|  |
| --- |
| <template>  <view class="container">  <view class="a">登记界面</view>    <view>  <button type="primary" class="button" @click="yuyue">登记</button>  </view>  </view>  </template>  <script>  import { cqwlyy } from '../../api/request'  export default {  data() {  return {  isSame:true,  passwordRepeat:undefined,  data:{    }    }  },  methods: {  yuyue(){  cqwlyy(this.data).then(resp=>{  if(resp.data.code==200){    uni.showToast({  icon:"success",  title:"预约成功"  })    setTimeout(x=>{  uni.switchTab({  url:"/pages/zhuye/zhuye"  })  },1500)  }  })  }  }  }  </script>  <style>  .container {  padding: 20px;  font-size: 14px;  line-height: 24px;  display: flex;  flex-direction: column;  }  .a{  font-size: 25px;  text-align: center;  margin-top: 10px;  }  .item{  margin-top: 10px;  }  .c{  font-size: 20px;  margin-top: 20px;  }  .d{  border: solid;  color: gray;  border-width: 1px;  border-radius: 5px;  margin-top: 10px;  }  .button{  margin-top: 200px;  width: 100px;  height: 50px;  background-color: lightslategray;  }  </style>  <template>  <view class="container">  <view class="a">预约登记</view>  <view class="c">当日进出次数</view>  <view>  <uni-data-select  v-model="data.goOutCount"  :localdata="range"  @change="change"  :clear="false"  ></uni-data-select>  </view>  <view>  <button type="primary" class="button" @click="yuyue">申报</button>  </view>  </view>  </template>  <script>  import { dqwlyy } from '../../api/request.js'  export default {  data() {  return {  isSame:true,  passwordRepeat:undefined,  data:{  code:undefined,  goOutCount:undefined,  userId:undefined  },  value: 0,  range: [  { value: 0, text: "请选择"},  { value: 1, text: "1" },  { value: 2, text: "2" },  { value: 3, text: "3" },  ],  }  },  methods: {  change(e) {  console.log("e:", e);  },  yuyue(){  dqwlyy(this.data).then(resp=>{  if(resp.data.code==200){    uni.showToast({  icon:"success",  title:"预约成功"  })    setTimeout(x=>{  uni.switchTab({  url:"/pages/zhuye/zhuye"  })  },1500)  }  })  }  }  }  </script>  <style>  .container {  padding: 20px;  font-size: 14px;  line-height: 24px;  display: flex;  flex-direction: column;  }  .a{  font-size: 25px;  text-align: center;  margin-top: 10px;  margin-bottom: 10px;  }  .c{  font-size: 20px;  margin-top: 15px;  }  .d{  border: solid;  color: gray;  border-width: 1px;  border-radius: 5px;  margin-top: 10px;  }  .button{  margin-top: 70px;  width: 100px;  height: 50px;  background-color: lightslategray;  }  </style>  <template>  <view>  <view class="item">  <view class="title">姓名</view>  <view class="content">{{userInfo.name}}</view>  </view>  <view class="item">  <view class="title">所属公司</view>  <view class="content">{{userInfo.company}}</view>  </view>  <view class="item">  <view class="title">手机号</view>  <view class="content">{{userInfo.phone}}</view>  </view>  <view class="item">  <view class="title">车牌号</view>  <view class="content">{{userInfo.carnum}}</view>  </view>  <view class="item">  <view class="title">进入时间</view>  <view class="content">{{examineInfo.enterTime}}</view>  </view>  <view class="item">  <view class="title">货物照片</view>  </view>  <uni-file-picker v-model="picture" @select="select" file-mediatype="image" limit="1" v-if="examineInfo.picture==null"></uni-file-picker>  <view class="image" v-else>  <image style="height: 100%;width: 100%;" :src="picture"></image>  </view>  <view class="bottom" v-if="examineInfo.examineResult==null && loginUser.deptId==6">  <button type="primary" @click="isagree(0)">放行</button>  </view>  <view class="bottom" v-if="examineInfo.picture!=null && examineInfo.baoanResult==null && loginUser.deptId==5">  <button type="primary" @click="isagree(0)">放行</button>  </view>    </view>  </template>  <script>  import {queryUserById,commonExamine,carLongPictureUp,carLongPictureExamine,queryLongPictureById,ofMe,carLongBaoAnExamine} from "../../api/request.js"  export default {  onLoad(data) {  this.examineInfo=JSON.parse(data.data)  queryUserById(this.examineInfo.userId).then(resp=>{  this.userInfo=resp.data.data  })  this.queryLongPictureById(this.examineInfo.id)  ofMe().then(resp=>{  this.loginUser=resp.data.data  })  },  data() {  return {  examineInfo:{},  userInfo:{},  picture:null,  loginUser:{}  }  },  methods: {  queryLongPictureById(id){  queryLongPictureById(id).then(resp=>{  this.picture=resp.data.data  })  },  upload(img){  const tempFilePaths = img.tempFiles[0].file;//e是获取的图片源  this.convertImageToBase64(tempFilePaths,resp=>{  //resp为base64编码  carLongPictureUp({  id:this.examineInfo.id,  image:resp  })  })    },  convertImageToBase64(file, callback) {  var reader = new FileReader();  reader.onload = function() {  var base64 = reader.result.split(',')[1];  callback(base64);  };  reader.readAsDataURL(file);  },    isagree(opinion){  if(this.loginUser.deptId==6){  if(this.picture==null || this.picture==''){  uni.showToast({  icon:"none",  title:"请先上传图片"  })  return  }  this.upload(this.picture)  this.examineInfo.examineResult=opinion  carLongPictureExamine(this.examineInfo).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  setTimeout(resp=>{  this.$router.go(-1)  },1000)    }  })  }else{  //baoan  this.examineInfo.baoanResult=opinion  carLongBaoAnExamine(this.examineInfo).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  setTimeout(resp=>{  this.$router.go(-1)  },1000)    }  })  }    },  select(img){  this.picture=img  }  }  }  </script>  <style lang="scss" scoped>    .item{  display: flex;  justify-content: space-between;  margin-top: 10px;  font-size: 18px;  .title{  margin-left: 5px;  }  .content{  margin-right: 5px;  }    }  .bottom{  margin-top: 20px;  display: flex;  justify-content: space-around;  }    .image{  width: 100px;  height: 100px;  margin-top: 10px;  margin-left: 5px;  }  </style>  <template>  <view>  <view class="item">  <view class="title">姓名</view>  <view class="content">{{userInfo.name}}</view>  </view>  <view class="item">  <view class="title">所属公司</view>  <view class="content">{{userInfo.company}}</view>  </view>  <view class="item">  <view class="title">手机号</view>  <view class="content">{{userInfo.phone}}</view>  </view>  <view class="item">  <view class="title">车牌号</view>  <view class="content">{{userInfo.carnum}}</view>  </view>  <view class="item">  <view class="title">进入时间</view>  <view class="content">{{examineInfo.enterTime}}</view>  </view>  <view class="item">  <view class="title">货物照片</view>  </view>  <uni-file-picker v-model="picture" @select="select" file-mediatype="image" limit="1" v-if="examineInfo.picture==null"></uni-file-picker>  <view class="image" v-else>  <image style="height: 100%;width: 100%;" :src="picture"></image>  </view>  <view class="bottom" v-if="examineInfo.examineResult==null && loginUser.deptId==6">  <button type="primary" @click="isagree(0)">放行</button>  </view>  <view class="bottom" v-if="examineInfo.picture!=null && examineInfo.baoanResult==null && loginUser.deptId==5">  <button type="primary" @click="isagree(0)">放行</button>  </view>    </view>  </template>  <script>  import {queryUserById,commonExamine,carShortPictureUp,carShortPictureExamine,queryShortPictureById,carShortBaoAnExamine,ofMe} from "../../api/request.js"  export default {  onLoad(data) {  this.examineInfo=JSON.parse(data.data)  queryUserById(this.examineInfo.userId).then(resp=>{  this.userInfo=resp.data.data  })  this.queryShortPictureById(this.examineInfo.id)  ofMe().then(resp=>{  this.loginUser=resp.data.data  })  },  data() {  return {  examineInfo:{},  userInfo:{},  picture:null,  loginUser:{}  }  },  methods: {  queryShortPictureById(id){  queryShortPictureById(id).then(resp=>{  this.picture=resp.data.data  })  },  upload(img){  const tempFilePaths = img.tempFiles[0].file;//e是获取的图片源  this.convertImageToBase64(tempFilePaths,resp=>{  //resp为base64编码  carShortPictureUp({  id:this.examineInfo.id,  image:resp  })  })    },  convertImageToBase64(file, callback) {  var reader = new FileReader();  reader.onload = function() {  var base64 = reader.result.split(',')[1];  callback(base64);  };  reader.readAsDataURL(file);  },    isagree(opinion){  if(this.loginUser.deptId==6){  if(this.picture==null || this.picture==''){  uni.showToast({  icon:"none",  title:"请先上传图片"  })  return  }  this.upload(this.picture)  this.examineInfo.examineResult=opinion  carShortPictureExamine(this.examineInfo).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  setTimeout(resp=>{  this.$router.go(-1)  },1000)    }  })  }else{  //baoan  this.examineInfo.baoanResult=opinion  carShortBaoAnExamine(this.examineInfo).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  setTimeout(resp=>{  this.$router.go(-1)  },1000)    }  })  }  },  select(img){  this.picture=img  }  }  }  </script>  <style lang="scss" scoped>    .item{  display: flex;  justify-content: space-between;  margin-top: 10px;  font-size: 18px;  .title{  margin-left: 5px;  }  .content{  margin-right: 5px;  }    }  .bottom{  margin-top: 20px;  display: flex;  justify-content: space-around;  }    .image{  width: 100px;  height: 100px;  margin-top: 10px;  margin-left: 5px;  }  </style>  <template>  <view>  <view class="item">  <view class="title">姓名</view>  <view class="content">{{userInfo.name}}</view>  </view>  <view class="item">  <view class="title">所属公司</view>  <view class="content">{{userInfo.company}}</view>  </view>  <view class="item">  <view class="title">手机号</view>  <view class="content">{{userInfo.phone}}</view>  </view>  <view class="item">  <view class="title">访问的部门</view>  <view class="content">{{visitDeptInfo.deptName}}</view>  </view>  <view class="item">  <view class="title">访问的人</view>  <view class="content">{{employeeInfo.name}}</view>  </view>  <view class="item">  <view class="title">访问开始时间</view>  <view class="content">{{examineInfo.visiteStartTime}}</view>  </view>  <view class="item">  <view class="title">访问结束时间</view>  <view class="content">{{examineInfo.visiteEndTime}}</view>  </view>  <view class="item">  <view class="title">预约时间</view>  <view class="content">{{examineInfo.appointTime}}</view>  </view>    <view class="bottom" v-if="examineInfo.isagree==-1">  <button type="primary" @click="isagree(0)">同意</button>  <button type="warn" @click="isagree(1)">拒绝</button>  </view>  <view class="bottom">  <button type="primary" v-if="examineInfo.isagree==0">已同意</button>  <button type="warn" v-if="examineInfo.isagree==1">已拒绝</button>  </view>    </view>  </template>  <script>  import {queryDeptById,queryUserById,commonExamine} from "../../api/request.js"  export default {  onLoad(data) {  this.examineInfo=JSON.parse(data.data)  queryUserById(this.examineInfo.userId).then(resp=>{  this.userInfo=resp.data.data  })  queryUserById(this.examineInfo.visiteEmployeeId).then(resp=>{  this.employeeInfo=resp.data.data  })  queryDeptById(this.examineInfo.visiteDeptId).then(resp=>{  this.visitDeptInfo=resp.data.data  })  },  data() {  return {  examineInfo:{},  userInfo:{},  employeeInfo:{},  visitDeptInfo:{}  }  },  methods: {  isagree(opinion){  this.examineInfo.isagree=opinion  commonExamine(this.examineInfo).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  setTimeout(resp=>{  this.$router.go(-1)  },1000)    }  })  }    }  }  </script>  <style lang="scss" scoped>  .item{  display: flex;  justify-content: space-between;  margin-top: 10px;  font-size: 18px;  .title{  margin-left: 5px;  }  .content{  margin-right: 5px;  }    }    .bottom{  margin-top: 20px;  display: flex;  justify-content: space-around;  }  </style>  <template>  <view>  <view class="item">  <view class="title">车牌号</view>  <view class="content">{{examineInfo.carnum}}</view>  </view>  <view class="item">  <view class="title">访问公司</view>  <view class="content">{{examineInfo.company}}</view>  </view>  <view class="item">  <view class="title">访问的部门</view>  <view class="content">{{visitDeptInfo.deptName}}</view>  </view>  <view class="item">  <view class="title">访问的人员</view>  <view class="content">{{employeeInfo.name}}</view>  </view>    <view class="item">  <view class="title">预约时间</view>  <view class="content">{{examineInfo.appointTime}}</view>  </view>    <view class="bottom" v-if="!isNotExamine()">  <button type="primary" @click="isagree(0)">同意</button>  <button type="warn" @click="isagree(1)">拒绝</button>  </view>  <view class="bottom" v-else>  <button type="primary" disabled="true">不需要审批</button>  </view>  </view>  </template>  <script>  import {queryUserById,queryDeptById,vipExamine, ofMe, login} from "../../api/request.js"  export default {  onLoad(data) {  this.examineInfo=JSON.parse(data.data)  queryUserById(this.examineInfo.visiteEmployeeId).then(resp=>{  this.employeeInfo=resp.data.data  })  queryDeptById(this.examineInfo.visiteDeptId).then(resp=>{  this.visitDeptInfo=resp.data.data  })  },  onShow() {  ofMe().then(resp=>{  this.loginInfo=resp.data.data  })  },  data() {  return {  examineInfo:{},  employeeInfo:{},  visitDeptInfo:{},  loginInfo:{}  }  },  methods: {  isNull(data){  if(data==null || data==undefined || data==''){  return true  }else{  return false  }  },  isagree(opinion){  var data={  id:this.examineInfo.id,  appointmentId:this.examineInfo.appointmentId,  visiteDeptOpinion:this.examineInfo.visiteDeptOpinion,  renshiOpinion:this.examineInfo.renshiOpinion,  manageOpinion:this.examineInfo.manageOpinion,  manageId:this.examineInfo.manageId,  renshiId:this.examineInfo.renshiId,  deptExamineId:this.examineInfo.deptExamineId,  visiteEmployeeId:this.examineInfo.visiteEmployeeId,  visiteDeptId:this.examineInfo.visiteDeptId,  code:this.examineInfo.code,  isgo:this.examineInfo.isgo,  }  if(this.isNull(this.examineInfo.deptExamineId)){  //部长  data.deptExamineId=this.loginInfo.id  data.visiteDeptOpinion=opinion  vipExamine(data).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  }  })  }else if(this.isNull(this.examineInfo.renshiId)){  data.renshiId=this.loginInfo.id  data.renshiOpinion=opinion  vipExamine(data).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  }  })    }else{  data.manageId=this.loginInfo.id  data.manageOpinion=opinion  vipExamine(data).then(resp=>{  if(resp.data.code==200){  uni.showToast({  icon:"success",  title:"审批成功"  })  }  })  }  setTimeout(resp=>{  this.$router.go(-1)  },1000)  },  /\*\*  \* 是否不需要审批  \*/  isNotExamine(){  //部长  if(this.loginInfo.deptId!=3 && this.loginInfo.deptId!=1){  if(this.isNull(this.examineInfo.deptExamineId)){  return false  }  return true  }else if(this.loginInfo.deptId==3){  //人事  if(this.employeeInfo.deptId==3 && (this.isNull(this.examineInfo.deptExamineId) || this.isNull(this.examineInfo.renshiId) )){  return false  }else if(this.employeeInfo.deptId!=3 && !this.isNull(this.examineInfo.deptExamineId) && this.isNull(this.examineInfo.renshiId)){  return false  }  return true    }else{  //经理  if(this.employeeInfo.deptId==1 && (this.isNull(this.examineInfo.deptExamineId) || this.isNull(this.examineInfo.manageId) )){  return false  }else if(this.employeeInfo.deptId!=1 && !this.isNull(this.examineInfo.renshiId) && this.isNull(this.examineInfo.manageId)){  return false  }  return true    }    }  }  }  </script>  <style>  .item{  display: flex;  justify-content: space-between;  margin-top: 10px;  font-size: 18px;  .title{  margin-left: 5px;  }  .content{  margin-right: 5px;  }        }  .bottom{  margin-top: 20px;  display: flex;  justify-content: space-around;  }  </style>  <template>  <view class="container">  <view class="a">{{"张三"}}</view>  <view class="a">{{"123456789"}}</view>  <view class="a">{{"苏F000001"}}</view>  <view class="a">{{"百度公司"}}</view>  <view class="a">{{"普通访客"}}</view>  <view v-if="loginUser.isEmployee==0">  <view class="c">拜访时间段</view>  <view class="item">  <view class="right">  <view class="example-body">  <uni-datetime-picker type="date" :clear-icon="false" v-model="data.visiteStartTime" @maskClick="maskClick" />  </view>  </view>  </view>  <view class="item">  <view class="right">  <view class="example-body">  <uni-datetime-picker type="date" :clear-icon="false" v-model="data.visiteEndTime" @maskClick="maskClick" />  </view>  </view>  </view>  </view>  <view class="b">  <view>  <button type="primary" class="button" @click="ty">同意</button>  </view>  <view>  <button type="primary" class="button" @click="jj">拒绝</button>  </view>  </view>    </view>  </template>  <script>  import {ofMe} from "../../api/request.js"  export default {  onShow() {  ofMe().then(resp=>{  this.loginUser=resp.data.data  })  },  data() {  return {  loginUser:{},  data:{  visiteEndTime:undefined,  visiteStartTime:undefined,  }  }  },  methods: {  ty(){  uni.navigateTo({  url:"/pages/yycg/yycg"  })  },  jj(){  uni.navigateTo({  url:"/pages/yysb/yysb"  })  }  }  }  </script>  <style>  .container {  padding: 20px;  font-size: 14px;  line-height: 24px;  display: flex;  flex-direction: column;  }  .a{  font-size: 20px;  text-align: center;  margin-top: 10px;  margin-bottom: 10px;  }  .b{  display: flex;  justify-content: space-between;  }  .c{  font-size: 20px;  margin-top: 15px;  }  .d{  border: solid;  color: gray;  border-width: 1px;  border-radius: 5px;  margin-top: 10px;  }  .button{  margin-top: 50px;  width: 100px;  height: 50px;  background-color: lightslategray;  }  .right{  margin-top: 10px;  }  </style>  <template>  <view class="container">  <view class="title"><h1>访问记录</h1></view>  <view class="no" v-if="have">没有任何访问记录</view>  <view v-for="(list,type) in visiteList">  <view class="item" v-for="item in list" @click="toDetail(type,item)">  <view class="title">  <view class="content">{{visitType(type)}}</view>  </view>  <view class="time">{{item.appointTime}}</view>  </view>  </view>  </view>  </template>  <script>  import {cxyy,queryVisitMe} from '../../api/request.js'  export default {  onShow() {  queryVisitMe().then(resp=>{  this.visiteList=resp.data.data  if(this.visiteList["0"].length==0 && this.visiteList["1"].length==0 ){  this.have=true  }  })  },  data() {  return {  visiteList:{},  have:false  }  },  methods: {  visitType(type){  switch(type){  case "0":  return "普通"  break;  case "1":  return "vip"  break;    }  },    toDetail(type,data){  uni.navigateTo({  url:"/pages/visitMe/visiteMe?data="+JSON.stringify({  type:type,  data:data  })  })      }  }  }  </script>  <style>  .container {  padding: 20px;  font-size: 14px;  line-height: 24px;  }  .title{  color: seagreen;  text-align: center;  }  .a{  display: flex;  justify-content: space-between;  color: seagreen;  text-align: center;  margin-top: 20px;  }  .c{  font-size: 20px;  margin-top: 10px;  box-shadow:0 0 2rpx #000000;  }    .no{  display: flex;justify-content: center;  margin-top: 50px;  padding-bottom: 10px;  border-bottom: 1px solid #c6c6c6;  }    .item{  display: flex;  justify-content: space-between;  box-shadow:0 0 2rpx #000000;  align-items: center;  margin-top: 10px;  .title{  width: 50px;  height: 50px;  background-color: #00aaff;  text-align: center;  display: flex;  justify-content: center;  align-items: center;  .content{  color: aliceblue;  font-size: 20px;  }    }  }  </style>  <template>  <view>  <view class="item" v-if="loginUser.type==0 || loginUser.type==4">  <view class="title">访问的部门</view>  <view class="content">{{deptName}}</view>  </view>  <view class="item" v-if="loginUser.type==0 || loginUser.type==4">  <view class="title">访问的人员</view>  <view class="content">{{userName}}</view>  </view>  <view class="item" v-if="loginUser.type==0">  <view class="title">访问开始时间</view>  <view class="content">{{appointmentData.visiteStartTime}}</view>  </view>  <view class="item" v-if="loginUser.type==0">  <view class="title">访问结束时间</view>  <view class="content">{{appointmentData.visiteEndTime}}</view>  </view>  <view class="item" v-if="loginUser.isEmployee==0">  <view class="title">车牌号</view>  <view class="content">{{appointmentData.carnum}}</view>  </view>  <view class="item" v-if="loginUser.isEmployee==0">  <view class="title">来访单位</view>  <view class="content">{{appointmentData.company}}</view>  </view>  <view class="item">  <view class="title">预约时间</view>  <view class="content">{{appointmentData.appointTime}}</view>  </view>    <view class="result">  <view class="wait" v-if="isWait()">审核中</view>  <view class="fail" v-if="isFail()">预约失败</view>  <view class="success" v-if="appointmentData.code!=null">预约成功</view>  <view class="success" v-if="appointmentData.code!=null" @click="Qrcode">查看二维码</view>  </view>    <view class="code" v-if="isShow">  <image :src="code" style="height: 100%;width: 100%;" ></image>  <button style="margin-top: 10px;" type="primary" @click="goin">进入</button>  <button style="margin-top: 10px;" type="primary" @click="goout">离厂</button>  </view>  </view>  </template>  <script>  import {ofMe,queryDeptById,queryUserById,queryCodeById,goin} from "../../api/request.js"  export default {  onLoad(options) {  this.appointmentData=JSON.parse(options.details)  ofMe().then(resp=>{  this.loginUser=resp.data.data  if(resp.data.data.type==0||resp.data.data.type==4){  this.queryDeptById(this.appointmentData.visiteDeptId)  this.queryUserById(this.appointmentData.visiteEmployeeId)  }  })    },  data() {  return {  appointmentData:{},  loginUser:{},  deptName:'',  userName:"",  code:'',  isShow:false  }  },  methods: {  queryDeptById(id) {  queryDeptById(id).then(resp=>{  this.deptName=resp.data.data.deptName  })  },  queryUserById(id) {  queryUserById(id).then(resp=>{  this.userName=resp.data.data.name  })  },  /\*\*  \* 审核是否失败  \*/  isFail(){  //不是内部人员  if(this.loginUser.isEmployee==1 && this.appointmentData.isagree==1 && this.loginUser.type!=2 && this.loginUser.type!=3){  return true  }  if(this.appointmentData.visiteDeptOpinion==1 || this.appointmentData.renshiOpinion==1||this.appointmentData.manageOpinion==1){  return true  }else{  return false  }  },  /\*\*  \* 是否在审核中  \*/  isWait(){  //不是内部人员  if(this.loginUser.isEmployee==1 && this.appointmentData.isagree==-1 && this.loginUser.type!=2 && this.loginUser.type!=3){  return true  }  //为内部人员  if(this.loginUser.isEmployee==0){  if(this.appointmentData.visiteDeptOpinion!=1 && this.appointmentData.renshiOpinion!=1 && this.appointmentData.manageOpinion==null){  return true  }else{  return false  }  }      },  Qrcode(){  queryCodeById(this.appointmentData.id).then(resp=>{  this.code=resp.data.data  })  this.isShow=true  },  goin(){  goin(this.appointmentData.id).then(resp=>{  if(resp.data.code==200){  uni.showToast({  title:"进入成功"  })  }  })  },  goout(){  uni.showToast({  title:"离场成功"  })  }  }  }  </script>  <style scoped lang="scss">  .item{  display: flex;  justify-content: space-between;  margin-top: 10px;  font-size: 18px;  .title{  margin-left: 5px;  }  .content{  margin-right: 5px;  }    }    .result{  display: flex;  justify-content: space-around;  align-items: center;  font-size: 18px;  margin-top: 50px;  .wait{  color: blue;  }  .fail{  color: red;  }  .success{  color: green;  }  }    .code{  margin: 20px auto;  width: 300px;  height:300px;  display: flex;  justify-content: center;  align-items: center;  flex-direction: column;  }    </style>  <template>  <view>  <view class="b">离场二维码</view>  <view class="c">  <image class="image2" src="../../static/ewm.png">  </view>  <view class="b">欢迎您的下次访问</view>  <view>  <button type="primary" class="button" @click="tuichu">退出</button>  </view>  </view>  </template>  <script>  export default {  data() {  return {    }  },  methods: {  tuichu(){  uni.navigateTo({  url:"/pages/login/login"  })  }  }  }  </script>  <style>  .b{  text-align: center;  margin-top: 70px;  font-size: 22px;    }  .c{  text-align: center;  margin-top: 10px;  }  .image2{  width: 200px;  height: 200px;  margin-top: 10px;  }  .button{  margin-top: 100px;  width: 100px;  height: 50px;  background-color: lightslategray;  }  </style>  <template>  <view>  <view class="a"><h1>访客系统</h1></view>  <view class="item">  <view class="left">手机号</view>  <view class="right">  <input type="text" v-model="data.phone" placeholder="请输入手机号" @input="change"/>  </view>  </view>  <view class="item">  <view class="left">密码</view>  <view class="right">  <input type="password" v-model="data.password" placeholder="请输入密码" @input="change"/>  </view>  </view>  <view >  <button class="bottom" type="primary" @click="login">登录</button>  </view>  <view class="b">  <view @click="zhuce">立即注册</view>  <view>忘记密码</view>  </view>    </view>  </template>  <script>  import {login} from "../../api/request.js"  export default {  mounted() {  },  data() {  return {  isSame:true,  passwordRepeat:undefined,  data:{  password:undefined,  phone:undefined  },  count:0    }  },  methods: {  login(){  const v=uni.createInnerAudioContext();  v.autoplay=true  v.src="../../static/gm.mp3"  v.onError(resp=>{  console.log(resp)  })  login(this.data).then(resp=>{  if(resp.data.code==200){  sessionStorage.setItem("token",resp.data.data)  uni.showToast({  icon:"success",  title:"登录成功"  })    setTimeout(x=>{  uni.switchTab({  url:"/pages/zhuye/zhuye"  })  },1500)  }  }).catch(rep=>{  uni.showToast({  title:"出错"  })  })  },  change(a){  var url="../../static/j.mp3"  if(this.data.phone.length<this.count.length){  url="../../static/a.mp3"  }  this.count=this.data.phone  const v=uni.createInnerAudioContext();  v.autoplay=true  v.src=url  v.onError(resp=>{  console.log(resp)  })  },  zhuce(){  uni.navigateTo({  url:"/pages/register/register"  })  },  check(){  if(this.data.password==this.passwordRepeat){  this.isSame=true  }else{  this.isSame=false  }  },  }  }  </script>  <style>  page{  background: url("../../static/dl.png");  }  </style>  <style>  input {  width: 200px;  border-radius: 5rpx;  box-shadow:0 0 2rpx #000;  height: 100%;  }  </style>  <style>  .a{  text-align: center;  margin-top: 30px;  margin-bottom: 40px;  }    </style>  <style lang="scss" scoped>  .item{  font-size: 20px;  margin: 20px 30px;  margin-top: 10px;  display: flex;    .left{  width: 80px;  }  .right{  margin-left: 10px;  display: flex;  flex: 1;  }    .label{  width: 10px;  height: 10px;  border-radius: 50%;  margin: auto 0;    }  .fail{  background-color: red;  }  .success{  background-color: green;  }  }    .bottom{  margin-top: 40px;  background:linear-gradient(white, pink, grey);  }    .b{  display: flex;  justify-content: space-between;  }  </style>  <template>  <view class="container">  <view class="a">预约登记</view>  <!-- <view class="item">  <view class="c">姓名</view>  <view class="d">  <input type="text" v-model="data.name" placeholder="请输入"/>  </view>  </view>  <view class="item">  <view class="c">电话号码</view>  <view class="d">  <input type="text" v-model="data.name" placeholder="请输入"/>  </view>  </view> -->  <!-- <view class="item">  <view class="c">所属公司</view>  <view class="d">  <input type="text" v-model="data.name" placeholder="请输入"/>  </view>  </view> -->  <view class="item">  <view class="c">拜访部门</view>  <view class="d">  <uni-data-select  v-model="data.visiteDeptId"  :localdata="range1"  @change="change"  :clear="false"  ></uni-data-select>  </view>  </view>  <view class="c">被访问人</view>  <view class="d">  <uni-data-select  v-model="data.visiteEmployeeId"  :localdata="range2"  @change="change"  :clear="false"  ></uni-data-select>  </view>  <!-- <view class="item">  <view class="c">车牌号（可选填）</view>  <view class="d">  <input type="text" v-model="data.name" placeholder="请输入"/>  </view>  </view> -->  <view class="c">拜访时间段</view>  <view class="item">  <view class="right">  <view class="example-body">  <uni-datetime-picker v-model="data.visiteStartTime" />    </view>  </view>  </view>  <view class="item">  <view class="right">  <view class="example-body">  <uni-datetime-picker v-model="data.visiteEndTime" />    </view>  </view>  </view>  <view>  <button type="primary" class="button" @click="yuyue">提交</button>  </view>    </view>  </template>  <script>  import {ptfkyy,cxbm,cxry} from '../../api/request.js'  export default {  onShow() {  cxbm().then(resp=>{  var list=resp.data.data  var result=[]  list.forEach(x=>{  result.push({  "text":x.deptName,  "value":x.deptId  })  })  this.range1=result  })  },  data() {  return {  isSame:true,  passwordRepeat:undefined,  data:{  code:undefined,  isagree:undefined,  isgo:undefined,  isvisite:undefined,  userId:undefined,  visiteDeptId:undefined,  visiteEmployeeId:undefined,  visiteEndTime:undefined,  visiteStartTime:undefined  },  value: 0,  range1: [  ],  range2:[  ],  }  },  methods: {  change(e) {  cxry(e).then(resp=>{  var list=resp.data.data  var result=[]  list.forEach(x=>{  result.push({  "text":x.name,  "value":x.id  })  })  this.range2=result  })  },  yuyue(){  ptfkyy(this.data).then(resp=>{  if(resp.data.code==200){    uni.showToast({  icon:"success",  title:"预约成功"  })    setTimeout(x=>{  uni.switchTab({  url:"/pages/zhuye/zhuye"  })  package visite\_system.demo.Service.Impl;  import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service; import org.springframework.util.ObjectUtils; import visite\_system.demo.Entity.BuildAppointment; import visite\_system.demo.Entity.User; import visite\_system.demo.GlobalUtils.QrCodeUtils; import visite\_system.demo.GlobalUtils.ThreadLocalUtil; import visite\_system.demo.Mapper.Build\_AppointmentMapper; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Service.BuildService;  import java.util.Date;  @Service public class BuildServiceImpl implements BuildService {  @Autowired  private Build\_AppointmentMapper buildAppointmentMapper;   @Autowired  private QrCodeUtils qrCodeUtils;   @Override  public Result buildAppoint(BuildAppointment buildAppointment) {  User user = ThreadLocalUtil.*get*();  buildAppointment.setUserId(user.getId());  buildAppointment.setAppointTime(new Date());  buildAppointmentMapper.insert(buildAppointment);  return Result.*ok*();  }   @Override  public Result buildExamine(BuildAppointment buildAppointment) throws Exception {  Long appointmentId = buildAppointment.getId();  if(ObjectUtils.*isEmpty*(appointmentId)){  throw new Exception("没有预约的id");  }  BuildAppointment appointment = buildAppointmentMapper.selectById(appointmentId);  User user = ThreadLocalUtil.*get*();  appointment.setExaminerId(user.getId());  Integer isagree = buildAppointment.getIsagree();  appointment.setIsagree(isagree);  //生成二维码  if(isagree==0){  String qrCode = qrCodeUtils.createQrCode(String.*valueOf*(appointment.getId()));  appointment.setCode(qrCode);  }  buildAppointmentMapper.updateById(appointment);  return Result.*ok*();  } }  package visite\_system.demo.Service.Impl; import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper; import com.baomidou.mybatisplus.core.conditions.update.LambdaUpdateWrapper; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service; import org.springframework.util.ObjectUtils; import org.springframework.web.multipart.MultipartFile; import org.springframework.web.multipart.MultipartHttpServletRequest; import visite\_system.demo.Entity.CarLongAppointment; import visite\_system.demo.Entity.CarLongRecord; import visite\_system.demo.Entity.User; import visite\_system.demo.GlobalUtils.QrCodeUtils; import visite\_system.demo.GlobalUtils.ThreadLocalUtil; import visite\_system.demo.Mapper.CarLong\_AppointmentMapper; import visite\_system.demo.Mapper.CarLong\_RecordMapper; import visite\_system.demo.Pojo.CarPicture; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Service.CarLongService;  import javax.servlet.http.HttpServletRequest; import java.io.File; import java.util.Date; import java.util.HashMap; import java.util.Map; import java.util.UUID;  @Service public class CarLongServiceImpl implements CarLongService {   @Autowired  private CarLong\_AppointmentMapper carLongAppointmentMapper;   @Autowired  private QrCodeUtils qrCodeUtils;   @Autowired  private CarLong\_RecordMapper carLongRecordMapper;     @Override  public Result carLongAppoint() throws Exception {  User user = ThreadLocalUtil.*get*();  LambdaQueryWrapper<CarLongAppointment> wrapper = new LambdaQueryWrapper<>();  wrapper.eq(CarLongAppointment::getUserId,user.getId());  CarLongAppointment one = carLongAppointmentMapper.selectOne(wrapper);  if(!ObjectUtils.*isEmpty*(one)){  throw new Exception("当前用户已登记");  }  CarLongAppointment carLongAppointment = new CarLongAppointment();  carLongAppointment.setUserId(user.getId());  carLongAppointment.setAppointTime(new Date());  carLongAppointmentMapper.insert(carLongAppointment);  //获取id  //生成二维码  one = carLongAppointmentMapper.selectOne(wrapper);  String qrCode = qrCodeUtils.createQrCode(String.*valueOf*(one.getId()));  one.setCode(qrCode);  carLongAppointmentMapper.updateById(one);  CarLongRecord carLongRecord = CarLongRecord.*builder*().appointmentId(one.getId()).build();  carLongRecordMapper.insert(carLongRecord);   return Result.*ok*(one);  }   @Override  public Result carLongPictureUp(CarPicture carPicture) {  Long id = carPicture.getId();  CarLongRecord carLongRecord = carLongRecordMapper.selectById(id);  //获取base64编码  String base64 = carPicture.getImage();  //组装  String picture="data:image/png;base64,"+base64;  carLongRecord.setPicture(picture);  carLongRecordMapper.updateById(carLongRecord);  return Result.*ok*();  }   @Override  public Result carLongPictureExamine(CarLongRecord carLongRecord) {  User user = ThreadLocalUtil.*get*();  carLongRecord.setExamineId(user.getId());  LambdaUpdateWrapper<CarLongRecord> wrapper = new LambdaUpdateWrapper<>();  wrapper.set(CarLongRecord::getExamineId,user.getId())  .set(CarLongRecord::getExamineResult,carLongRecord.getExamineResult());  carLongRecordMapper.update(carLongRecord,wrapper);  return Result.*ok*();  }   @Override  public Result carLongBaoAnExamine(CarLongRecord carLongRecord) {  User user = ThreadLocalUtil.*get*();  CarLongRecord carLongRecord1 = carLongRecordMapper.selectById(carLongRecord.getId());  carLongRecord1.setBaoanId(user.getId());  carLongRecord1.setBaoanResult(carLongRecord.getBaoanResult());  carLongRecord1.setGoTime(new Date());  carLongRecordMapper.updateById(carLongRecord1);  return Result.*ok*();  }   @Override  public Result queryLongPictureById(Long id) {  CarLongRecord carLongRecord = carLongRecordMapper.selectById(id);  return Result.*ok*(carLongRecord.getPicture());  } }  package visite\_system.demo.Service.Impl;  import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper; import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service; import org.springframework.util.ObjectUtils; import org.springframework.web.multipart.MultipartFile; import visite\_system.demo.Entity.CarLongRecord; import visite\_system.demo.Entity.CarShortAppointment; import visite\_system.demo.Entity.CarShortRecord; import visite\_system.demo.Entity.User; import visite\_system.demo.GlobalUtils.QrCodeUtils; import visite\_system.demo.GlobalUtils.ThreadLocalUtil; import visite\_system.demo.Mapper.CarShort\_AppointmentMapper; import visite\_system.demo.Mapper.CarShort\_RecordMapper; import visite\_system.demo.Pojo.CarPicture; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Service.CarShortService;  import java.io.File; import java.util.Date; import java.util.UUID;  @Service public class CarShortServiceImpl implements CarShortService {   @Autowired  private CarShort\_AppointmentMapper carShortAppointmentMapper;   @Autowired  private CarShort\_RecordMapper carShortRecordMapper;    @Autowired  private QrCodeUtils qrCodeUtils;  @Override  public Result carShortAppoint(CarShortAppointment carShortAppointment) throws Exception {  User user = ThreadLocalUtil.*get*();  LambdaQueryWrapper<CarShortAppointment> wrapper = new LambdaQueryWrapper<>();  wrapper.eq(CarShortAppointment::getUserId,user.getId());  CarShortAppointment one = carShortAppointmentMapper.selectOne(wrapper);  if(!ObjectUtils.*isEmpty*(one)){  throw new Exception("当前账号已登记");  }  carShortAppointment.setUserId(user.getId());  carShortAppointment.setAppointTime(new Date());  carShortAppointmentMapper.insert(carShortAppointment);  one = carShortAppointmentMapper.selectOne(wrapper);  //生成二维码  String qrCode = qrCodeUtils.createQrCode(String.*valueOf*(one.getId()));  one.setCode(qrCode);  carShortAppointmentMapper.updateById(one);  CarShortRecord carShortRecord = CarShortRecord.*builder*()  .appointmentId(one.getId())  .build();  carShortRecordMapper.insert(carShortRecord);  return Result.*ok*(one);  }   @Override  public Result carShortPictureUp(CarPicture carPicture) {  Long id = carPicture.getId();  CarShortRecord carShortRecord = carShortRecordMapper.selectById(id);  //获取base64编码  String base64 = carPicture.getImage();  //组装  String picture="data:image/png;base64,"+base64;  carShortRecord.setPicture(picture);  carShortRecordMapper.updateById(carShortRecord);  return Result.*ok*();  }   @Override  public Result carShortPictureExamine(CarShortRecord carShortRecord) {  User user = ThreadLocalUtil.*get*();  CarShortRecord carShortRecord1 = carShortRecordMapper.selectById(carShortRecord.getId());  carShortRecord1.setExamineId(user.getId());  carShortRecord1.setExamineResult(carShortRecord.getExamineResult());  carShortRecordMapper.updateById(carShortRecord1);  return Result.*ok*();  }   @Override  public Result carShortBaoAnExamine(CarShortRecord carShortRecord) {  User user = ThreadLocalUtil.*get*();  CarShortRecord carShortRecord1 = carShortRecordMapper.selectById(carShortRecord.getId());  carShortRecord1.setBaoanId(user.getId());  carShortRecord1.setBaoanResult(carShortRecord.getBaoanResult());  carShortRecord1.setGoTime(new Date());  carShortRecordMapper.updateById(carShortRecord1);  return Result.*ok*();  }   @Override  public Result queryShortPictureById(Long id) {  CarShortRecord carShortRecord = carShortRecordMapper.selectById(id);  return Result.*ok*(carShortRecord.getPicture());  }  }  package visite\_system.demo.Service.Impl;  import cn.hutool.extra.qrcode.QrCodeUtil; import cn.hutool.extra.qrcode.QrConfig; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service; import org.springframework.util.ObjectUtils; import visite\_system.demo.Entity.CommonAppointment; import visite\_system.demo.Entity.User; import visite\_system.demo.GlobalUtils.ImageUtils; import visite\_system.demo.GlobalUtils.QrCodeUtils; import visite\_system.demo.GlobalUtils.ThreadLocalUtil; import visite\_system.demo.Mapper.Common\_AppointmentMapper; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Service.CommonService;  import java.awt.image.BufferedImage; import java.text.SimpleDateFormat; import java.time.LocalDate; import java.time.LocalDateTime; import java.time.format.DateTimeFormatter; import java.util.Date;  @Service public class CommonServiceImpl implements CommonService {   @Autowired  private Common\_AppointmentMapper commonAppointmentMapper;   @Autowired  private QrCodeUtils qrCodeUtils;   @Override  public Result commonAppoint(CommonAppointment commonAppointment) {  //获取登录用户信息  User user = ThreadLocalUtil.*get*();  Long userId = user.getId();  commonAppointment.setUserId(userId);  commonAppointment.setAppointTime(new Date());  //存入数据库  int insert = commonAppointmentMapper.insert(commonAppointment);  return Result.*ok*();  }   @Override  public Result commonExamine(CommonAppointment commonAppointment) throws Exception {  User user = ThreadLocalUtil.*get*();  if(ObjectUtils.*isEmpty*(commonAppointment.getId())){  throw new Exception("未传入id");  }  CommonAppointment commonAppointment1 = commonAppointmentMapper.selectById(commonAppointment.getId());  Integer isagree = commonAppointment.getIsagree();  commonAppointment1.setIsagree(isagree);  //同意，生成二维码  if(isagree==0){  String qrCode = qrCodeUtils.createQrCode(String.*valueOf*(commonAppointment1.getId()));  commonAppointment1.setCode(qrCode);  }  commonAppointmentMapper.updateById(commonAppointment1);  return Result.*ok*(commonAppointment1);  }   @Override  public Result commonVisiteEnd(Long id) throws Exception {  CommonAppointment commonAppointment = commonAppointmentMapper.selectById(id);  if(ObjectUtils.*isEmpty*(commonAppointment)){  throw new Exception("预约id不存在");  }  if(commonAppointment.getIsagree()!=0){  throw new Exception("该预约未同意");  }  commonAppointment.setIsgo(0);  commonAppointmentMapper.updateById(commonAppointment);  return Result.*ok*();  } }  package visite\_system.demo.Service.Impl;  import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service; import visite\_system.demo.Entity.Dept; import visite\_system.demo.Mapper.DeptMapper; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Service.DeptService;  import java.util.List;  @Service public class DeptServiceImpl implements DeptService {   @Autowired  private DeptMapper deptMapper;   @Override  public Result queryAllDept() {  List<Dept> depts = deptMapper.selectList(null);  return Result.*ok*(depts);  }   @Override  public Result queryDeptById(Long id) {  Dept dept = deptMapper.selectById(id);  return Result.*ok*(dept);  } }  package visite\_system.demo.Service.Impl;  import com.baomidou.mybatisplus.core.conditions.Wrapper; import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper; import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service; import visite\_system.demo.Entity.\*; import visite\_system.demo.GlobalUtils.ThreadLocalUtil; import visite\_system.demo.Mapper.\*; import visite\_system.demo.Pojo.CarLongExamineInfo; import visite\_system.demo.Pojo.CarShortExamineInfo; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Pojo.VipExamineInfo; import visite\_system.demo.Service.GlobalService;  import java.util.\*;  @Service public class GlobalServiceImpl implements GlobalService {  @Autowired  private Build\_AppointmentMapper buildAppointmentMapper;   @Autowired  private CarLong\_AppointmentMapper carLongAppointmentMapper;   @Autowired  private CarLong\_RecordMapper carLongRecordMapper;   @Autowired  private CarShort\_AppointmentMapper carShortAppointmentMapper;   @Autowired  private CarShort\_RecordMapper carShortRecordMapper;   @Autowired  private Common\_AppointmentMapper commonAppointmentMapper;   @Autowired  private VIP\_AppointmentMapper vipAppointmentMapper;   @Autowired  private VIP\_ExamineMapper vipExamineMapper;   @Autowired  private UserMapper userMapper;   @Autowired  private DeptMapper deptMapper;   @Autowired  private EnterRecordMapper enterRecordMapper;   @Override  public Result queryMyAppointment() {  User user = ThreadLocalUtil.*get*();  Long userId = user.getId();  Integer type = user.getType();  //内部人员  if(user.getIsEmployee()==0){  List<VipExamineInfo> vipExamineInfos = vipAppointmentMapper.queryVipAppointmentByUserId(userId);  return Result.*ok*(vipExamineInfos);  }  //不是内部人员  //普通访客  if(type==0){  LambdaQueryWrapper<CommonAppointment> commonAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  commonAppointmentLambdaQueryWrapper.eq(CommonAppointment::getUserId,userId);  List<CommonAppointment> commonAppointments = commonAppointmentMapper.selectList(commonAppointmentLambdaQueryWrapper);  return Result.*ok*(commonAppointments);  }  //临时物流司机  if(type==3){  LambdaQueryWrapper<CarShortAppointment> carShortAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  carShortAppointmentLambdaQueryWrapper.eq(CarShortAppointment::getUserId,userId);  List<CarShortAppointment> carShortAppointments = carShortAppointmentMapper.selectList(carShortAppointmentLambdaQueryWrapper);  return Result.*ok*(carShortAppointments);  }   //长期物流司机  if(type==2){  LambdaQueryWrapper<CarLongAppointment> carLongAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  carLongAppointmentLambdaQueryWrapper.eq(CarLongAppointment::getUserId,userId);  List<CarLongAppointment> carLongAppointments = carLongAppointmentMapper.selectList(carLongAppointmentLambdaQueryWrapper);  return Result.*ok*(carLongAppointments);  }  //建筑工人  if(type==4){  LambdaQueryWrapper<BuildAppointment> buildAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  buildAppointmentLambdaQueryWrapper.eq(BuildAppointment::getUserId,userId);  List<BuildAppointment> buildAppointments = buildAppointmentMapper.selectList(buildAppointmentLambdaQueryWrapper);  return Result.*ok*(buildAppointments);  }  return Result.*fail*(500,"账号异常");  }   @Override  public Result queryCodeById(Long id) {  User user = ThreadLocalUtil.*get*();  Long userId = user.getId();  Integer type = user.getType();  //内部人员  if(user.getIsEmployee()==0){  VipExamine vipExamine = vipExamineMapper.selectById(id);  return Result.*ok*(vipExamine.getCode());  }  //不是内部人员  //普通访客  if(type==0){  CommonAppointment commonAppointment = commonAppointmentMapper.selectById(id);  return Result.*ok*(commonAppointment.getCode());  }  //临时物流司机  if(type==3){  CarShortAppointment carShortAppointment = carShortAppointmentMapper.selectById(id);  return Result.*ok*(carShortAppointment.getCode());  }   //长期物流司机  if(type==2){  CarLongAppointment carLongAppointment = carLongAppointmentMapper.selectById(id);  return Result.*ok*(carLongAppointment.getCode());  }  //建筑工人  if(type==4){  BuildAppointment buildAppointment = buildAppointmentMapper.selectById(id);  return Result.*ok*(buildAppointment.getCode());  }  return Result.*fail*(500,"账号异常");  }   @Override  public Result queryMyExamine() throws Exception {  User user = ThreadLocalUtil.*get*();  Long id = user.getId();  //是否是内部人员  if(user.getIsEmployee()==1){  throw new Exception("不是内部人员,没有权限");  }   //普通访客  LambdaQueryWrapper<CommonAppointment> commonAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  commonAppointmentLambdaQueryWrapper.eq(CommonAppointment::getVisiteEmployeeId,id);  List<CommonAppointment> commonAppointments = commonAppointmentMapper.selectList(commonAppointmentLambdaQueryWrapper);  //查询登录者是否是部门部长  List<CarLongExamineInfo> carLongExamineInfos=new ArrayList<>();  List<CarShortExamineInfo> carShortExamineInfos=new ArrayList<>();  Dept dept = deptMapper.selectById(user.getDeptId());  Set vipExamineInfos=new HashSet();  if(dept.getManagerId()==id){  List<VipExamineInfo> vipExamineInfo=new ArrayList<>();  //是部长  //vip访客  if(dept.getDeptId()==3){  //是人事部部长  vipExamineInfo= vipAppointmentMapper.queryAllExamin();  }  if(dept.getDeptId()==1){  //是董事部部长  vipExamineInfo= vipAppointmentMapper.queryAllExamin();  }  //是其他部门部长，查询是否有需要我审批的  List<VipExamineInfo> vipExamine = vipAppointmentMapper.queryMyExamineByDeptId(dept.getDeptId());   //合并  vipExamineInfos.addAll(vipExamineInfo);  vipExamineInfos.addAll(vipExamine);   //是物流部部长(审批物流)  if(dept.getDeptId()==6){  carLongExamineInfos= carLongAppointmentMapper.queryMyExamine();  carShortExamineInfos= carShortAppointmentMapper.queryMyExamine();  }  }  //是保安  if(user.getDeptId()==5){  carLongExamineInfos= carLongAppointmentMapper.queryMyExamine();  carShortExamineInfos= carShortAppointmentMapper.queryMyExamine();  }   //建筑工人  LambdaQueryWrapper<BuildAppointment> buildAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  buildAppointmentLambdaQueryWrapper  .eq(BuildAppointment::getVisiteEmployeeId,id);  List<BuildAppointment> buildAppointments = buildAppointmentMapper.selectList(buildAppointmentLambdaQueryWrapper);  HashMap<Integer, Object> hashMap = new HashMap<>();  hashMap.put(0,commonAppointments);  hashMap.put(1,vipExamineInfos);  hashMap.put(2,carLongExamineInfos);  hashMap.put(3,carShortExamineInfos);  hashMap.put(4,buildAppointments);  return Result.*ok*(hashMap);  }   @Override  public Result queryVisitMe() throws Exception {  User user = ThreadLocalUtil.*get*();  if(user.getIsEmployee()==1){  throw new Exception("不是内部人员");  }  //普通访客  LambdaQueryWrapper<CommonAppointment> commonAppointmentLambdaQueryWrapper = new LambdaQueryWrapper<>();  commonAppointmentLambdaQueryWrapper  .eq(CommonAppointment::getVisiteEmployeeId,user.getId())  .eq(CommonAppointment::getIsagree,0)  .eq(CommonAppointment::getIsgo,null); ;  List<CommonAppointment> commonAppointments = commonAppointmentMapper.selectList(commonAppointmentLambdaQueryWrapper);  //vip访客  List<VipExamineInfo> vipExamineInfos = vipAppointmentMapper.queryVisitMe(user.getId());  HashMap<String, Object> hashMap = new HashMap<>();  hashMap.put("0",commonAppointments);  hashMap.put("1",vipExamineInfos);  return Result.*ok*(hashMap);  }   @Override  public Result goin(Long id) {  User user = ThreadLocalUtil.*get*();  //普通访客  if(user.getType()==0){  CommonAppointment commonAppointment = commonAppointmentMapper.selectById(id);  commonAppointment.setIsvisite(0);  commonAppointmentMapper.updateById(commonAppointment);   }else if(user.getType()==2){  //长期物流  CarLongRecord carLongRecord = carLongRecordMapper.selectById(id);  carLongRecord.setEnterTime(new Date());  carLongRecordMapper.updateById(carLongRecord);   }else if(user.getType()==3){  //短期物流  CarShortRecord carShortRecord = carShortRecordMapper.selectById(id);  carShortRecord.setEnterTime(new Date());  carShortRecordMapper.updateById(carShortRecord);   }else if(user.getType()==4){  //施工  BuildAppointment buildAppointment = buildAppointmentMapper.selectById(id);  buildAppointment.setIsvisite(0);  buildAppointmentMapper.updateById(buildAppointment);   }   //插入数据  EnterRecord enterRecord = new EnterRecord();  enterRecord.setUserId(user.getId());  enterRecord.setEnterTime(new Date());  enterRecord.setCarnum(user.getCarnum());  enterRecordMapper.insert(enterRecord);  return Result.*ok*();  }   }  package visite\_system.demo.Service.Impl; import cn.hutool.json.JSONUtil; import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper; import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.data.redis.core.StringRedisTemplate; import org.springframework.stereotype.Service; import org.springframework.util.ObjectUtils; import visite\_system.demo.Constant.RedisKey; import visite\_system.demo.Entity.User; import visite\_system.demo.Mapper.Login\_RegisterMapper; import visite\_system.demo.Pojo.LoginDetail; import visite\_system.demo.Pojo.Result; import visite\_system.demo.Service.Login\_RegisterService;  import java.util.UUID; import java.util.concurrent.TimeUnit;  @Service public class Login\_RegisterServiceImpl extends ServiceImpl<Login\_RegisterMapper,User> implements Login\_RegisterService {  @Autowired  private StringRedisTemplate stringRedisTemplate;  @Autowired  private Login\_RegisterMapper loginRegisterMapper;  @Override  public Result login(LoginDetail loginDetail) throws Exception {  LambdaQueryWrapper<User> userLambdaQueryWrapper = new LambdaQueryWrapper<User>();  userLambdaQueryWrapper.eq(User::getPhone, loginDetail.getPhone());  User user = loginRegisterMapper.selectOne(userLambdaQueryWrapper);  if(ObjectUtils.*isEmpty*(user)  || !user.getPassword().equals(loginDetail.getPassword())){  throw new Exception("用户名或密码错误");  }  //生成token  String token = UUID.*randomUUID*().toString();  //用户信息存入redis  String KEY=RedisKey.*LOGIN\_KEY*+token;  String value=JSONUtil.*toJsonStr*(user);  stringRedisTemplate.opsForValue().set(KEY,value,30, TimeUnit.*MINUTES*);  return Result.*ok*(token);  }   @Override  public Result register(User user) throws Exception {  String phone=user.getPhone();  //检验手机号是否正确  String regx="(13[0-9]|14[01456879]|15[0-35-9]|16[2567]|17[0-8]|18[0-9]|19[0-35-9])\\d{8}";  if(!phone.matches(regx)){  throw new Exception("手机号格式不正确");  }  //查询是否注册过  LambdaQueryWrapper<User> userLambdaQueryWrapper = new LambdaQueryWrapper<>();  userLambdaQueryWrapper.eq(User::getPhone,phone);  User u = loginRegisterMapper.selectOne(userLambdaQueryWrapper);  if(!ObjectUtils.*isEmpty*(u)){  throw new Exception("用户已存在");  }  //注册  user.setType(0);  loginRegisterMapper.insert(user);  return Result.*ok*();  } } |