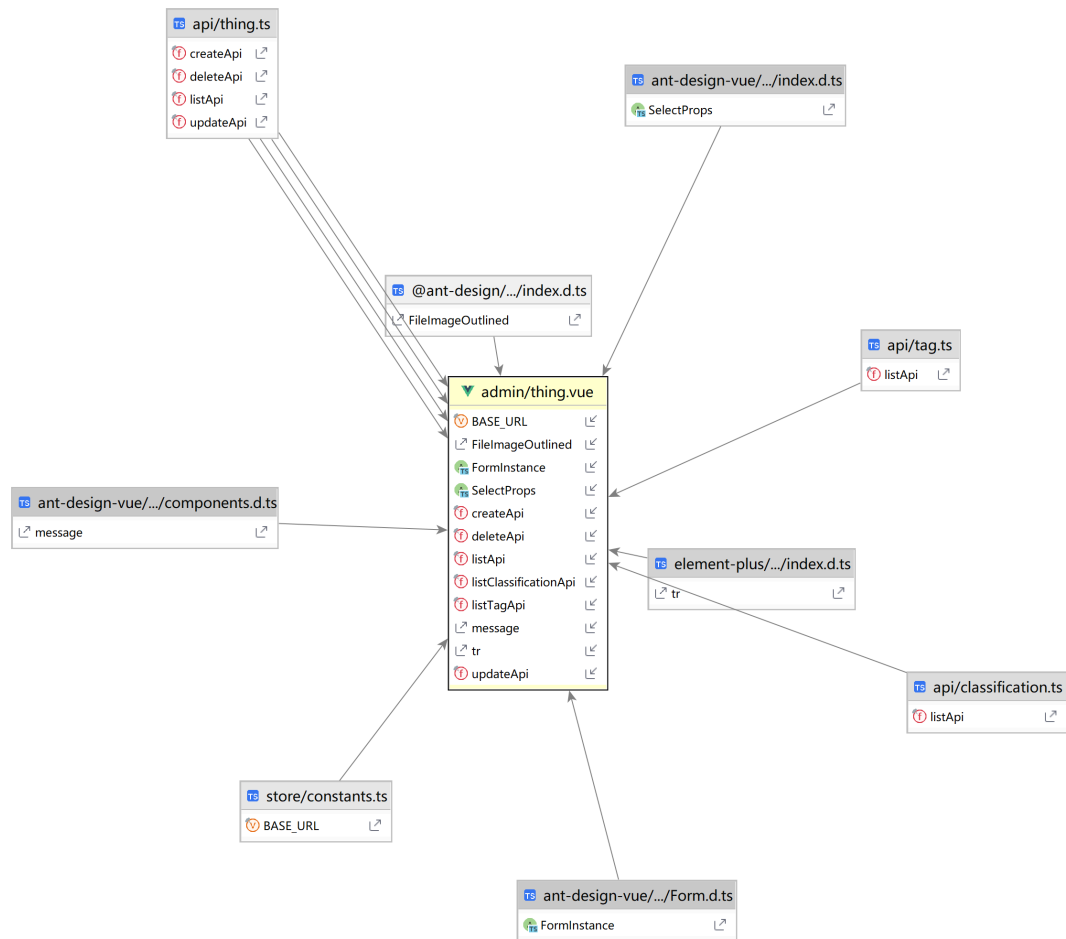
（一）主要技术细节

- **Vue.js**：作为前端框架，负责组件的声明和数据绑定。
- **ant-design-vue**：作为UI组件库，提供了表格、表单、弹窗等常用组件，确保界面的一致性和美观性。
- **API接口**：通过与后端API的交互，实现数据的增删改查。主要接口包括listApi、createApi、updateApi、deleteApi、listClassificationApi和listTagApi。
- **文件上传**：使用Upload组件实现图片文件的上传，并通过BASE_URL配置文件路径，实现图片的存储和展示。

四、目录结构

```
1 src/
2   └─ api/
3       └─ thing.ts # 教师相关API
4       └─ classification.ts # 分类相关API
5       └─ tag.ts # 标签相关API
6   └─ components/
7       └─ thing.vue # 教师管理组件
8   └─ store/
9       └─ constants.ts # 常量文件
```

五、关系图



六、代码详情

(一) 模板部分

- 包含数据表格，支持新增、编辑、删除和搜索功能。
- 弹窗用于数据的详细编辑与添加。
- 使用Ant Design Vue组件库进行UI构建。

```
1 <template>
2 <div>
3   <!-- 页面区域 -->
4   <div class="page-view">
5     <div class="table-operations">
6       <a-space>
7         <a-button type="primary" @click="handleAdd">新增</a-
button>
8         <a-button @click="handleBatchDelete">批量删除</a-button>
9         <a-input-search addon-before="名称" enter-button
@search="onSearch" @change="onSearchChange" />
10      </a-space>
11    </div>
12    <a-table
13      size="middle"
14      rowKey="id"
15      :loading="data.loading"
16      :columns="columns"
17      :data-source="data.dataList"
18      :scroll="{ x: 'max-content' }"
19      :row-selection="rowSelection"
20      :pagination="{
21        size: 'default',
22        current: data.page,
23        pageSize: data.pageSize,
24        onChange: (current) => (data.page = current),
25        showSizeChanger: false,
26        showTotal: (total) => `共${total}条数据`,
27      }"
28    >
29      <template #bodyCell="{ text, record, index, column }">
30        <template v-if="column.key === 'operation'">
31          <span>
32            <a @click="handleEdit(record)">编辑</a>
33            <a-divider type="vertical" />
34            <a-popconfirm title="确定删除?" ok-text="是" cancel-
text="否" @confirm="confirmDelete(record)">
35              <a href="#">删除</a>
36            </a-popconfirm>
```

```

37         </span>
38     </template>
39 </template>
40 </a-table>
41 </div>
42
43 <!-- 弹窗区域 -->
44 <div>
45     <a-modal
46         :visible="modal.visible"
47         :forceRender="true"
48         :title="modal.title"
49         width="880px"
50         ok-text="确认"
51         cancel-text="取消"
52         @cancel="handleCancel"
53         @ok="handleOk"
54     >
55         <div>
56             <a-form ref="myform" :label-col="{ style: { width: '80px'
57 } }" :model="modal.form" :rules="modal.rules">
58                 <a-row :gutter="24">
59                     <a-col span="24">
60                         <a-form-item label="家教姓名" name="title">
61                             <a-input placeholder="请输入" v-
62 model:value="modal.form.title" />
63                         </a-form-item>
64                     </a-col>
65                     <a-col span="12">
66                         <a-form-item label="分类" name="classificationId">
67                             <a-select
68                                 placeholder="请选择"
69                                 allowClear
70                                 :options="modal.cData"
71                                 :field-names="{ label: 'title', value: 'id' }"
72                                 v-model:value="modal.form.classificationId"
73                             />
74                         </a-form-item>
75                     </a-col>
76                     <a-col span="12">
77                         <a-form-item label="标签">
78                             <a-select mode="multiple" placeholder="请选择"
79 allowClear v-model:value="modal.form.tags">
80                                 <template v-for="item in modal.tagData">
81                                     <a-select-option :value="item.id">{{
82 item.title }}</a-select-option>
83                                 </template>
84                             </a-select>

```



```

81         </a-form-item>
82     </a-col>
83     <a-col span="24">
84         <a-form-item label="封面">
85             <a-upload-dragger
86                 name="file"
87                 accept="image/*"
88                 :multiple="false"
89                 :before-upload="beforeUpload"
90                 v-model:file-list="fileList"
91             >
92                 <p class="ant-upload-drag-icon">
93                     <template v-if="modal.form.coverUrl">
94                         
96                     </template>
97                     <template v-else>
98                         <file-image-outlined />
99                     </template>
100                 </p>
101                 <p class="ant-upload-text"> 请选择要上传的封面图
片 </p>
102             </a-upload-dragger>
103         </a-form-item>
104     </a-col>
105     <a-col span="24">
106         <a-form-item label="家教简介">
107             <a-textarea placeholder="请输入" v-
model:value="modal.form.description" />
108         </a-form-item>
109     </a-col>
110     <a-col span="12">
111         <a-form-item label="小时价格" name="price">
112             <a-input-number placeholder="请输入" :min="0" v-
model:value="modal.form.price" style="width: 100%" />
113         </a-form-item>
114     </a-col>
115     <a-col span="12">
116         <a-form-item label="手机号">
117             <a-input-number placeholder="请输入" :min="0" v-
model:value="modal.form.mobile" style="width: 100%" />
118         </a-form-item>
119     </a-col>
120     <a-col span="12">
121         <a-form-item label="年龄">
122             <a-input-number placeholder="请输入" :min="0" v-
model:value="modal.form.age" style="width: 100%" />

```

```

123         </a-form-item>
124     </a-col>
125     <a-col span="12">
126         <a-form-item label="性别">
127             <a-input placeholder="请输入" v-
model:value="modal.form.sex" style="width: 100%" />
128         </a-form-item>
129     </a-col>
130     <a-col span="12">
131         <a-form-item label="所在地区">
132             <a-input placeholder="请输入" v-
model:value="modal.form.location" style="width: 100%" />
133         </a-form-item>
134     </a-col>
135     <a-col span="12">
136         <a-form-item label="状态" name="status">
137             <a-select placeholder="请选择" allowClear v-
model:value="modal.form.status">
138                 <a-select-option key="0" value="0">上架</a-
select-option>
139                 <a-select-option key="1" value="1">下架</a-
select-option>
140             </a-select>
141         </a-form-item>
142     </a-col>
143 </a-row>
144 </a-form>
145 </div>
146 </a-modal>
147 </div>
148 </div>
149 </template>

```

(二) 脚本部分

- getDataList、getCDataList、getTagDataList分别从不同的API获取数据。
 - handleAdd、handleEdit、confirmDelete、handleBatchDelete负责物品的增删改操作。
 - handleOk验证并提交表单数据到服务器。
 - resetModal、hideModal重置并关闭物品管理模态框。
-

```

1 <script setup lang="ts">
2 import { FormInstance, message, SelectProps } from 'ant-design-
vue';
3 import { createApi, listApi, updateApi, deleteApi } from
'./@/api/thing';

```

```
4 import { ListApi as listClassificationApi } from
  '@api/classification';
5 import { ListApi as listTagApi } from '@api/tag';
6 import { BASE_URL } from '@store/constants';
7 import { FileImageOutlined } from '@ant-design/icons-vue';
8
9 const columns = reactive([
10   {
11     title: '序号',
12     dataIndex: 'index',
13     key: 'index',
14     width: 60,
15   },
16   {
17     title: '姓名',
18     dataIndex: 'title',
19     key: 'title',
20   },
21   {
22     title: '价格',
23     dataIndex: 'price',
24     key: 'price',
25   },
26   {
27     title: '性别',
28     dataIndex: 'sex',
29     key: 'sex',
30   },
31   {
32     title: '年龄',
33     dataIndex: 'age',
34     key: 'age',
35   },
36   {
37     title: '地区',
38     dataIndex: 'location',
39     key: 'location',
40   },
41   {
42     title: '简介',
43     dataIndex: 'description',
44     key: 'description',
45     customRender: ({ text, record, index, column }) => (text ?
text.substring(0, 10) + '...' : '--'),
46   },
47   {
48     title: '状态',
49     dataIndex: 'status',
```

```

50     key: 'status',
51     customRender: ({ text, record, index, column }) => (text ===
'0' ? '上架' : '下架'),
52   },
53   {
54     title: '操作',
55     dataIndex: 'action',
56     key: 'operation',
57     align: 'center',
58     fixed: 'right',
59     width: 140,
60   },
61   ]);
62
63   const beforeUpload = (file: File) => {
64     // 改文件名
65     const fileName = new Date().getTime().toString() + '.' +
file.type.substring(6);
66     const copyFile = new File([file], fileName);
67     console.log(copyFile);
68     modal.form.imageFile = copyFile;
69     return false;
70   };
71
72   // 文件列表
73   const fileList = ref<any[]>([]);
74
75   // 页面数据
76   const data = reactive({
77     dataList: [],
78     loading: false,
79     keyword: '',
80     selectedRowKeys: [] as any[],
81     pageSize: 10,
82     page: 1,
83   });
84
85   // 弹窗数据源
86   const modal = reactive({
87     visible: false,
88     editFlag: false,
89     title: '',
90     cData: [],
91     tagData: [{}],
92     form: {
93       id: undefined,
94       title: undefined,
95       classificationId: undefined,

```

```

96     tags: [],
97     repertory: undefined,
98     price: undefined,
99     mobile: undefined,
100    age: undefined,
101    sex: undefined,
102    location: undefined,
103    status: undefined,
104    cover: undefined,
105    coverUrl: undefined,
106    imageFile: undefined,
107  },
108  rules: {
109    title: [{ required: true, message: '请输入名称', trigger:
'change' }],
110    classificationId: [{ required: true, message: '请选择分类',
trigger: 'change' }],
111    price: [{ required: true, message: '请输入定价', trigger:
'change' }],
112    status: [{ required: true, message: '请选择状态', trigger:
'change' }],
113  },
114  });
115
116  const myform = ref<FormInstance>();
117
118  onMounted(() => {
119    getDataList();
120    getCDataList();
121    getTagDataList();
122  });
123
124  const getDataList = () => {
125    data.loading = true;
126    listApi({
127      keyword: data.keyword,
128    })
129      .then((res) => {
130        data.loading = false;
131        console.log(res);
132        res.data.forEach((item: any, index: any) => {
133          item.index = index + 1;
134          item.price = item.price + '元/时';
135        });
136        data.dataList = res.data;
137      })
138      .catch((err) => {
139        data.loading = false;

```

```

140     console.log(err);
141   });
142 };
143
144 const getCDataList = () => {
145   listClassificationApi({}).then((res) => {
146     modal.cData = res.data;
147   });
148 };
149 const getTagDataList = () => {
150   listTagApi({}).then((res) => {
151     res.data.forEach((item, index) => {
152       item.index = index + 1;
153     });
154     modal.tagData = res.data;
155   });
156 };
157
158 const onSearchChange = (e: Event) => {
159   data.keyword = e?.target?.value;
160   console.log(data.keyword);
161 };
162
163 const onSearch = () => {
164   getDataList();
165 };
166
167 const rowSelection = ref({
168   onChange: (selectedRowKeys: (string | number)[], selectedRows:
169     DataItem[]) => {
170     console.log(`selectedRowKeys: ${selectedRowKeys}`,
171       'selectedRows: ', selectedRows);
172     data.selectedRowKeys = selectedRowKeys;
173   },
174 });
175
176 const handleAdd = () => {
177   resetModal();
178   modal.visible = true;
179   modal.editFlag = false;
180   modal.title = '新增';
181   // 重置
182   for (const key in modal.form) {
183     modal.form[key] = undefined;
184   }
185   modal.form.cover = undefined;
186 };
187
188 const handleEdit = (record: any) => {

```

```

186   resetModal();
187   modal.visible = true;
188   modal.editFlag = true;
189   modal.title = '编辑';
190   // 重置
191   for (const key in modal.form) {
192     modal.form[key] = undefined;
193   }
194   for (const key in record) {
195     if (record[key]) {
196       modal.form[key] = record[key];
197     }
198   }
199   if (modal.form.cover) {
200     modal.form.coverUrl = BASE_URL + '/api/staticfiles/image/' +
modal.form.cover;
201     modal.form.cover = undefined;
202   }
203 };
204
205 const confirmDelete = (record: any) => {
206   console.log('delete', record);
207   deleteApi({ ids: record.id })
208     .then((res) => {
209       getDataList();
210     })
211     .catch((err) => {
212       message.error(err.msg || '操作失败');
213     });
214 };
215
216 const handleBatchDelete = () => {
217   console.log(data.selectedRowKeys);
218   if (data.selectedRowKeys.length <= 0) {
219     console.log('hello');
220     message.warn('请勾选删除项');
221     return;
222   }
223   deleteApi({ ids: data.selectedRowKeys.join(',') })
224     .then((res) => {
225       message.success('删除成功');
226       data.selectedRowKeys = [];
227       getDataList();
228     })
229     .catch((err) => {
230       message.error(err.msg || '操作失败');
231     });
232 };

```

```

233
234 const handleOk = () => {
235   myform.value
236   ?.validate()
237   .then(() => {
238     const formData = new FormData();
239     if (modal.editFlag) {
240       formData.append('id', modal.form.id);
241     }
242     formData.append('title', modal.form.title);
243     if (modal.form.classificationId) {
244       formData.append('classificationId',
modal.form.classificationId);
245     }
246     if (modal.form.tags) {
247       modal.form.tags.forEach(function (value) {
248         if (value) {
249           formData.append('tags[]', value);
250         }
251       });
252     }
253     if (modal.form.imageFile) {
254       formData.append('imageFile', modal.form.imageFile);
255     }
256     formData.append('description', modal.form.description ||
''');
257     formData.append('price', modal.form.price || '');
258     if (modal.form.mobile) {
259       formData.append('mobile', modal.form.mobile);
260     }
261     if (modal.form.age) {
262       formData.append('age', modal.form.age);
263     }
264     if (modal.form.sex) {
265       formData.append('sex', modal.form.sex);
266     }
267     if (modal.form.location) {
268       formData.append('location', modal.form.location);
269     }
270     if (modal.form.description) {
271       formData.append('description', modal.form.description);
272     }
273     if (modal.form.status) {
274       formData.append('status', modal.form.status);
275     }
276     if (modal.editFlag) {
277       updateApi(formData)
278       .then((res) => {

```



```

279         hideModal();
280         getDataList();
281     })
282     .catch((err) => {
283         console.log(err);
284         message.error(err.msg || '操作失败');
285     });
286 } else {
287     createApi(formData)
288     .then((res) => {
289         hideModal();
290         getDataList();
291     })
292     .catch((err) => {
293         console.log(err);
294         message.error(err.msg || '操作失败');
295     });
296 }
297 })
298 .catch((err) => {
299     console.log('不能为空');
300 });
301 };
302
303 const handleCancel = () => {
304     hideModal();
305 };
306
307 // 恢复表单初始状态
308 const resetModal = () => {
309     myform.value?.resetFields();
310     fileList.value = [];
311 };
312
313 // 关闭弹窗
314 const hideModal = () => {
315     modal.visible = false;
316 };
317 </script>

```

（三）样式部分

- `.page-view`: 设置了页面视图的样式，包括最小高度、背景色、内边距、显示方式和主轴方向。
- `.table-operations`: 设置了表格操作区域的样式，包括底部边距和文本对齐方式。

- `.table-operations > button`: 设置了表格操作区域中按钮的样式，包括右边距。

```
1 <style scoped lang="less">
2 .page-view {
3   min-height: 100%;
4   background: #fff;
5   padding: 24px;
6   display: flex;
7   flex-direction: column;
8 }
9
10 .table-operations {
11   margin-bottom: 16px;
12   text-align: right;
13 }
14
15 .table-operations > button {
16   margin-right: 8px;
17 }
18 </style>``
```

七、controller部分

(一) ThingController

1. 接口说明

- 查询物品列表接口
 - URL: `/thing/list`
 - 方法: GET
 - 参数: String keyword、String sort、String c、String tag
 - 响应: 物品列表

查询物品详情接口

- URL: `/thing/detail`
- 方法: GET
- 参数: String id (物品ID)
- 响应: 物品详情

创建物品接口

- URL: /thing/create
- 方法: POST
- 权限: 管理员
- 参数: Thing thing (物品实体)
- 响应: 创建成功提示
- 事务: 确保创建操作的原子性

删除物品接口

- URL: /thing/delete
- 方法: POST
- 权限: 管理员
- 参数: String ids (物品ID列表)
- 响应: 删除成功提示

更新物品接口

- URL: /thing/update
- 方法: POST
- 权限: 管理员
- 参数: Thing thing (物品实体)
- 响应: 更新成功提示
- 事务: 确保更新操作的原子性

查询用户物品列表接口

- URL: /thing/listUserThing
- 方法: GET
- 参数: String userId (用户ID)
- 响应: 用户物品列表

2. 代码

```
1 @RestController
2 @RequestMapping("/thing")
3 public class ThingController {
4     @Autowired
5     ThingService service;
6
7     @Value("${File.uploadPath}")
8     private String upLoadPath;
9
10    @RequestMapping(value = "/list", method = RequestMethod.GET)
```

```

11     public APIResponse list(String keyword, String sort, String
c, String tag) {
12         List<Thing> list = service.getThingList(keyword, sort,
c, tag);
13         return new APIResponse(ResponseCode.SUCCESS, "查询成功",
list);
14     }
15
16     @RequestMapping(value = "/detail", method =
RequestMethod.GET)
17     public APIResponse detail(String id) {
18         Thing thing = service.getThingById(id);
19         return new APIResponse(ResponseCode.SUCCESS, "查询成功",
thing);
20     }
21
22     @Access(level = AccessLevel.ADMIN)
23     @RequestMapping(value = "/create", method =
RequestMethod.POST)
24     @Transactional
25     public APIResponse create(Thing thing) throws IOException {
26         String url = saveThing(thing);
27         if (!StringUtils.isEmpty(url)) {
28             thing.cover = url;
29         }
30         service.createThing(thing);
31         return new APIResponse(ResponseCode.SUCCESS, "创建成
功");
32     }
33
34     @Access(level = AccessLevel.ADMIN)
35     @RequestMapping(value = "/delete", method =
RequestMethod.POST)
36     public APIResponse delete(String ids) {
37         String[] arr = ids.split(",");
38         for (String id : arr) {
39             service.deleteThing(id);
40         }
41         return new APIResponse(ResponseCode.SUCCESS, "删除成
功");
42     }
43
44     @Access(level = AccessLevel.ADMIN)
45     @RequestMapping(value = "/update", method =
RequestMethod.POST)
46     @Transactional
47     public APIResponse update(Thing thing) throws IOException {
48         String url = saveThing(thing);

```

```

49         if (!StringUtils.isEmpty(url)) {
50             thing.cover = url;
51         }
52         service.updateThing(thing);
53         return new APIResponse(ResponseCode.SUCCESS, "更新成功");
54     }
55
56     @RequestMapping(value = "/listUserThing", method =
RequestMethod.GET)
57     public APIResponse listUserThing(String userId) {
58         List<Thing> list = service.getUserThing(userId);
59         return new APIResponse(ResponseCode.SUCCESS, "查询成功",
list);
60     }
61
62     private String saveThing(Thing thing) throws IOException {
63         MultipartFile file = thing.getImageFile();
64         String newFileName = null;
65         if (file != null && !file.isEmpty()) {
66             String oldFileName = file.getOriginalFilename();
67             String randomStr = UUID.randomUUID().toString();
68             newFileName = randomStr +
oldFileName.substring(oldFileName.lastIndexOf("."));
69             String filePath = uploadPath + File.separator +
"image" + File.separator + newFileName;
70             File destFile = new File(filePath);
71             if (!destFile.getParentFile().exists()) {
72                 destFile.getParentFile().mkdirs();
73             }
74             file.transferTo(destFile);
75         }
76         if (!StringUtils.isEmpty(newFileName)) {
77             thing.cover = newFileName;
78         }
79         return newFileName;
80     }
81 }

```

3. 关系图

4. 其他略

八、组件说明

(一) 家教管理表格

表格用于显示教师的基本信息，包括序号、姓名、价格、性别、年龄、地区、简介和状态。提供编辑和删除操作按钮。

九、弹窗表格

用于新增和编辑教师信息的弹窗表单，包含教师的姓名、分类、标签、封面、简介、价格、手机号、年龄、性别、所在地区和状态等字段。

十、主要方法

- `getDataList`: 获取教师列表数据。
- `getCDataList`: 获取分类列表数据。
- `getTagDataList`: 获取标签列表数据。
- `onSearchChange`: 更新搜索关键字。
- `onSearch`: 执行搜索操作。
- `handleAdd`: 打开新增教师弹窗。
- `handleEdit`: 打开编辑教师弹窗，并填充表单数据。
- `confirmDelete`: 确认删除教师信息。
- `handleBatchDelete`: 批量删除教师信息。
- `handleOk`: 确认弹窗表单的操作（新增或编辑）。
- `handleCancel`: 取消弹窗操作。
- `resetModal`: 重置表单和文件列表。
- `hideModal`: 隐藏弹窗。

十一、API说明

- `listApi`: 获取教师列表数据的API。
- `createApi`: 创建新教师的API。
- `updateApi`: 更新教师信息的API。
- `deleteApi`: 删除教师信息的API。
- `listClassificationApi`: 获取分类列表的API。
- `listTagApi`: 获取标签列表的API。

十二、使用指南

（一）引入组件

```
1 import TeacherManagement from  
  '@@/components/TeacherManagement.vue';
```

（二）在页面中使用组件

```
1 <TeacherManagement />
```

（三）配置API接口

确保/api/thing、/api/classification、/api/tag等接口正常工作。

（四）配置基础URL

在/store/constants中配置BASE_URL（注意端口号）。

十三、注意事项

- 确保API接口返回的数据格式与组件中的预期一致。
- 组件使用了ant-design-vue的a-table、a-modal、a-form等组件，需要确保项目中安装了ant-design-vue。

十四、总结

该组件实现了家教管理的主要功能，包括新增、编辑、删除和搜索家教信息。通过直观的界面和便捷的操作方式，管理员可以轻松管理家教信息，提升工作效率。该组件的设计和实现充分考虑了用户体验，使用现代化的前端技术栈，确保系统的可维护性和可扩展性。开发者可以根据具体需求对组件进行定制和扩展，满足不同场景下的教师管理需求。