.daily+.daily {

margin-top: 1.5em;

}

.daily .info {

color: black;

overflow-wrap: break-word;

}

.daily .text {

font-size: 14px;

}

.daily .heading {

font-size: 12px;

color: #888;

}

.daily .item {

padding: 10px 0;

}

.daily .clearfix:before,

.daily .clearfix:after {

display: table;

content: "";

}

.daily .clearfix:after {

clear: both

}

.daily .box-card {

width: 100%;

}

.daily .btn-add {

float: right;

width: 36px;

text-align: center;

height: 36px;

padding: 0;

border-radius: 50%;

}

.daily .btn-edit,.daily .btn-delete {

width: 18px;

text-align: center;

height: 18px;

padding: 0;

border-radius: 50%;

}

.daily .text-right {

text-align: right;

}

.daily .el-form-item\_\_content{

width: 100%;

}

.daily-mask{

position:fixed;

width:100%;

height:100%;

top:0;

left:0;

z-index:10000;

background-color: hsla(0,0%,100%,.9);

}

.daily-mask-text{

position: absolute;

top: 50%;

left: 50%;

margin-left: -30px;

margin-top: -10px;

}

.dailycc-group{

display: inline-block;

font-size: 12px;

margin: 3px;

border-radius: 5px;

border: #c3c3c3 solid;

border-width: 1px;

}

.dailycc-group span{

padding: 0 3px;

}

.dailycc-group .dailycc-read{

background-color: #c3c3c3 ;

color: white;

}

.dailycc-group.hasreaded{

border-color: #37C355;

}

.dailycc-read.hasreaded{

background-color: #37C355;

}

/\* jshint esversion: 6 \*/

import Vue from 'vue'

import VueRouter from 'vue-router'

import VueResource from 'vue-resource'

// import ElementUI from 'element-ui'

// import 'element-ui/lib/theme-default/index.css'

import {

Button, Rate, Icon, Checkbox, Tag, Card, Radio, Form, Message, Dialog, CheckboxGroup, FormItem, Input, RadioGroup

} from 'element-ui'

import './css/daily.css'

import App from './App.vue'

Vue.use(VueRouter)

Vue.use(VueResource)

// Vue.use(ElementUI)

Vue.use(Button)

Vue.use(Rate)

Vue.use(Icon)

Vue.use(Checkbox)

Vue.use(Tag)

Vue.use(Card)

Vue.use(Radio)

Vue.use(Form)

Vue.use(Dialog)

Vue.use(CheckboxGroup)

Vue.use(FormItem)

Vue.use(Input)

Vue.use(RadioGroup)

Vue.prototype.$message = Message

import routes from './routes.js'

const router = new VueRouter({

mode: 'history',

base: 'todolist',

// base: \_\_dirname,

routes // （缩写）相当于 routes: routes

})

new Vue({

router ,

render: h => h(App)

}).$mount('#app');

<template>

<div class="board">

<div class="board-content">

<div class="board-head">

<h1>今年任务</h1>

<div class="task-add el-icon-plus" @touchend.stop.prevent="openCreateTask" v-if="operate"></div>

</div>

<div class="no-task" v-if="!tasks.length"><h2>还没有任务!<span v-if="operate">新建一个吧↗</span></h2></div>

<ul>

<li v-for="task in tasks">

<div class="task" @touchend="showToolbar(task)">

<div class="content">

<h2 class="task-name">{{task.taskName}}</h2>

<div class="task-info">

<p class="create-date">创建日期：<span>{{task.createDate}}</span></p>

<p class="schedule-date">计划日期：<span>{{task.scheduleDate}}</span></p>

<div class="rate" v-if="task.status == 1" @touchend="rate">

<el-rate v-model="task.rate" :colors="['#99A9BF', '#F7BA2A', '#FF9900']">

</el-rate>

</div>

</div>

</div>

<img src="../../assets/finish.png" v-if="task.status==1" class="badge finish-badge">

<img src="../../assets/cancel.png" v-if="task.status==3" class="badge cancel-badge">

<div class="finish-date">{{task.finishDate ? task.finishDate.replace(/[0-9]{4}-/, '') : null}}</div>

</div>

<transition name="toggle">

<div class="toolbar" v-if="operate && task.toolbar && task.status==0" @touchend="showToolbar(task)">

<div class="toolbar-button toolbar-edit" @touchend="openUpdateTask(task)">修改</div>

<div class="toolbar-button toolbar-delete" @touchend="deleteTask(task)">删除</div>

<div class="toolbar-button toolbar-close" @touchend="cancelTask(task)">取消</div>

<div class="toolbar-button toolbar-check" @touchend="finishTask(task)">完成</div>

</div>

</transition>

</li>

</ul>

</div>

<task-popup :task="task" :tasks="tasks" :taskPopupShow="taskPopupShow" @createTask="createTask" @updateTask="updateTask" @closeTaskDialog="closeTaskDialog" v-if="taskPopupShow"></task-popup>

</div>

</template>

<script>

// import taskPopup from './task-popup.vue'

export default {

name: 'board',

components: {

taskPopup: (resolve) => require(['./task-popup.vue'], resolve)

},

props: ['operate', 'other'],

mounted () {

if(!this.other)

this.retrieveData(\_user.emplId);

else

this.retrieveData(this.other.user);

},

data() {

return {

taskPopupShow: false,

task: {

id: null,

taskName: null,

createDate: null,

scheduleDate: null,

finishDate: null,

rate: null,

toolbar: false,

user: \_user.emplId,

status: 0

},

tasks: []

}

},

watch: {

other: {

deep: true,

handler: function(val, oldVal){

console.log(val);

this.retrieveData(val.user.emplId || val.user);

}

}

},

methods: {

rate (task, e) {

},

showToolbar (task) {

task.toolbar = !task.toolbar;

},

openCreateTask () {

this.taskPopupShow = true;

},

openUpdateTask (task) {

this.task = task;

this.taskPopupShow = true;

},

createTask (task) {

this.tasks.push(task);

},

finishTask (task) {

this.endTask(1, task);

},

retrieveTask (task) {

console.log('retrieve');

},

updateTask (task) {

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

if(this.tasks[i].id == task.id)

for( let t in this.tasks[i] ){

if( typeof this.tasks[i][t] != 'function' )

// console.log( this.tasks[i][r].toString() );

this.tasks[i][t] = task[t];

}

}

},

deleteTask (task) {

this.postData(0, {id: task.id}, (data)=>{

if(!data.error){

let tasks = [];

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

if(this.tasks[i].id!=data.id){

tasks.push( this.tasks[i] );

}

}

this.tasks = tasks;

}

});

},

cancelTask (task) {

this.endTask(3, task);

},

rate () {

this.$message({ showClose: true, message: '评级功能尚未开放', type: 'warning' });

},

closeTaskDialog () {

this.taskPopupShow = false;

this.task = {

id: null,

taskName: null,

createDate: null,

scheduleDate: null,

finishDate: null,

rate: null,

toolbar: false,

user: \_user.emplId,

status: 0

};

},

endTask (status, task) {

let date = new Date(),

temp = {};

for( let t in task ){

if( typeof task[t] != 'function' && t!='toolbar' )

// console.log( task[r].toString() );

temp[t] = task[t];

}

temp.finishDate = date.getFullYear() + '-' + (date.getMonth()+1) + '-' + date.getDate();

temp.status = status;

this.postData(1, {task: temp}, (data)=>{

if(data.error==0){

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

if(this.tasks[i].id == data.task.id){

this.tasks[i].finishDate = date.getFullYear() + '-' + (date.getMonth()+1) + '-' + date.getDate();;

this.tasks[i].status = status;

}

}

}

});

},

retrieveData (user) {

if(!user){

// this.$message.error({showClose: true, message: '用户参数出错!'});

return false;

}

this.$http.get(HOST+'/php/task/task-retrieve.php?user='+user).then( (respones)=>{

this.tasks = respones.data.tasks.map((task)=>{

task.toolbar = false;

return task

});

}, (respones)=>{

this.tasks = [];

this.$message.error({showClose: true, message: '任务模块通信失败!'});

} );

},

postData (method, data, callback) {

let that = this,

url = method ? HOST+'/php/task/task-update.php' : HOST+'/php/task/task-delete.php';

this.$http.post(url, data, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

callback(response.data);

}, (response)=>{

this.$message.error({showClose: true, message: '任务模块通信失败!'});

});

}

}

}

</script>

<style>

.board{

width: 100%; height: 100%;

}

.board-content{

background-color: white;

height: auto; width: 100%;

margin: 0 auto;

border: 1px solid #d1dbe5;

border-radius: 5px;

box-shadow: 0 2px 4px 0 rgba(0,0,0,.12), 0 0 6px 0 rgba(0,0,0,.04);

position: relative;

}

.board .board-head{

margin: 0 auto; padding: .5em 0;

width: 100%;

text-align: center;

border-bottom: 1px solid #d1dbe5;

/\*border: 1px solid black;

border-bottom: none;

border-radius: 5px 5px 0 0;\*/

/\*background: rgba(0, 188, 212, .3);\*/

}

.board .board-head h1{

font-size: 1.5em;

font-weight: normal;

text-align: center;

display: inline-block;

}

.board .board-head .task-add{

position: absolute;

background: rgba(0, 124, 255, 0.7);

top: .5em; right: 1em;

width: 2em; height: 2em;

line-height: 2em;

border-radius: 50%;

color: white;

}

.no-task h2 {

font-size: 100%;

font-weight: normal;

margin: 0 auto; padding: 0;

text-align: center;

}

.board ul{

margin: 0 auto; padding: 0;

list-style: none;

height: auto;

width: 100%;

/\*border: 1px solid black;

border-radius: 0 0 3px 5px;\*/

}

.board ul li{

border-bottom: 1px solid #d1dbe5;

padding: .5em .5em;

}

.board ul li:last-child{

border-bottom: none;

}

.board ul li .task{

display: flex;

justify-content: space-between;

align-items: center;

}

.board ul li .task h2.task-name{

max-width: 12em;

font-size: 1.25em;

font-weight: normal;

}

.board ul li .task .task-info{

font-size: .7em;

color: gray;

}

.board ul li .task .task-info p{

display: inline-block;

}

.board ul li .task .finish-date{

text-align: center;

font-size: 1em;

width: 3em; height: 3em;

line-height: 3em;

color: rgba(120, 120, 120, 1);

}

.badge{

position: absolute;

right: .5em;

width: 3em; height: 3em;

}

.toolbar{

display: flex;

justify-content: space-between;

margin-top: .5em;

}

.toolbar .toolbar-button{

font-size: .75em;

color: white;

text-align: center;

display: inline-block;

margin: 0;

width: 20%; height: 2em;

line-height: 2em;

border-radius: 5px;

}

.toolbar .toolbar-check{

background: #47d847;

}

.toolbar-edit{

background: #5cabff;

}

.toolbar-delete{

background: #ff5555;

}

.toolbar .toolbar-close{

background: #f7db4e;

}

.toolbar i{

color: white;

}

/\* Vue动画 \*/

.toggle-enter-active,

.toggle-leave-active{

transition: all .2s;

}

.toggle-enter,

.toggle-leave-active{

opacity: 0;

}

</style>

<template>

<div class="daily">

<el-card class="box-card">

<div slot="header" class="clearfix">

<span style="line-height: 36px;">{{createDate}} 工作日志</span>

<el-button type="primary" class="btn-add el-icon-plus" @click="openDial()"></el-button>

</div>

<el-checkbox-group v-model="attendance">

<el-checkbox label="1">本地</el-checkbox>

<el-checkbox label="2">加班</el-checkbox>

<el-checkbox label="3">外勤</el-checkbox>

<el-checkbox label="4">出差</el-checkbox>

</el-checkbox-group>

<p v-if="!dailys.length" class="info">请增加工作内容</p>

<transition-group name="list" tag="div">

<div v-for="(o,index) in dailys" :key="index" class="text item toggle">

<div class="heading">任务{{ index+1 }} <i v-if="o.status" class="el-icon-check"></i><i v-else class="el-icon-more"></i></div>

<p class="info">{{o.content}}</p>

<div class="text-right toggleContent">

<i class="el-icon-edit" @click="dailyEdit(o,index)"></i>

<i class="el-icon-delete" style="margin-left:10px;" @click="dailyDelete(index)"></i>

</div>

</div>

</transition-group>

<div v-if="dailys.length" class="cc-picker-content">

<label>抄送</label>

<div class="cc-picker-list">

<div class="cc-picker-item"><el-tag type="primary">陈朝晖</el-tag></div>

<div class="cc-picker-item" v-for="(item,index) in ccUsers" @click="deleteCCPicker(index)" :key="index">

<el-tag type="primary">{{item.name}}</el-tag>

</div>

<div @click="openCCPicker" class="cc-picker-item"><el-tag type="warning" >添加抄送人</el-tag></div>

</div>

</div>

<div v-if="dailys.length" class="text-right">

<p id="submit-tips">点击“提交”按钮后生效</p>

<el-button @click="dailySubmit('checkForm')">提交</el-button>

</div>

</el-card>

<el-dialog title="新增" :visible.sync="dialogVisible" size="large">

<el-form :model="daily" :rules="rules" ref="dailyForm">

<el-form-item label="工作内容" prop="content">

<el-input type="textarea" autosize placeholder="请输入内容" v-model="daily.content">

</el-input>

</el-form-item>

<el-form-item label="完成情况" prop="status">

<el-radio-group v-model="daily.status">

<el-radio label="1">已完成</el-radio>

<el-radio label="0">进行中</el-radio>

</el-radio-group>

</el-form-item>

</el-form>

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

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

<el-button type="primary" @click="addToCard('dailyForm')">确 定</el-button>

</span>

</el-dialog>

</div>

</template>

<script>

export default {

name: 'daily',

mounted: function() {

// `this` 指向 vm 实例

this.currentUser = \_user.emplId;

this.$emit('loadingChange',true);

let currentDate = new Date();

this.createDate = currentDate.getFullYear()+'-'+(currentDate.getMonth()+1)+'-'+currentDate.getDate();

var url = HOST+'/php/daily/daily-retrieve.php'

var param = {

user: this.currentUser,

createDate:this.createDate

};

console.log(param);

this.$http.post(url, param, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

if(response.data.error == 0 && response.data.attendance.attendance){

this.$emit('dailySubmit',{response:response.data,attId:this.attId});

this.attId = response.data.attendance.id;

}

this.$emit('loadingChange',false);

}, (response)=>{

this.attId = -1;

this.$emit('loadingChange',false);

this.$message.error({showClose: true, message: '日志模块通信失败!'});

});

},

data() {

return {

editIndex:-1,

createDate: '',

attendance: [],

check1: {},

dialogVisible: false,

daily: {

content: '',

status: '1'

},

dailys: [],

rules: {

content: [{

required: true,

message: '请填写工作内容',

trigger: 'blur'

}],

status: [{

required: true,

message: '请选择完成情况',

trigger: 'change'

}]

},

ccUsers:[],

ccUserIds:[],

attId:-2,//初始状态，-1初始化失败，>=0则为id

currentUser:"",

};

},

methods: {

openDial: function() {

this.daily.content='';//每次打开置空

this.dialogVisible = true;

this.editIndex=-1;//每次打开判断为增加纪录而不是编辑

},

addToCard: function(formName) {

this.$refs[formName].validate((valid) => {

if (valid) {

let i = {

content: this.daily.content,

status: parseInt(this.daily.status)

};

if(this.editIndex != -1){this.dailys.splice(this.editIndex, 1, i);}

else{this.dailys.push(i);}

this.dialogVisible = false;

} else {

console.log('error submit!!');

return false;

}

});

},

dailyEdit: function(daily,index) {

this.dialogVisible = true;

this.editIndex=index;

this.daily.content = daily.content;

this.daily.status = daily.status + '';

},

dailyDelete: function(index) {

this.dailys.splice(index, 1);

console.log(this.dailys);

},

dailySubmit: function(checkForm) {

if (this.attendance.length) {

if(this.attId == -1){

this.$message.error('初始化失败，请退出重新进入');

}else{

this.ccUserIds.push({user:'03401659233316'}); // 加上固定的抄送人 陈总

// this.ccUserIds.push({user:'03424264076698'}); // 加上固定的抄送人 小红

let att=this.attendance.toString();

var url;

var param;

if(this.attId==-2){

url = HOST+'/php/daily/daily-add.php'

param = {

attendance: {attendance:att,user:this.currentUser,createDate:this.createDate},

dailys: this.dailys,

dailyCc: this.ccUserIds

};

}else{

url = HOST+'/php/daily/daily-update.php'

param = {

attendance: {id:this.attId,attendance:att},

dailys: this.dailys,

dailyCc: this.ccUserIds

};

}

this.$emit('loadingChange',true);

this.$http.post(url, param, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

if(response.data.error == 0){

this.$emit('loadingChange',false);

this.$message.success('添加日志成功！');

this.$emit('dailySubmit',{response:response.data,attId:this.attId});

this.attId = response.data.attendance.id;

}

}, (response)=>{

this.$emit('loadingChange',false);

this.$message.error({showClose: true, message: '日志模块通信失败!'});

});

this.dailys.splice(0,this.dailys.length);

this.ccUsers.splice(0,this.ccUsers.length);

this.ccUserIds.splice(0,this.ccUserIds.length);

}

} else {

this.$message.error('请选择考勤');

return false;

}

},

openCCPicker: function(){

let that = this;

//引入钉钉后可用

dd.biz.contact.choose({

startWithDepartmentId: 0, //-1表示打开的通讯录从自己所在部门开始展示, 0表示从企业最上层开始，(其他数字表示从该部门开始:暂时不支持)

multiple: true, //是否多选： true多选 false单选； 默认true

users: null, //默认选中的用户列表，userid；成功回调中应包含该信息

disabledUsers: ['03401659233316'], // 不能选中的用户列表，员工userid (默认选中陈总)

corpId: window.\_config.corpId, //企业id

// max: , //人数限制，当multiple为true才生效，可选范围1-1500

limitTips: "挑太多啦！", //超过人数限制的提示语可以用这个字段自定义

isNeedSearch: true, // 是否需要搜索功能

title: "挑个人呗~", // 如果你需要修改选人页面的title，可以在这里赋值

local: "false", // 是否显示本地联系人，默认false

onSuccess: function(data) {

// todo

data.forEach( (user, index, data)=>{

that.ccUsers.push(user);

that.ccUserIds.push({user: user.emplId});

} );

console.log(that.ccUsers, that.ccUserIds);

},

onFail: function(err) {

// todo

that.$message.error('非常抱歉！您的通信录打不开。。。');

}

});

},

deleteCCPicker: function(index){

this.ccUsers.splice(index,1);

this.ccUserIds.splice(index,1);

}

}

}

</script>

<style>

.toggleContent{display: none;}

.toggle:hover .toggleContent{display: block;}

.daily .cc-picker-item{

display: inline-block;

margin: 5px 5px 5px 0;

}

.list-item {

display: inline-block;

margin-right: 10px;

}

.list-enter-active, .list-leave-active {

transition: all 1s;

}

.list-enter, .list-leave-active {

opacity: 0;

transform: translateY(30px);

}

#submit-tips {

font-size: .5em;

color: gray;

}

</style>

<template>

<div :class="{'task-popup': !taskPopupShow}">

<div class="task-popup-main">

<header class="task-popup-header">

<h1>{{ title }}</h1>

</header>

<div class="task-popup-content">

<div class="input-field text-field">

<label class="input-field-label">任务名称:</label>

<input type="text" placeholder="请输入任务名称" v-model="taskName">

</div>

<div class="input-field date-field">

<div class="open-date-picker" @touchend="openDatePicker"></div>

<label class="input-field-label">预计完成:</label>

<input type="text" placeholder="点击选择日期" v-model="scheduleDate" @focus="openDatePicker">

</div>

</div>

<footer class="task-popup-footer">

<div class="button button-cancel" @touchend="cancel">取消</div>

<div class="button button-confirm" @touchend="confirm">确定</div>

</footer>

</div>

<div class="task-popup-mask mask" @touchend="cancel"></div>

</div>

</template>

<script>

export default {

props: ['task', 'tasks', 'taskPopupShow'],

data () {

return {

scheduleDate: this.task.scheduleDate,

taskName: this.task.taskName

}

},

watch: {

scheduleDate: {

deep: true,

handler: function(val, oldVal){

console.log(val);

this.thisTask.scheduleDate = val;

}

},

taskName: {

deep: true,

handler: function(val, oldVal){

console.log(val);

this.thisTask.taskName = val;

}

}

},

computed: {

title () {

return this.task.id ? '修改' : '新建';

},

thisTask (){

let task = {},

date = new Date();

for( let p in this.task ){

if(typeof this.task[p] !== 'function')

task[p] = this.task[p];

}

if( !this.task.id )

task.createDate = date.getFullYear() + '-' + (date.getMonth()+1) + '-' + date.getDate();

return task;

}

},

methods: {

openDatePicker () {

let that = this;

// 引入钉钉后可用

dd.biz.util.datepicker({

format: 'yyyy-MM-dd',

value: '', //默认显示

onSuccess : function(result) {

that.scheduleDate = result.value;

},

onFail : function() {}

});

},

cancel () {

this.$emit('closeTaskDialog');

},

confirm () {

if( this.thisTask.taskName && this.thisTask.scheduleDate )

this.postData(!this.thisTask.id);

else

this.$message({ message: '任务名称和日期不能为空！', type: 'warning', showClose: true });

},

postData (method) {

let that = this,

task = {},

url = method ? HOST+'/php/task/task-add.php' : HOST+'/php/task/task-update.php';

for( let t in this.thisTask ){

if( typeof this.thisTask[t] != 'function' && t!='toolbar' )

// console.log( this.thisTask[r].toString() );

task[t] = this.thisTask[t];

}

// console.log(task);

this.$http.post(url, {task: task}, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

if(response.data.error==0){

task = response.data.task;

task.toolbar = false;

that.$emit(method?'createTask':'updateTask', task);

}

that.$emit('closeTaskDialog');

}, (response)=>{

this.$message.error('通信失败!');

});

}

}

}

</script>

<style>

.task-popup-main{

position: fixed;

z-index: 9999;

width: calc( 100% - 1em );

/\*height: calc( 100% - 1em );\*/

left: .5em; top: 20%;

background: rgba(255, 255, 255, 1);

/\*padding: .5em 1em;\*/

border-radius: 3px;

box-shadow: 1px 0 5px hsla(0,0%,59%,.9);

}

.task-popup{

display: none;

}

.task-popup-main > \* {

margin: 0 1em;

}

header.task-popup-header h1{

margin: .5em 0;

font-size: 1.25em;

}

.task-popup-content{

}

.input-field{

margin: .5em auto;

}

.input-field-label{

display: inline-block;

width: 5em;

}

.input-field input{

border: 1px solid gray;

border-radius: 3px;

padding: 1px;

background: rgba(255, 255, 255, 0);

outline: none;

width: calc( 100% - 6em );

}

.date-field .open-date-picker{

position: absolute;

width: calc( 100% - 2em ); height: 1.5em;

background: rgba(100, 100, 100, 0);

}

footer.task-popup-footer{

display: flex;

justify-content: flex-end;

}

.button{

/\*color: white;\*/

font-size: 1em;

display: inline-block;

width: 5em; height: 2em;

line-height: 2em;

border-radius: 5px;

text-align: center;

box-sizing:border-box;

margin: .5em .5em;

}

.button-cancel{

border: 1px solid black;

}

.button-confirm{

background: #5cabff;

}

.task-popup-mask {

display: block;

position: fixed;

top: 0; left: 0;

width: 100%; height: 100%;

background: rgba(100, 100, 100, .7);

z-index: 998;

}

</style>

<template>

<div class="watch-other-popup">

<div class="watch-other-popup-main">

<header class="watch-other-popup-header">

<h1>想看谁？</h1>

</header>

<div class="watch-other-popup-content">

<div class="input-field colleague-field">

<div class="open-colleague-picker picker" @touchend="openColleaguePicker"></div>

<label class="input-field-label">选择同事</label>

<input type="text" placeholder="点击选择同事" v-model="user.name">

</div>

<div class="input-field date-field">

<div class="open-date-picker picker" @touchend="openDatePicker"></div>

<label class="input-field-label">选择日期</label>

<input type="text" placeholder="点击选择日期" v-model="date">

</div>

</div>

<footer class="watch-other-popup-footer">

<div class="button button-cancel" @touchend="cancel">取消</div>

<div class="button button-confirm" @touchend="confirm">确定</div>

</footer>

</div>

<div class="watch-other-popup-mask mask" @touchend="cancel"></div>

</div>

</template>

<script>

export default {

data () {

return {

user: {name: null, userid: null},

date: null,

}

},

methods: {

openColleaguePicker (){

let that = this;

// 引入钉钉后可用

dd.biz.contact.choose({

startWithDepartmentId: -1, //-1表示打开的通讯录从自己所在部门开始展示, 0表示从企业最上层开始，(其他数字表示从该部门开始:暂时不支持)

multiple: false, //是否多选： true多选 false单选； 默认true

users: null, //默认选中的用户列表，userid；成功回调中应包含该信息

disabledUsers: null, // 不能选中的用户列表，员工userid

corpId: window.\_config.corpId, //企业id

// max: , //人数限制，当multiple为true才生效，可选范围1-1500

limitTips: "挑太多啦！", //超过人数限制的提示语可以用这个字段自定义

isNeedSearch: true, // 是否需要搜索功能

title: "挑个人呗~", // 如果你需要修改选人页面的title，可以在这里赋值

local: "false", // 是否显示本地联系人，默认false

onSuccess: function(data) {

// todo

that.user = data[0];

},

onFail: function(err) {

// todo

}

});

},

openDatePicker () {

let that = this;

// 引入钉钉后可用

dd.biz.util.datepicker({

format: 'yyyy-MM-dd',

value: '', //默认显示

onSuccess : function(result) {

that.date = result.value;

},

onFail : function() {

}

});

},

cancel () {

this.$emit('watchOther', null);

},

confirm () {

let other = {user: this.user, date: this.date};

this.$emit('watchOther', other);

}

}

}

</script>

<style>

.watch-other-popup {

display: block;

color: black;

text-align: left;;

}

.watch-other-popup-main{

position: fixed;

justify-content: flex-start;

z-index: 9999;

width: calc( 100% - 1em );

/\*height: calc( 100% - 1em );\*/

left: .5em; top: 20%;

background: rgba(255, 255, 255, 1);

/\*padding: .5em 1em;\*/

border-radius: 3px;

box-shadow: 1px 0 5px hsla(0,0%,59%,.9);

}

.watch-other-popup{

/\*display: none;\*/

}

.watch-other-popup-main > \* {

margin: 0 1em;

}

header.watch-other-popup-header h1{

margin: .5em 0;

font-size: 1.25em;

}

.watch-other-popup-content{

}

.input-field{

margin: .5em auto;

}

.input-field-label{

display: inline-block;

width: 5em;

}

.input-field input{

border: 1px solid gray;

border-radius: 3px;

padding: 1px;

background: rgba(255, 255, 255, 0);

outline: none;

width: calc( 100% - 6em );

}

.picker{

position: absolute;

width: calc( 100% - 2em ); height: 1.5em;

background: rgba(100, 100, 100, 0);

}

footer.watch-other-popup-footer{

display: flex;

justify-content: flex-end;

}

.button{

/\*color: white;\*/

font-size: 1em;

display: inline-block;

width: 5em; height: 2em;

line-height: 2em;

border-radius: 5px;

text-align: center;

box-sizing:border-box;

margin: .5em .5em;

}

.button-cancel{

border: 1px solid black;

}

.button-confirm{

background: #5cabff;

}

.watch-other-popup-mask {

display: block;

position: fixed;

top: 0; left: 0;

width: 100%; height: 100%;

background: rgba(100, 100, 100, .7);

z-index: 998;

}

</style>

<template>

<div class="note-daily">

<board :operate="true"></board>

<hr>

<div v-show="loading" class="daily-mask">

<div class="daily-mask-text">加载中...</div>

</div>

<div style="text-align:center;color: #adadad;font-size: .8em; height: 2em; line-height: 2em;">

<span @click="loadMore">加载历史</span></div>

<history v-for="(item, index) in historys" :key="item.attendance.id" :history="item"></history>

<daily v-on:dailySubmit="submit" v-on:loadingChange="loadingChange"></daily>

</div>

</template>

<script>

import daily from '../modules/daily.vue'

import history from '../modules/history.vue'

import board from '../modules/board.vue'

export default {

data() {

return {

historys: [],

isUpdate: true,

currentUser: '',

loading: false

}

},

components: { // 注册组件，这很重要

// daily: (resolve) => require(['../modules/daily.vue'], resolve),

daily,

history,

board

},

mounted:function () {

console.log('mounted', new Date());

this.currentUser = \_user.emplId;

},

methods:{

submit:function(param){

//字符串转字符串数组

var response=param.response;

this.isUpdate=false;

if(param.attId==-2){

response.attendance.attendance = response.attendance.attendance.split(",");

this.historys.push(response);

}

else{

for(var i=0; i<this.historys.length; i++){

if(response.attendance.id == this.historys[i].attendance.id){

this.historys[i].dailys.splice(0,this.historys[i].dailys.length);

for(var j=0; j<response.dailys.length; j++)

{

this.historys[i].dailys.push(response.dailys[j]);

}

this.historys[i].dailyCc.splice(0,this.historys[i].dailyCc.length);

for(var k=0; k<response.dailyCc.length; k++)

{

this.historys[i].dailyCc.push(response.dailyCc[k]);

}

}

}

}

},

loadMore:function(){

this.loading=true;

var param;

if(this.historys.length){

param={user:this.currentUser,offset:this.historys.length};

}

else{

param={user:this.currentUser,offset:0};

}

var url = HOST+'/php/daily/daily-loadhistory.php'

this.$http.post(url, param, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

if(response.data.error == 0){

var results = response.data.results;

if(!results.length){

this.$message({message: '已经没有纪录了哦',type: 'warning'});

}

for(var i=0; i<results.length; i++)

{

results[i].attendance.attendance=results[i].attendance.attendance.split(",");

this.historys.unshift(results[i]);

}

}

this.loading=false;

}, (response)=>{

this.loading=false;

this.$message.error({showClose: true, message: '日志模块通信失败!'});

});

},

loadingChange:function(loading) {

this.loading=loading

}

}

}

</script>

<style>

.note-daily {

padding: 1.5em;

background-color: #EBEBEB;

}

hr{

margin: 1em auto;

border: none;

display: block;

width: 100%;

height: 1px;

background: rgba(120, 120, 120, 0.4);

}

</style>

<template>

<div class="read">

<header class="read-component-header">

<span @touchend.prevent="watchSelf">看自己</span>

<span @touchend.prevent="popupWatchOther">看别人</span>

<watch-other-popup v-if="watchOtherPopupVisible" @watchOther="watchOther"></watch-other-popup>

</header>

<!-- 看板 -->

<board id="board" :operate="false" :other="other"></board>

<hr>

<div v-show="loading" class="daily-mask">

<div class="daily-mask-text">加载中...</div>

</div>

<!-- 日志 -->

<history v-for="(item,index) in historys" :key="index" :history="item"></history>

<div v-if="!historys.length" class="read-component-info">

当天纪录为空。。。

</div>

<div v-if="isloadMore" class="read-component-info">

<span @click="loadMore">点击加载历史</span>

</div>

</div>

</template>

<script>

import board from '../modules/board.vue'

import history from '../modules/history.vue'

import watchOtherPopup from '../modules/watch-other-popup.vue'

export default {

data () {

return {

historys: [],

other: null,

watchOtherPopupVisible: false,

currentUser: '',

loading:true,

selectDate: new Date().getFullYear() + '-' + (new Date().getMonth()+1) + '-' + new Date().getDate(),

isloadMore:true

}

},

components: {

history,

board,

watchOtherPopup

},

mounted () {

this.currentUser = \_user.emplId;

let otherUser = localStorage.getItem('otherUser'),

otherDate = localStorage.getItem('otherDate');

if(otherDate&&otherUser){

this.other = {user: otherUser, date: otherDate};

localStorage.removeItem('otherUser'); localStorage.removeItem('otherDate');

this.currentUser = this.other.user;

this.selectDate = this.other.date;

}

else{

this.other = null;

}

this.selectARecord();

localStorage.removeItem('otherUser'); localStorage.removeItem('otherDate');

},

methods:{

loadMore:function(){

this.historysLoad();

},

watchSelf () {

this.historys.splice(0,this.historys.length);//清空历史纪录

this.currentUser = \_user.emplId;

this.historysLoad();

this.isloadMore = true;

this.other = {user: \_user.emplId};

},

popupWatchOther () {

console.log(\_user.department);

if( \_user.department.indexOf(40500828) != -1 ){ // 按照权限打开

this.watchOtherPopupVisible = !this.watchOtherPopupVisible;

}

else{

this.$message.error({showClose: true, message: '您不是管理干部，不能随便看别人'});

}

},

watchOther (other) {

this.watchOtherPopupVisible = false;

this.other = other;

this.currentUser=other ? other.user.emplId : \_user.userid;

this.selectDate=other ? other.date : new Date().getFullYear() + '-' + (new Date().getMonth()+1) + '-' + new Date().getDate();

this.isloadMore=false;

this.historys.splice(0,this.historys.length);//清空历史纪录

this.selectARecord();

},

historysLoad: function(){

this.loading=true;

var param;

if(this.historys.length){

param={user:this.currentUser,offset:this.historys.length};

}

else{

param={user:this.currentUser,offset:0};

}

var url = HOST+'/php/daily/daily-loadhistory.php'

this.$http.post(url, param, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

if(response.data.error == 0){

var results = response.data.results;

if(!results.length){

this.$message({message: '已经没有纪录了哦',type: 'warning'});

}

for(var i=0; i<results.length; i++)

{

results[i].attendance.attendance=results[i].attendance.attendance.split(",");

this.historys.push(results[i]);

}

}

this.loading=false;

}, (response)=>{

this.loading=false;

this.$message.error({showClose: true, message: '日志模块通信失败!'});

});

},

selectARecord: function(){

var url = HOST+'/php/daily/daily-retrieve.php';

var param = {

user: this.currentUser,

createDate:this.selectDate

};

let delay = setInterval( ()=>{

if(\_user.emplId){

clearInterval(delay);

if(this.currentUser != \_user.emplId){ param.currentUser=\_user.emplId };

// param.currentUser='mj';

console.log(param);

this.loading=true;

this.$http.post(url, param, {

emulateJSON: true,

headers: {

'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

}

}).then((response)=>{

if(response.data.error == 0){

if(!response.data.attendance.attendance){

this.$message({message: '已经没有纪录了哦',type: 'warning'});

}else{

response.data.attendance.attendance = response.data.attendance.attendance.split(",");

this.historys.push(response.data);

}

}

this.loading=false;

}, (response)=>{

this.loading=false;

this.$message.error({showClose: true, message: '日志查询失败!'});

});

}

}, 1000);

// if(this.currentUser != \_user.emplId){ param.currentUser=\_user.emplId };

// // param.currentUser='mj';

// console.log(param);

// this.loading=true;

// this.$http.post(url, param, {

// emulateJSON: true,

// headers: {

// 'Content-Type': 'enctype="application/x-www-form-urlencoded; charset=utf-8"'

// }

// }).then((response)=>{

// if(response.data.error == 0){

// if(!response.data.attendance.attendance){

// this.$message({message: '已经没有纪录了哦',type: 'warning'});

// }else{

// response.data.attendance.attendance = response.data.attendance.attendance.split(",");

// this.historys.push(response.data);

// }

// }

// this.loading=false;

// }, (response)=>{

// this.loading=false;

// this.$message.error({showClose: true, message: '日志查询失败!'});

// });

},

}

}

</script>

<style>

.read {

padding: 2.5em 1em 1em 1em;

background-color: #EBEBEB;

}

header.read-component-header {

z-index: 999;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: auto;

display: flex;

justify-content: space-around;

box-sizing: border-box;

border-bottom: 1px solid #d1dbe5;

background: rgba(255, 255, 255, .9);

color: rgba(80, 175, 250, 1);

text-align: center;

}

header.read-component-header span {

display: inline-block;

font-size: 1em;

line-height: 2em;

width: 100%;

border-right: 1px solid #d1dbe5;

}

header.read-component-header span:last-child {

border: none;

}

header.read-component-header span:active {

background: rgba(220, 220, 220, .9);

}

#board{

/\*margin-bottom: 1.5em;\*/

}

hr{

margin: 1em auto;

border: none;

display: block;

width: 100%;

height: 1px;

background: rgba(120, 120, 120, 0.4);

}

.read-component-info{

text-align:center;

color: #adadad;

font-size: .8em;

height: 2em;

line-height: 2em;

margin: 1em auto

}

</style>

<?php

/\*\*

\* require libs

\*/

require\_once( \_\_DIR\_\_.'/../../php/config/server-config.php');

require\_once( \_\_DIR\_\_.'/../../php/lib/yang-lib/yang-class-mysql.php');

require\_once( \_\_DIR\_\_.'/../../php/api/Msg.php');

/\*\*

\* recieve POST data

\*/

$attendance = $\_POST['attendance'];

$dailys = $\_POST['dailys'];

$dailyCc = $\_POST['dailyCc'];

// date filter

/\*\*

\* instance a new yangMysql class

\*/

$atdQuery = new yangMysql(); $dailyQuery = new yangMysql(); $dailyCcQuery = new yangMysql(); $userQuery = new yangMysql();

$dailyQuery->selectDb(DB\_DATABASE); //

$atdQuery->selectTable("attendance"); $dailyQuery->selectTable("daily"); $dailyCcQuery->selectTable("dailyCc"); $userQuery->selectTable("user");

/\*\*

\* insert new attendance

\*/

$result = $attendance ? $atdQuery->insert($attendance) : false; // attendance can not be null

if(!$result){

$error = '1';

$errorMsg = '新增失败';

}

else{

$result =$atdQuery->query('select @@IDENTITY');

$attendance['id'] = $result[0]['@@IDENTITY'];

for($i=0; $i<count($dailys); $i++){

$dailys[$i]['attendance'] = $attendance['id'];

$dailyQuery->insert($dailys[$i]);

}

for($i=0; $i<count($dailyCc); $i++){

$dailyCc[$i]['attendance'] = $attendance['id'];

$dailyCcQuery->insert($dailyCc[$i]);

}

$id=$attendance['id'];

$dailys = $dailyQuery->simpleSelect(null, "`attendance` = '$id'", null, null);

$dailyCc = $dailyCcQuery->query("select user.name,dailyCc.\* from dailyCc LEFT JOIN user on user.emplId=dailyCc.user where attendance = '$id'");

$error = '0';

$errorMsg = '';

}

$response = [

"attendance"=> $attendance,

"dailys"=> $dailys,

"dailyCc"=> $dailyCc,

"error" => $error,

"errorMsg" => $errorMsg

];

echo json\_encode( $response );

/\*\*

\* [发钉钉企业消息]

\*/

if($error == '0'){

$user = $userQuery->simpleSelect(null, "`emplId` = '".$attendance['user']."'", null, null)[0];

// echo json\_encode($dailyCc);

/\*\*

\* send Msg

\*/

$msg = new Msg(null);

$respond = $msg->sendMsg([

"title" => $user['name']."的工作看板",

"touser" => array\_map(function($user){return $user['user'];}, $dailyCc),

"message\_url" => SERVER\_HOST."/msg-redirect.html?user=".$user['emplId']."&date=".$attendance['createDate']."&signature=".randomIdFactory(10),

"image"=> "", // 图片

"rich" => ["num" => '', "unit" => $attendance['createDate']],

"content" => "摘要："."\n".mb\_substr($dailys[0]['content'], 0, 20, 'utf-8')."\n……\n",

"bgcolor" => "ff40B782"

]);

}

<?php

/\*\*

\* require libs

\*/

require\_once( \_\_DIR\_\_.'/../../php/config/server-config.php');

require\_once( \_\_DIR\_\_.'/../../php/lib/yang-lib/yang-class-mysql.php');

/\*\*

\* recieve GET data

\*/

$user = $\_POST['user'];

$offset = $\_POST['offset'];

//

/\*\*

\* instance a new yangMysql class

\*/

$atdQuery = new yangMysql();$dailyQuery = new yangMysql();$dailyCcQuery = new yangMysql();

$atdQuery->selectTable("attendance");

$dailyQuery->selectTable("daily");

$dailyCcQuery->selectTable("dailyCc");

$condition = "user = '$user'";

$attendances = $atdQuery->simpleSelect(null, $condition, ['`createDate`', 'DESC'], [$offset,5]);

$results = array();

$dailys = [];

if($attendances === false){

$error = '1';

$errorMsg = '查询失败';

}else{

$error = '0';

if(count($attendances)){

for($i=0; $i<count($attendances); $i++){

$attendanceId = $attendances[$i]['id'];

$condition2 = "attendance = '$attendanceId'";

$dailys = $dailyQuery->simpleSelect(null, $condition2, null, null);

$dailyCc = $dailyCcQuery->query("select user.name,dailyCc.\* from dailyCc LEFT JOIN user on user.emplId=dailyCc.user where attendance = '$attendanceId'");

array\_push($results,["attendance"=>$attendances[$i], "dailys"=>$dailys,"dailyCc"=>$dailyCc]);

}

$errorMsg = '';

}

else{

$errorMsg = '查询纪录为空';

}

}

$response = [

"results"=> $results,

"error" => $error,

"errorMsg" => $errorMsg

];

echo json\_encode($response);

<?php

/\*\*

\* require libs

\*/

require\_once( \_\_DIR\_\_.'/../../php/config/server-config.php');

require\_once( \_\_DIR\_\_.'/../../php/lib/yang-lib/yang-class-mysql.php');

/\*\*

\* recieve GET data

\*/

$recieve = $\_POST;

$user = $\_POST['user'];

$createDate = $\_POST['createDate'];

@$currentUser = $recieve['currentUser']?$\_POST['currentUser']:false;

/\*\*

\* instance a new yangMysql class

\*/

$atdQuery = new yangMysql();$dailyQuery = new yangMysql();$dailyCcQuery = new yangMysql();

$atdQuery->selectTable("attendance");

$dailyQuery->selectTable("daily");

$dailyCcQuery->selectTable("dailyCc");

$condition = "user = '$user' AND createDate = '$createDate'";

$attendances = $atdQuery->simpleSelect(null, $condition, null, null);

$dailys = [];

if($attendances === false){

$error = '1';

$errorMsg = '查询失败';

}else{

$error = '0';

if(count($attendances)){

$attendanceId = $attendances[0]['id'];

$condition2 = "attendance = '$attendanceId'";

$dailys = $dailyQuery->simpleSelect(null, $condition2, null, null);

// $dailyCc = $dailyCcQuery->simpleSelect(null, $condition2, null, null);

if($currentUser){

$updateCc = $dailyCcQuery ->query("update dailyCc set `read`=1 where attendance='$attendanceId' AND user='$currentUser'");

}

$dailyCc = $dailyCcQuery->query("select user.name,dailyCc.\* from dailyCc LEFT JOIN user on user.emplId=dailyCc.user where attendance = '$attendanceId'");

$errorMsg = '';

}

else{

$errorMsg = '查询纪录为空';

}

}

if(!(count($attendances) && count($dailys))){

}else{

$error = '0';

$errorMsg = '';

}

$response = [

"attendance"=> count($attendances)?$attendances[0]:'no attendance',

"dailys"=> count($dailys)?$dailys:'no dailys',

"dailyCc"=>count($dailyCc)?$dailyCc:'no dailyCc',

"error" => $error,

"errorMsg" => $errorMsg

];

echo json\_encode($response);

<?php

require\_once( \_\_DIR\_\_.'/../../php/config/server-config.php');

require\_once( \_\_DIR\_\_.'/../../php/lib/yang-lib/yang-class-mysql.php');

require\_once( \_\_DIR\_\_.'/../../php/api/Msg.php');

/\*\*

\* recieve POST data

\*/

$attendance = $\_POST['attendance'];

$dailys = $\_POST['dailys'];

$dailyCc = $\_POST['dailyCc'];

// date filter

/\*\*

\* instance a new yangMysql class

\*/

$atdQuery = new yangMysql(); $dailyQuery = new yangMysql(); $dailyCcQuery = new yangMysql(); $userQuery = new yangMysql();

$dailyQuery->selectDb(DB\_DATABASE); //

$atdQuery->selectTable("attendance"); $dailyQuery->selectTable("daily"); $dailyCcQuery->selectTable("dailyCc"); $userQuery->selectTable("user");

/\*\*

\* insert new attendance

\*/

$id=$attendance['id'];

$att=$attendance['attendance'];

$result = $atdQuery->query("update attendance set `attendance` = '$att' where id = '$id'");

$condition = "`id` = '$id'";

$attendances = $atdQuery->simpleSelect(null, $condition, null, null);

if(!count($attendances) && !$result){

$error = '1';

$errorMsg = '新增失败';

}

else{

for($i=0; $i<count($dailys); $i++){

$dailys[$i]['attendance'] = $attendance['id'];

$dailyQuery->insert($dailys[$i]);

}

for($i=0; $i<count($dailyCc); $i++){

$dailyCc[$i]['attendance'] = $attendance['id'];

$ccUser=$dailyCc[$i]['user'];

$test = $dailyCcQuery->simpleSelect(null, "`attendance` = '$id' and user = '$ccUser'", null, null);

if(count($test) == 0){$dailyCcQuery->insert($dailyCc[$i]);}

}

$error = '0';

$errorMsg = '';

}

$dailys = $dailyQuery->simpleSelect(null, "`attendance` = '$id'", null, null);

$dailyCc = $dailyCcQuery->query("select user.name,dailyCc.\* from dailyCc LEFT JOIN user on user.emplId=dailyCc.user where attendance = '$id'");

$response = [

"attendance"=> count($attendances)?$attendances[0]:'',

"dailys"=> count($dailys)?$dailys:'',

"dailyCc"=> count($dailyCc)?$dailyCc:'',

"error" => $error,

"errorMsg" => $errorMsg

];

echo json\_encode( $response );

/\*\*

\* [发钉钉企业消息]

\*/

if($error == '0'){

$user = $userQuery->simpleSelect(null, "`emplId` = '".$attendances[0]['user']."'", null, null)[0];

// echo $dailys[0]['content'].'\n'.mb\_substr($dailys[0]['content'], 0, 20, 'utf-8');

/\*\*

\* send Msg

\*/

$msg = new Msg(null);

$respond = $msg->sendMsg([

"title" => $user['name']."的工作看板",

"touser" => array\_map(function($user){return $user['user'];}, $dailyCc),

"message\_url" => SERVER\_HOST."/msg-redirect.html?user=".$user['emplId']."&date=".$attendances[0]['createDate']."&signature=".randomIdFactory(10),

"image"=> "", // 图片

"rich" => ["num" => '', "unit" => $attendances[0]['createDate']],

"content" => "摘要："."\n".mb\_substr($dailys[0]['content'], 0, 20, 'utf-8')."\n……\n",

"bgcolor" => "ff40B782"

]);

}

<?php

/\*\*

\* 更新一条task记录

\* http方法： POST

\* 2017/06/12 Yangholmes

\*/

/\*\*

\* require libs

\*/

require\_once( \_\_DIR\_\_.'/../../php/config/server-config.php');

require\_once( \_\_DIR\_\_.'/../../php/lib/yang-lib/yang-class-mysql.php');

/\*\*

\* recieve POST data

\*/

$task = $\_POST['task'];

// date filter

/\*\*

\* instance a new yangMysql class

\*/

$taskQuery = new yangMysql(); //

$taskQuery->selectDb(DB\_DATABASE); //

$taskQuery->selectTable("task");

/\*\*

\* update the item

\*/

$condition = "id = '".$task['id']."'";

$result = $taskQuery->update($task, $condition, null, null);

if(!$result){

$error = '2';

$errorMsg = '修改任务失败';

}

else{

$task = $taskQuery->simpleSelect(null, $condition, null, null)[0];

$error = '0';

$errorMsg = '';

}

$response = [

"task" => $task,

"error" => $error,

"errorMsg" => $errorMsg

];

echo json\_encode( $response );

<?php

/\*\*

\* 新增一条task记录

\* http方法： POST

\* 2017/06/09 Yangholmes

\*/

/\*\*

\* require libs

\*/

require\_once( \_\_DIR\_\_.'/../../php/config/server-config.php');

require\_once( \_\_DIR\_\_.'/../../php/lib/yang-lib/yang-class-mysql.php');

/\*\*

\* recieve POST data

\*/

$task = $\_POST['task'];

// date filter

/\*\*

\* instance a new yangMysql class

\*/

$taskQuery = new yangMysql(); //

$taskQuery->selectDb(DB\_DATABASE); //

$taskQuery->selectTable("task");

/\*\*

\* insert new task

\*/

$result = $task ? $taskQuery->insert($task) : false; // task can not be null

if(!$result){

$error = '1';

$errorMsg = '新增任务失败';

}

else{

$result = $taskQuery->query('select @@IDENTITY');

$condition = "id = '".$result[0]['@@IDENTITY']."'";

$task = $taskQuery->simpleSelect(null, $condition, null, null)[0];

$error = '0';

$errorMsg = '';

}

$response = [

"task" => $task,

"error" => $error,

"errorMsg" => $errorMsg

];

echo json\_encode( $response );