<template>

<div class="login">

<div class="loginbox">

<!-- 标题 -->

<div class="title">

<span class="el-icon-menu"></span>

<span>启课教学助手</span>

</div>

<el-form :model="loginForm" status-icon :rules="rules" ref="loginForm" label-width="100px" class="demo-ruleForm">

<!-- 账号 -->

<el-form-item label="账号" prop="account">

<el-input type="text" v-model="loginForm.account" auto-complete="off"></el-input>

</el-form-item>

<!-- 密码 -->

<el-form-item label="密码" prop="password">

<el-input type="password" v-model="loginForm.password" auto-complete="off"></el-input>

</el-form-item>

<!-- 确定密码 -->

<!-- <el-form-item label="确认密码" prop="checkPass">

<el-input type="password" v-model="loginForm.checkPass" auto-complete="off"></el-input>

</el-form-item> -->

<el-form-item>

<el-button type="success" @click="submitForm('loginForm')">登&emsp;录</el-button>

<el-button @click="resetForm('loginForm')">重&emsp;置</el-button>

</el-form-item>

</el-form>

</div>

</div>

</template>

<script>

// 引入验证用户名函数

import { accountReg,passwordReg } from "@/utils/validator/validator.js";

// 引入本地存储localStorage函数

import local from '@/utils/local/local';

export default {

data(){

// 用户名验证

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

if(!value){

callback(new Error("账号不能为空"));//输出错误提示

}else if(!accountReg(value)){

callback(new Error("只能输入5-20个以字母开头、可带数字、“\_”、“.”"));//输出错误提示

}else{

callback();//成功回调

}

}

// 密码验证

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

if(!value){

callback(new Error("密码不能为空"));//输出错误提示

}else if(!passwordReg(value)){

callback(new Error("密码不能含有非法字符，长度在4-10之间"));//输出错误提示

}else{

/\* if(this.loginForm.checkPass){//如果确认密码不为空触发一致性验证

// 触发一致性验证

this.$refs.loginForm.validateField("checkPass");

} \*/

callback();//成功回调

}

}

/\* // 确认密码验证

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

if(!value){

callback(new Error("确认密码不能为空"));//输出错误提示

}else if(value !== this.loginForm.password){

callback(new Error("两次输入不一致"));//输出错误提示

}else{

callback();//成功回调

}

} \*/

return{

// 表单数据

loginForm:{

account:"",

password:"",

checkpass:"",

},

// 验证规则

rules:{

// 账号验证

account:[

{required:true,validator: validateAccount, trigger: 'blur' }

],

// 密码验证

password:[

{required:true,validator: validatePassword, trigger: 'blur' }

],

}

}

},

methods:{

// 重置按钮事件

resetForm(formName) {

this.$refs[formName].resetFields();

},

// 提交按钮事件

submitForm(formName) {

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

// 如果成功

if (valid) {

// 提交数据给后端

let params = {

account:this.loginForm.account,

password:this.loginForm.password

};

// 提交数据给后台

this.request.post('/login/checklogin',params)

.then(res=>{

// 接收后端传回来的数据

let {code,msg,token}=res;

if(code===0){

// 将数据存储到本地储存

local.save('qike\_2019\_system',token);

// 弹成功提示

this.$message({

type:"success",

message:msg,

duration:1000

})

// 跳转路由

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

}else if(code === 1){

this.$message.error(msg);

}

})

.catch(err=>{

console.log(err);

})

// 路由跳转

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

} else {

return false;

}

});

},

}

}

</script>

<style lang="less">

@import "./login.less";

</style>