# 学生部分：

## 添加：

### 按钮：

<el-button type='success' icon='el-icon-plus' @click='showAddStuForm()'>新增学生</el-button>

### 表单控制：

showAddStuForm() {

this.stuFormTitle = "新增学生";

this.sIDInputDisabled = false;

this.rstButtonVisible = true;

this.addButtonVisible = true;

this.edtButtonVisible = false;

this.stuFormVisible = true;

},

### 提交请求：

addStudent(student) {

this.hideStuForm();

this.$http

.post("/admin/student", student)

.then(res => {

if (res.data.restype == "success") {

this.$message({

type: "success",

message: "添加学生成功",

showClose: true,

center: true

});

this.getStudents();

this.resetStuFormModel();

} else {…

});

});

},

## 删除

### 按钮

<el-button type='danger' icon='el-icon-minus' @click='handleDeleteSels()'>删除所选</el-button>

<el-button icon='el-icon-delete' size='mini' type='danger' @click='handleDeleteFromMenu(scope.$index, scope.row)'>删除</el-button>

### 提交请求

handleDeleteSels() {

this.selectedStudents.forEach(student => {

this.delStudent(student.sID);

});

},

handleDeleteFromMenu(index, row) {

this.$confirm("确定删除" + row.sID + "？", "警告", {

confirmButtonText: "确定删除",

cancelButtonText: "取消",

type: "warning"

}).then(() => {

this.delStudent(row.sID);

});

},

delStudent(sID) {

this.$http

.delete("/admin/student/" + sID)

.then(res => {

if (res.data.restype == "success") ….

}

## 修改

### 按钮

<el-button icon='el-icon-edit' size='mini' @click='handleEditFromMenu(scope.$index, scope.row)'>编辑</el-button>

### 表单控制

handleEditFromMenu(index, row) {

this.stuFormModel.sID = row.sID;

this.stuFormModel.sName = row.sName;

this.stuFormModel.classno = row.classno;

this.stuFormModel.sex = row.sex;

this.stuFormModel.password = row.password;

this.sIDInputDisabled = true;

this.showEdtStuForm();

},

### 提交请求

handleEditFromForm() {

this.$refs.stuForm.validate(valid => {

if (valid) {

this.edtStudent(this.stuFormModel);

} else {

return false;

}

});

},

edtStudent(student) {

this.hideStuForm();

this.$http

.put("/admin/student/" + student.sID, student)

.then(res => {

if (res.data.restype == "success") …

},

## 查询

### 按钮

<el-button type='primary' icon='el-icon-refresh' @click='getStudents()'>刷新表格</el-button>

### 提交请求

getStudents() {

this.$http

.get("/admin/student")

.then(res => {

this.stuTableData = res.data;

})….

},

# 教师部分：

## 添加：

### 按钮：

<el-button type='success' icon='el-icon-plus' @click='showAddTeaForm()'>新增教师</el-button>

### 表单控制：

showAddTeaForm() {

this.teaFormTitle = "新增教师";

this.tIDInputDisabled = false;

this.rstButtonVisible = true;

this.addButtonVisible = true;

this.edtButtonVisible = false;

this.teaFormVisible = true;

},

### 提交请求：

addTeacher(teacher) {

this.hideTeaForm();

this.$http

.post("/admin/teacher", teacher)

.then(res => {

if (res.data.restype == "success") …

## 删除

### 按钮

<el-button type='danger' icon='el-icon-minus' @click='handleDeleteSels()'>删除所选</el-button>

<el-button icon='el-icon-delete' size='mini' type='danger' @click='handleDeleteFromMenu(scope.$index, scope.row)'>删除</el-button>

### 提交请求

delTeacher(tID) {

this.$http

.delete("/admin/teacher/" + tID)

.then(res => {

if (res.data.restype == "success")

## 修改

### 按钮

<el-button icon='el-icon-edit' size='mini' @click='handleEditFromMenu(scope.$index, scope.row)'>编辑</el-button>

### 表单控制

handleEditFromForm() {

this.$refs.teaForm.validate(valid => {

if (valid) {

this.edtTeacher(this.teaFormModel);

} else {

return false;

}

});

},

### 提交请求

edtTeacher(teacher) {

this.hideTeaForm();

this.$http

.put("/admin/teacher/" + teacher.tID, teacher)

.then(res => {

if (res.data.restype == "success")

## 查询

### 按钮

<el-button type='primary' icon='el-icon-refresh' @click='getTeachers()'>刷新表格</el-button>

### 提交请求

getTeachers() {

this.$http

.get("/admin/teacher")

.then(res => {…

# 课程部分：

## 添加：

### 按钮：

<el-button type='success' icon='el-icon-plus' @click='showAddCrsForm()'>新增课程</el-button>

### 表单控制：

showAddCrsForm() {

this.crsFormTitle = "新增课程";

this.cIDInputDisabled = false;

this.rstButtonVisible = true;

this.addButtonVisible = true;

this.edtButtonVisible = false;

this.crsFormVisible = true;

},

### 提交请求：

addCourse(course) {

this.hideCrsForm();

this.$http

.post("/admin/course", course)

.then(res => {

if (res.data.restype == "success") {

## 删除

### 按钮

<el-button type='danger' icon='el-icon-minus' @click='handleDeleteSels()'>删除所选</el-button>

<el-button icon='el-icon-delete' size='mini' type='danger' @click='handleDeleteFromMenu(scope.$index, scope.row)'>删除</el-button>

### 提交请求

handleDeleteSels() {

this.selectedCourses.forEach(course => {

this.delCourse(course.cID);

});

},

handleDeleteFromMenu(index, row) {

this.$confirm("确定删除" + row.cID + "？", "警告", {

confirmButtonText: "确定删除",

cancelButtonText: "取消",

type: "warning"

}).then(() => {

this.delCourse(row.cID);

});

},

delCourse(cID) {

this.$http

.delete("/admin/course/" + cID)

.then(res => {

if (res.data.restype == "success") {

## 修改

### 按钮

<el-button icon='el-icon-edit' size='mini' @click='handleEditFromMenu(scope.$index, scope.row)'>编辑</el-button>

### 表单控制

handleEditFromMenu(index, row) {

this.crsFormModel.cID = row.cID;

this.crsFormModel.tID = row.tID;

this.crsFormModel.cName = row.cName;

this.crsFormModel.credit = row.credit;

this.crsFormModel.semester = row.semester;

this.cIDInputDisabled = true;

this.showEdtCrsForm();

},

### 提交请求

handleEditFromForm() {

this.$refs.crsForm.validate(valid => {

if (valid) {

this.edtCourse(this.crsFormModel);

} else {

return false;

}

});

},

edtCourse(course) {

this.hideCrsForm();

this.$http

.put("/admin/course/" + course.cID, course)

.then(res => {

if (res.data.restype == "success") {

## 查询

### 按钮

<el-button type='primary' icon='el-icon-refresh' @click='getCourses()'>刷新表格</el-button>

### 提交请求

getCourses() {

this.$http

.get("/admin/course")

.then(res => {