<template>

<view class="login">

<view class="content">

<view class="header">

<image :src="logoImage"></image>

</view>

<view class="main">

<wInput

v-model="username"

type="text"

maxlength="11"

placeholder="用户名"

:focus="isFocus"

></wInput>

<wInput

v-model="password"

type="password"

maxlength="11"

placeholder="密码"

></wInput>

</view>

<wButton

class="wbutton"

text="登 录"

:rotate="isRotate"

@click="startLogin"

></wButton>

<view class="footer">

<navigator url="register" open-type="navigate">注册账号</navigator>

</view>

</view>

</view>

</template>

<script>

let \_this;

import wInput from '../../components/watch-login/watch-input.vue' //input

import wButton from '../../components/watch-login/watch-button.vue' //button

export default {

data() {

return {

logoImage: '',

username:'',

password:'',

isRotate: false,

isFocus: true

};

},

components:{

wInput,

wButton,

},

mounted() {

\_this= this;

//this.isLogin();

},

methods: {

login() {

this.$queue.showLoading('登录中...');

this.$Request

.login('/loginApp', {

username: this.username,

password: this.password

})

.then(res => {

if (res.code == 200) {

console.log(res)

this.$queue.setData('token', res.token);

this.$queue.setData("username", this.username);

this.$queue.setData("password", this.password);

// this.getUserInfo()

uni.hideLoading();

uni.switchTab({

url: "../home/home"

})

} else {

uni.hideLoading();

this.$queue.showToast(res.msg);

console.log(res.msg)

}

})

.catch(res => {

uni.hideLoading();

});

},

startLogin() {

if (this.username.length == 0) {

this.$queue.showToast('请输入账号');

return;

}

if (this.password.length == 0) {

this.$queue.showToast('请输入密码');

return;

}

this.login();

},

getUserInfo() {

this.$Request.get('/getInfo').then(f => {

console.log(f)

if (f.code == 200) {

console.log(f)

let userInfo = f.userInfo

userInfo.userId = f.user.userId

userInfo.userName = f.user.userName

console.log(userInfo)

this.$queue.setData("UserInfo", userInfo)

}

})

},

}

}

</script>

<style>

@import url("../../components/watch-login/css/icon.css");

@import url("./css/main.css");

</style>

<template>

<view class="register">

<view class="content">

<view class="header">

<image :src="logoImage"></image>

</view>

<view class="main">

<wInput

v-model="nickName"

type="text"

maxlength="8"

placeholder="姓名"

></wInput>

<wInput

v-model="userName"

type="text"

maxlength="12"

placeholder="用户名"

></wInput>

<wInput

v-model="password"

type="password"

maxlength="11"

placeholder="登录密码"

isShowPass

></wInput>

</view>

<wButton

class="wbutton"

text="注 册"

:rotate="isRotate"

@click.native="startReg()"

></wButton>

<view class="footer">

<text

@tap="isShowAgree"

class="cuIcon"

:class="showAgree?'cuIcon-radiobox':'cuIcon-round'"

> 同意</text>

<navigator url="../agreement/agreement" open-type="navigate">《用户协议》</navigator>

</view>

</view>

</view>

</template>

<script>

let \_this;

import wInput from '../../components/watch-login/watch-input.vue' //input

import wButton from '../../components/watch-login/watch-button.vue' //button

import neilModal from '@/components/neil-modal/neil-modal.vue';

export default {

components: {

neilModal,

},

data() {

return {

//logo图片 base64

logoImage: '',

userName:'',

nickName:'',

password:'',

verCode:"",

showAgree:true,

isRotate: false,

}

},

components:{

wInput,

wButton,

},

mounted() {

\_this= this;

},

methods: {

isShowAgree(){

\_this.showAgree = !\_this.showAgree;

},

startReg() {

if(this.isRotate){

return false;

}

if (this.showAgree == false) {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '请先同意《协议》'

});

return false;

}

if (this.nickName =='') {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '姓名不能为空'

});

return false;

}

if (this.userName =='') {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '用户名不能为空'

});

return false;

}

if (this.password =='') {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '密码不能为空'

});

return false;

}

if (this.nickName.length <2) {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '请输入正确的姓名'

});

return false;

}

if (this.userName.length <6||this.userName.length >12) {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '请输入6到12位的用户名'

});

return false;

}

if (this.password.length < 6 ||this.password.length >10) {

uni.showToast({

icon: 'none',

position: 'bottom',

title: '请输入6到10位的密码'

});

return false;

}

console.log("注册成功")

\_this.isRotate=true

this.register()

},

register(){

let user = {

userName:this.userName,

password:this.password,

nickName:this.nickName

}

this.$Request.post('/register',user).then(f => {

console.log(f)

if (f.code == 200) {

console.log(f)

\_this.isRotate=false

this.$queue.showToast('注册成功')

uni.navigateTo({

url:"./login"

})

}else{

\_this.isRotate=false

uni.showToast({

icon: 'none',

position: 'bottom',

title: '注册失败!'

});

}

})

}

}

}

</script>

<style>

@import url("../../components/watch-login/css/icon.css");

@import url("./css/main.css");

</style>

<template>

<view class="formContainer">

<u-form :model="form" :rules="rules" ref="uForm" :errorType="errorType">

<u-form-item label="视力类别" label-width="150" prop="visionType" :required='true'>

<u-input v-model="form.visionType" type="select" placeholder="请选择视力类别" :select-open="visiontypeListShow" @click="visiontypeListShow = true"/>

</u-form-item>

<u-form-item label="左眼视力" label-width="150" prop="visionLeft" :required='true'>

<u-input v-model="form.visionLeft" type="float" placeholder="请输入视力" />

</u-form-item>

<u-form-item label="右眼视力" label-width="150" prop="visionRight" :required='true'>

<u-input v-model="form.visionRight" type="float" placeholder="请输入视力" />

</u-form-item>

<u-form-item label="左眼眼压" label-width="150" prop="pressureLeft" :required='true'>

<u-input v-model="form.pressureLeft" type="number" placeholder="请输入眼压" />

</u-form-item>

<u-form-item label="右眼眼压" label-width="150" prop="pressureRight" :required='true'>

<u-input v-model="form.pressureRight" type="number" placeholder="请输入眼压" />

</u-form-item>

<u-form-item label="眼压测量时间" label-width="180" :required='true' prop="pressureTime">

<u-input v-model="form.pressureTime" type="select" placeholder="请选择测量时间" @click="eyePressurePickerShow=true" />

</u-form-item>

<u-form-item label="运动类型" label-width="150" :required='true' prop="motion">

<u-input v-model="form.motion" type="select" placeholder="请选择运动类型" @click="motionTypeListShow=true" />

</u-form-item>

<u-form-item label="运动时长" label-width="150" prop="motionDuration" :required='true'>

<u-input v-model="form.motionDuration" placeholder="请输入运动时长" />

</u-form-item>

<u-form-item label="运动备注" label-width="150">

<u-input v-model="form.motionRemark" placeholder="请填写运动备注" />

</u-form-item>

<u-form-item label="用药名称" label-width="150" :required='true' prop="medication">

<u-input v-model="form.medication" placeholder="请输入用药名称" />

</u-form-item>

<u-form-item label="用药时刻" label-width="150" :required='true' prop="medicationMoment">

<u-input v-model="form.medicationMoment" placeholder="请选择用药时刻" type="select" @click="medicationMomentPickerShow=true" />

<view v-if="form.medicationMoment">

<u-icon name="backspace" size="36" @click="clearMedicationMoment"></u-icon>

</view>

</u-form-item>

<u-form-item label="不良反应" label-width="150">

<u-input v-model="form.medicationAdr" placeholder="请输入不良反应" />

</u-form-item>

<u-form-item label="停药时刻" label-width="150">

<u-input v-model="form.medicationSuspend" placeholder="请选择停药时刻" type="select" @click="medicationSuspendPickerShow=true" />

</u-form-item>

<u-form-item label="用药备注" label-width="150">

<u-input v-model="form.medicationRemark" placeholder="请输入用药备注" />

</u-form-item>

<u-form-item label="情绪类型" label-width="150" :required='true' prop="emotion">

<u-input v-model="form.emotion" placeholder="请输入情绪类型" />

</u-form-item>

<u-form-item label="记录日期" label-width="150" :required='true' prop="recordTime">

<u-input v-model="form.recordTime" placeholder="请选择记录日期" type="select" @click="recordTimePickerShow=true" />

</u-form-item>

</u-form>

<u-gap height="20"></u-gap>

<u-section :title="getHistoryTitle()" color="#2979ff" sub-title="查看更多" @click="selectHistory"></u-section>

<u-gap height="20"></u-gap>

<u-button @click="submit">提交</u-button>

<u-select @confirm="selectVisionTypeConfirm" v-model="visiontypeListShow" :list="visiontypeList"></u-select>

<u-select @confirm="selectMotionTypeConfirm" v-model="motionTypeListShow" :list="motionTypeList"></u-select>

<u-picker mode="time" v-model="eyePressurePickerShow" :params="timePickerParams" @confirm="eyePressureDateConfirm"></u-picker>

<u-picker mode="time" v-model="recordTimePickerShow" :params="timePickerParams" @confirm="recordTimeDateConfirm"></u-picker>

<u-picker mode="time" v-model="medicationMomentPickerShow" :params="medicationMomentTimePickerParams" @confirm="selectMedicationMomentConfirm"></u-picker>

<u-picker mode="time" v-model="medicationSuspendPickerShow" :params="medicationMomentTimePickerParams" @confirm="medicationSuspendConfirm"></u-picker>

</view>

</template>

<script>

export default {

data() {

return {

form: {

visionType: '',

visionLeft: '',

visionRight: '',

pressureLeft: '',

pressureRight: '',

pressureTime: '',

motion:'',

motionDuration:'',

motionRemark:'',

medication:'',

medicationMoment:'',

medicationAdr:'',

medicationSuspend:'',

medicationRemark:'',

emotion:'',

recordTime:'',

},

visiontypeList: [

{label:'矫正',value:0},

{label:'裸眼',value:1},

],

visiontypeListShow: false,

motionTypeList: [

{label:'散步',value:0},

{label:'快走',value:1},

{label:'慢跑',value:2},

{label:'骑自行车',value:3},

{label:'游泳',value:4},

{label:'其他',value:5},

],

motionTypeListShow:false,

timePickerParams: {

year: true,

month: true,

day: true,

hour: true,

minute: true,

},

eyePressurePickerShow: false,

recordTimePickerShow: false,

medicationMomentPickerShow: false,

medicationMomentTimePickerParams:{

hour: true,

minute: true,

},

medicationSuspendPickerShow: false,

rules: {

visionType: [

{

required: true,

message: '此为必填字段',

// blur和change事件触发检验

trigger: ['blur', 'change'],

}

],

visionLeft: [

// 必填规则

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

{

type: 'float',

message: '只能输入数字',

trigger: ['change','blur'],

}

],

visionRight: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

{

type: 'float',

message: '只能输入数字',

trigger: ['change','blur'],

}

],

pressureLeft: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

{

type: 'number',

message: '只能输入数字',

trigger: ['change','blur'],

}

],

pressureRight: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

{

type: 'number',

message: '只能输入数字',

trigger: ['change','blur'],

}

],

pressureTime: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

],

motion: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

],

motionDuration: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

{

type: 'number',

message: '只能输入数字',

trigger: ['change','blur'],

}

],

medication: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

],

medicationMoment: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

],

emotion:[

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

],

recordTime: [

{

required: true,

message: '此为必填字段',

trigger: ['blur', 'change'],

},

],

},

errorType: ['message'],

history: {},

};

},

onReady() {

this.$refs.uForm.setRules(this.rules);

},

onLoad() {

this.getHistory()

},

methods: {

selectVisionTypeConfirm(e) {

if(e){

if(e.length>0){

this.form.visionType=e[0].label

}

}

},

selectMotionTypeConfirm(e){

if(e){

if(e.length>0){

this.form.motion=e[0].label

}

}

},

submit() {

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

if (valid) {

console.log('验证通过');

this.$queue.showLoading("正在修改...")

let param = this.form

param.userId=this.$queue.getData('UserInfo').userId;

console.log(param)

this.$Request.post("/system/record",

param

).then(res => {

console.log(res)

uni.hideLoading()

if (res.code == 200) {

this.$queue.showToast("记录成功")

this.history.add(param)

} else {

this.$queue.showToast(res.msg)

}

})

console.log("提交记录")

} else {

console.log('验证失败');

uni.showToast({

title:'请完善记录信息',

icon:"none"

})

}

});

},

selectHistory() {

console.log("选择历史记录")

if(this.history && this.history.length>0){

uni.navigateTo({

url:'history',

})

}else{

uni.showToast({

title:'暂时没有历史记录',

icon:"none"

})

}

},

eyePressureDateConfirm(e){

let date = new Date();

if(e.year<=date.getFullYear()

&& e.month<=(date.getMonth()+1)

&& e.day <= date.getDate()){

this.form.pressureTime=e.year+"-"+e.month+"-"+e.day+" "+e.hour+":"+e.minute

}else{

uni.showToast({

title:'时间不可超前选择',

icon:"none"

})

}

},

recordTimeDateConfirm(e){

let date = new Date();

if(e.year<=date.getFullYear()

&& e.month<=(date.getMonth()+1)

&& e.day <= date.getDate()){

this.form.recordTime=e.year+"-"+e.month+"-"+e.day+" "+e.hour+":"+e.minute

}else{

uni.showToast({

title:'时间不可超前选择',

icon:"none"

})

}

},

selectMedicationMomentConfirm(e){

this.form.medicationMoment=this.form.medicationMoment+e.hour+":"+e.minute+";";

console.log(e)

},

clearMedicationMoment(){

this.form.medicationMoment='';

},

medicationSuspendConfirm(e){

this.form.medicationSuspend=e.hour+":"+e.minute+";";

},

getHistory(){

let userInfo= this.$queue.getData("UserInfo")

if(userInfo)

{

this.$Request.get("/system/record/getInfoById/"+userInfo.userId).then(res=>{

if(res.code == 200)

{

this.history=res.data

console.log(JSON.stringify(this.history))

}

}).catch(res=>{console.log("getHistory error")})

}

},

getHistoryTitle(){

let title="历史记录";

if(this.history && this.history.length>0){

title+=" ("+this.history.length+")"

}

return title

},

},

};

</script>

<style>

.formContainer {

margin-left: 20rpx;

margin-right: 20rpx;

}

.formLabel {

width: 200rpx;

}

</style>

<template>

<view>

<u-tabs name="cate\_name" count="cate\_count" :list="tyepList" :is-scroll="false" :current="currentType" @change="onTypeChanged"></u-tabs>

<view v-if="currentType===0">

<visionHistory :history="history"></visionHistory>

</view>

<view v-else-if="currentType===1">

<pressureHistory :history="history"></pressureHistory>

</view>

<view v-else-if="currentType===2">

<motionHistory :history="history"></motionHistory>

</view>

<view v-else-if="currentType===3">

<medicationHistory :history="history"></medicationHistory>

</view>

<view v-else-if="currentType===4">

<emotionHistory :history="history"></emotionHistory>

</view>

</view>

</template>

<script>

import visionHistory from './vision-history'

import pressureHistory from './pressure-history'

import motionHistory from './motion-history'

import medicationHistory from './medication-history'

import emotionHistory from './emotion-history'

export default {

components:{

visionHistory,pressureHistory,motionHistory,medicationHistory,emotionHistory,

},

data() {

return {

tyepList: [{cate\_name: '视力'}, {cate\_name: '眼压'}, {cate\_name: '运动'}, {cate\_name: '用药'}, {cate\_name: '情绪'}],

currentType: 0,

history:[],

tableInfo:[

{tr:['时间','左眼','右眼','裸眼/矫正']}

],

selectTable: {}

}

},

onLoad(options) {

this.getHistory()

this.selectTable=this.tableInfo[0]

},

methods: {

onTypeChanged(index) {

this.currentType = index;

this.selectTable=this.tableInfo[index]

},

getHistory(){

let userInfo= this.$queue.getData("UserInfo")

if(userInfo)

{

this.$Request.get("/system/record/getInfoById/"+userInfo.userId).then(res=>{

if(res.code == 200)

{

this.history=res.data

uni.setStorageSync("history",this.history)

console.log(JSON.stringify(this.history))

this.$forceUpdate()

}

}).catch(res=>{console.log("getHistory error")})

}

},

}

}

</script>

<style>

.container{

width: 1000rpx;

}

</style>

<template>

<view>

<u-table>

<u-tr>

<u-th>日期</u-th>

<u-th>情绪类型</u-th>

</u-tr>

<view v-for="item in history">

<u-tr>

<u-td>{{item.recordTime}}</u-td>

<u-td>{{getEmotion(item.emotion)}}</u-td>

</u-tr>

</view>

</u-table>

</view>

</template>

<script>

export default {

props:["history"],

data() {

return {

name: 'emotion-history',

}

},

onShow() {

},

methods: {

getEmotion(emotion){

if(emotion){}else{

emotion="-";

}

return emotion;

}

}

}

</script>

<style>

</style>  
<template>

<view>

<scroll-view scroll-x='true' show-scrollbar='true' class="container">

<u-table>

<u-tr>

<u-th >用药时间</u-th>

<u-th>药品名</u-th>

<u-th>频次</u-th>

<u-th>不良反应</u-th>

<u-th>停药时间</u-th>

<u-th>备注</u-th>

</u-tr>

<view v-for="item in history">

<u-tr>

<u-td>{{item.recordTime}}</u-td>

<u-td>{{getMedication(item.medication)}}</u-td>

<u-td>{{getFrequency(item.medicationMoment)}}</u-td>

<u-td>{{getMedicationAdr(item.medicationAdr)}}</u-td>

<u-td>{{getMedicationSuspend(item.medicationSuspend)}}</u-td>

<u-td>{{getMedicationRemark(item.medicationRemark)}}</u-td>

</u-tr>

</view>

</u-table>

</scroll-view>

</view>

</template>

<script>

export default {

props:["history"],

data() {

return {

}

},

onShow() {

},

methods: {

getMedication(medication){

if(medication){}else{

medication="-"

}

return medication;

},

getFrequency(medicationMoment){

let frequency=0

if(medicationMoment){

let args = medicationMoment.split(';')

frequency=args.length

}

return frequency

},

getMedicationAdr(medicationAdr){

if(medicationAdr){}else{

medicationAdr="-"

}

return medicationAdr;

},

getMedicationSuspend(medicationSuspend){

if(medicationSuspend){}else{

medicationSuspend='-'

}

return medicationSuspend

},

getMedicationRemark(medicationRemark){

if(medicationRemark){}else{

medicationRemark="-"

}

return medicationRemark;

},

}

}

</script>

<style>

.container{

/\* width: 1200rpx; \*/

white-space: nowrap;

}

</style>

<template>

<view>

<u-table>

<u-tr>

<u-th>时间</u-th>

<u-th>类型</u-th>

<u-th>时长</u-th>

<u-th>备注</u-th>

</u-tr>

<view v-for="item in history">

<u-tr>

<u-td>{{item.recordTime}}</u-td>

<u-td>{{getMotion(item.motion)}}</u-td>

<u-td>{{getDuration(item.motionDuration)}}</u-td>

<u-td>{{getMark(item.motionRemark)}}</u-td>

</u-tr>

</view>

</u-table>

</view>

</template>

<script>

export default {

props:["history"],

data() {

return {

}

},

onShow() {

},

methods: {

getMotion(motion){

if(motion){}else{

motion="-";

}

return motion;

},

getDuration(duration){

if(duration && duration>0){}else{

duration=0;

}

return duration;

},

getMark(mark){

if(mark){}else{

mark="-";

}

return mark;

},

}

}

</script>

<style>

</style>

<template>

<view class="content">

<view style="background-color: #FFFFFF;">

<qiun-title-bar title="眼压折线图" />

<view class="charts-box">

<qiun-data-charts

type="line"

:chartData="getChartData()"

background="none"

/>

</view>

</view>

<uni-section></uni-section>

<view class="itme-box">

<qiun-title-bar title="眼压数据表"/>

<view class="h-table h-table-primary">

<view class="h-tr h-tr-2 h-thead">

<view class="h-td">时间</view>

<view class="h-td">左眼</view>

<view class="h-td">右眼</view>

<view class="h-td">已用药物</view>

<view class="h-td">备注</view>

</view>

<view class="h-tr h-tr-2" v-for="(item ,index) in history" :index="index">

<view class="h-td">{{item.recordTime}}</view>

<view class="h-td">{{item.pressureLeft}}</view>

<view class="h-td">{{item.pressureRight}}</view>

<view class="h-td">{{item.medication}}</view>

<view class="h-td">{{item.motion}} {{item.emotion}}</view>

</view>

</view>

</view>

</view>

</template>

<script>

export default {

props:["history"],

data() {

return {

}

},

methods: {

getChartData(){

let chartData={};

let categories=[];

let leftSeries=[];

let rightSeries=[];

for(let i=0;i<this.history.length;i++){

let item = this.history[i]

categories.push(item.recordTime)

leftSeries.push(item.pressureLeft)

rightSeries.push(item.pressureRight)

}

let series = []

series.push({"name":"左眼","data":leftSeries})

series.push({"name":"右眼","data":rightSeries})

chartData.categories=categories

chartData.series=series

return chartData

},

}

}

</script>

<style lang="scss">

@import "../../static/css/helang-table.scss";

.content {

font-size: 28rpx;

background-color: #f3f3f3;

padding: 20rpx;

.red {

background-color: #DD524D;

color: #fff;

text-align: center;

}

.green {

background-color: #42b983;

color: #fff;

text-align: center;

}

.blue {

background-color: #007AFF;

color: #fff;

text-align: center;

}

.yellow {

background-color: #ffaa00;

color: #fff;

text-align: center;

}

.itme-box {

padding: 20rpx;

background-color: #fff;

margin-bottom: 20rpx;

.title {

padding-bottom: 20rpx;

margin-bottom: 20rpx;

border-bottom: #e5e5e5 solid 1px;

line-height: normal;

}

.width-fill {

width: 100%;

}

.preview-box {

height: 400rpx;

box-shadow: 0 0 4px rgba(0, 0, 0, 0.3);

}

}

.charts-box {

width: 100%;

height: 300px;

}

}

.charts-box{

width: 100%;

height:300px;

}

</style>

<template>

<view>

<u-table>

<u-tr>

<u-th>时间</u-th>

<u-th>左眼</u-th>

<u-th>右眼</u-th>

<u-th>裸眼/矫正</u-th>

</u-tr>

<view v-for="item in history">

<u-tr>

<u-td>{{item.recordTime}}</u-td>

<u-td>{{item.visionLeft}}</u-td>

<u-td>{{item.visionRight}}</u-td>

<u-td>{{getVisionType(item.visionType)}}</u-td>

</u-tr>

</view>

</u-table>

</view>

</template>

<script>

export default {

props:["history"],

data() {

return {

}

},

onShow() {

},

methods: {

getVisionType(visionType){

if(visionType){

} else{

visionType='裸眼'

}

return visionType

},

},

}

</script>

<style>

</style>

<template>

<view class="warp">

<view class="example-body">

<uni-grid :column="3" :show-border="false" :square="false" @change="change">

<uni-grid-item v-if="index < 6" v-for="(item,index) in list" :index="index" :key="index">

<view v-if="index==1" class="grid-item-box">

<image class="image" :src="item.url" mode="aspectFill" />

<text class="text">{{item.text}}</text>

<view v-if="item.badge" class="grid-dot">

<uni-badge :text="item.badge" :type="item.type" />

</view>

<text v-if="currentPressureType<=1" class="pressureText"> {{ currentPressure }} </text>

<text v-else class="pressureOverflowText">{{ currentPressure }}</text>

</view>

<view v-if="index!=1" class="grid-item-box">

<image class="image" :src="item.url" mode="aspectFill" />

<text class="text">{{item.text}}</text>

<view v-if="item.badge" class="grid-dot">

<uni-badge :text="item.badge" :type="item.type" />

</view>

</view>

</uni-grid-item>

</uni-grid>

</view>

<view class="uni-list">

<view class="uni-list-cell" hover-class="uni-list-cell-hover" v-for="(value,key) in listData" :key="key"

@click="goDetail(value)">

<view class="uni-media-list" v-if="key<4&&value.title">

<!-- <image class="uni-media-list-logo" :src="value.cover"></image> -->

<u-parse class="uni-media-list-logo" :content="value.cover" />

<view class="uni-media-list-body">

<view class="uni-media-list-text-top">{{value.title}}</view>

<view class="uni-media-list-text-bottom">

<text>{{value.author\_name}}</text>

<text>{{value.published\_at}}</text>

</view>

</view>

</view>

</view>

</view>

</view>

</template>

<script>

import uParse from '@/components/gaoyia-parse/parse.vue'

import neilModal from '@/components/neil-modal/neil-modal.vue';

import LbPicker from '@/components/lb-picker';

import cmdNavBar from "@/components/cmd-nav-bar/cmd-nav-bar.vue"

import cmdPageBody from "@/components/cmd-page-body/cmd-page-body.vue"

import cmdTransition from "@/components/cmd-transition/cmd-transition.vue"

import cmdCelItem from "@/components/cmd-cell-item/cmd-cell-item.vue"

import uniSection from '@/components/uni-section/uni-section.vue'

import uniGrid from '@/components/uni-grid/uni-grid.vue'

import uniGridItem from '@/components/uni-grid-item/uni-grid-item.vue'

import uniBadge from '@/components/uni-badge/uni-badge.vue'

import wInput from '../../components/watch-login/watch-input.vue' //input

var dateUtils = require('../../common/util.js').dateUtils;

export default {

components: {

uniSection,

uniGrid,

uniGridItem,

uniBadge,

cmdNavBar,

cmdPageBody,

cmdTransition,

cmdCelItem,

neilModal,

LbPicker,

uParse

},

data() {

return {

showModal1: false,

userInfo: {},

list: [{

url: '/static/img/home/date.png',

text: '眼压数据'

},

{

url: '/static/img/home/pen.png',

text: '记录眼压'

},

{

url: '/static/img/home/eye.png',

text: '就诊提醒'

},

{

url: '/static/img/home/1.png',

text: '药物依从性量表'

},

{

url: '/static/img/home/2.png',

text: '感知压力量表'

},

{

url: '/static/img/home/clock.png',

text: '用药提醒'

}

],

banner: {},

listData: [{

cover: '',

title: '',

author\_name: '',

published\_at: ''

},

{

cover: '',

title: '',

author\_name: '',

published\_at: ''

},

{

cover: '',

title: '',

author\_name: '',

published\_at: ''

},

{

cover: '',

title: '',

author\_name: '',

published\_at: ''

}

],

last\_id: "",

reload: false,

currentPressureType: 0,

currentPressure:"",

}

},

onLoad() {

// this.getBanner();

// this.getList();

this.loadData();

},

onShow() {

this.getUserInfo()

},

onPullDownRefresh() {

this.reload = true;

this.last\_id = "";

this.getBanner();

this.getList();

},

onReachBottom() {

this.getList();

},

methods: {

loadData() {

this.userInfo = this.$queue.getData('UserInfo')

this.currentPressure=this.getCurrentPressure()

this.$Request.get('/system/notice/list').then(

res => {

if (res.code == 200) {

console.log(res)

for (let i = 0; i < res.total; i++) {

if (i >= 4) {

return

}

if (res.total < 4) {

this.listData[i].cover = res.rows[i].picture

this.listData[i].title = res.rows[i].noticeTitle

this.listData[i].author\_name = res.rows[i].auther

this.listData[i].published\_at = res.rows[i].releasetime

this.listData[i].content = res.rows[i].noticeContent

console.log(this.listData[i].content)

} else {

this.listData[i].cover = res.rows[res.total - 4 + i].picture

this.listData[i].title = res.rows[res.total - 4 + i].noticeTitle

this.listData[i].author\_name = res.rows[res.total - 4 + i].auther

this.listData[i].published\_at = res.rows[res.total - 4 + i].releasetime

this.listData[i].content = res.rows[res.total - 4 + i].noticeContent

}

}

}

})

},

change(e) {

let {

index

} = e.detail

this.list[index].badge && this.list[index].badge++

if (index == 0) {

uni.navigateTo({

url: '../Intraocular-pressure/Intraocular-pressure'

})

} else if (index == 1) {

uni.reLaunch({

url: '../record/record'

})

} else if (index == 2) {

uni.navigateTo({

url: '../visit-reminder/visit-reminder'

})

} else if (index == 3) {

uni.navigateTo({

url: '../questionnaire1/questionnaire1'

})

} else if (index == 4) {

uni.navigateTo({

url: '../questionnaire2/questionnaire2'

})

} else if (index == 5) {

uni.navigateTo({

url: '../medication-reminder/medication-reminder'

})

}

uni.showToast({

title: `点击第${index+1}个宫格`,

icon: 'none'

})

},

add() {

if (this.dynamicList.length < 9) {

this.dynamicList.push({

url: `/static/c${this.dynamicList.length+1}.png`,

text: `Grid ${this.dynamicList.length+1}`,

color: this.dynamicList.length % 2 === 0 ? '#f5f5f5' : "#fff"

})

} else {

uni.showToast({

title: '最多添加9个',

icon: 'none'

});

}

},

del() {

this.dynamicList.splice(this.dynamicList.length - 1, 1)

},

getBanner() {

let data = {

column: "id,post\_id,title,author\_name,cover,published\_at" //需要的字段名

};

uni.request({

url: 'https://unidemo.dcloud.net.cn/api/banner/36kr',

data: data,

success: (data) => {

uni.stopPullDownRefresh();

if (data.statusCode == 200) {

this.banner = data.data;

}

},

fail: (data, code) => {

console.log('fail' + JSON.stringify(data));

}

})

},

getList() {

var data = {

column: "id,post\_id,title,author\_name,cover,published\_at" //需要的字段名

};

if (this.last\_id) { //说明已有数据，目前处于上拉加载

data.minId = this.last\_id;

data.time = new Date().getTime() + "";

data.pageSize = 10;

}

uni.request({

url: 'https://unidemo.dcloud.net.cn/api/news',

data: data,

success: (data) => {

if (data.statusCode == 200) {

let list = this.setTime(data.data);

this.listData = this.reload ? list : this.listData.concat(list);

this.last\_id = list[list.length - 1].id;

this.reload = false;

}

},

fail: (data, code) => {

console.log('fail' + JSON.stringify(data));

}

})

},

goDetail: function(e) {

// if (!/前|刚刚/.test(e.published\_at)) {

// e.published\_at = dateUtils.format(e.published\_at);

// }

let detail = {

author\_name: e.author\_name,

cover: e.cover,

id: e.id,

post\_id: e.post\_id,

published\_at: e.published\_at,

title: e.title,

content: e.content

}

uni.navigateTo({

url: "../list2detail-detail/list2detail-detail?detailDate=" + encodeURIComponent(JSON

.stringify(

detail))

})

},

setTime: function(items) {

var newItems = [];

items.forEach((e) => {

newItems.push({

author\_name: e.author\_name,

cover: e.cover,

id: e.id,

post\_id: e.post\_id,

published\_at: dateUtils.format(e.published\_at),

title: e.title

});

});

return newItems;

},

changeEye(e) {

this[e] = true

},

closeModal() {

this.showModal1 = false

this.showModal2 = false

},

cancelModal(e) {

if (e == 'showModal1') {

this.visionLeft = ''

this.visionRight = ''

} else if (e == 'showModal2') {

this.pressureLeft = ''

this.pressureRight = ''

}

},

confirmModal(e) {

},

handleTap(name) {

this.$refs[name].show()

},

handleChange(e) {

console.log('change::', e)

},

handleConfirm(e) {

console.log('confirm::', e)

},

handleCancel(e) {

console.log('cancel::', e)

},

getUserInfo() {

this.$Request.get('/getInfo').then(f => {

if (f.code == 200) {

console.log(f)

let userInfo = {}

userInfo.userId = f.user.userId

userInfo.userName = f.user.userName

this.$Request.get("/system/record/getInfoById/" + userInfo.userId).then(res => {

if (res.code == 200) {

if (res.data.length!=0) {

console.log(res)

userInfo.pressureLeft = res.data[res.data.length - 1].pressureLeft

userInfo.pressureRight = res.data[res.data.length - 1].pressureRight

}else{

userInfo.pressureLeft = ''

userInfo.pressureRight = ''

}

this.$queue.setData("UserInfo", userInfo)

this.loadData()

}

})

}

})

},

getCurrentPressure()

{

var pressureText=""

this.currentPressureType=0

if (this.userInfo)

{

if(this.userInfo.pressureLeft)

{

if (this.userInfo.pressureLeft<10||this.userInfo.pressureRight<10)

{

this.currentPressureType=2

}

else if(this.userInfo.pressureLeft>21||this.userInfo.pressureRight>21)

{

this.currentPressureType=3

}

else

{

this.currentPressureType=1

}

}

}

if(this.currentPressureType==0)

{

pressureText="正常眼压值：10~21mmHg"

}

else if(this.currentPressureType==2)

{

pressureText='左眼:'+ this.userInfo.pressureLeft+'右眼'+this.userInfo.pressureRight+'\n低眼压'

}

else if(this.currentPressureType==3)

{

pressureText='左眼:'+this.userInfo.pressureLeft+'右眼'+this.userInfo.pressureRight+'\n高眼压'

}

else

{

pressureText='左眼:'+ this.userInfo.pressureLeft+'右眼'+this.userInfo.pressureRight+'\n正常眼压'

}

return pressureText

},

}

}

</script>

<style>

@font-face {

font-family: uniicons;

font-weight: normal;

font-style: normal;

src: url('~@/static/uni.ttf') format('truetype');

}

page {

display: flex;

flex-direction: column;

box-sizing: border-box;

background-color: #efeff4;

min-height: 100%;

height: auto;

}

view {

font-size: 14px;

line-height: inherit;

}

.word-btn-white {

font-size: 18px;

color: #FFFFFF;

}

.word-btn {

/\* #ifndef APP-NVUE \*/

display: flex;

/\* #endif \*/

flex-direction: row;

align-items: center;

justify-content: center;

border-radius: 6px;

height: 48px;

margin: 15px;

background-color: #007AFF;

}

.word-btn--hover {

background-color: #4ca2ff;

}

.image {

width: 80rpx;

height: 80rpx;

}

.text {

font-size: 30rpx;

margin-top: 10rpx;

}

.example-body {

/\* #ifndef APP-NVUE \*/

display: block;

/\* #endif \*/

}

.grid-dynamic-box {

margin-bottom: 15px;

}

.grid-item-box {

flex: 1;

/\* position: relative;

/\* #ifndef APP-NVUE \*/

display: flex;

/\* #endif \*/

flex-direction: column;

align-items: center;

justify-content: center;

padding: 10px 0;

}

.grid-dot {

position: absolute;

top: 5px;

right: 15px;

}

.swiper {

height: 420px;

}

.banner {

height: 360upx;

overflow: hidden;

position: relative;

background-color: #ccc;

}

.banner-img {

width: 100%;

}

.banner-title {

max-height: 84upx;

overflow: hidden;

position: absolute;

left: 30upx;

bottom: 30upx;

width: 90%;

font-size: 32upx;

font-weight: 400;

line-height: 42upx;

color: white;

z-index: 11;

}

.uni-list {

background-color: #FFFFFF;

position: relative;

width: 100%;

display: flex;

flex-direction: column;

}

.uni-list:after {

position: absolute;

z-index: 10;

right: 0;

bottom: 0;

left: 0;

height: 1px;

content: '';

-webkit-transform: scaleY(.5);

transform: scaleY(.5);

background-color: #c8c7cc;

}

.uni-list::before {

position: absolute;

z-index: 10;

right: 0;

top: 0;

left: 0;

height: 1px;

content: '';

-webkit-transform: scaleY(.5);

transform: scaleY(.5);

background-color: #c8c7cc;

}

.uni-list-cell {

position: relative;

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

}

.uni-list-cell-hover {

background-color: #eee;

}

.uni-list-cell::after {

position: absolute;

z-index: 3;

right: 0;

bottom: 0;

left: 30upx;

height: 1px;

content: '';

-webkit-transform: scaleY(.5);

transform: scaleY(.5);

background-color: #c8c7cc;

}

.uni-list .uni-list-cell:last-child::after {

height: 0upx;

}

.uni-media-list {

padding: 30upx 30upx;

box-sizing: border-box;

display: flex;

width: 100%;

flex-direction: row;

}

.uni-navigate-right.uni-media-list {

padding-right: 74upx;

}

.input-password {

height: 80upx;

width: 100%;

display: flex;

flex-direction: row;

justify-content: center;

align-items: center;

position: relative;

padding-left: 30upx;

box-sizing: border-box;

}

.input-name::after {

content: " ";

position: absolute;

left: 30upx;

bottom: 0;

right: 0;

height: 1px;

border-top: 1px solid #e5e5e5;

transform-origin: 0 0;

transform: scaleY(.5);

}

.input-name view,

.input-password view {

width: 140upx;

height: 50upx;

line-height: 50upx;

font-size: 28upx;

color: #333333;

}

.input-name input,

.pressureText

{

width: 260rpx;

font-size: 12px;

margin: auto;

text-align: center;

}

.pressureOverflowText

{

color: #FF0000;

width: 260rpx;

font-size: 12px;

margin: auto;

text-align: center;

}

.pressureHighlightText

{

color: #00BFFF;

}

</style>

package com.ruoyi.system.controller;

import java.util.List;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.ruoyi.common.annotation.Log;

import com.ruoyi.common.core.controller.BaseController;

import com.ruoyi.common.core.domain.AjaxResult;

import com.ruoyi.common.enums.BusinessType;

import com.ruoyi.system.domain.SysMedicalRecord;

import com.ruoyi.system.service.ISysMedicalRecordService;

import com.ruoyi.common.utils.poi.ExcelUtil;

import com.ruoyi.common.core.page.TableDataInfo;

@RestController

@RequestMapping("/system/medicalRecord")

public class SysMedicalRecordController extends BaseController

{

@Autowired

private ISysMedicalRecordService sysMedicalRecordService;

@GetMapping("/list")

public TableDataInfo list(SysMedicalRecord sysMedicalRecord)

{

startPage();

List<SysMedicalRecord> list = sysMedicalRecordService.selectSysMedicalRecordList(sysMedicalRecord);

return getDataTable(list);

}

@Log(title = "就诊记录", businessType = BusinessType.EXPORT)

@GetMapping("/export")

public AjaxResult export(SysMedicalRecord sysMedicalRecord)

{

List<SysMedicalRecord> list = sysMedicalRecordService.selectSysMedicalRecordList(sysMedicalRecord);

ExcelUtil<SysMedicalRecord> util = new ExcelUtil<SysMedicalRecord>(SysMedicalRecord.class);

return util.exportExcel(list, "就诊记录数据");

}

@GetMapping(value = "/{recordId}")

public AjaxResult getInfo(@PathVariable("recordId") Long recordId)

{

return AjaxResult.success(sysMedicalRecordService.selectSysMedicalRecordById(recordId));

}

@Log(title = "就诊记录", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody SysMedicalRecord sysMedicalRecord)

{

return toAjax(sysMedicalRecordService.insertSysMedicalRecord(sysMedicalRecord));

}

@Log(title = "就诊记录", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody SysMedicalRecord sysMedicalRecord)

{

return toAjax(sysMedicalRecordService.updateSysMedicalRecord(sysMedicalRecord));

}

@Log(title = "就诊记录", businessType = BusinessType.DELETE)

@DeleteMapping("/{recordIds}")

public AjaxResult remove(@PathVariable Long[] recordIds)

{

return toAjax(sysMedicalRecordService.deleteSysMedicalRecordByIds(recordIds));

}

}

package com.ruoyi.system.controller;

import java.util.List;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.ruoyi.common.annotation.Log;

import com.ruoyi.common.core.controller.BaseController;

import com.ruoyi.common.core.domain.AjaxResult;

import com.ruoyi.common.enums.BusinessType;

import com.ruoyi.system.domain.SysTreatmentMethod;

import com.ruoyi.system.service.ISysTreatmentMethodService;

import com.ruoyi.common.utils.poi.ExcelUtil;

import com.ruoyi.common.core.page.TableDataInfo;

@RestController

@RequestMapping("/system/method")

public class SysTreatmentMethodController extends BaseController

{

@Autowired

private ISysTreatmentMethodService sysTreatmentMethodService;

@GetMapping("/list")

public TableDataInfo list(SysTreatmentMethod sysTreatmentMethod)

{

startPage();

List<SysTreatmentMethod> list = sysTreatmentMethodService.selectSysTreatmentMethodList(sysTreatmentMethod);

return getDataTable(list);

}

@Log(title = "就诊记录", businessType = BusinessType.EXPORT)

@GetMapping("/export")

public AjaxResult export(SysTreatmentMethod sysTreatmentMethod)

{

List<SysTreatmentMethod> list = sysTreatmentMethodService.selectSysTreatmentMethodList(sysTreatmentMethod);

ExcelUtil<SysTreatmentMethod> util = new ExcelUtil<SysTreatmentMethod>(SysTreatmentMethod.class);

return util.exportExcel(list, "就诊记录数据");

}

@PreAuthorize("@ss.hasPermi('system:method:query')")

@GetMapping(value = "/{methodId}")

public AjaxResult getInfo(@PathVariable("methodId") Long methodId)

{

return AjaxResult.success(sysTreatmentMethodService.selectSysTreatmentMethodById(methodId));

}

@Log(title = "就诊记录", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody SysTreatmentMethod sysTreatmentMethod)

{

return toAjax(sysTreatmentMethodService.insertSysTreatmentMethod(sysTreatmentMethod));

}

@Log(title = "就诊记录", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody SysTreatmentMethod sysTreatmentMethod)

{

return toAjax(sysTreatmentMethodService.updateSysTreatmentMethod(sysTreatmentMethod));

}

@Log(title = "就诊记录", businessType = BusinessType.DELETE)

@DeleteMapping("/{methodIds}")

public AjaxResult remove(@PathVariable Long[] methodIds)

{

return toAjax(sysTreatmentMethodService.deleteSysTreatmentMethodByIds(methodIds));

}

}

package com.ruoyi.system.controller;

import java.util.List;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.ruoyi.common.annotation.Log;

import com.ruoyi.common.core.controller.BaseController;

import com.ruoyi.common.core.domain.AjaxResult;

import com.ruoyi.common.enums.BusinessType;

import com.ruoyi.system.domain.SysUserInfo;

import com.ruoyi.system.service.ISysUserInfoService;

import com.ruoyi.common.utils.poi.ExcelUtil;

import com.ruoyi.common.core.page.TableDataInfo;

@RestController

@RequestMapping("/system/info")

public class SysUserInfoController extends BaseController

{

@Autowired

private ISysUserInfoService sysUserInfoService;

@GetMapping("/list")

public TableDataInfo list(SysUserInfo sysUserInfo)

{

startPage();

List<SysUserInfo> list = sysUserInfoService.selectSysUserInfoList(sysUserInfo);

return getDataTable(list);

}

@Log(title = "个人信息", businessType = BusinessType.EXPORT)

@GetMapping("/export")

public AjaxResult export(SysUserInfo sysUserInfo)

{

List<SysUserInfo> list = sysUserInfoService.selectSysUserInfoList(sysUserInfo);

ExcelUtil<SysUserInfo> util = new ExcelUtil<SysUserInfo>(SysUserInfo.class);

return util.exportExcel(list, "个人信息数据");

}

@GetMapping(value = "/{userId}")

public AjaxResult getInfo(@PathVariable("userId") Long userId)

{

return AjaxResult.success(sysUserInfoService.selectSysUserInfoById(userId));

}

@Log(title = "个人信息", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody SysUserInfo sysUserInfo)

{

return toAjax(sysUserInfoService.insertSysUserInfo(sysUserInfo));

}

@Log(title = "个人信息", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody SysUserInfo sysUserInfo)

{

return toAjax(sysUserInfoService.updateSysUserInfo(sysUserInfo));

}

@Log(title = "个人信息", businessType = BusinessType.DELETE)

@DeleteMapping("/{userIds}")

public AjaxResult remove(@PathVariable Long[] userIds)

{

return toAjax(sysUserInfoService.deleteSysUserInfoByIds(userIds));

}

}

package com.ruoyi.system.controller;

import java.util.List;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.ruoyi.common.annotation.Log;

import com.ruoyi.common.core.controller.BaseController;

import com.ruoyi.common.core.domain.AjaxResult;

import com.ruoyi.common.enums.BusinessType;

import com.ruoyi.system.domain.SysUserRecord;

import com.ruoyi.system.service.ISysUserRecordService;

import com.ruoyi.common.utils.poi.ExcelUtil;

import com.ruoyi.common.core.page.TableDataInfo;

@RestController

@RequestMapping("/system/record")

public class SysUserRecordController extends BaseController

{

@Autowired

private ISysUserRecordService sysUserRecordService;

@GetMapping("/list")

public TableDataInfo list(SysUserRecord sysUserRecord)

{

startPage();

List<SysUserRecord> list = sysUserRecordService.selectSysUserRecordList(sysUserRecord);

return getDataTable(list);

}

@Log(title = "记录信息", businessType = BusinessType.EXPORT)

@GetMapping("/export")

public AjaxResult export(SysUserRecord sysUserRecord)

{

List<SysUserRecord> list = sysUserRecordService.selectSysUserRecordList(sysUserRecord);

ExcelUtil<SysUserRecord> util = new ExcelUtil<SysUserRecord>(SysUserRecord.class);

return util.exportExcel(list, "记录信息数据");

}

@GetMapping(value = "/{recordId}")

public AjaxResult getInfo(@PathVariable("recordId") Long recordId)

{

return AjaxResult.success(sysUserRecordService.selectSysUserRecordById(recordId));

}

@GetMapping(value = "/getInfoById/{userId}")

public AjaxResult getInfoById(@PathVariable("userId") Long userId)

{

SysUserRecord sysUserRecord = new SysUserRecord();

sysUserRecord.setUserId(userId);

List<SysUserRecord> list = sysUserRecordService.selectSysUserRecordList(sysUserRecord);

return AjaxResult.success(list);

}

@Log(title = "记录信息", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody SysUserRecord sysUserRecord)

{

return toAjax(sysUserRecordService.insertSysUserRecord(sysUserRecord));

}

@Log(title = "记录信息", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody SysUserRecord sysUserRecord)

{

return toAjax(sysUserRecordService.updateSysUserRecord(sysUserRecord));

}

@PreAuthorize("@ss.hasPermi('system:record:remove')")

@Log(title = "记录信息", businessType = BusinessType.DELETE)

@DeleteMapping("/{recordIds}")

public AjaxResult remove(@PathVariable Long[] recordIds)

{

return toAjax(sysUserRecordService.deleteSysUserRecordByIds(recordIds));

}

}

package com.ruoyi.system.controller;

import java.util.List;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.ruoyi.common.annotation.Log;

import com.ruoyi.common.core.controller.BaseController;

import com.ruoyi.common.core.domain.AjaxResult;

import com.ruoyi.common.enums.BusinessType;

import com.ruoyi.system.domain.SysUserRemind;

import com.ruoyi.system.service.ISysUserRemindService;

import com.ruoyi.common.utils.poi.ExcelUtil;

import com.ruoyi.common.core.page.TableDataInfo;

@RestController

@RequestMapping("/system/remind")

public class SysUserRemindController extends BaseController

{

@Autowired

private ISysUserRemindService sysUserRemindService;

@GetMapping("/list")

public TableDataInfo list(SysUserRemind sysUserRemind)

{

startPage();

List<SysUserRemind> list = sysUserRemindService.selectSysUserRemindList(sysUserRemind);

return getDataTable(list);

}

@Log(title = "提醒信息", businessType = BusinessType.EXPORT)

@GetMapping("/export")

public AjaxResult export(SysUserRemind sysUserRemind)

{

List<SysUserRemind> list = sysUserRemindService.selectSysUserRemindList(sysUserRemind);

ExcelUtil<SysUserRemind> util = new ExcelUtil<SysUserRemind>(SysUserRemind.class);

return util.exportExcel(list, "提醒信息数据");

}

@GetMapping(value = "/{remindId}")

public AjaxResult getInfo(@PathVariable("remindId") Long remindId)

{

return AjaxResult.success(sysUserRemindService.selectSysUserRemindById(remindId));

}

@GetMapping(value = "/getInfoByUserId/{userId}")

public AjaxResult getInfoByUserId(@PathVariable("userId") Long userId)

{

SysUserRemind sysUserRemind = new SysUserRemind();

sysUserRemind.setUserId(userId);

List<SysUserRemind> list = sysUserRemindService.selectSysUserRemindList(sysUserRemind);

return AjaxResult.success(list);

}

@Log(title = "提醒信息", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody SysUserRemind sysUserRemind)

{

return toAjax(sysUserRemindService.insertSysUserRemind(sysUserRemind));

}

@Log(title = "提醒信息", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody SysUserRemind sysUserRemind)

{

return toAjax(sysUserRemindService.updateSysUserRemind(sysUserRemind));

}

@Log(title = "提醒信息", businessType = BusinessType.DELETE)

@DeleteMapping("/{remindIds}")

public AjaxResult remove(@PathVariable Long[] remindIds)

{

return toAjax(sysUserRemindService.deleteSysUserRemindByIds(remindIds));

}

}

package com.ruoyi.system.domain;

import java.math.BigDecimal;

import java.util.Date;

import com.fasterxml.jackson.annotation.JsonFormat;

import org.apache.commons.lang3.builder.ToStringBuilder;

import org.apache.commons.lang3.builder.ToStringStyle;

import com.ruoyi.common.annotation.Excel;

import com.ruoyi.common.core.domain.BaseEntity;

public class SysMedicalRecord extends BaseEntity

{

private static final long serialVersionUID = 1L;

private Long recordId;

@Excel(name = "用户ID")

private Long userId;

@Excel(name = "MD值")

private BigDecimal md;

@Excel(name = "PSD值")

private BigDecimal psd;

@Excel(name = "杯盘比值")

private BigDecimal bh;

@Excel(name = "角膜厚度")

private BigDecimal um;

@Excel(name = "RNFL值")

private BigDecimal rnfl;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "就诊日期", width = 30, dateFormat = "yyyy-MM-dd")

private Date medicalTime;

@Excel(name = "就诊医院")

private String hospital;

public void setRecordId(Long recordId)

{

this.recordId = recordId;

}

public Long getRecordId()

{

return recordId;

}

public void setUserId(Long userId)

{

this.userId = userId;

}

public Long getUserId()

{

return userId;

}

public void setMd(BigDecimal md)

{

this.md = md;

}

public BigDecimal getMd()

{

return md;

}

public void setPsd(BigDecimal psd)

{

this.psd = psd;

}

public BigDecimal getPsd()

{

return psd;

}

public void setBh(BigDecimal bh)

{

this.bh = bh;

}

public BigDecimal getBh()

{

return bh;

}

public void setUm(BigDecimal um)

{

this.um = um;

}

public BigDecimal getUm()

{

return um;

}

public void setRnfl(BigDecimal rnfl)

{

this.rnfl = rnfl;

}

public BigDecimal getRnfl()

{

return rnfl;

}

public void setMedicalTime(Date medicalTime)

{

this.medicalTime = medicalTime;

}

public Date getMedicalTime()

{

return medicalTime;

}

public void setHospital(String hospital)

{

this.hospital = hospital;

}

public String getHospital()

{

return hospital;

}

@Override

public String toString() {

return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)

.append("recordId", getRecordId())

.append("userId", getUserId())

.append("md", getMd())

.append("psd", getPsd())

.append("bh", getBh())

.append("um", getUm())

.append("rnfl", getRnfl())

.append("medicalTime", getMedicalTime())

.append("hospital", getHospital())

.append("remark", getRemark())

.toString();

}

}

package com.ruoyi.system.domain;

import java.util.Date;

import com.fasterxml.jackson.annotation.JsonFormat;

import org.apache.commons.lang3.builder.ToStringBuilder;

import org.apache.commons.lang3.builder.ToStringStyle;

import com.ruoyi.common.annotation.Excel;

import com.ruoyi.common.core.domain.BaseEntity;

public class SysTreatmentMethod extends BaseEntity

{

private static final long serialVersionUID = 1L;

private Long methodId;

@Excel(name = "用户ID")

private Long userId;

@Excel(name = "药品")

private String drugs;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "药物治疗开始时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date drugsStartTime;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "药物治疗结束时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date drugsEndTime;

@Excel(name = "激光治疗")

private String laser;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "激光治疗时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date laserTime;

@Excel(name = "手术治疗术式")

private String operation;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "手术治疗时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date operationTime;

public void setMethodId(Long methodId)

{

this.methodId = methodId;

}

public Long getMethodId()

{

return methodId;

}

public void setUserId(Long userId)

{

this.userId = userId;

}

public Long getUserId()

{

return userId;

}

public void setDrugs(String drugs)

{

this.drugs = drugs;

}

public String getDrugs()

{

return drugs;

}

public void setDrugsStartTime(Date drugsStartTime)

{

this.drugsStartTime = drugsStartTime;

}

public Date getDrugsStartTime()

{

return drugsStartTime;

}

public void setDrugsEndTime(Date drugsEndTime)

{

this.drugsEndTime = drugsEndTime;

}

public Date getDrugsEndTime()

{

return drugsEndTime;

}

public void setLaser(String laser)

{

this.laser = laser;

}

public String getLaser()

{

return laser;

}

public void setLaserTime(Date laserTime)

{

this.laserTime = laserTime;

}

public Date getLaserTime()

{

return laserTime;

}

public void setOperation(String operation)

{

this.operation = operation;

}

public String getOperation()

{

return operation;

}

public void setOperationTime(Date operationTime)

{

this.operationTime = operationTime;

}

public Date getOperationTime()

{

return operationTime;

}

@Override

public String toString() {

return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)

.append("methodId", getMethodId())

.append("userId", getUserId())

.append("drugs", getDrugs())

.append("drugsStartTime", getDrugsStartTime())

.append("drugsEndTime", getDrugsEndTime())

.append("laser", getLaser())

.append("laserTime", getLaserTime())

.append("operation", getOperation())

.append("operationTime", getOperationTime())

.append("remark", getRemark())

.toString();

}

}

package com.ruoyi.system.domain;

import java.math.BigDecimal;

import java.util.Date;

import com.fasterxml.jackson.annotation.JsonFormat;

import org.apache.commons.lang3.builder.ToStringBuilder;

import org.apache.commons.lang3.builder.ToStringStyle;

import com.ruoyi.common.annotation.Excel;

import com.ruoyi.common.core.domain.BaseEntity;

public class SysUserInfo extends BaseEntity

{

private static final long serialVersionUID = 1L;

private Long userId;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "确诊日期", width = 30, dateFormat = "yyyy-MM-dd")

private Date diagnosis;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "出生日期", width = 30, dateFormat = "yyyy-MM-dd")

private Date birthday;

@Excel(name = "左眼视力")

private BigDecimal visionLeft;

@Excel(name = "右眼视力")

private BigDecimal visionRight;

@Excel(name = "家族史")

private String history;

@Excel(name = "青光眼档案")

private String archives;

@Excel(name = "眼压")

private Integer pressure;

@Excel(name = "确诊照片")

private String photo;

@Excel(name = "感知压力量表分数")

private BigDecimal stress;

@Excel(name = "药物依从性量表分数")

private Integer compliance;

@Excel(name = "昵称")

private String nickName;

@Excel(name = "头像地址")

private String avatar;

@Excel(name = "性别")

private String sex;

public void setUserId(Long userId)

{

this.userId = userId;

}

public Long getUserId()

{

return userId;

}

public void setDiagnosis(Date diagnosis)

{

this.diagnosis = diagnosis;

}

public Date getDiagnosis()

{

return diagnosis;

}

public void setBirthday(Date birthday)

{

this.birthday = birthday;

}

public Date getBirthday()

{

return birthday;

}

public void setVisionLeft(BigDecimal visionLeft)

{

this.visionLeft = visionLeft;

}

public BigDecimal getVisionLeft()

{

return visionLeft;

}

public void setVisionRight(BigDecimal visionRight)

{

this.visionRight = visionRight;

}

public BigDecimal getVisionRight()

{

return visionRight;

}

public void setHistory(String history)

{

this.history = history;

}

public String getHistory()

{

return history;

}

public void setArchives(String archives)

{

this.archives = archives;

}

public String getArchives()

{

return archives;

}

public void setPressure(Integer pressure)

{

this.pressure = pressure;

}

public Integer getPressure()

{

return pressure;

}

public void setPhoto(String photo)

{

this.photo = photo;

}

public String getPhoto()

{

return photo;

}

public void setStress(BigDecimal stress)

{

this.stress = stress;

}

public BigDecimal getStress()

{

return stress;

}

public void setCompliance(Integer compliance)

{

this.compliance = compliance;

}

public Integer getCompliance()

{

return compliance;

}

public void setNickName(String nickName)

{

this.nickName = nickName;

}

public String getNickName()

{

return nickName;

}

public void setAvatar(String avatar)

{

this.avatar = avatar;

}

public String getAvatar()

{

return avatar;

}

public void setSex(String sex)

{

this.sex = sex;

}

public String getSex()

{

return sex;

}

@Override

public String toString() {

return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)

.append("userId", getUserId())

.append("diagnosis", getDiagnosis())

.append("birthday", getBirthday())

.append("visionLeft", getVisionLeft())

.append("visionRight", getVisionRight())

.append("history", getHistory())

.append("archives", getArchives())

.append("pressure", getPressure())

.append("photo", getPhoto())

.append("stress", getStress())

.append("compliance", getCompliance())

.append("nickName", getNickName())

.append("avatar", getAvatar())

.append("sex", getSex())

.toString();

}

}

package com.ruoyi.system.domain;

import java.math.BigDecimal;

import java.util.Date;

import com.fasterxml.jackson.annotation.JsonFormat;

import org.apache.commons.lang3.builder.ToStringBuilder;

import org.apache.commons.lang3.builder.ToStringStyle;

import com.ruoyi.common.annotation.Excel;

import com.ruoyi.common.core.domain.BaseEntity;

public class SysUserRecord extends BaseEntity

{

private static final long serialVersionUID = 1L;

private Long recordId;

@Excel(name = "用户ID")

private Long userId;

@Excel(name = "左眼视力")

private BigDecimal visionLeft;

@Excel(name = "右眼视力")

private BigDecimal visionRight;

@Excel(name = "左眼压力")

private BigDecimal pressureLeft;

@Excel(name = "右眼压力")

private BigDecimal pressureRight;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "记录日期", width = 30, dateFormat = "yyyy-MM-dd")

private Date recordTime;

@Excel(name = "运动记录")

private String medication;

@Excel(name = "运动情况")

private String emotion;

@Excel(name = "情绪状况")

private String motion;

@Excel(name = "情绪状况")

private String visiontype;

@JsonFormat(pattern = "yyyy-MM-dd hh:mm")

@Excel(name = "眼压测量时刻", width = 30, dateFormat = "yyyy-MM-dd hh:mm")

private Date pressureTime;

@Excel(name = "运动时长")

private Long motionDuration;

@Excel(name = "运动备注")

private String motionRemark;

@Excel(name = "用药时刻")

private String medicationMoment;

@Excel(name = "不良反应")

private String medicationAdr;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "停药时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date medicationSuspend;

@Excel(name = "用药备注")

private String medicationRemark;

public void setRecordId(Long recordId)

{

this.recordId = recordId;

}

public Long getRecordId()

{

return recordId;

}

public void setUserId(Long userId)

{

this.userId = userId;

}

public Long getUserId()

{

return userId;

}

public void setVisionLeft(BigDecimal visionLeft)

{

this.visionLeft = visionLeft;

}

public BigDecimal getVisionLeft()

{

return visionLeft;

}

public void setVisionRight(BigDecimal visionRight)

{

this.visionRight = visionRight;

}

public BigDecimal getVisionRight()

{

return visionRight;

}

public void setPressureLeft(BigDecimal pressureLeft)

{

this.pressureLeft = pressureLeft;

}

public BigDecimal getPressureLeft()

{

return pressureLeft;

}

public void setPressureRight(BigDecimal pressureRight)

{

this.pressureRight = pressureRight;

}

public BigDecimal getPressureRight()

{

return pressureRight;

}

public void setRecordTime(Date recordTime)

{

this.recordTime = recordTime;

}

public Date getRecordTime()

{

return recordTime;

}

public void setMedication(String medication)

{

this.medication = medication;

}

public String getMedication()

{

return medication;

}

public void setEmotion(String emotion)

{

this.emotion = emotion;

}

public String getEmotion()

{

return emotion;

}

public void setMotion(String motion)

{

this.motion = motion;

}

public String getMotion()

{

return motion;

}

public void setVisiontype(String visiontype)

{

this.visiontype = visiontype;

}

public String getVisiontype()

{

return visiontype;

}

public void setPressureTime(Date pressureTime)

{

this.pressureTime = pressureTime;

}

public Date getPressureTime()

{

return pressureTime;

}

public void setMotionDuration(Long motionDuration)

{

this.motionDuration = motionDuration;

}

public Long getMotionDuration()

{

return motionDuration;

}

public void setMotionRemark(String motionRemark)

{

this.motionRemark = motionRemark;

}

public String getMotionRemark()

{

return motionRemark;

}

public void setMedicationMoment(String medicationMoment)

{

this.medicationMoment = medicationMoment;

}

public String getMedicationMoment()

{

return medicationMoment;

}

public void setMedicationAdr(String medicationAdr)

{

this.medicationAdr = medicationAdr;

}

public String getMedicationAdr()

{

return medicationAdr;

}

public void setMedicationSuspend(Date medicationSuspend)

{

this.medicationSuspend = medicationSuspend;

}

public Date getMedicationSuspend()

{

return medicationSuspend;

}

public void setMedicationRemark(String medicationRemark)

{

this.medicationRemark = medicationRemark;

}

public String getMedicationRemark()

{

return medicationRemark;

}

@Override

public String toString() {

return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)

.append("recordId", getRecordId())

.append("userId", getUserId())

.append("visionLeft", getVisionLeft())

.append("visionRight", getVisionRight())

.append("pressureLeft", getPressureLeft())

.append("pressureRight", getPressureRight())

.append("recordTime", getRecordTime())

.append("medication", getMedication())

.append("emotion", getEmotion())

.append("motion", getMotion())

.append("visiontype", getVisiontype())

.append("pressureTime", getPressureTime())

.append("motionDuration", getMotionDuration())

.append("motionRemark", getMotionRemark())

.append("medicationMoment", getMedicationMoment())

.append("medicationAdr", getMedicationAdr())

.append("medicationSuspend", getMedicationSuspend())

.append("medicationRemark", getMedicationRemark())

.toString();

}

}

package com.ruoyi.system.domain;

import java.util.Date;

import com.fasterxml.jackson.annotation.JsonFormat;

import org.apache.commons.lang3.builder.ToStringBuilder;

import org.apache.commons.lang3.builder.ToStringStyle;

import com.ruoyi.common.annotation.Excel;

import com.ruoyi.common.core.domain.BaseEntity;

public class SysUserRemind extends BaseEntity

{

private static final long serialVersionUID = 1L;

private Long remindId;

@Excel(name = "用户ID")

private Long userId;

@Excel(name = "提醒类型(0-就诊提醒，1-用药提醒)")

private Long remindType;

@Excel(name = "提醒事件")

private String remind;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "提醒时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date remindTime;

@Excel(name = "药品")

private String drugs;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "药物治疗开始时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date drugsStartTime;

@JsonFormat(pattern = "yyyy-MM-dd")

@Excel(name = "药物治疗结束时间", width = 30, dateFormat = "yyyy-MM-dd")

private Date drugsEndTime;

@Excel(name = "药物使用方式")

private String drugsMode;

public void setRemindId(Long remindId)

{

this.remindId = remindId;

}

public Long getRemindId()

{

return remindId;

}

public void setUserId(Long userId)

{

this.userId = userId;

}

public Long getUserId()

{

return userId;

}

public void setRemindType(Long remindType)

{

this.remindType = remindType;

}

public Long getRemindType()

{

return remindType;

}

public void setRemind(String remind)

{

this.remind = remind;

}

public String getRemind()

{

return remind;

}

public void setRemindTime(Date remindTime)

{

this.remindTime = remindTime;

}

public Date getRemindTime()

{

return remindTime;

}

public void setDrugs(String drugs)

{

this.drugs = drugs;

}

public String getDrugs()

{

return drugs;

}

public void setDrugsStartTime(Date drugsStartTime)

{

this.drugsStartTime = drugsStartTime;

}

public Date getDrugsStartTime()

{

return drugsStartTime;

}

public void setDrugsEndTime(Date drugsEndTime)

{

this.drugsEndTime = drugsEndTime;

}

public Date getDrugsEndTime()

{

return drugsEndTime;

}

public void setDrugsMode(String drugsMode)

{

this.drugsMode = drugsMode;

}

public String getDrugsMode()

{

return drugsMode;

}

@Override

public String toString() {

return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)

.append("remindId", getRemindId())

.append("userId", getUserId())

.append("remindType", getRemindType())

.append("remind", getRemind())

.append("remindTime", getRemindTime())

.append("drugs", getDrugs())

.append("drugsStartTime", getDrugsStartTime())

.append("drugsEndTime", getDrugsEndTime())

.append("drugsMode", getDrugsMode())

.toString();

}

}

package com.ruoyi.system.mapper;

import java.util.List;

import com.ruoyi.system.domain.SysMedicalRecord;

public interface SysMedicalRecordMapper

{

public SysMedicalRecord selectSysMedicalRecordById(Long recordId);

public List<SysMedicalRecord> selectSysMedicalRecordList(SysMedicalRecord sysMedicalRecord);

public int insertSysMedicalRecord(SysMedicalRecord sysMedicalRecord);

public int updateSysMedicalRecord(SysMedicalRecord sysMedicalRecord);

public int deleteSysMedicalRecordById(Long recordId);

public int deleteSysMedicalRecordByIds(Long[] recordIds);

}

package com.ruoyi.system.mapper;

import java.util.List;

import com.ruoyi.system.domain.SysUserRecord;

public interface SysUserRecordMapper

{

public SysUserRecord selectSysUserRecordById(Long recordId);

public List<SysUserRecord> selectSysUserRecordList(SysUserRecord sysUserRecord);

public int insertSysUserRecord(SysUserRecord sysUserRecord);

public int updateSysUserRecord(SysUserRecord sysUserRecord);

public int deleteSysUserRecordById(Long recordId);

public int deleteSysUserRecordByIds(Long[] recordIds);

}

package com.ruoyi.system.mapper;

import java.util.List;

import com.ruoyi.system.domain.SysUserRemind;

public interface SysUserRemindMapper

{

public SysUserRemind selectSysUserRemindById(Long remindId);

public List<SysUserRemind> selectSysUserRemindList(SysUserRemind sysUserRemind);

public int insertSysUserRemind(SysUserRemind sysUserRemind);

public int updateSysUserRemind(SysUserRemind sysUserRemind);

public int deleteSysUserRemindById(Long remindId);

public int deleteSysUserRemindByIds(Long[] remindIds);

}

package com.ruoyi.system.service;

import java.util.List;

import com.ruoyi.system.domain.SysUserRecord;

public interface ISysUserRecordService

{

public SysUserRecord selectSysUserRecordById(Long recordId);

public List<SysUserRecord> selectSysUserRecordList(SysUserRecord sysUserRecord);

public int insertSysUserRecord(SysUserRecord sysUserRecord);

public int updateSysUserRecord(SysUserRecord sysUserRecord);

public int deleteSysUserRecordByIds(Long[] recordIds);

public int deleteSysUserRecordById(Long recordId);

}

package com.ruoyi.system.service;

import java.util.List;

import com.ruoyi.system.domain.SysUserRemind;

public interface ISysUserRemindService

{

public SysUserRemind selectSysUserRemindById(Long remindId);

public List<SysUserRemind> selectSysUserRemindList(SysUserRemind sysUserRemind);

public int insertSysUserRemind(SysUserRemind sysUserRemind);

public int updateSysUserRemind(SysUserRemind sysUserRemind);

public int deleteSysUserRemindByIds(Long[] remindIds);

public int deleteSysUserRemindById(Long remindId);

}

@PostMapping("/loginApp")

public AjaxResult loginApp(@RequestParam Map<String,Object> map)

{

System.out.println(map.get("username").toString());

AjaxResult ajax = AjaxResult.success();

// 生成令牌

String token = loginService.loginApp(map.get("username").toString(), map.get("password").toString());

ajax.put(Constants.TOKEN, token);

return ajax;

}

@PostMapping("/register")

public AjaxResult register(@Validated @RequestBody SysUser user)

{

if (UserConstants.NOT\_UNIQUE.equals(userService.checkUserNameUnique(user.getUserName())))

{

return AjaxResult.error("注册账号'" + user.getUserName() + "'失败，登录账号已存在");

}

user.setPassword(SecurityUtils.encryptPassword(user.getPassword()));

return AjaxResult.success(userService.insertUser(user));

}

public String loginApp(String username, String password) {

Authentication authentication = null;

try

{

// 该方法会去调用UserDetailsServiceImpl.loadUserByUsername

authentication = authenticationManager

.authenticate(new UsernamePasswordAuthenticationToken(username, password));

}

catch (Exception e)

{

if (e instanceof BadCredentialsException)

{

AsyncManager.me().execute(AsyncFactory.recordLogininfor(username, Constants.LOGIN\_FAIL, MessageUtils.message("user.password.not.match")));

throw new UserPasswordNotMatchException();

}

else

{

AsyncManager.me().execute(AsyncFactory.recordLogininfor(username, Constants.LOGIN\_FAIL, e.getMessage()));

throw new CustomException(e.getMessage());

}

}

AsyncManager.me().execute(AsyncFactory.recordLogininfor(username, Constants.LOGIN\_SUCCESS, MessageUtils.message("user.login.success")));

LoginUser loginUser = (LoginUser) authentication.getPrincipal();

recordLoginInfo(loginUser.getUser());

return tokenService.createToken(loginUser);

}

package com.ruoyi.system.service.impl;

import java.math.BigDecimal;

import java.util.List;

import com.ruoyi.system.domain.SysUserInfo;

import com.ruoyi.system.mapper.SysUserInfoMapper;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.ruoyi.system.mapper.SysUserRecordMapper;

import com.ruoyi.system.domain.SysUserRecord;

import com.ruoyi.system.service.ISysUserRecordService;

@Service

public class SysUserRecordServiceImpl implements ISysUserRecordService

{

@Autowired

private SysUserRecordMapper sysUserRecordMapper;

@Autowired

private SysUserInfoMapper sysUserInfoMapper;

@Override

public SysUserRecord selectSysUserRecordById(Long recordId)

{

return sysUserRecordMapper.selectSysUserRecordById(recordId);

}

@Override

public List<SysUserRecord> selectSysUserRecordList(SysUserRecord sysUserRecord)

{

return sysUserRecordMapper.selectSysUserRecordList(sysUserRecord);

}

public int insertSysUserRecord(SysUserRecord sysUserRecord)

{

if(sysUserRecord.getUserId()!=null&&sysUserRecord.getVisionLeft()!=null&&sysUserRecord.getVisionRight()!=null){

SysUserInfo sysUserInfo = new SysUserInfo();

sysUserInfo.setUserId(sysUserRecord.getUserId());

sysUserInfo.setVisionLeft(sysUserRecord.getVisionLeft());

sysUserInfo.setVisionRight(sysUserRecord.getVisionRight());

sysUserInfoMapper.updateSysUserInfo(sysUserInfo);

}

return sysUserRecordMapper.insertSysUserRecord(sysUserRecord);

}

@Override

public int updateSysUserRecord(SysUserRecord sysUserRecord)

{

return sysUserRecordMapper.updateSysUserRecord(sysUserRecord);

}

@Override

public int deleteSysUserRecordByIds(Long[] recordIds)

{

return sysUserRecordMapper.deleteSysUserRecordByIds(recordIds);

}

@Override

public int deleteSysUserRecordById(Long recordId)

{

return sysUserRecordMapper.deleteSysUserRecordById(recordId);

}

}

<template>

<div class="app-container">

<el-form :model="queryParams" ref="queryForm" :inline="true" v-show="showSearch" label-width="68px">

<el-form-item label="用户ID" prop="userId">

<el-input

v-model="queryParams.userId"

placeholder="请输入用户ID"

clearable

size="small"

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="左眼视力" prop="visionLeft">

<el-input

v-model="queryParams.visionLeft"

placeholder="请输入左眼视力"

clearable

size="small"

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="右眼视力" prop="visionRight">

<el-input

v-model="queryParams.visionRight"

placeholder="请输入右眼视力"

clearable

size="small"

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="左眼压力" prop="pressureLeft">

<el-input

v-model="queryParams.pressureLeft"

placeholder="请输入左眼压力"

clearable

size="small"

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="右眼压力" prop="pressureRight">

<el-input

v-model="queryParams.pressureRight"

placeholder="请输入右眼压力"

clearable

size="small"

@keyup.enter.native="handleQuery">

</el-form-item>

<el-form-item label="记录日期" prop="recordTime">

<el-date-picker clearable size="small"

v-model="form.recordTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="选择记录日期">

</el-date-picker>

</el-form-item>

<el-form-item label="运动记录" prop="medication">

<el-input v-model="form.medication" placeholder="请输入运动记录" />

</el-form-item>

<el-form-item label="运动情况" prop="emotion">

<el-input v-model="form.emotion" placeholder="请输入运动情况" />

</el-form-item>

<el-form-item label="情绪状况" prop="motion">

<el-input v-model="form.motion" placeholder="请输入情绪状况" />

</el-form-item>

<el-form-item label="情绪状况" prop="visiontype">

<el-select v-model="form.visiontype" placeholder="请选择情绪状况">

<el-option label="请选择字典生成" value="" />

</el-select>

</el-form-item>

<el-form-item label="眼压测量的具体时刻" prop="pressureTime">

<el-date-picker clearable size="small"

v-model="form.pressureTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="选择眼压测量的具体时刻">

</el-date-picker>

</el-form-item>

<el-form-item label="运动时长" prop="motionDuration">

<el-input v-model="form.motionDuration" placeholder="请输入运动时长" />

</el-form-item>

<el-form-item label="运动备注" prop="motionRemark">

<el-input v-model="form.motionRemark" placeholder="请输入运动备注" />

</el-form-item>

<el-form-item label="用药时刻" prop="medicationMoment">

<el-input v-model="form.medicationMoment" placeholder="请输入用药时刻" />

</el-form-item>

<el-form-item label="不良反应" prop="medicationAdr">

<el-input v-model="form.medicationAdr" placeholder="请输入不良反应" />

</el-form-item>

<el-form-item label="停药时间" prop="medicationSuspend">

<el-date-picker clearable size="small"

v-model="form.medicationSuspend"

type="date"

value-format="yyyy-MM-dd"

placeholder="选择停药时间">

</el-date-picker>

</el-form-item>

<el-form-item label="用药备注" prop="medicationRemark">

<el-input v-model="form.medicationRemark" placeholder="请输入用药备注" />

</el-form-item>

</el-form>

<div slot="footer" class="dialog-footer">

<el-button type="primary" @click="submitForm">确 定</el-button>

<el-button @click="cancel">取 消</el-button>

</div>

</el-dialog>

</div>

</template>

<script>

import { listRecord, getRecord, delRecord, addRecord, updateRecord, exportRecord } from "@/api/system/record";

export default {

name: "Record",

components: {

},

data() {

return {

loading: true,

exportLoading: false,

ids: [],

single: true,

multiple: true,

showSearch: true,

total: 0,

recordList: [],

title: "",

open: false,

queryParams: {

pageNum: 1,

pageSize: 10,

userId: null,

visionLeft: null,

visionRight: null,

pressureLeft: null,

pressureRight: null,

recordTime: null,

medication: null,

emotion: null,

motion: null,

visiontype: null,

pressureTime: null,

motionDuration: null,

motionRemark: null,

medicationMoment: null,

medicationAdr: null,

medicationSuspend: null,

medicationRemark: null

},

form: {},

rules: {

}

};

},

created() {

this.getList();

},

methods: {

getList() {

this.loading = true;

listRecord(this.queryParams).then(response => {

this.recordList = response.rows;

this.total = response.total;

this.loading = false;

});

},

cancel() {

this.open = false;

this.reset();

},

reset() {

this.form = {

recordId: null,

userId: null,

visionLeft: null,

visionRight: null,

pressureLeft: null,

pressureRight: null,

recordTime: null,

medication: null,

emotion: null,

motion: null,

visiontype: null,

pressureTime: null,

motionDuration: null,

motionRemark: null,

medicationMoment: null,

medicationAdr: null,

medicationSuspend: null,

medicationRemark: null

};

this.resetForm("form");

},

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

handleSelectionChange(selection) {

this.ids = selection.map(item => item.recordId)

this.single = selection.length!==1

this.multiple = !selection.length

},

handleAdd() {

this.reset();

this.open = true;

this.title = "添加记录信息";

},

handleUpdate(row) {

this.reset();

const recordId = row.recordId || this.ids

getRecord(recordId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改记录信息";

});

},

submitForm() {

this.$refs["form"].validate(valid => {

if (valid) {

if (this.form.recordId != null) {

updateRecord(this.form).then(response => {

this.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

addRecord(this.form).then(response => {

this.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

handleDelete(row) {

const recordIds = row.recordId || this.ids;

this.$confirm('是否确认删除记录信息编号为"' + recordIds + '"的数据项?', "警告", {

confirmButtonText: "确定",

cancelButtonText: "取消",

type: "warning"

}).then(function() {

return delRecord(recordIds);

}).then(() => {

this.getList();

this.msgSuccess("删除成功");

})

},

handleExport() {

const queryParams = this.queryParams;

this.$confirm('是否确认导出所有记录信息数据项?', "警告", {

confirmButtonText: "确定",

cancelButtonText: "取消",

type: "warning"

}).then(() => {

this.exportLoading = true;

return exportRecord(queryParams);

}).then(response => {

this.download(response.msg);

this.exportLoading = false;

})

}

}

};

</script>

import request from '@/utils/request'

export function listRecord(query) {

return request({

url: '/system/record/list',

method: 'get',

params: query

})

}

export function getRecord(recordId) {

return request({

url: '/system/record/' + recordId,

method: 'get'

})

}

export function addRecord(data) {

return request({

url: '/system/record',

method: 'post',

data: data

})

}

export function updateRecord(data) {

return request({

url: '/system/record',

method: 'put',

data: data

})

}

export function delRecord(recordId) {

return request({

url: '/system/record/' + recordId,

method: 'delete'

})

}

export function exportRecord(query) {

return request({

url: '/system/record/export',

method: 'get',

params: query

})

}

import request from '@/utils/request'

export function listRecord(query) {

return request({

url: '/system/record/list',

method: 'get',

params: query

})

}

export function getRecord(recordId) {

return request({

url: '/system/record/' + recordId,

method: 'get'

})

}

export function addRecord(data) {

return request({

url: '/system/record',

method: 'post',

data: data

})

}

export function updateRecord(data) {

return request({

url: '/system/record',

method: 'put',

data: data

})

}

export function delRecord(recordId) {

return request({

url: '/system/record/' + recordId,

method: 'delete'

})

}

export function exportRecord(query) {

return request({

url: '/system/record/export',

method: 'get',

params: query

})

}