江西理工大学应用科学学院

安全帽用户手册

[建筑工地安全管控系统]

**系 别： 信息工程系**

**班 级： 2018级嵌入式实验班**

**开发单位： 工地软件组**

**技术总监： 谢小云**

**时 间： 2021年6月27日**

文档编号 01

版本号

文档名称 安全帽用户手册

项目名称 建筑工地安全管控系统

项目负责人 张连鑫

编写 刘润发 2021 年 6 月 27 日

校对 年 月 日

批准 年 月 日

开发单位  年 月 日

**版 本 历 史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **起止日期** | **备注** |
| 01 | 刘润发 |  | 2019.6.21——2019.6.27 |  |
|  |  |  |  |  |
|  |  |  |  |  |

**内容提要**

本文为“工地安全管控系统安全帽”的使用手册。在本文中，介绍了安全帽的主要功能、特点、系统结构；它具有以下功能，当你在安全管控系统点击安全帽模块按钮，你会进入安全帽登录页面，在首页你可以看到电子围栏、人员定位、人员体温、报警、签到、人员管理、设备管理等按钮点击各个按钮就可以查看各种信息，可以对各个管理和信息进行设置与修改，你还可以实时查看各种信息的变化。

**使用说明**

本手册使用WPS格式，用户可以安装可阅读WPS文档的相关工具软件，推荐安装WPS office。文档可通过WPS officer等软件的“标签”进行主题的快速跳转。

**版本说明**

使用手册版本：V0.1

**版权申明**

本软件由工地软件组开发，任何个人和单位需要使用本软件，均应获得授权后才能使用，否则视为侵犯版权。

**目 录**

[**目 录** 4](#_Toc75806946)

[1、系统功能特点与结构 1](#_Toc75806947)

[1.1 主要功能 1](#_Toc75806948)

[1.2 系统特点 1](#_Toc75806949)

[1.3 系统结构 1](#_Toc75806950)

[系统结构 1](#_Toc75806951)

[2、系统部署 1](#_Toc75806952)

[2.1 安装要求 2](#_Toc75806953)

[2.1.1 硬件配置 2](#_Toc75806954)

[2.1.2 系统软件配置 2](#_Toc75806955)

[2.2 部署步骤 2](#_Toc75806956)

[2.2.1服务器部署 2](#_Toc75806957)

[2.2.2 配套软件的部署 2](#_Toc75806958)

[3、 使用方法： 2](#_Toc75806959)

[3.1 操作界面结构 2](#_Toc75806960)

[4、出错处理和恢复 7](#_Toc75806961)

[5、系统升级与维护 8](#_Toc75806962)

1、系统功能特点与结构

1.1 主要功能

（1）首页：登录之后先看到首页信息，上面为一张包含给弟弟范围的大地图，可以看到一星期内的报警情况与当前员工在线数。

（2）电子围栏：点击电子围栏按钮，你可以查看自己所设置的围栏及范围，你可以对围栏进行修改与删除。

（3）人员定位：帽子上自带GPS定位系统，点击该按钮你可以查看每一个员工的所在地，并可以查看他们的运动轨迹。

（4）人员体温：帽子上安装有温度传感器，点击人员体温，你可以查看每个员工的体温情况。

（5）报警：上面装有自动报警系统，只要工地人员出现SOS、脱帽、跌落、体温异常、靠近围栏等情况，报警就会启动。

（6）签到：人员戴安全帽从入口进入工地，系统会对该人员进行签到处理。

（7）人员管理：可查看佩戴安全帽的人员信息，如工号、姓名、年龄等。

（8）设备管理：记录了安全帽的基本信息，如id、编号、使用者、生产日期、生产地址等。

1.2 系统特点

本系统具有如下的特点：

（1）方便性

系统利用网络进行数据的监控，不用人为的去实施检查，出现问题可以自动进行报警预防。

（2）自动性

根据设备上传的数据自动存储并转发给客户端，实时展示动态数据，无需用户介入操作，你就可以看到数据的变化。

（3）实时性

该系统能够第一时间得到现场数据并进行展现，用户只需要登录点击自己想要知道的一些情况就得到现场的数据。

1.3 系统结构

系统结构

管理员在网页上进行登录注册，可以查看员工的考勤信息、人员当前位置、人员当前的体温状况、报警次数、设备的一些使用情况和一些员工基本信息；登录之后你可以对很多东西进行修改，比如新增加员工的信息的添加，你可以通过地图对工地范围进行修改，使电子围栏更加精确；对设备信息进行查看，以做更新互换，防止人员的事故；系统自动报警，减少了管理员的工作量保障了员工的人生安全。

2、系统部署

本章介绍服务器的部署方法，将从安装要求和部署步骤两个方面进行详细介绍。

2.1 安装要求

本服务器安装所需最基本的配置。

2.1.1 硬件配置

服务器的基础配置即可，硬盘空间20GB左右，内存2GB以上。

2.1.2 系统软件配置

利用窄带物联网NB-IoT技术,实现数据的直接“上云”。通过Vue.js框架搭建用户界面，Servlet容器使用SpringBoot框架内嵌的Tomcat容器。通过WebSocket连接来将平台接收的数据实时转发到websocket客户端，使用MQTT队列订阅的形式来获取设备上传的数据。数据库使用SpringBoot jpa 映射自动生成数据表，通过node.js实现前后交互。

2.2 部署步骤

2.2.1服务器部署

将springboot打包成可执行的jar文件，使用scp命令将文件上传至服务器，在服务器中安装docker环境，通过编写的Dockerfile来构建运行时镜像，同时创建mysql镜像，通过容器中的相互通信来实现程序的运行，同时打开防火墙的对应端口提供外部访问。

2.2.2 配套软件的部署

将前端Vuejs项目生成生产环境下的代码，将其中的dist文件夹拷贝到服务器中，将文件夹构建到Docker镜像中，提供外部访问的端口映射，再使用Http-server或nginx来搭建静态的页面访问。

1. 使用方法：

3.1 操作界面结构

如图3.1安全帽登录页面，输入账号密码进行登录



图3.1

如图3.2首页页面，在这个页面，你可以看工地所在地区的大地图，，你还可以报警信息，如谁报的警，什么原因报警。你还可以通过首页进入其他页面，然后进行相关操作。



图3.2

如图3.3电子围栏页面，在此页面，你可以看到围栏范围和围栏信息，你不仅可以查看有多少个围栏，你还可以自定义划分围栏，和改变原来的围栏范围，你还可以删除围栏。



图3.3

如图3.4，为人员定位，在此页面，你可以看到一张工地范围的大地图，在大地图上，你可以看到每一个工人的信息，以及他当前所在的位置，你可以看到每一个工人在的工作范围。



图3.4

如图3.5人员体温，下图你可以看到，它可以查看每一个工人的基本信息和他的现在温度情况，它会记录，你也可以查看每一个你需要查看的工人的体温，系统会帮你筛选出你所要查询的信息。



图3.5

如图3.6为报警页面，通过图片你可以看有哪些报警开关，和当前开关的报警情况，你还可以看到历史的报警信息，如是谁报的警，何时报的警，因什么报警，报警时间等。



图3.6

如下图3.7为签到页面，在此页面，你可以看到每一个的签到情况，如你可以看到签到人的具体时间，它的工号，签到的次数，历史签到记录；而且你还可以通过地图看到签到范围和增加签到区域。



图3.7

如图3.8为人员管理界面，在此页面，你可以看到每一个员工的基本信息，如你可以id、姓名、性别、身份证号、家庭住址、联系电话、所在部门、职务等；你还可以单独查看一个工人的信息，你还可以对信息工人进行修改和删除。



图3.8

如图3.9为设备管理界面，在此界面，你可以看到你所用设备的基本信息，你还可以查看设备型号和当前所佩戴的人员，你可以通过此页面看到安全帽的损害情况和基本信息。



图3.9

4、出错处理和恢复

数据库不存在时或数据库账号密码程序会启动失败，只需创建对应的数据库即可，数据表会在生成的过程中自动创建起来。

5、系统升级与维护

系统运行时间过长的话输出的日志文件会比较多，对无用的一些日志可以每周清理一次，防止服务器硬盘空间不足

**安全帽系统代码**

前端代码：

<template>  
 <div>  
 <div class="switchBox boxBorder">  
 <div class="title fontStyle">{{title}}报警</div>  
 <div class="switch">  
 <span class="fontStyle">报警开关</span>  
 <el-switch  
 v-model="alarmSwitchData.switch\_state"  
 @change="getState"  
 :active-value = 1  
 :inactive-value = 0  
 active-color="#13ce66">  
 </el-switch>  
 </div>  
 <div class="footerBorder"></div>  
 </div>  
 </div>  
</template>  
<script>  
import http from '@/utils/http'  
export default {  
 name: 'alarmSwitch',  
 props: ['alarmSwitchData'],  
 data () {  
 return {  
 title: this.alarmSwitchData.switch\_name  
 }  
 },  
 methods: {  
 getState (state) {  
 http.post('/warning-switch/setBYid', { switch\_id: this.alarmSwitchData.switch\_id, switch\_state: state })  
 .then(resp => {  
 if (state === 1) {  
 this.$message.success('开启' + resp.message)  
 }  
 if (state === 0) {  
 this.$message.success('关闭' + resp.message)  
 }  
 })  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
.switchBox {  
 width: 270px;  
 height: 160px;  
 .title {  
 font-size: 24px;  
 text-align: center;  
 top: 30px;  
 position: relative  
 }  
 .switch {  
 position: relative;  
 top: 40px;  
 text-align: center;  
 span {  
 margin-right: 20px;  
 font-size: 20px;  
 }  
 }  
}  
</style>

<template>  
 <baidu-map class="map" :center="center" :scroll-wheel-zoom="true" :zoom="zoom" @click="addMarker" @ready="handler">  
 <bm-polygon  
 :path="polygonPath"  
 fillColor="#a0cff6"  
 :editing="isEditing"  
 stroke-color="#fdd1f5"  
 :stroke-opacity="0"  
 :stroke-weight="8"  
 @lineupdate="updatePolygonPath">  
 </bm-polygon>  
 <bm-marker v-for="(item,index) of positionData" :mass-clear="true" :position="item" :key="index"></bm-marker>  
 </baidu-map>  
</template>  
<script>  
export default {  
 name: 'addElectricFence',  
 data () {  
 return {  
 current: 0,  
 polygonPath: [],  
 positionData: [],  
 center: '',  
 zoom: 0,  
 isEditing: false,  
 removeClick: true,  
 isSelect: true  
 }  
 },  
 methods: {  
 addMarker (e) {  
 this.isSelect = false  
 this.$emit('ListenAddMarker', this.isSelect)  
 if (this.current < 5 && this.removeClick === true) {  
 this.positionData.push(e.point)  
 this.current += 1  
 }  
 if (this.current === 5) {  
 this.$message.error('只能添加五个点')  
 }  
 },  
 handler ({ BMap, map }) {  
 ***console***.log(BMap, map)  
 this.center = '赣南科技学院'  
 this.zoom = 17  
 },  
 polygon () {  
 this.polygonPath = this.positionData  
 this.$emit('ListenAddFence', this.polygonPath)  
 this.removeClick = false  
 this.positionData = []  
 },  
 clearMarker () {  
 this.positionData = []  
 this.polygonPath = []  
 this.current = 0  
 this.removeClick = true  
 this.isEditing = false  
 },  
 startEditPolygon () {  
 this.isEditing = true  
 this.$emit('ListenAddFence', this.polygonPath)  
 },  
 updatePolygonPath (e) {  
 this.polygonPath = e.target.getPath()  
 },  
 endEditPolygon () {  
 this.isEditing = false  
 this.$emit('ListenAddFence', this.polygonPath)  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
.map {  
 width: 100%;  
 height: 100%;  
}  
/deep/.BMap\_cpyCtrl {  
 display:none;  
}  
/deep/.anchorBL {  
 display:none;  
}  
</style>

<template>  
 <baidu-map class="map" :center="center" :scroll-wheel-zoom="true" :zoom="zoom" @ready="handler">  
 <bm-polygon  
 :path="polygonPathData.point"  
 fill-color="#ff7675"  
 stroke-color="blue"  
 :stroke-opacity="0"  
 :stroke-weight="2">  
 </bm-polygon>  
 </baidu-map>  
</template>  
<script>  
export default {  
 name: 'electricFenceMap',  
 props: ['polygonPathData'],  
 data () {  
 return {  
 center: {},  
 zoom: 0  
 }  
 },  
 methods: {  
 handler ({ BMap, map }) {  
 ***console***.log(BMap, map)  
 this.center = '赣南科技学院'  
 this.zoom = 17  
 },  
 getCurrentCenter (val) {  
 this.center = val  
 ***console***.log(val)  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
.map {  
 width: 100%;  
 height: 100%;  
}  
/deep/.BMap\_cpyCtrl {  
 display:none;  
}  
/deep/.anchorBL {  
 display:none;  
}  
</style>

<template>  
 <div>  
 <el-table  
 :data="waringTableData"  
 height="520px"  
 empty-text="暂无报警信息"  
 :show-header="false"  
 style="width: 100%">  
 <el-table-column  
 prop="user\_name"  
 label="姓名"  
 align="center"  
 width="100">  
 </el-table-column>  
 <el-table-column  
 prop="warning\_time"  
 label="时间"  
 align="center"  
 width="130">  
 </el-table-column>  
 <el-table-column  
 prop="warning\_name"  
 align="center"  
 :show-overflow-tooltip="true"  
 label="类型">  
 </el-table-column>  
 </el-table>  
 </div>  
</template>  
<script>  
import http from '@/utils/http'  
export default {  
 name: 'alarmInfo',  
 data () {  
 return {  
 waringTableData: []  
 }  
 },  
 created () {  
 http.get('/index/getWaring').then(resp => {  
 this.waringTableData = resp.data  
 })  
 },  
 methods: {  
 }  
}  
</script>  
<style scoped lang="less">  
</style>

<template>  
 <div>  
 <div :id="lineChart" style="width: 650px;height: 130px"></div>  
 </div>  
</template>  
<script>  
import \* as echarts from 'echarts'  
export default {  
 name: 'lineChart',  
 data () {  
 return {  
 charts: null,  
 lineChart: 'lineChart'  
 }  
 },  
 beforeDestroy () {  
 if (!this.charts) {  
 return ''  
 }  
 this.charts.dispose()  
 this.charts = null  
 },  
 mounted () {  
 this.initCharts()  
 },  
 methods: {  
 initCharts () {  
 this.charts = echarts.init(***document***.getElementById(this.lineChart))  
 const fontColor = '#30eee9'  
 const options = {  
 title: {  
 text: '七天内\n报警变化'  
 },  
 grid: {  
 left: '25%',  
 right: '10%',  
 top: '25%',  
 bottom: '15%'  
 // containLabel: true  
 },  
 tooltip: {  
 show: true,  
 trigger: 'item'  
 },  
 xAxis: [{  
 type: 'category',  
 boundaryGap: false,  
 axisLabel: {  
 color: fontColor  
 },  
 axisLine: {  
 show: true,  
 lineStyle: {  
 color: '#397cbc'  
 }  
 },  
 axisTick: {  
 show: false  
 },  
 splitLine: {  
 show: false,  
 lineStyle: {  
 color: '#195384'  
 }  
 },  
 data: ['6月5号', '6月6号', '6月7号', '6月8号', '6月9号', '6月10号', '6月11号']  
 }],  
 yAxis: [{  
 type: 'value',  
 name: '',  
 min: 0,  
 max: 100,  
 axisLabel: {  
 formatter: '{value}',  
 textStyle: {  
 color: '#2ad1d2'  
 }  
 },  
 axisLine: {  
 lineStyle: {  
 color: '#27b4c2'  
 }  
 },  
 axisTick: {  
 show: false  
 },  
 splitLine: {  
 show: true,  
 lineStyle: {  
 color: '#11366e'  
 }  
 }  
 }],  
 series: [{  
 name: '报警数',  
 type: 'line',  
 symbol: 'circle',  
 symbolSize: 8,  
 itemStyle: {  
 normal: {  
 color: '#0092f6',  
 lineStyle: {  
 color: '#0092f6',  
 width: 1  
 }  
 }  
 },  
 markPoint: {  
 data: [{  
 type: 'max',  
 name: '最大值',  
 symbolSize: 20  
 },  
 {  
 type: 'min',  
 name: '最小值',  
 symbolSize: 20  
 }],  
 itemStyle: {  
 normal: {  
 color: '#0092f6'  
 }  
 }  
 },  
 data: [12, 13, 10, 34, 90, 30, 20, 82, 11, 34, 90, 33]  
 }]  
 }  
 this.charts.setOption(options)  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
</style>

<template>  
 <baidu-map class="map" :center="center" :zoom="zoom" :scroll-wheel-zoom="true" @ready="handler">  
 <bm-marker  
 :dragging="true"  
 v-for="(item, index) of markerPoint"  
 @click="infoWindowOpen"  
 :position="item.point"  
 :key="index"  
 :icon="{url: 'http://10.168.200.65:8082/hys/icon/anquanmao.png', size: {width: 32, height: 26}}">  
 <bm-info-window :show="show" @close="infoWindowClose" @open="infoWindowOpen">{{content}}</bm-info-window>  
 </bm-marker>  
 </baidu-map>  
</template>  
<script>  
export default {  
 name: 'locationMap',  
 data () {  
 return {  
 show: false,  
 content: '安全帽',  
 markerPoint: [{  
 user\_id: 1,  
 point: {  
 lng: 114.922,  
 lat: 25.8452  
 }  
 }],  
 center: '',  
 zoom: 0  
 }  
 },  
 methods: {  
 handler ({ BMap, map }) {  
 ***console***.log(BMap, map)  
 this.center = '赣南科技学院'  
 this.zoom = 15  
 },  
 infoWindowClose () {  
 this.show = false  
 },  
 infoWindowOpen () {  
 this.show = true  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
/deep/.BMap\_cpyCtrl {  
 display:none;  
}  
/deep/.anchorBL {  
 display:none;  
}  
</style>

<template>  
 <div>  
 <div :id="pieChart" style="height: 130px;width: 400px"></div>  
 </div>  
</template>  
<script>  
import \* as echarts from 'echarts'  
export default {  
 name: 'pieChart',  
 data () {  
 return {  
 charts: null,  
 pieChart: 'pieChart',  
 pieData: [  
 { value: 1048, name: '在线数量' },  
 { value: 735, name: '未在线数量' }  
 ]  
 }  
 },  
 beforeDestroy () {  
 if (!this.charts) {  
 return ''  
 }  
 this.charts.dispose()  
 this.charts = null  
 },  
 mounted () {  
 this.initCharts()  
 },  
 methods: {  
 initCharts () {  
 this.charts = echarts.init(***document***.getElementById(this.pieChart))  
 const pieData = this.pieData  
 const options = {  
 color: ['#46cdbd', '#3f565f'],  
 title: {  
 text: '安全帽在线数',  
 left: 'left'  
 },  
 tooltip: {  
 trigger: 'item'  
 },  
 grid: {  
 left: 100  
 },  
 legend: {  
 orient: 'vertical',  
 left: 300  
 },  
 series: [  
 {  
 center: ['50%', '50%'],  
 name: '人数',  
 type: 'pie',  
 radius: '50%',  
 data: pieData,  
 emphasis: {  
 itemStyle: {  
 shadowBlur: 10,  
 shadowOffsetX: 0,  
 shadowColor: 'rgba(0, 0, 0, 0.5)'  
 }  
 }  
 }  
 ]  
 }  
 this.charts.setOption(options)  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
</style>

<template>  
 <baidu-map class="map" :center="center" :zoom="zoom" :scroll-wheel-zoom="true" @ready="handler">  
 <bml-lushu  
 :path="path"  
 :play="true"  
 :speed="500"  
 @stop="reset"  
 :rotation="true">  
 </bml-lushu>  
 </baidu-map>  
</template>  
<script>  
import { BmlLushu } from 'vue-baidu-map'  
export default {  
 name: 'positionMap',  
 components: { BmlLushu },  
 data () {  
 return {  
 content: '安全帽',  
 center: '',  
 path: [],  
 zoom: 0  
 }  
 },  
 created () {  
 },  
 methods: {  
 handler ({ BMap, map }) {  
 ***console***.log(BMap, map)  
 this.center = '赣南科技学院'  
 this.zoom = 17  
 },  
 reset () {  
 this.path = [{  
 lng: 116.297611,  
 lat: 40.047363  
 }, {  
 lng: 116.302839,  
 lat: 40.048219  
 }, {  
 lng: 116.308301,  
 lat: 40.050566  
 }, {  
 lng: 116.305732,  
 lat: 40.054957  
 }, {  
 lng: 116.304754,  
 lat: 40.057953  
 }, {  
 lng: 116.306487,  
 lat: 40.058312  
 }, {  
 lng: 116.307223,  
 lat: 40.056379  
 }]  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
/deep/.BMap\_cpyCtrl {  
 display:none;  
}  
/deep/.anchorBL {  
 display:none;  
}  
</style>

<template>  
 <div>  
 <div class="titleContainer"><img src="../../assets/images/globalImages/安全帽管理系统.png" alt=""></div>  
 <div class="navContainer">  
 <div class="navImgContainer">  
 <img :id="homeId" :class="{imgWidth: homeId===current}" @click="toHome" src="../../assets/images/globalImages/首页.png" alt="">  
 <img :class="{imgWidth: electricFenceId===current}" @click="toElectricFence" src="../../assets/images/globalImages/电子围栏.png" alt="">  
 <img :class="{imgWidth: personnelPositionId===current}" @click="toPersonnelPosition" src="../../assets/images/globalImages/人员定位.png" alt="">  
 <img :class="{imgWidth: personnelTemperatureId===current}" @click="toPersonnelTemperature" src="../../assets/images/globalImages/人员体温.png" alt="">  
 <img :class="{imgWidth: alarmId===current}" @click="toAlarm" style="margin-left: 540px" src="../../assets/images/globalImages/报警.png" alt="">  
 <img :class="{imgWidth: signId===current}" @click="toSign" src="../../assets/images/globalImages/签到.png" alt="">  
 <img :class="{imgWidth: personnelManagementId===current}" @click="toPersonnelManagement" src="../../assets/images/globalImages/人员管理.png" alt="">  
 <img :class="{imgWidth: deviceManagement===current}" @click="toDeviceManagement" src="../../assets/images/globalImages/设备管理.png" alt="">  
 </div>  
 </div>  
 </div>  
</template>  
<script>  
export default {  
 name: 'navigationBar',  
 data () {  
 return {  
 current: 0,  
 homeId: 1,  
 electricFenceId: 2,  
 personnelPositionId: 3,  
 personnelTemperatureId: 4,  
 alarmId: 5,  
 signId: 6,  
 personnelManagementId: 7,  
 deviceManagement: 8  
 }  
 },  
 created () {  
 this.current = 1  
 },  
 methods: {  
 toHome () {  
 this.current = 1  
 this.$router.push('/home')  
 },  
 toElectricFence () {  
 this.current = 2  
 this.$router.push('/electricFence')  
 },  
 toPersonnelPosition () {  
 this.current = 3  
 this.$router.push('/personnelPosition')  
 },  
 toPersonnelTemperature () {  
 this.current = 4  
 this.$router.push('/personnelTemperature')  
 },  
 toAlarm () {  
 this.current = 5  
 this.$router.push('/alarm')  
 },  
 toSign () {  
 this.current = 6  
 this.$router.push('/sign')  
 },  
 toPersonnelManagement () {  
 this.current = 7  
 this.$router.push('/personnelManagement')  
 },  
 toDeviceManagement () {  
 this.current = 8  
 this.$router.push('/deviceManagement')  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
.titleContainer {  
 min-width: 1500px;  
 max-width: 1920px;  
 img {  
 width: 1536px;  
 height: 70px;  
 }  
}  
.navContainer {  
 min-width: 1500px;  
 max-width: 1920px;  
 display: flex;  
 padding: 5px;  
 .navImgContainer {  
 height: 40px;  
 img {  
 width: 100px;  
 margin-left: 20px;  
 cursor: pointer;  
 }  
 .imgWidth {  
 width: 120px;  
 }  
 }  
}  
</style>

<template>  
 <baidu-map class="map" :center="center" :scroll-wheel-zoom="true" :zoom="zoom" @click="addMarker" @ready="handler">  
 <bm-polygon  
 :path="polygonPath"  
 fillColor="#a0cff6"  
 :editing="isEditing"  
 stroke-color="#fdd1f5"  
 :stroke-opacity="0"  
 :stroke-weight="8"  
 @lineupdate="updatePolygonPath">  
 </bm-polygon>  
 <bm-marker v-for="(item,index) of positionData" :position="item" :key="index"></bm-marker>  
 </baidu-map>  
</template>  
<script>  
export default {  
 name: 'addSignMap',  
 data () {  
 return {  
 current: 0,  
 polygonPath: [],  
 positionData: [],  
 center: {},  
 zoom: 0,  
 isEditing: false,  
 newSelect: true,  
 removeClick: true  
 }  
 },  
 methods: {  
 addMarker (e) {  
 this.newSelect = false  
 this.$emit('ListenAddSignMarker', this.newSelect)  
 if (this.current < 5 && this.removeClick === true) {  
 this.positionData.push(e.point)  
 this.current += 1  
 }  
 if (this.current === 5) {  
 this.$message.error('只能添加五个点')  
 }  
 },  
 handler ({ BMap, map }) {  
 ***console***.log(BMap, map)  
 this.center = '赣南科技学院'  
 this.zoom = 17  
 },  
 polygon () {  
 this.polygonPath = this.positionData  
 this.$emit('ListenAddSignArea', this.polygonPath)  
 this.removeClick = false  
 this.positionData = []  
 },  
 clearMarker () {  
 this.positionData = []  
 this.polygonPath = []  
 this.current = 0  
 this.removeClick = true  
 this.isEditing = false  
 },  
 startEditPolygon () {  
 this.isEditing = true  
 this.$emit('ListenAddSignArea', this.polygonPath)  
 },  
 updatePolygonPath (e) {  
 this.polygonPath = e.target.getPath()  
 },  
 endEditPolygon () {  
 this.isEditing = false  
 this.$emit('ListenAddSignArea', this.polygonPath)  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
.map {  
 height: 500px;  
}  
/deep/.BMap\_cpyCtrl {  
 display:none;  
}  
/deep/.anchorBL {  
 display:none;  
}  
</style>

<template>  
 <baidu-map class="map" :center="center" :scroll-wheel-zoom="true" :zoom="zoom" @ready="handler">  
 <bm-polygon  
 :path="polygonPathData.point"  
 fill-color="#74b9ff"  
 stroke-color="#fd79a8"  
 :stroke-opacity="0"  
 :stroke-weight="2">  
 </bm-polygon>  
 </baidu-map>  
</template>  
<script>  
export default {  
 name: 'signMap',  
 props: ['polygonPathData'],  
 data () {  
 return {  
 center: {},  
 zoom: 0  
 }  
 },  
 methods: {  
 handler ({ BMap, map }) {  
 ***console***.log(BMap, map)  
 this.center = '赣南科技学院'  
 this.zoom = 17  
 },  
 getCurrentCenter (val) {  
 this.center = val  
 ***console***.log(val)  
 }  
 }  
}  
</script>  
<style scoped lang="less">  
/deep/.BMap\_cpyCtrl {  
 display:none;  
}  
/deep/.anchorBL {  
 display:none;  
}  
</style>

后端代码：

import com.qrs.pojo.Department;  
import com.qrs.pojo.RespBean;  
import com.qrs.service.IDepartmentService;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.List;  
@RestController  
@RequestMapping("//department")  
@Api(value = "安全帽部门模块" ,tags = "hat部门接口")  
public class DepartmentController {  
 @Autowired  
 private IDepartmentService departmentService;  
 @GetMapping("/list")  
 @ApiOperation(value = "获得部门列表")  
 public RespBean getList(){  
 List<Department> list = departmentService.list();  
 return RespBean.*success*("成功",list);  
 }  
}

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.qrs.pojo.DTO.EquipmentDTO;  
import com.qrs.pojo.DTO.PageDTO;  
import com.qrs.pojo.Equipment;  
import com.qrs.pojo.RespBean;  
import com.qrs.pojo.User;  
import com.qrs.service.IEquipmentService;  
import com.qrs.service.IUserService;  
import com.qrs.utils.PageUtils;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.Collections;  
import java.util.List;  
@RestController  
@RequestMapping("//equipment")  
@Api(value = "安全帽模块", tags = "hat设备管理接口")  
public class EquipmentController {  
 @Autowired  
 private IEquipmentService equipmentService;  
 @Autowired  
 private IUserService userService;  
 @GetMapping("/getSize")  
 @ApiOperation("获得设备数")  
 public RespBean getSize(){  
 List<Equipment> list = equipmentService.list();  
 return RespBean.*success*("成功",list.size());  
 }  
 @PostMapping("/getList")  
 @ApiOperation("查询全部设备,分页")  
 public RespBean getList(@RequestBody PageDTO pageDTO){  
 int current=pageDTO.getCurrent();  
 int size=pageDTO.getSize();  
 List<Object>list=Collections.*singletonList*(equipmentService.getList());  
 List<Object> page = PageUtils.*page*(list, current, size);  
 return RespBean.*success*("成功",page);  
 }  
 @PostMapping("/equipment")  
 @ApiOperation("查询单个设备")  
 public RespBean getListById(@RequestBody String ch){  
 String[] split = ch.split("**\"**");  
 ch=split[1];  
 List<EquipmentDTO>equipment= equipmentService.getSelectById(ch);  
 return RespBean.*success*("成功",equipment);  
 }  
 @PostMapping("/add")  
 @ApiOperation("添加设备")  
 public RespBean add(@RequestBody Equipment equipment){  
 QueryWrapper<User> wrapper = new QueryWrapper();  
 wrapper.eq("user\_id",equipment.getUser\_id());  
 List<User> list = userService.list(wrapper);  
 if(!(list.size()>0)){  
 return RespBean.*error*("用户id不存在");  
 }  
 if(equipmentService.save(equipment)){  
 return RespBean.*success*("添加成功");  
 }  
 return RespBean.*success*("添加失败");  
 }  
 @PostMapping("delete")  
 @ApiOperation("删除设备")  
 public RespBean delete(@RequestBody Integer equipment\_id){  
 QueryWrapper<Equipment>wrapper=new QueryWrapper();  
 wrapper.eq("equipment\_id",equipment\_id);  
 if(equipmentService.remove(wrapper)){  
 return RespBean.*success*("删除成功");  
 }  
 return RespBean.*success*("删除失败");  
 }  
 @PostMapping("/update")  
 @ApiOperation("更新设备")  
 public RespBean update(@RequestBody Equipment equipment){  
 QueryWrapper<User> wrapper1 = new QueryWrapper();  
 wrapper1.eq("user\_id",equipment.getUser\_id());  
 List<User> list1 = userService.list(wrapper1);  
 if(!(list1.size()>0)){  
 return RespBean.*error*("用户id不存在");  
 }  
 QueryWrapper<Equipment>wrapper=new QueryWrapper();  
 wrapper.eq("equipment\_id",equipment.getEquipment\_id());  
 List<Equipment> list = equipmentService.list(wrapper);  
 equipmentService.remove(wrapper);  
 equipmentService.save(equipment);  
 return RespBean.*success*("更新成功");  
 }  
}

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.qrs.pojo.DTO.FenceDTO;  
import com.qrs.pojo.DTO.PointDTO;  
import com.qrs.pojo.Fence;  
import com.qrs.pojo.FencePoint;  
import com.qrs.pojo.RespBean;  
import com.qrs.service.IFencePointService;  
import com.qrs.service.IFenceService;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.ArrayList;  
import java.util.List;  
@RestController  
@RequestMapping("//fence")  
@Api(value = "围栏接口", tags = "hat围栏接口")  
public class FenceController {  
 @Autowired  
 private IFenceService fenceService;  
 @Autowired  
 private IFencePointService fencePointService;  
 @GetMapping("/get")  
 @ApiOperation("获得围栏列表")  
 public RespBean get(){  
 List<Fence> list = fenceService.list();  
 List<FenceDTO> list1=new ArrayList<>();  
 for(Fence fence:list){  
 List<PointDTO> listByID = fencePointService.getListByID(fence.getFence\_id());  
 PointDTO stringObjectMap = gravityPoint(listByID);  
 System.*out*.println(stringObjectMap);  
 FenceDTO fenceDTO=new FenceDTO(fence.getFence\_id(),fence.getFence\_number(),fence.getFence\_name(),fence.getFence\_node(),stringObjectMap,fencePointService.getListByID(fence.getFence\_id()));  
 list1.add(fenceDTO);  
 }  
 return RespBean.*success*("成功",list1);  
 }  
 @PostMapping("/add")  
 @ApiOperation("添加围栏")  
 public RespBean add(@RequestBody FenceDTO fenceDTO){  
 System.*out*.println(fenceDTO);  
 Fence fence=Fence.*builder*()  
 .fence\_id(fenceDTO.getFence\_id())  
 .fence\_name(fenceDTO.getFence\_name())  
 .fence\_node(fenceDTO.getFence\_node())  
 .fence\_number(fenceDTO.getFence\_number())  
 .build();  
 List<PointDTO> list=fenceDTO.getPoint();  
 for(PointDTO fencePointDTO:list){  
 FencePoint fencePoint=FencePoint.*builder*()  
 .fence\_id(fence.getFence\_id())  
 .lng(fencePointDTO.getLng())  
 .lat(fencePointDTO.getLat())  
 .build();  
 fencePointService.save(fencePoint);  
 }  
 if(fenceService.save(fence)){  
 return RespBean.*success*("添加成功");  
 }  
 return RespBean.*success*("添加失败");  
 }  
 @PostMapping("/delete")  
 @ApiOperation("删除围栏")  
 public RespBean delete(@RequestBody Integer fence\_id){  
 QueryWrapper<FencePoint> wrapper1=new QueryWrapper<>();  
 wrapper1.eq("fence\_id",fence\_id);  
 fencePointService.remove(wrapper1);  
  
 QueryWrapper<Fence> wrapper=new QueryWrapper<>();  
 wrapper.eq("fence\_id",fence\_id);  
 if(fenceService.remove(wrapper)){  
 return RespBean.*success*("删除成功");  
 }  
 return RespBean.*error*("删除失败");  
 }  
 public PointDTO gravityPoint(List<PointDTO> mPoints){  
 double area = 0.0;//多边形面积  
 double Gx = 0.0, Gy = 0.0;// 重心的x、y  
 for (int i = 1; i <= mPoints.size(); i++) {  
 double iLat = mPoints.get(i % mPoints.size()).getLat();  
 double iLng = mPoints.get(i % mPoints.size()).getLng();  
 double nextLat = mPoints.get(i - 1).getLat();  
 double nextLng = mPoints.get(i - 1).getLng();  
 double temp = (iLat \* nextLng - iLng \* nextLat) / 2.0;  
 area += temp;  
 Gx += temp \* (iLat + nextLat) / 3.0;  
 Gy += temp \* (iLng + nextLng) / 3.0;  
 }  
 Gx = Gx / area;  
 Gy = Gy / area;  
 PointDTO pointDTO=new PointDTO(Double.*valueOf*(String.*valueOf*(Gy).substring(0, String.*valueOf*(Gy).indexOf(".")+7)),Double.*valueOf*(String.*valueOf*(Gx).substring(0, String.*valueOf*(Gx).indexOf(".")+7)));  
 return pointDTO;  
 }  
}

import com.qrs.pojo.DTO.PointDTO;  
import com.qrs.pojo.RespBean;  
import com.qrs.service.IFencePointService;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.List;  
@RestController  
@RequestMapping("//fence-point")  
public class FencePointController {  
 @Autowired  
 private IFencePointService fencePointService;  
 @PostMapping("/getList")  
 @ApiOperation("通过围栏id获得点列表")  
 public RespBean getListByID(Integer fence\_id){  
 List<PointDTO> list = fencePointService.getListByID(fence\_id);  
 return RespBean.*success*("成功",list);  
 }  
}

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.qrs.pojo.\*;  
import com.qrs.pojo.DTO.PointDTO;  
import com.qrs.pojo.DTO.PointDTO2;  
import com.qrs.pojo.DTO.UserDTO4;  
import com.qrs.service.IEquipmentService;  
import com.qrs.service.IPositioningService;  
import com.qrs.service.IUserService;  
import com.qrs.service.IWarningService;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
@RestController  
@RequestMapping("//index")  
@Api(value = "首页接口", tags = "hat首页接口")  
public class IndexController {  
 @Autowired  
 private IPositioningService positioningService;  
 @Autowired  
 private IEquipmentService equipmentService;  
 @Autowired  
 private IUserService userService;  
 @Autowired  
 private IWarningService warningService;  
 @GetMapping("/getPoint")  
 @ApiOperation("获得人员定位列表")  
 public RespBean getPoint(){  
 QueryWrapper<Equipment> wrapper=new QueryWrapper<>();  
 wrapper.eq("employ",1);  
 List<Equipment> list = equipmentService.list(wrapper);  
 List<PointDTO2> list1=new ArrayList<>();  
 for(Equipment equipment:list){  
 QueryWrapper<Positioning> wrapper1=new QueryWrapper<>();  
 wrapper1.eq("equipment\_id",equipment.getEquipment\_id());  
 List<Positioning> list2 = positioningService.list(wrapper1);  
 Positioning positioning = list2.get(list2.size() - 1);  
 PointDTO2 pointDTO2=new PointDTO2(positioning.getEquipment\_id(),new PointDTO(positioning.getLng(),positioning.getLat()));  
 list1.add(pointDTO2);  
 }  
 return RespBean.*success*("成功",list1);  
 }  
 @PostMapping("/getById")  
 @ApiOperation("通过id获得信息")  
 public RespBean getPoint(@RequestBody Integer user\_id){  
 QueryWrapper<User> wrapper=new QueryWrapper<>();  
 wrapper.eq("user\_id",user\_id);  
 List<User> list = userService.list(wrapper);  
 QueryWrapper<Equipment> wrapper2=new QueryWrapper<>();  
 wrapper2.eq("user\_id",user\_id);  
 List<Equipment> list1 = equipmentService.list(wrapper2);  
  
 UserDTO4 userDTO4= UserDTO4.*builder*()  
 .user\_id(user\_id)  
 .user\_name(list.get(0).getUser\_name())  
 .equipment\_id(list1.get(0).getEquipment\_id())  
 .build();  
 return RespBean.*success*("成功",userDTO4);  
 }  
 @GetMapping("/getWaring")  
 @ApiOperation("获得警告")  
 public RespBean getWaring(){  
 QueryWrapper<Warning> wrapper=new QueryWrapper<>();  
 wrapper.last("limit 10");  
 List<Warning> list = warningService.list(wrapper);  
 return RespBean.*success*("成功",list);  
 }  
 @GetMapping("/get2")  
 @ApiOperation("近七天报警统计")  
 public RespBean get2(){  
 Map<String,Object> map=new HashMap<>();  
 List<Equipment> list = equipmentService.list();  
 QueryWrapper<Equipment> wrapper=new QueryWrapper<>();  
 wrapper.eq("employ",1);  
 List<Equipment> list1 = equipmentService.list(wrapper);  
 map.put("size",list.size());  
 map.put("online",list1.size()/(float)list.size());  
 List<Integer>list2=new ArrayList<>();  
 list2.add(10);  
 list2.add(8);  
 list2.add(12);  
 list2.add(15);  
 list2.add(11);  
 list2.add(6);  
 list2.add(5);  
 map.put("list",list2);  
 return RespBean.*success*("成功",map);  
 }  
}

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import com.qrs.annotation.PassToken;  
import com.qrs.annotation.UserLoginToken;  
import com.qrs.pojo.DTO.UserDTO;  
import com.qrs.pojo.RespBean;  
import com.qrs.pojo.User;  
import com.qrs.service.impl.UserServiceImpl;  
import com.qrs.utils.TokenUtils;  
import io.swagger.annotations.Api;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpSession;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
@RestController  
@RequestMapping("/Login")  
@Api(value = "登录接口", tags = "hat登录接口")  
public class LogoController {  
 @Autowired  
 private UserServiceImpl userService;  
 @PassToken  
 @RequestMapping(value = "/login",method = RequestMethod.*POST*)  
 public RespBean userLogin(HttpServletRequest request, @RequestBody UserDTO userDTO){  
 System.*out*.println(userDTO);  
 QueryWrapper<User> wrapper = new QueryWrapper();  
 wrapper.eq("user\_id",userDTO.getId());  
 List<User> list=userService.list(wrapper);  
 User user=list.get(0);  
 HttpSession session = request.getSession();  
 if (user==null) {  
 session.setAttribute("isError", true);  
 return RespBean.*error*("isError");  
 }  
 else{  
 Map<String,Object> tokenMap = new HashMap<>();  
 tokenMap.put("token", TokenUtils.*getUserToken*(user));  
 System.*out*.println(TokenUtils.*getUserToken*(user));  
 tokenMap.put("user",user);  
 return RespBean.*success*("登录成功",tokenMap);  
 }  
 }  
 @UserLoginToken  
 @PostMapping("/getMessage")  
 public RespBean getMessage(){  
 String aaa="hello word";  
 return RespBean.*success*("你已经通过验证",aaa);  
 }  
}

import com.qrs.pojo.DTO.PointDTO;  
import com.qrs.pojo.DTO.PointDTO3;  
import com.qrs.pojo.DTO.UserDTO4;  
import com.qrs.pojo.RespBean;  
import com.qrs.service.IEquipmentService;  
import com.qrs.service.IPositioningService;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
@RestController  
@RequestMapping("//positioning")  
@Api(value = "定位接口", tags = "hat定位接口")  
public class PositioningController {  
 @Autowired  
 private IPositioningService positioningService;  
 @Autowired  
 private IEquipmentService equipmentService;  
 @GetMapping("/getUserList")  
 @ApiOperation("获得定位人员列表")  
 private RespBean getList(){  
 List<UserDTO4> list=equipmentService.getUserList();  
 return RespBean.*success*("成功",list);  
 }  
 @PostMapping("/getPoint")  
 @ApiOperation("获得人员当天定位")  
 private RespBean getPoint(@RequestBody PointDTO3 pointDTO3){  
 List<PointDTO> list=equipmentService.getPoint(pointDTO3.getEquipment\_id(),pointDTO3.getTime());  
 return RespBean.*success*("成功",list);  
 }  
}

import com.qrs.pojo.DTO.PointDTO;  
import com.qrs.pojo.RespBean;  
import com.qrs.service.ISignPointService;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.List;  
@RestController  
@RequestMapping("//sign-point")  
public class SignPointController {  
 @Autowired  
 private ISignPointService signPointService;  
 @PostMapping("/get")  
 @ApiOperation("通过签到区域id获得点列表")  
 public RespBean get(Integer sign\_id){  
 List<PointDTO> list = signPointService.getListById(sign\_id);  
 return RespBean.*success*("成功",list);  
 }  
}

import com.baomidou.mybatisplus.core.conditions.update.UpdateWrapper;  
import com.qrs.pojo.DTO.WarningSwitchDTO;  
import com.qrs.pojo.RespBean;  
import com.qrs.pojo.Warning;  
import com.qrs.pojo.WarningSwitch;  
import com.qrs.service.IWarningService;  
import com.qrs.service.IWarningSwitchService;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
@RestController  
@Api(value = "安全帽报警开关模块" ,tags = "hat报警开关接口")  
@RequestMapping("//warning-switch")  
public class WarningSwitchController {  
 @Autowired  
 private IWarningSwitchService warningSwitchService;  
 @Autowired  
 private IWarningService warningService;  
 @GetMapping("/getList")  
 @ApiOperation(value = "获得开关列表")  
 public RespBean getList(){  
 List<WarningSwitch> list = warningSwitchService.list();  
 return RespBean.*success*("成功",list);  
 }  
 @PostMapping("/setBYid")  
 @ApiOperation(value = "修改状态")  
 public RespBean setById(@RequestBody WarningSwitchDTO warningSwitchDTO){  
 UpdateWrapper<WarningSwitch> wrapper = new UpdateWrapper();  
 wrapper.eq("switch\_id",warningSwitchDTO.getSwitch\_id())  
 .set("switch\_state",warningSwitchDTO.getSwitch\_state());  
 if(warningSwitchService.update(wrapper)){  
 return RespBean.*success*("成功");  
 }  
 return RespBean.*success*("失败");  
 }  
 @GetMapping("/getListWarning")  
 @ApiOperation(value = "获得警报列表")  
 public RespBean getListWarning(){  
 List<Warning> list = warningService.list();  
 return RespBean.*success*("成功",list);  
 }  
}

@PostMapping("/addUser")  
 @ApiOperation(value = "添加用户")  
 public RespBean addUser(@RequestBody User user){  
 QueryWrapper<User> wrapper = new QueryWrapper();  
 wrapper.eq("user\_id",user.getUser\_id());  
 List<User> list = userService.list(wrapper);  
 if(list.size()>0){  
 return RespBean.*error*("添加失败，用户id已经存在");  
 }  
 user.setUser\_password(DigestUtils.*md5DigestAsHex*("123456".getBytes()));  
 if (userService.save(user)){  
 return RespBean.*success*("添加成功");  
 }  
 return RespBean.*success*("添加失败");  
 }  
 @PostMapping("/update")  
 @ApiOperation(value = "修改某个用户")  
 public RespBean update(@RequestBody User user){  
 QueryWrapper<User> wrapper = new QueryWrapper();