<script>

new Vue({

el: '#login-app',

data() {

return {

loginForm:{

username: 'admin',

password: '123456'

},

loading: false

}

},

computed: {

loginRules() {

//设置一个检查函数

//rule 表示当前验证规则的信息，可以包含验证规则的属性，例如 required、min、max 等//等value 表示当前字段的值，它将被验证是否符合规则

//callback 是一个函数，用于在验证结束后通知 Vue.js 表单验证系统。如果验证成功，可以//不带参数调用它；如果验证失败，则需要传递一个 Error 对象作为参数，并在该对象的消//息中包含有关验证失败的详细信息。

const validateUsername = (rule, value, callback) => {

if (value.length < 1 ) {

callback(new Error('请输入用户名'))

} else {

callback()

}

}

const validatePassword = (rule, value, callback) => {

if (value.length < 6) {

callback(new Error('密码必须在6位以上'))

} else {

callback()

}

}

//validator 是一个函数，用于验证当前字段的值是否符合规则。在这个例子中，//validateUsername 和 validatePassword 函数分别用于验证 username 和 password 字段。

//trigger 定义了触发验证的事件。在这个例子中，它被设置为 'blur'，表示当字段失去焦点//时将触发验证。也可以设置为其他事件，例如 'change'、'input' 等。

return {

'username': [{ 'validator': validateUsername, 'trigger': 'blur' }],

'password': [{ 'validator': validatePassword, 'trigger': 'blur' }]

}

}

},

created() {

},

methods: {

async handleLogin() {

this.$refs.loginForm.validate(async (valid) => {

if (valid) {

this.loading = true

let res = await loginApi(this.loginForm)

if (String(res.code) === '1') {

localStorage.setItem('userInfo',JSON.stringify(res.data))

window.location.href= '../../index.html'

} else {

this.$message.error(res.msg)

this.loading = false

}

}

})

}

}

})

</script>