硅谷课堂第五天-讲师管理模块前端

一、设置路由定义

1、修改路由

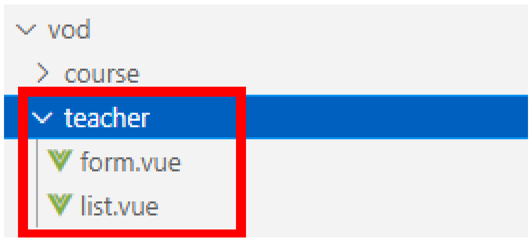
修改 src/router/index.js 文件，重新定义 constantRouterMap

注意： 每个路由的 name 不能相同

```
export const constantRouterMap = [
  {
    path: "/login",
    component: () => import("@/views/login/index"),
    hidden: true,
  },
  { path: "/404", component: () => import("@/views/404"), hidden: true },
  // 首页
  {
    path: "/",
    component: Layout,
    redirect: "/dashboard",
    name: "Dashboard",
    children: [
      {
        path: "dashboard",
        component: () => import("@/views/dashboard/index"),
        meta: { title: "硅谷课堂后台管理系统", icon: "dashboard" },
      },
    ],
  },
  // 讲师管理
  {
    path: "/vod",
    component: Layout,
    redirect: "/vod/course/list",
    name: "Vod",
    meta: {
      title: "点播管理",
      icon: "el-icon-bank-card",
    },
    alwaysShow: true,
    children: [
      {
        path: "teacher/list",
        name: "TeacherList",
        component: () => import("@/views/vod/teacher/list"),
        meta: { title: "讲师列表" },
      },
      {
        path: "teacher/create",
        name: "TeacherCreate",
        component: () => import("@/views/vod/teacher/form"),
        meta: { title: "添加讲师" },
        hidden: true,
      },
      {
        path: "teacher/edit/:id",
        name: "TeacherEdit",
        component: () => import("@/views/vod/teacher/form"),
        meta: { title: "编辑讲师" },
        hidden: true,
      },
    ],
  },
  { path: "*", redirect: "/404", hidden: true },
];
```

2、创建 vue 组件

在 src/views 文件夹下创建以下文件夹和文件



3、form.vue

```
<template>
  <div class="app-container">讲师表单</div>
</template>
```

4、list.vue

```
<template>
  <div class="app-container">讲师列表</div>
</template>
```

二、讲师分页列表

1、定义 api

创建文件 src/api/vod/teacher.js

```
import request from "@/utils/request";

const api_name = "/admin/vod/teacher";

export default {
  //讲师列表
  pageList(page, limit, searchObj) {
    return request({
      url: `${api_name}/${page}/${limit}`,
      method: `post`,
      data: searchObj,
    });
  },
};
```

2、初始化 vue 组件

src/views/vod/teacher/list.vue

```
<template>
  <div class="app-container">设置列表</div>
</template>
<script>
  import teacherApi from "@/api/vod/teacher";
  export default {
    // 定义数据模型
    data() {
      return {};
    },
    // 页面渲染成功后获取数据
    created() {
      this.fetchData();
    },
    // 定义方法
    methods: {
      fetchData() {},
    },
  };
</script>
```

3、定义 data

```
// 定义数据模型
data() {
  return {
    list: [], // 讲师列表
    total: 0, // 总记录数
    page: 1, // 页码
    limit: 10, // 每页记录数
    searchObj: {}, // 查询条件
    multipleSelection: []// 批量删除选中的记录列表
  }
},
```

4、定义 methods

```
methods: {
  fetchData() {
    // 调用api
    teacherApi.pageList(this.page, this.limit, this.searchObj).then(response => {
      debugger
      this.list = response.data.records
      this.total = response.data.total
    })
  },
}
```

5、表格渲染

```
<!-- 表格 -->
<el-table :data="list" border stripe @selection-change="handleSelectionChange">
  <el-table-column type="selection" />
  <el-table-column label="#" width="50">
    <template slot-scope="scope">
      {{ (page - 1) * limit + scope.$index + 1 }}
    </template>
  </el-table-column>
  <el-table-column prop="name" label="名称" width="80" />
  <el-table-column label="头衔" width="90">
    <template slot-scope="scope">
      <el-tag v-if="scope.row.level === 1" type="success" size="mini"
        >高级讲师</el-tag
      >
      <el-tag v-if="scope.row.level === 0" size="mini">首席讲师</el-tag>
    </template>
  </el-table-column>
  <el-table-column prop="intro" label="简介" />
  <el-table-column prop="sort" label="排序" width="60" />
  <el-table-column prop="joinDate" label="入驻时间" width="160" />
  <el-table-column label="操作" width="200" align="center">
    <template slot-scope="scope">
      <el-button type="text" size="mini" @click="removeById(scope.row.id)"
        >删除</el-button
      >
      <router-link :to="'/vod/teacher/edit/'+scope.row.id">
        <el-button type="text" size="mini">修改</el-button>
      </router-link>
    </template>
  </el-table-column>
</el-table>
```

6、分页组件

```
<!-- 分页组件 -->
<el-pagination
  :current-page="page"
  :total="total"
  :page-size="limit"
  :page-sizes="[5, 10, 20, 30, 40, 50, 100]"
  style="padding: 30px 0; text-align: center;"
  layout="total, sizes, prev, pager, next, jumper"
  @size-change="changePageSize"
  @current-change="changeCurrentPage"
/>
```

7、顶部查询表单

```
<!--查询表单-->
<el-card class="operate-container" shadow="never">
  <el-form :inline="true" class="demo-form-inline">
    <el-form-item label="名称">
      <el-input v-model="searchObj.name" placeholder="讲师名" />
    </el-form-item>

    <el-form-item label="头衔">
      <el-select v-model="searchObj.level" clearable placeholder="头衔">
        <el-option value="1" label="高级讲师" />
        <el-option value="0" label="首席讲师" />
      </el-select>
    </el-form-item>

    <el-form-item label="入驻时间">
      <el-date-picker
        v-model="searchObj.joinDateBegin"
        placeholder="开始时间"
        value-format="yyyy-MM-dd"
      />
    </el-form-item>
    <el-form-item label="-">
      <el-date-picker
        v-model="searchObj.joinDateEnd"
        placeholder="结束时间"
        value-format="yyyy-MM-dd"
      />
    </el-form-item>

    <el-button type="primary" icon="el-icon-search" @click="fetchData()"
      >查询</el-button
    >
    <el-button type="default" @click="resetData()">清空</el-button>
  </el-form>
</el-card>
```

分页和清空方法

```
// 每页记录数改变，size：回调参数，表示当前选中的“每页条数”
changePageSize(size) {
  this.limit = size
  this.fetchData()
},

// 改变页码，page：回调参数，表示当前选中的“页码”
changeCurrentPage(page) {
  this.page = page
  this.fetchData()
},

// 重置表单
resetData() {
  this.searchObj = {}
  this.fetchData()
},
```

三、讲师删除

1、定义 api

src/api/vod/teacher.js

```
removeById(id) {
  return request({
    url: `${api_name}/remove/${id}`,
    method: `delete`
  })
},
```

2、定义 methods

src/views/vod/teacher/list.vue

使用 MessageBox 弹框组件

```
// 根据id删除数据
removeById(id) {
  this.$confirm('此操作将永久删除该记录, 是否继续?', '提示', {
    confirmButtonText: '确定',
    cancelButtonText: '取消',
    type: 'warning'
  }).then(() => {
    return teacherApi.removeById(id)
  }).then((response) => {
    this.fetchData()
    this.$message.success(response.message)
  })
},
```

四、讲师新增

1、定义 api

src/api/vod/teacher.js

```
save(teacher) {
  return request({
    url: `${api_name}/save`,
    method: `post`,
    data: teacher
  })
},
```

2、初始化组件

src/views/vod/teacher/form.vue

```
<template>
  <div class="app-container">
    <!-- 输入表单 -->
    <el-form label-width="120px">
      <el-form-item label="讲师名称">
        <el-input v-model="teacher.name" />
      </el-form-item>
      <el-form-item label="入驻时间">
        <el-date-picker v-model="teacher.joinDate" value-format="yyyy-MM-dd" />
      </el-form-item>
      <el-form-item label="讲师排序">
        <el-input-number v-model="teacher.sort" :min="0" />
      </el-form-item>
      <el-form-item label="讲师头衔">
        <el-select v-model="teacher.level">
          <!--
            数据类型一定要和取出的json中的一致，否则没法回填
            因此，这里value使用动态绑定的值，保证其数据类型是number
            -->
          <el-option :value="1" label="高级讲师" />
          <el-option :value="2" label="首席讲师" />
        </el-select>
      </el-form-item>
      <el-form-item label="讲师简介">
        <el-input v-model="teacher.intro" />
      </el-form-item>
      <el-form-item label="讲师资历">
        <el-input v-model="teacher.career" :rows="10" type="textarea" />
      </el-form-item>

      <!-- 讲师头像 -->
      <el-form-item label="讲师头像"> </el-form-item>

      <el-form-item>
        <el-button type="primary" @click="saveOrUpdate()">保存</el-button>
      </el-form-item>
    </el-form>
  </div>
</template>
```

3、**实现**新增功能

```
<script>
import teacherApi from '@/api/vod/teacher'
export default {
  data() {
    return {
      BASE_API: 'http://localhost:8301',
      // 初始化讲师默认数据
      teacher: {
        sort: 0,
        level: 1
      },
      saveBtnDisabled: false // 保存按钮是否禁用，防止表单重复提交
    }
  },
  // 页面渲染成功
  created() {

  },
  methods: {
    saveOrUpdate() {
    // 禁用保存按钮
      this.saveBtnDisabled = true
      if (!this.teacher.id) {
        this.saveData()
      } else {
        this.updateData()
      }
    },
    // 新增讲师
    saveData() {
    // debugger
      teacherApi.save(this.teacher).then(response => {
        this.$message({
          type: 'success',
          message: response.message
        })
        this.$router.push({ path: '/vod/teacher/list' })
      })
    },
    // 根据id更新记录
    updateData() {

    }
  }
}
</script>
```

五、讲师修改-数据回显

1、定义 api

src/api/vod/teacher.js

```
getById(id) {
  return request({
    url: `${api_name}/get/${id}`,
    method: `get`
  })
},
```

2、组件中调用 api

methods 中定义 fetchDataById

```
// 根据id查询记录
fetchDataById(id) {
  teacherApi.getById(id).then(response => {
    this.teacher = response.data
  })
},
```

3、页面渲染前调用 fetchDataById

```
// 页面渲染成功
created() {
  if (this.$route.params.id) {
    this.fetchDataById(this.$route.params.id)
  }
},
```

六、讲师修改-更新

1、定义 api

```
updateById(teacher) {
  return request({
    url: `${api_name}/update`,
    method: `put`,
    data: teacher
  })
},
```

2、组件中调用 api

methods 中定义 updateData

```
// 根据id更新记录
updateData() {
  // teacher数据的获取
  teacherApi.updateById(this.teacher).then(response => {
    this.$message({
      type: 'success',
      message: response.message
    })
    this.$router.push({ path: '/vod/teacher/list' })
  })
},
```

3、完善 saveOrUpdate 方法

```
saveOrUpdate() {
  // 禁用保存按钮
    this.saveBtnDisabled = true
    if (!this.teacher.id) {
      this.saveData()
    } else {
      this.updateData()
    }
},
```

七、讲师批量删除

1、定义 api

src/api/vod/teacher.js

```
batchRemove(idList) {
  return request({
    url: `${api_name}/batch-remove`,
    method: `delete`,
    data: idList
  })
},
```

2、初始化组件

src/views/vod/teacher/list.vue

在 table 组件上添加 **批量删除按钮**

```
<!-- 工具按钮 -->
<el-card class="operate-container" shadow="never">
  <i class="el-icon-tickets" style="margin-top: 5px"></i>
  <span style="margin-top: 5px">数据列表</span>
  <el-button class="btn-add" @click="add()" style="margin-left: 10px;"
    >添加</el-button
  >
  <el-button class="btn-add" @click="batchRemove()">批量删除</el-button>
</el-card>
```

在 table 组件上添加复选框

```
<!-- 表格 -->
<el-table :data="list" border stripe @selection-change="handleSelectionChange">
  <el-table-column type="selection"
/></el-table>
```

3、实现功能

data 定义数据

```
multipleSelection: []; // 批量删除选中的记录列表
```

完善方法

```
// 批量删除
batchRemove() {
  if (this.multipleSelection.length === 0) {
    this.$message.warning('请选择要删除的记录! ')
    return
  }
  this.$confirm('此操作将永久删除该记录, 是否继续?', '提示', {
    confirmButtonText: '确定',
    cancelButtonText: '取消',
    type: 'warning'
  }).then(() => {
    // 点击确定, 远程调用ajax
    // 遍历selection, 将id取出放入id列表
    var idList = []
    this.multipleSelection.forEach(item => {
      idList.push(item.id)
    })
    // 调用api
    return teacherApi.batchRemove(idList)
  }).then((response) => {
    this.fetchData()
    this.$message.success(response.message)
  }).catch(error => {
    if (error === 'cancel') {
      this.$message.info('取消删除')
    }
  })
},
// 当多选选项发生变化的时候调用
handleSelectionChange(selection) {
  console.log(selection)
  this.multipleSelection = selection
},
```