

Cola-Ui 前端文档

序

项目地址

- 后端地址: <https://gitee.com/xiaolifeizei/cola-admin>
- 前端地址: <https://gitee.com/xiaolifeizei/cola-ui>

在线演示

- 演示地址: <http://www.cola-admin.vip>
- 默认用户: admin
- 默认密码: 123123

前端介绍

Cola-Ui基于Vue-Element-Admin框架开发, 参考了Eladmin-Web的部分代码, 项目地址如下

Vue-Element-Admin: <https://gitee.com/panjiachen/vue-element-admin>

Eladmin-web: <https://gitee.com/elunez/eladmin-web>

感谢两个优秀的开源前端框架, 大佬V5!

目录结构

```
1  cola-ui
2  |—build      # 构建配置目录
3  |—mock       # mock数据
4  |—public     # 静态资源
5  |—src
6  |   |—api      # 与后端交互的接口
7  |   |—components # 公共组件
8  |   |—layout    # 页面布局
9  |   |—libs      # 自定义UI
10 |   |—router     # 路由
11 |   |—store      # vuex
12 |   |—styles     # 自定义样式
13 |   |—utils      # axios封装、鉴权封装等
14 |   |—views      # 页面
15 |—nginx.conf    # nginx配置文件
```

环境要求

基础开发环境

- NodeJs v16+
- npm v8+

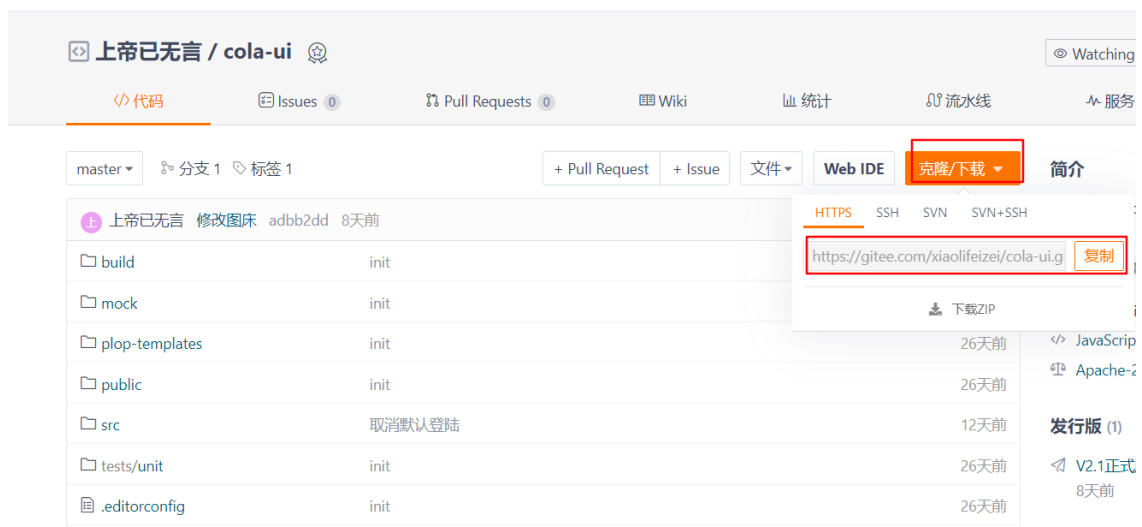
推荐IDE

- WebStorm

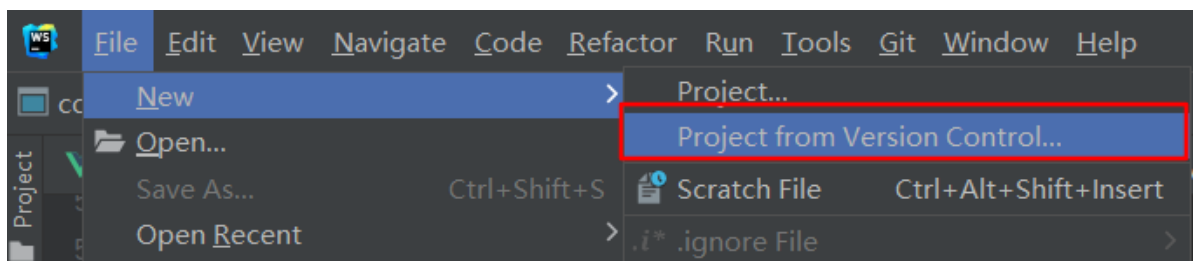
环境准备

导入项目

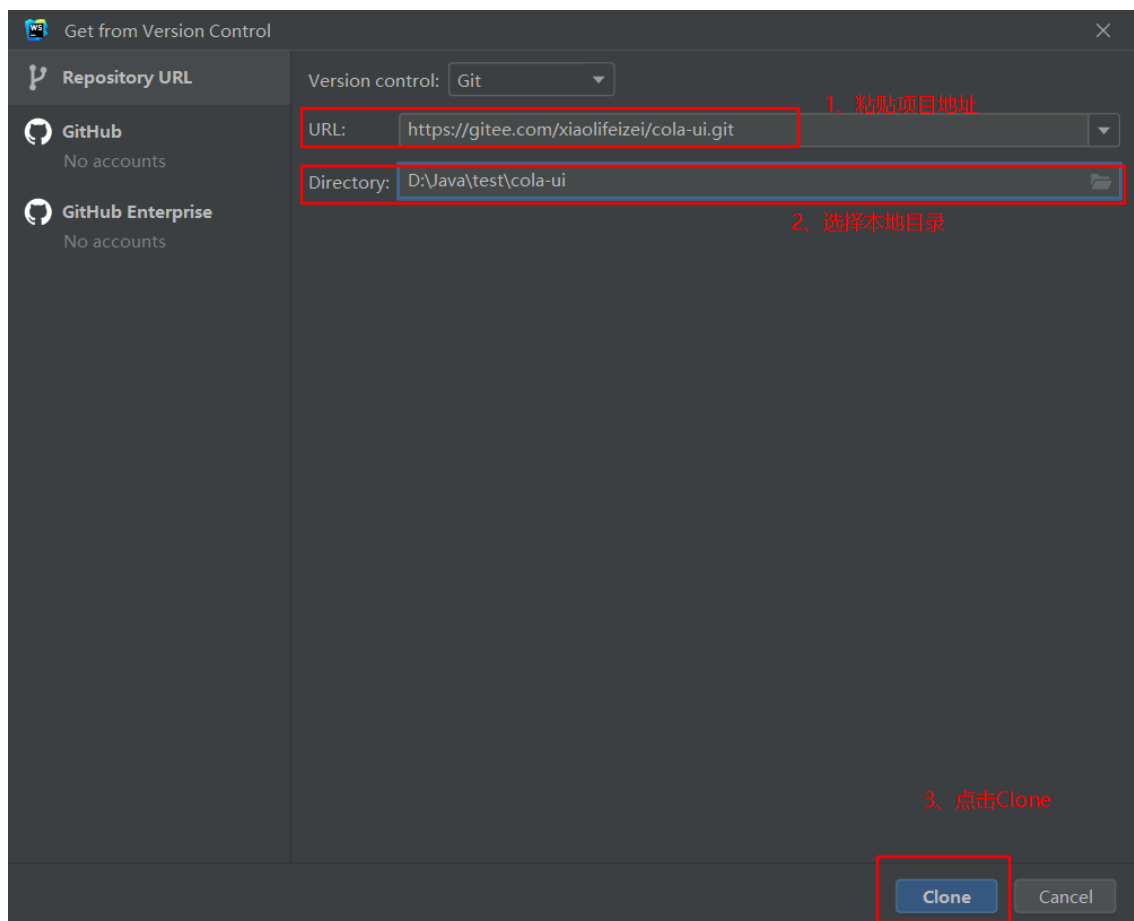
1. 进入cola-ui项目首页<https://gitee.com/xiaolifeize/cola-ui>
2. 复制项目地址



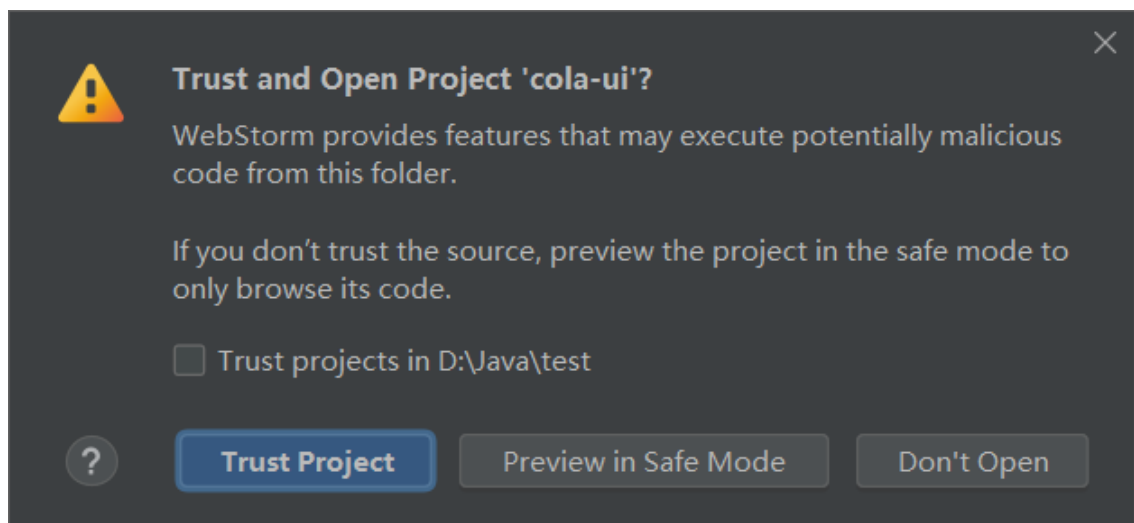
3. 打开WebStorm，依次点击File->New->Project from Version Control...



4. 在弹出的对话框中粘贴复制的项目地址

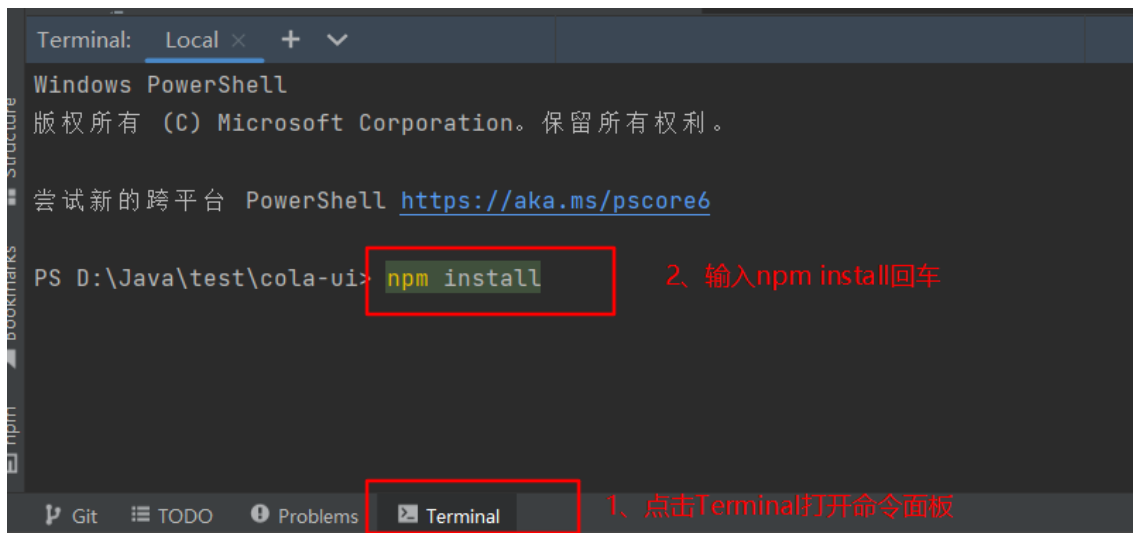


5. IDE可能会弹出对话框，点击“Trust Project”



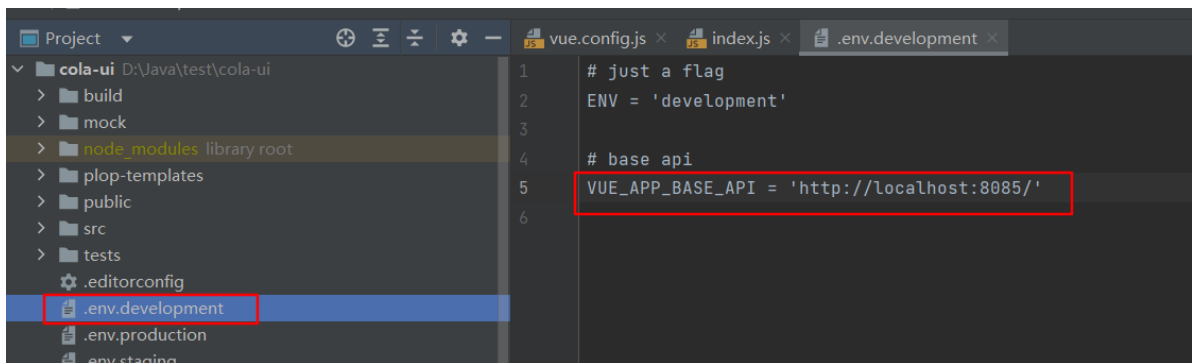
6. 等待代码下载完成

7. 安装依赖

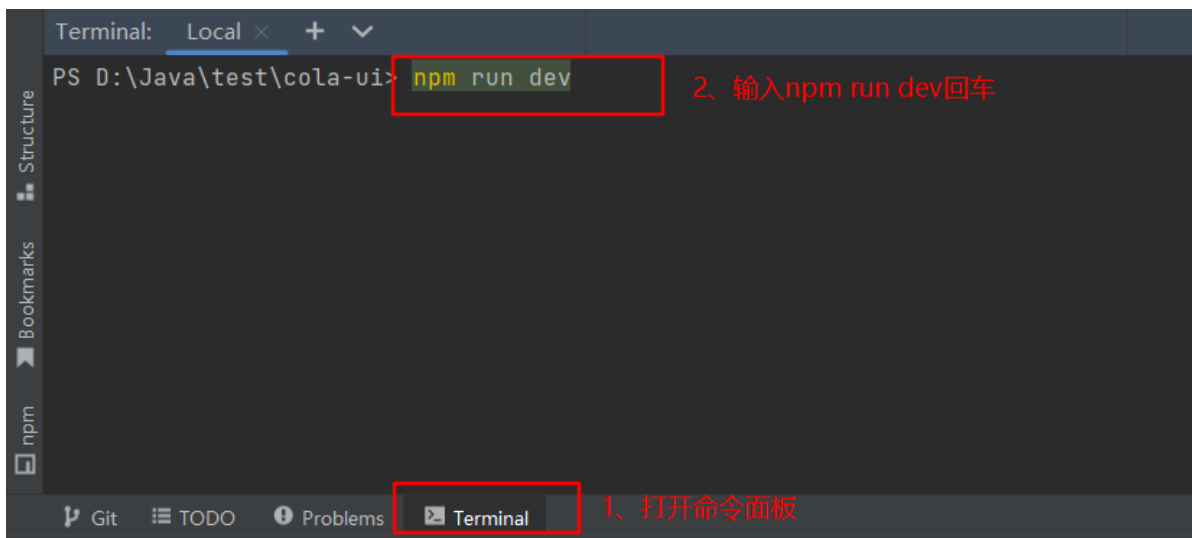


运行项目

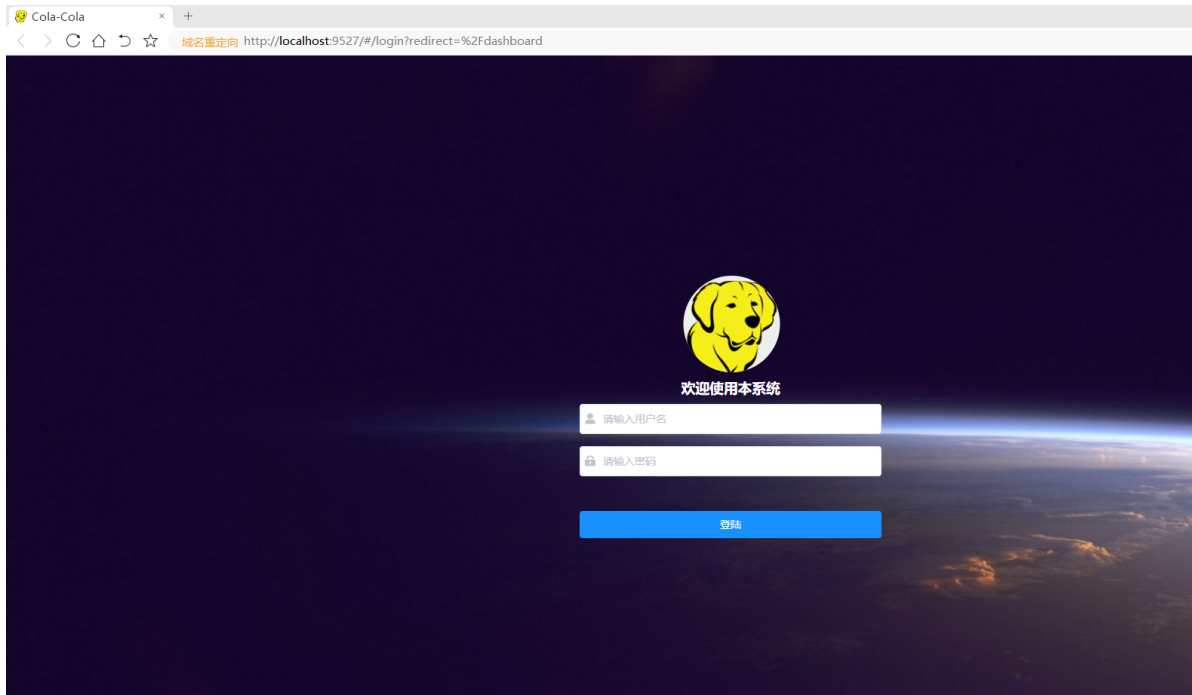
1. 运行前请先启动cola-admin后台服务，请在.env.development文件中修改后台地址及端口，默认为8085端口



2. 输入命令npm run dev启动服务



3. 启动成功后会自动打开浏览器




开发初探

第一个CURD

以后端项目中的学生管理为例，后端具体内容请参考文档：<https://gitee.com/xiaolifeize/cola-admin/wikis/%E5%90%8E%E7%AB%AF%E6%96%87%E6%A1%A3>

添加菜单

进入系统后点击“系统后台管理”--“菜单管理”，再点击新增按钮

 Cola-Admin

首页

订单管理

基础信息管理

系统权限管理

系统运维管理

系统后台管理

通知公告

字典管理

菜单管理

用户管理

机构管理

导航 / 系统后台管理 / 菜单管理






首页 菜单管理

请输入菜单名称 请输入菜单编码 请选择菜单类型 查询

+ 新增

修改

删除

	名称	图标	编码	url
<input type="checkbox"/>	> 订单管理		order	/order
<input type="checkbox"/>	> 基础信息管理		basics	/basics
<input type="checkbox"/>	> 系统权限管理		permission	/permission
<input type="checkbox"/>	> 系统运维管理		operation	/operation
<input type="checkbox"/>	> 系统后台管理		system	/system

填写菜单内容后点击确定按钮完成添加

添加菜单

×

父节点

根节点

* 菜单名称

学生信息管理

图标

el-icon-s-flag

* 菜单编码

studentInfo

* 组件

Layout

第一级菜单必填Layout

* URL

/studentInfo

菜单类型

菜单

打开类型

内置

是否隐藏

否

排序

-

0

+

取消

确定

添加子菜单，在列表中选中刚添加的学生信息管理菜单项，再点击新增按钮

请输入菜单名称

请输入菜单编码

请选择菜单类型

查询

重置

+ 新增

修改

删除

	名称	图标	编码	url	组件	菜单
<input type="checkbox"/>	> 订单管理		order	/order	Layout	菜单
<input checked="" type="checkbox"/>	学生信息管理		studentInfo	/studentInfo	Layout	菜单
<input type="checkbox"/>	> 基础信息管理		basics	/basics	Layout	菜单
<input type="checkbox"/>	> 系统权限管理		permission	/permission	Layout	菜单
<input type="checkbox"/>	> 系统运维管理		operation	/operation	Layout	菜单
<input type="checkbox"/>	> 系统后台管理		system	/system	Layout	菜单

填写子菜单信息

添加菜单



父节点

学生信息管理

* 菜单名称

学生管理

图标

 el-icon-s-flag

* 菜单编码

studentInfoStudent
上级编码 本级编码

* 组件

student_infostudent
上级编码加本级编码，单词前用下滑线

* URL

/studentInfo/student
上级url 本级url

菜单类型

菜单

打开类型

内置

是否隐藏

否

排序

-

0

+

取消

确定

最终效果如下





请输入菜单名称

请输入菜单编码

请选择菜单类型

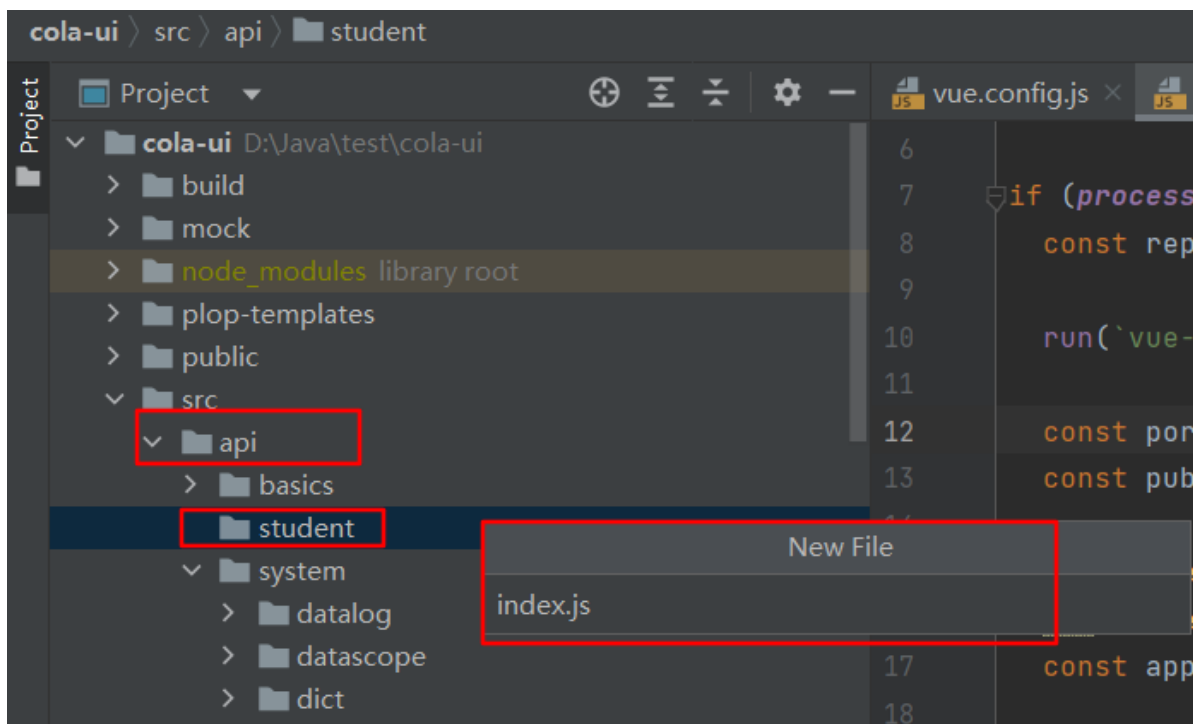
查询

重置

	名称	图标	编码	url	组件	菜单类型	打开方式	是否隐藏	排序
<input type="checkbox"/>	> 订单管理		order	/order	Layout	菜单	内置	否	0
<input type="checkbox"/>	√ 学生信息管理		studentInfo	/studentInfo	Layout	菜单	内置	否	0
<input type="checkbox"/>	学生管理		studentInfoStudent	/studentInfo/student	student_info_student	菜单	内置	否	0
<input type="checkbox"/>	> 基础信息管理		basics	/basics	Layout	菜单	内置	否	10

编写API

在src\api目录下新建student目录，并添加index.js文件



打开src\api\student\index.js

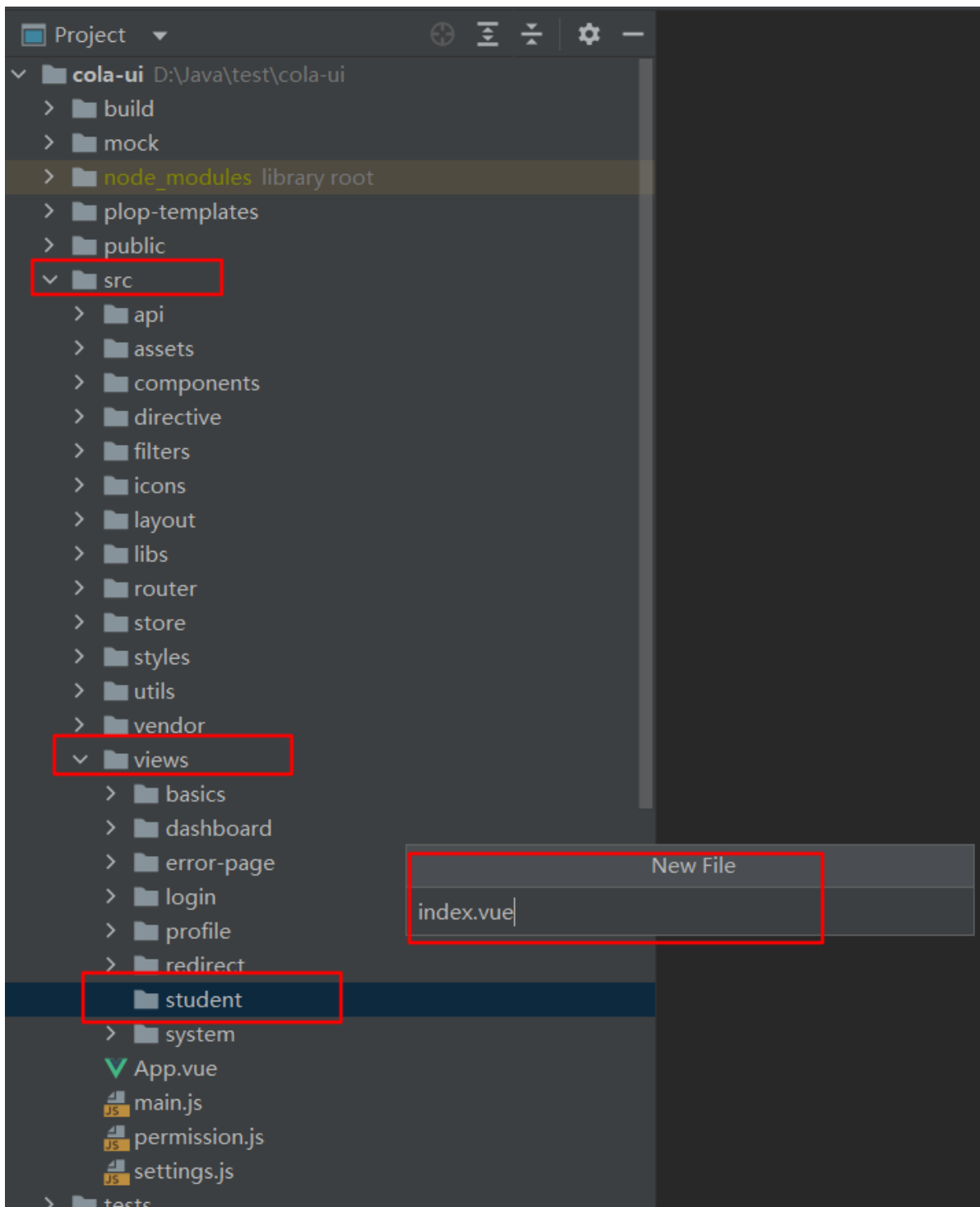
```
1  import request from '@utils/request'
2
3  /**
4   * 添加学生
5   * @param data
6   * @returns {*}
7   */
8  export function addStudent(data) {
9    return request({
10      url: '/student/addStudent',
11      method: 'post',
12      data
13    })
14  }
15
16  /**
17   * 修改学生信息
18   * @param data
19   * @returns {*}
20   */
21  export function editStudent(data) {
22    return request({
23      url: '/student/editStudent',
24      method: 'post',
25      data
26    })
27  }
28
29  /**
30   * 删除学生信息
31   * @param data
32   * @returns {*}
33   */
34  export function deleteStudent(data) {
35    return request({
```



```
36     url: '/student/deleteStudent',
37     method: 'post',
38     data
39   })
40 }
41
42 /**
43  * 分页查询
44  * @param data
45  * @returns {*}
46  */
47 export function getStudentPage(data) {
48   return request({
49     url: '/student/getStudentPage',
50     method: 'post',
51     data
52   })
53 }
54
```

编写主页面

打开src\views目录，新建student目录，然后再新建index.vue



index.vue为主页面，主要功能是查询列表的展示以及查询、修改、删除等组件的容器。复制如下代码到index.vue中

```
1 <template>
2   <!-- 主页面容器-->
3   <div class="app-container">
4     <!-- 查询容器-->
5     <table-query></table-query>
6     <!-- 表格工具条 -->
7     <table-header></table-header>
8     <!-- 表格 -->
9     <el-table
10      ref="indexTable"
11      v-loading="tableLoading"
12      :data="list"
13      fit
```

```

14     border
15     highlight-current-row
16     style="width: 100%"
17     class="indexTable"
18     size="mini"
19     @selection-change="handleSelectionChange"
20   >
21     <el-table-column align="center" type="selection" width="60" />
22     <!-- 显示字段 -->
23     <el-table-column label="id" prop="id" width="80" />
24     <el-table-column label="姓名" prop="name" width="200" />
25     <el-table-column label="年龄" prop="age" width="80" />
26     <el-table-column label="性别" prop="sex" width="80" />
27     <el-table-column label="地址" prop="address" />
28     <el-table-column label="创建时间" prop="createTime" width="150px" />
29     <el-table-column label="操作" width="120px" align="center">
30       <template v-slot="scope">
31         <div style="display: flex;">
32           <!-- 修改按钮 -->
33           <el-button type="primary" icon="el-icon-edit" size="mini"
@click="toEdit(scope.row)" />
34           <!-- 删除按钮 -->
35           <el-button type="danger" icon="el-icon-delete" size="mini"
@click="toDelete(scope.row)" />
36         </div>
37       </template>
38     </el-table-column>
39   </el-table>
40 </div>
41 </template>
42
43 <script>
44
45 import { getStudentPage } from '@/api/student'
46
47 export default {
48   name: 'Student',
49   data() {
50     return {
51       tableLoading: false,
52       list: []
53     }
54   },
55   mounted() {
56     // 增加刷新方法
57     this.$refs.indexTable.refresh = this.getTableData
58     // 页面加载完成后自动查询数据
59     this.getTableData()
60   },
61   methods: {
62     // 控制只能单选
63     handleSelectionChange(val) {
64       if (val.length > 1) {
65         this.$refs.indexTable.clearSelection()
66         this.$refs.indexTable.toggleRowSelection(val.pop())
67       }
68     },
69     getTableData() {

```

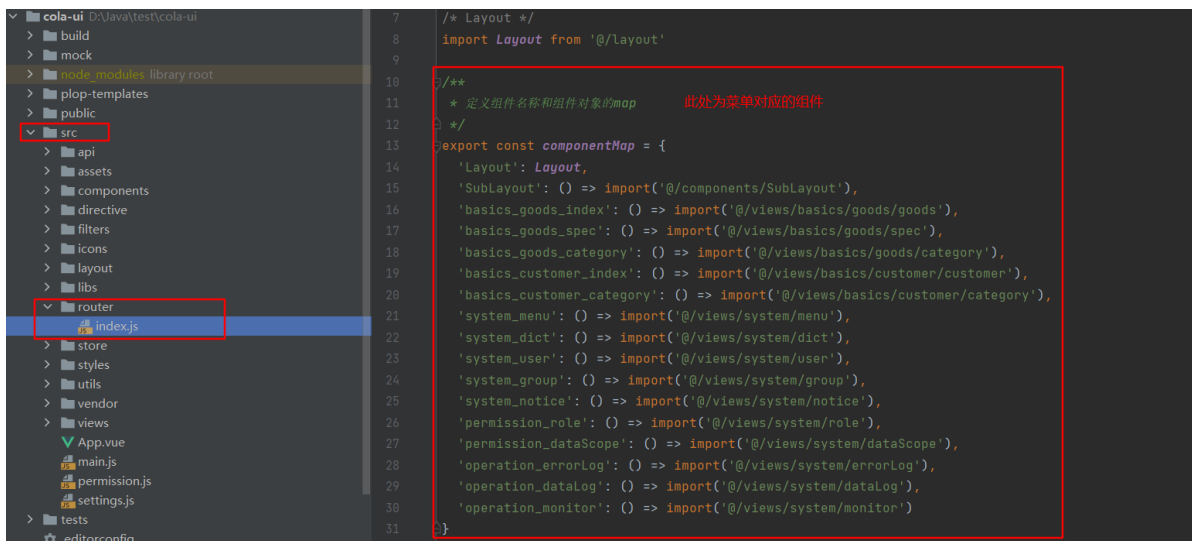
```

70     this.tableLoading = true
71     getStudentPage().then(response => {
72         if (response.data.page) {
73             this.list = response.data.page.records ?
response.data.page.records : []
74         }
75         this.tableLoading = false
76     }).catch(() => {
77         this.tableLoading = false
78     })
79 },
80 toEdit(row) {
81 },
82 toDelete(row) {
83 }
84 }
85 }
86 </script>
87
88 <style scoped>
89 /* 取消表头上的选择框 */
90 .indexTable /deep/ .el-table-column--selection.is-leaf .el-checkbox {
91     display: none;
92 }
93 </style>
94

```

添加路由

打开src\router\index.js



添加学生管理路由映射

$\left. \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right\}$

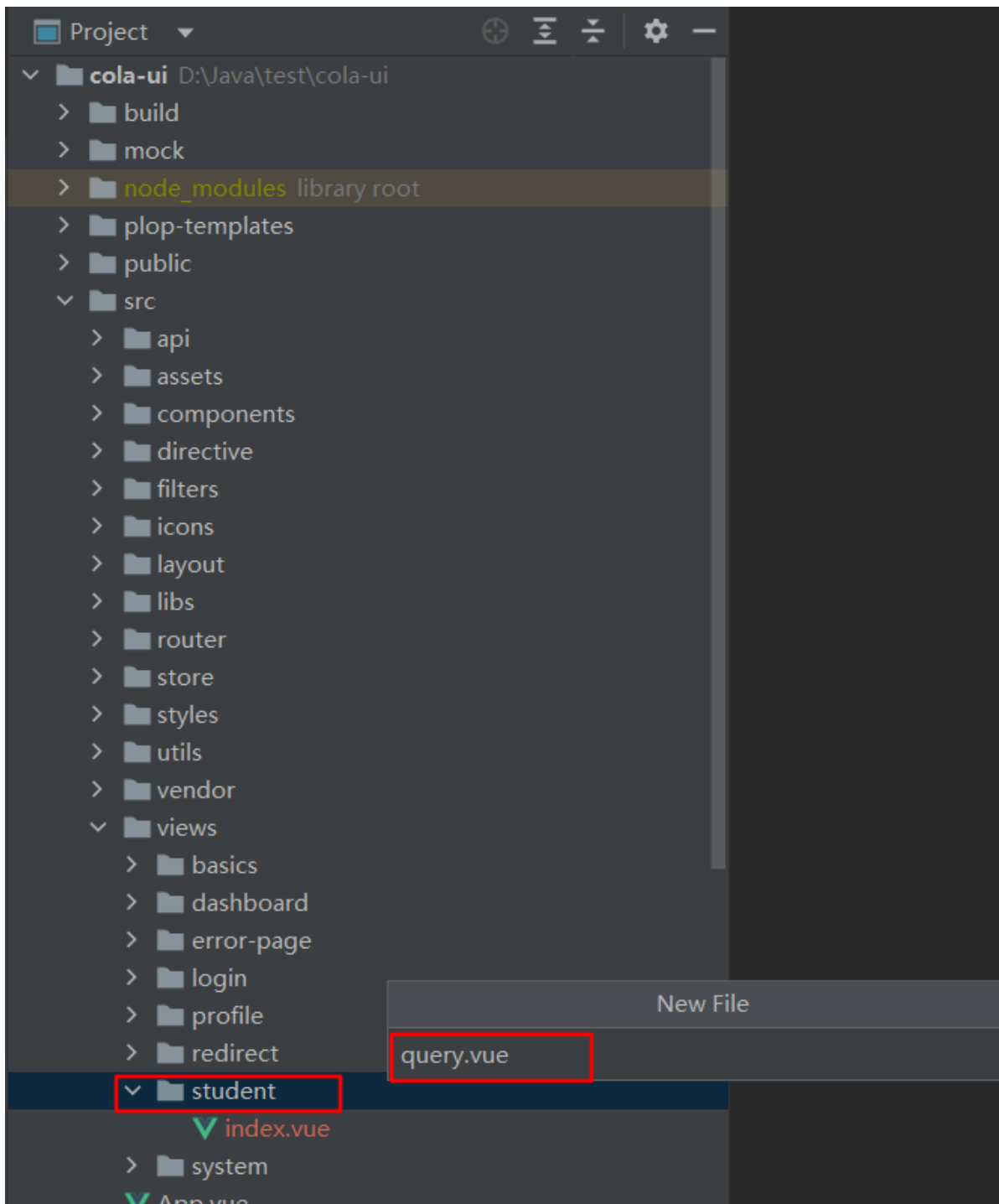

Cola-Admin

三
导航 / 学生信息管理 / 学生管理
个人中心

首页
订单管理
学生信息管理
学生管理
基础信息管理
系统权限管理

学生管理

	id	姓名	年龄	性别	地址	创建时间	操作
<input type="checkbox"/>	2	张三	16	1		2022-07-22 15:38:15	编辑 删除
<input type="checkbox"/>	3	王五	19	0		2022-07-22 15:38:15	编辑 删除
<input type="checkbox"/>	4	赵六	20	1		2022-07-22 15:38:16	编辑 删除
<input type="checkbox"/>	5	张生	19	0		2022-07-22 17:31:11	编辑 删除



2. 编写查询条件代码

```
1 <template>
2   <!-- 查询条件容器 -->
3   <div class="query-content">
4     <!-- 表单 -->
5     <el-form ref="queryForm" :model="query.data" :inline="true">
6       <!-- 查询条件 -->
7       <el-form-item prop="name">
8         <el-input v-model="query.data.name" :clearable="true"
placeholder="请输入学生姓名" />
9       </el-form-item>
10      <!-- 查询条件 -->
11      <el-form-item prop="menuType">
12        <el-select v-model="query.data.sex" :clearable="true"
placeholder="请选择性别">
13          <el-option label="女" :value="0" />
```

```

14         <el-option label="男" :value="1" />
15     </el-select>
16 </el-form-item>
17     <!-- 查询按钮 -->
18     <el-button type="primary" icon="el-icon-search" @click="doQuery"> 查
19 询 </el-button>
20     <el-button type="success" icon="el-icon-refresh-right"
21 @click="doReset"> 重 置 </el-button>
22 </el-form>
23 </div>
24 </template>
25
26 <script>
27 export default {
28   name: 'StudentQuery',
29   props: {
30     // 主页面传过来的查询对象
31     query: {
32       type: Object,
33       require: true,
34       default() {
35         return {
36           data: {
37             name: '',
38             sex: ''
39           },
40           currentPage: 1
41         }
42       }
43     },
44     data() {
45       return {
46         // 主页面table对象
47         table: {}
48       }
49     },
50     mounted() {
51       // 获取父页面table对象
52       this.table = this.$parent.$parent.$children.find(vc => vc.tableId)
53     },
54     methods: {
55       doQuery() {
56         // 调用父页面刷新方法执行查询操作
57         this.table.refresh()
58       },
59       doReset() {
60         // 重置时清空表单信息
61         this.$refs.queryForm.resetFields()
62         // 重置时重新刷新
63         this.table.refresh()
64       }
65     }
66   }
67 }
68 </script>
69
70 <style scoped>

```

```
70 | </style>
71 |
```

3. 在主页面中添加查询条件组件和查询条件对象

```
5 | import { getStudentPage } from '@api/student'
6 | import Query from './query' 引入查询条件组件
7 |
8 | export default {
9 |   name: 'Student',
10 |   components: { Query }, 引入查询组件
11 |   data() {
12 |     return {
13 |       tableLoading: false,
14 |       list: [],
15 |       query: { 查询条件对象
16 |         conditions: [ 表示data中的name字段使用like查询
17 |           { name: 'name', keyword: 'like' }
18 |         ],
19 |         data: {
20 |           name: '',
21 |           sex: '' | 默认是=号查询
22 |         },
23 |         total: 0, 总条数
24 |         currentPage: 0, 当前第几页
25 |         pageSize: 20 每页条数
26 |       } 分页属性
27 |     }
28 |   },
29 | }
```

4. 页面中添加引用

```
✓ student\index.vue x
1 | <template>
2 |   <!-- 主页面容器-->
3 |   <div class="app-container"> 引用查询组件
4 |     <!-- 查询容器-->
5 |     <table-query> 传递查询条件
6 |       <query :query.sync="query" />
7 |     </table-query>
8 |     <!-- 表格工具条容器 -->
9 |     <table-header></table-header>
10 |    <!-- 表格 -->
11 |    <el-table
```

5. 修改查询方法，将查询条件对象传入

```
getTableData() {
  this.tableLoading = true
  getStudentPage(this.query).then(response => {
    if (response.data.page) {
      this.list = response.data.page.records ? response.data.page.records : []
    }
    this.tableLoading = false
  }).catch(() => {
    this.tableLoading = false
  })
}
```

6. 最终效果

Cola-Admin

首页

订单管理

学生信息管理

学生管理

基础信息管理

系统权限管理

导航 / 学生信息管理 / 学生管理

超级管理员

学生管理

请输入学生姓名

请选择性别

查询

重置

	id	姓名	年龄	性别	地址	创建时间	操作
<input type="checkbox"/>	2	张三	16	1		2022-07-22 15:38:15	编辑 删除
<input type="checkbox"/>	3	王五	19	0		2022-07-22 15:38:15	编辑 删除
<input type="checkbox"/>	4	赵六	20	1		2022-07-22 15:38:16	编辑 删除
<input type="checkbox"/>	5	张生	19	0		2022-07-22 17:31:11	编辑 删除

7. 测试查询

Cola-Admin

首页

订单管理

学生信息管理

学生管理

基础信息管理

系统权限管理

导航 / 学生信息管理 / 学生管理

超级管理员

学生管理

张

请选择性别

查询

重置

	id	姓名	年龄	性别	地址	创建时间	操作
<input type="checkbox"/>	2	张三	16	1		2022-07-22 15:38:15	编辑 删除
<input type="checkbox"/>	5	张生	19	0		2022-07-22 17:31:11	编辑 删除

新增功能