/\*\*

\* 获取当前时间 格式：yyyy-MM-dd HH:MM:SS

\*/

function getCurrentTime() {

let date = new Date(); //当前时间

let month = zeroFill(date.getMonth() + 1); //月

let day = zeroFill(date.getDate()); //日

let hour = zeroFill(date.getHours()); //时

let minute = zeroFill(date.getMinutes()); //分

let second = zeroFill(date.getSeconds()); //秒

let curTime =

date.getFullYear() +

"-" +

month +

"-" +

day +

" " +

hour +

":" +

minute +

":" +

second;

return curTime;

}

/\*\*

\* 补零

\*/

function zeroFill(i) {

if (i >= 0 && i <= 9) {

return "0" + i;

} else {

return i;

}

}

module.exports = {

getCurrentTime,

};

<template>

<router-view />

</template>

<script>

export default {

name: "BlankLayout",

};

</script>

<style scoped></style>

<template>

<v-container fluid fill-height>

<notification></notification>

<v-dialog v-model="dialog" width="500">

<v-card class="elevation-12">

<v-toolbar dark color="primary">

<v-toolbar-title>检测人信息</v-toolbar-title>

</v-toolbar>

<v-card-text>

<v-form>

<v-text-field

prepend-icon="person"

name="login"

label="名称"

type="text"

:rules="rules"

v-model="name"

></v-text-field>

</v-form>

</v-card-text>

<v-card-actions>

<v-spacer></v-spacer>

<v-btn color="primary" @click="Login">确定</v-btn>

</v-card-actions>

</v-card>

</v-dialog>

<v-layout>

<div style="width: 100%">

<template v-if="screenWidth > 700">

<v-card>

<v-img

src="/api/file?file=background\_pc.jpg"

lazy-src="/api/file?file=background\_pc.jpg"

width="100%"

>

<v-card-title

style="

display: flex;

flex-direction: row;

align-items: center;

color: white;

"

class="text-h4"

>

连铸坯裂痕监测系统

</v-card-title>

</v-img>

</v-card>

</template>

<template v-else>

<v-img src="api/file?file=background\_mobile.jpg" height="100%" >

<v-card-title

style="

display: flex;

flex-direction: row;

align-items: center;

color: white;

"

class="text-h4"

>

<div>连铸坯裂痕监测系统</div>

</v-card-title>

</v-img>

</template>

</div>

<v-footer

style="

width: 100%;

position: fixed;

bottom: 0;

display: flex;

flex-direction: column;

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

"

>

<div

style="

display: flex;

flex-direction: column;

justify-content: center;

width: 250px;

background: rgba(255, 255, 255, 0.25);

margin-bottom: 5%;

"

>

<v-btn

outlined

color="black"

style="font-size: large"

@click="confirm"

>

开始查看

</v-btn>

</div>

<!-- <div style="color: white">Copyright © 2020 绝对能行小组</div>-->

</v-footer>

</v-layout>

</v-container>

</template>

<script>

import notification from "../components/notification.vue";

import { notify } from "../components/notification";

export default {

name: "LoginLayout",

components: { notification },

data() {

return {

show: false,

username: "",

password: "",

messageshow: true,

yzm: "",

type: "",

screenWidth: "",

screenHeight: "",

dialog: false,

rules: [(value) => !!value || "不许为空"],

name: "",

};

},

computed: {

width: function () {

return this.screenWidth \* 0.82;

},

},

created() {

},

watch: {

$route() {

},

},

mounted() {

this.screenWidth = document.body.clientWidth;

this.screenHeight = document.body.clientHeight;

window.onresize = () =>

(() => {

this.screenWidth = document.body.clientWidth;

this.screenHeight = document.body.clientHeight;

})();

},

methods: {

codeChange() {

const num = Math.ceil(Math.random() \* 10); // 生成一个随机数（防止缓存）

this.img = `${this.img}?${num}`;

},

confirm() {

this.dialog = true;

},

Login() {

if (this.name !== "") {

localStorage.setItem("Access-Token", this.name);

this.$router.push("/home");

notify("success", "欢迎登录，" + this.name);

} else {

notify("error", "请输入名字");

}

},

changeImg() {

return this.screenWidth / this.screenHeight + 0.04;

},

},

};

</script>

<style scoped>

.backgroud {

display: flex;

flex-direction: column;

align-items: center;

}

.loginpanel {

display: flex;

flex-direction: column;

align-items: center;

background: rgba(255, 255, 255, 0.75);

}

</style>

<template>

<div style="width: 100%; height: 100%">

<div style="display: flex; flex-direction: column; align-items: center">

<notification></notification>

</div>

<v-navigation-drawer

v-model="drawer"

app

:mini-variant="mini"

:color="colorRGB"

dark

>

<p></p>

<v-list-item> 连铸坯表面裂痕监测系统 </v-list-item>

<p></p>

<v-list dense>

<template v-for="router in routermaps[0].children">

<v-list-item

:key="router.name"

@click="jumpto(router)"

v-if="

router.meta.group === false

"

>

<v-list-item-action>

<v-icon>{{ router.meta.icon }}</v-icon>

</v-list-item-action>

<v-list-item-content>

<v-list-item-title>{{ router.name }}</v-list-item-title>

</v-list-item-content>

</v-list-item>

<v-list-group

:key="router.name"

:prepend-icon="router.meta.icon"

v-if="router.meta.group === true"

no-action

>

<template v-slot:activator>

<v-list-item-content>

<v-list-item-title v-text="router.name"></v-list-item-title>

</v-list-item-content>

</template>

<template v-for="children1 in router.children">

<v-list-item

v-if="children1.meta.group === false"

:key="children1.name"

@click="jumpto(children1)"

>

<v-list-item-content>

<v-list-item-title

v-text="children1.name"

></v-list-item-title>

</v-list-item-content>

</v-list-item>

</template>

</v-list-group>

</template>

</v-list>

</v-navigation-drawer>

<v-app-bar app height="60px" flat>

<v-app-bar-nav-icon @click="drawer = !drawer"></v-app-bar-nav-icon>

<v-toolbar-title></v-toolbar-title>

<v-spacer />

<v-toolbar-items>

<v-spacer />

<v-row align="center" class="mx-0">

<v-btn icon @click="back">

<v-icon> arrow\_back </v-icon>

</v-btn>

<v-btn icon @click="changeTheme">

<div v-if="this.darkmode">

<v-icon> brightness\_4 </v-icon>

</div>

<div v-if="!this.darkmode">

<v-icon> brightness\_5 </v-icon>

</div>

</v-btn>

<v-menu

bottom

offset-y

open-on-hover

transition="slide-y-reverse-transition"

>

<template v-slot:activator="{ on }">

<v-btn icon v-on="on">

<v-icon> person </v-icon>

</v-btn>

</template>

<v-list>

<v-list-item @click="loginout">

<v-list-item-title>退出</v-list-item-title>

</v-list-item>

</v-list>

</v-menu>

</v-row>

</v-toolbar-items>

</v-app-bar>

<v-layout align-center justify-center>

<v-flex>

<router-view />

</v-flex>

</v-layout>

</div>

</template>

<script>

import { asyncRouterMap } from "../router/routerConfig";

import notification from "../components/notification.vue";

export default {

name: "BasicLayout",

components: {

notification,

},

props: {

source: String,

},

data: () => ({

drawer: true,

mini: false,

colorRGB: "",

routermaps: {

...asyncRouterMap,

},

permission: "",

oldPassword: "",

newPassword: "",

dialogContent: false,

index: 0,

showO: false,

showN: false,

darkmode: false,

}),

created() {

this.permission = localStorage.getItem("permission");

if (localStorage.getItem("mode") === undefined) {

this.darkmode = false;

this.colorRGB = "#25506B";

localStorage.setItem("mode", 0);

} else {

this.darkmode = localStorage.getItem("mode");

if (this.darkmode === "1") {

this.colorRGB = "#303030";

this.$vuetify.theme.dark = true;

} else {

this.colorRGB = "#25506B";

this.$vuetify.theme.dark = false;

}

}

},

computed: {},

methods: {

back() {

localStorage.setItem("isBackButton", "1");

if (window.history.length <= 1) {

this.$router.push({ path: "/" });

return false;

} else {

this.$router.go(-1);

}

},

changeTheme() {

this.darkmode = !this.darkmode;

if (this.darkmode === true) {

this.colorRGB = "#303030";

localStorage.setItem("mode", 1);

this.$vuetify.theme.dark = true;

} else {

this.colorRGB = "#25506B";

localStorage.setItem("mode", 0);

this.$vuetify.theme.dark = false;

}

},

jumpto(e) {

if (!Object.prototype.hasOwnProperty.call(e.meta, "add")) {

this.$router.push(e.path);

} else {

let subStr = new RegExp("/:.\*"); //创建正则表达式对象

let path = e.path.replace(subStr, "/" + e.meta.add); //替换路由

this.$router.push(path);

}

},

loginout() {

localStorage.clear();

sessionStorage.clear();

this.$router.push("/login");

},

},

};

</script>

<style scoped>

.alert {

position: fixed;

top: 0;

z-index: 9999;

}

</style>

import BasicLayout from "../Layouts/BasicLayout";

import LoginLayout from "../Layouts/LoginLayout";

export const asyncRouterMap = [

{

path: "/",

name: "Basic",

component: BasicLayout,

redirect: "/login",

children: [

{

path: "/home",

name: "机制展示",

component: () => import("../views/Home"),

meta: {

permission: ["1", "2", "3", "4", "5"],

icon: "home",

group: false,

},

},

],

},

];

export const constantRouterMap = [

{

path: "/error",

component: BasicLayout,

children: [

{

path: "/error/404",

component: () => import("../views/error/404"),

meta: {

permission: "all",

},

},

{

path: "/error/403",

component: () => import("../views/error/403"),

meta: {

permission: "all",

},

},

],

},

{

path: "/",

redirect: "/login",

},

{

path: "/login",

component: LoginLayout,

meta: {

permission: "all",

},

},

];

import Vue from "vue";

import VueRouter from "vue-router";

import { constantRouterMap, asyncRouterMap } from "./routerConfig";

Vue.use(VueRouter);

const router = new VueRouter({

mode: "history",

base: process.env.BASE\_URL,

routes: constantRouterMap.concat(asyncRouterMap),

});

export default router;

<template>

<div>

<div style="margin: 1%">

<v-btn color="error" :disabled="errorBtn" @click="switchDialogTrigger">

异常图像

</v-btn>

</div>

<v-dialog v-model="dialogTrigger" max-width="400px">

<v-card>

<v-card-title>异常图像</v-card-title>

<v-card-text>

<!-- <img src="../assets/logo.png" />-->

<v-img src="/api/file?file=%E5%B1%95%E7%A4%BA.jpg"></v-img>

<v-simple-table>

<template v-slot:default>

<thead>

<tr>

<th class="text-left">缺陷类型</th>

<th class="text-left">长度</th>

<th class="text-left">位置</th>

</tr>

</thead>

<tbody>

<tr>

<td>纵裂纹</td>

<td>10mm</td>

<td>x y</td>

</tr>

</tbody>

</template>

</v-simple-table>

</v-card-text>

<v-card-actions>

<v-btn text color="primary" @click="switchDialogTrigger">

关闭

</v-btn>

</v-card-actions>

</v-card>

</v-dialog>

<div style="margin: 1%">

<v-row

style="display: flex; flex-direction: row; justify-content: center"

>

<v-col md="6" cols="12" sm="9" class="flex flex-column d-flex">

<div>

<video-player

class="video-player-box"

ref="videoPlayer"

:playsinline="true"

:options="playerOptions"

@play="onPlayerPlay($event)"

@ready="playerReadied"

@timeupdate="onPlayerTimeupdate($event)"

@pause="onPlayerPause($event)"

></video-player>

</div>

</v-col>

</v-row>

</div>

<div style="margin: 1%">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">相关信息</div>

</v-card>

</v-col>

</v-row>

<v-row no-gutters>

<v-col md="2" cols="4" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">铸坯编号</div>

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="4" cols="8" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

123456

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="2" cols="4" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">铸坯宽度</div>

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="4" cols="8" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

2.1m

</v-card>

</v-col>

</v-row>

</v-col>

</v-row>

<v-row no-gutters>

<v-col md="2" cols="4" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">拉速</div>

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="4" cols="8" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

0.8m/min

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="2" cols="4" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">相机运行状态</div>

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="4" cols="8" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

良好

</v-card>

</v-col>

</v-row>

</v-col>

</v-row>

<v-row no-gutters>

<v-col md="2" cols="4" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">检测人</div>

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="4" cols="8" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

{{ permission }}

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="2" cols="4" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

<div style="font-weight: bold">检测时间</div>

</v-card>

</v-col>

</v-row>

</v-col>

<v-col md="4" cols="8" class="flex flex-column d-flex">

<v-row no-gutters>

<v-col class="flex flex-column d-flex">

<v-card class="pa-2 flex flex-column d-flex" outlined tile>

{{ getTime() }}

</v-card>

</v-col>

</v-row>

</v-col>

</v-row>

</div>

</div>

</template>

<script>

import { notify } from "../components/notification";

import { getCurrentTime } from "../utils/index";

export default {

name: "Home",

data() {

return {

playerOptions: {

playbackRates: [0.7, 1.0, 1.5, 2.0], //播放速度

autoplay: false, //如果true,浏览器准备好时开始回放。

muted: false, // 默认情况下将会消除任何音频。

loop: false, // 导致视频一结束就重新开始。

preload: "auto", // 建议浏览器在<video>加载元素后是否应该开始下载视频数据。auto浏览器选择最佳行为,立即开始加载视频（如果浏览器支持）

language: "zh-CN",

aspectRatio: "16:9", // 将播放器置于流畅模式，并在计算播放器的动态大小时使用该值。值应该代表一个比例 - 用冒号分隔的两个数字（例如"16:9"或"4:3"）

fluid: true, // 当true时，Video.js player将拥有流体大小。换句话说，它将按比例缩放以适应其容器。

sources: [

{

type: "video/mp4",

src: "/api/file?file=展示.mp4", //url地址

},

],

poster: "/api/file?file=展示.jpg", //你的封面地址

notSupportedMessage: "此视频暂无法播放，请稍后再试", //允许覆盖Video.js无法播放媒体源时显示的默认信息。

controlBar: {

timeDivider: true,

durationDisplay: true,

remainingTimeDisplay: false,

fullscreenToggle: true, //全屏按钮

},

},

gklog: "", //视频观看时长

pauseTrigger: [

[5, false],

],

errorBtn: true,

dialogTrigger: false,

permission: "",

};

},

mounted() {

this.permission = localStorage.getItem("Access-Token");

},

components: {},

methods: {

onPlayerTimeupdate(player) {

this.gklog = player.cache\_.currentTime;

let i = 0;

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

if (this.gklog >= this.pauseTrigger[i][0] && !this.pauseTrigger[i][1]) {

this.$refs.videoPlayer.player.pause();

this.pauseTrigger[i][1] = true;

this.errorBtn = false;

notify("error", "异常出现，请点击错误图像查看大图");

break;

}

}

console.log(" onPlayerTimeupdate!", this.gklog);

},

/\* 设置视频进度 \*/

playerReadied(player) {

player.currentTime(this.gklog);

},

onPlayerPause() {},

onPlayerPlay() {},

switchDialogTrigger() {

this.dialogTrigger = !this.dialogTrigger;

},

getTime() {

return getCurrentTime();

},

},

};

</script>

<template>

<div class="about">

<h1>This is an about page</h1>

</div>

</template>

<template>

<div>

<div

style="

width: 100%;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

"

>

<div class="display-3" style="margin-top: 20%">404</div>

<div class="display-1">page not found</div>

</div>

</div>

</template>

<script>

export default {

name: "app",

created() {},

data() {},

methods: {},

};

</script>

<style scoped></style>

<template>

<div>

<div style="width: 100%;display: flex;flex-direction: column;justify-content: center;align-items: center">

<div class="display-3" style="margin-top: 20%">

403

</div>

<div class="display-1">

permission denied

</div>

</div>

</div>

</template>

<script>

export default {

name: "403"

};

</script>

<style scoped></style>

import "@mdi/font/css/materialdesignicons.css"; // Ensure you are using css-loader

import "material-design-icons-iconfont/dist/material-design-icons.css"; // 确保您正在使用 css-loader

import Vue from "vue";

import Vuetify from "vuetify/lib/framework";

Vue.use(Vuetify);

export default new Vuetify({

icons: {

iconfont: "md", // 默认值 - 仅用于展示目的

},

});

<template>

<v-app>

<v-main>

<router-view />

</v-main>

</v-app>

</template>

<script>

export default {

name: "App",

data: () => ({

}),

};

</script>

<template>

<v-container>

<v-row class="text-center">

<v-col cols="12">

<v-img

:src="require('../assets/logo.svg')"

class="my-3"

contain

height="200"

/>

</v-col>

<v-col class="mb-4">

<h1 class="display-2 font-weight-bold mb-3">Welcome to Vuetify</h1>

<p class="subheading font-weight-regular">

For help and collaboration with other Vuetify developers,

<br />please join our online

<a href="https://community.vuetifyjs.com" target="\_blank"

>Discord Community</a

>

</p>

</v-col>

<v-col class="mb-5" cols="12">

<h2 class="headline font-weight-bold mb-3">What's next?</h2>

<v-row justify="center">

<a

v-for="(next, i) in whatsNext"

:key="i"

:href="next.href"

class="subheading mx-3"

target="\_blank"

>

{{ next.text }}

</a>

</v-row>

</v-col>

<v-col class="mb-5" cols="12">

<h2 class="headline font-weight-bold mb-3">Important Links</h2>

<v-row justify="center">

<a

v-for="(link, i) in importantLinks"

:key="i"

:href="link.href"

class="subheading mx-3"

target="\_blank"

>

{{ link.text }}

</a>

</v-row>

</v-col>

<v-col class="mb-5" cols="12">

<h2 class="headline font-weight-bold mb-3">Ecosystem</h2>

<v-row justify="center">

<a

v-for="(eco, i) in ecosystem"

:key="i"

:href="eco.href"

class="subheading mx-3"

target="\_blank"

>

{{ eco.text }}

</a>

</v-row>

</v-col>

</v-row>

</v-container>

</template>

<script>

export default {

name: "HelloWorld",

data: () => ({

ecosystem: [

{

text: "vuetify-loader",

href: "https://github.com/vuetifyjs/vuetify-loader",

},

{

text: "github",

href: "https://github.com/vuetifyjs/vuetify",

},

{

text: "awesome-vuetify",

href: "https://github.com/vuetifyjs/awesome-vuetify",

},

],

importantLinks: [

{

text: "Documentation",

href: "https://vuetifyjs.com",

},

{

text: "Chat",

href: "https://community.vuetifyjs.com",

},

{

text: "Made with Vuetify",

href: "https://madewithvuejs.com/vuetify",

},

{

text: "Twitter",

href: "https://twitter.com/vuetifyjs",

},

{

text: "Articles",

href: "https://medium.com/vuetify",

},

],

whatsNext: [

{

text: "Explore components",

href: "https://vuetifyjs.com/components/api-explorer",

},

{

text: "Select a layout",

href: "https://vuetifyjs.com/getting-started/pre-made-layouts",

},

{

text: "Frequently Asked Questions",

href:

"https://vuetifyjs.com/getting-started/frequently-asked-questions",

},

],

}),

};

</script>

<template>

<v-snackbar :value="showmessage" :color="type">

{{ message }}

<template v-slot:action="{ attrs }">

<v-btn dark text v-bind="attrs" @click="close()">

<v-icon>close</v-icon>

</v-btn>

</template>

</v-snackbar>

</template>

<script>

export default {

name: "notification",

created() {},

computed: {

type() {

return this.$store.state.type;

},

showmessage() {

return this.$store.state.showmessage;

},

message() {

let { message } = this.$store.state;

if (message.length > 25) {

message = `${message.substring(0, 25)}..`;

}

return message;

},

},

methods: {

close() {

this.$store.commit("removeAlert");

},

},

};

</script>

<style scoped></style>

import store from "../store";

export function notify(Mtype, message) {

store.commit({

type: "alert",

Mtype,

message,

showmessage: true,

});

setTimeout(() => {

store.commit("removeAlert");

}, 1000);

}

import Vue from "vue";

import App from "./App.vue";

import router from "./router";

import store from "./store";

import "./permission";

import VueVideoPlayer from "vue-video-player";

import "video.js/dist/video-js.css";

import vuetify from "./plugins/vuetify";

Vue.config.productionTip = false;

Vue.use(

VueVideoPlayer /\* {

options: global default options,

events: global videojs events

} \*/

);

new Vue({

router,

store,

vuetify,

render: (h) => h(App),

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

import NProgress from "nprogress";

import router from "./router";

import "nprogress/nprogress.css";

const whitelist = ["/login", "/error/404", "/error/403"];

router.beforeEach((to, from, next) => {

NProgress.start();

const AccessToken = localStorage.getItem("Access-Token");

if (AccessToken === null && whitelist.indexOf(to.path) === -1) {

next("/login");

} else {

next();

}

});

router.afterEach(() => {

NProgress.done();

});

import Vue from "vue";

import Vuex from "vuex";

Vue.use(Vuex);

const store = new Vuex.Store({

state: {

type: "success",

message: "",

showmessage: false,

file\_upload: 0,

},

mutations: {

alert(state, payload) {

state.message = payload.message;

state.type = payload.Mtype;

state.showmessage = payload.showmessage;

},

removeAlert(state) {

state.showmessage = false;

},

process\_change(state, payload) {

state.file\_upload = payload.upload;

},

},

actions: {},

});

export default store;