1. 验证码组件
   1. 预期效果



图1-1 验证码组件效果图

* 1. 掌握技能

表1-1 任务说明

|  |  |  |
| --- | --- | --- |
| 任务名称 | 子任务 | 任务内容 |
| 验证码组件开发 | 1 | 验证码组件的html,css,js |

* 1. 任务分解
     1. 验证码组件的开发
        1. 在当前项目目录下的/src/component路径下创建verification.vue文件：
        2. 组件的html,css,js代码：

<!-- 这是模板 -->

<template>

<canvas v-on:click.stop="changVer()" class="verCanvas" width="100%" height="35px"></canvas>

</template>

<!-- 这是js代码 -->

<script>

export default {

name: 'login',

data () {

return {

canvas: "",

width:"",

height:"",

ctx:""

}

},

mounted:function(){

this.canvas=document.getElementsByClassName('verCanvas')[0];

this.width=this.canvas.width;

this.height=this.canvas.height;

this.ctx = this.canvas.getContext('2d');

this.ctx.textBaseline = 'bottom';

this.getVer( this.drawPic);

},

computed: {

},

methods:{

getVer:function(fn){

this.$reqs({//180.76.156.148

method: 'get',

url: this.$store.state.url.BASEURL+this.$store.state.url.GETVERICOD

}).then(function (response) {

fn( response.data.data);

});

},

randomNum:function (min,max){//生成随机数

return Math.floor( Math.random()\*(max-min)+min);

},

randomColor:function (min,max){//生成随机色

var r = this.randomNum(min,max);

var g = this.randomNum(min,max);

var b = this.randomNum(min,max);

return "rgb("+r+","+g+","+b+")";

},

drawPic:function (returnTxt){ /\*\*绘制验证码图片\*\*/

/\*\*绘制背景色\*\*/

this.ctx.fillStyle = this.randomColor(180,240); //颜色若太深可能导致看不清

this.ctx.fillRect(0,0,this.width,this.height);

/\*\*绘制文字\*\*/

for(var i=0; i<4; i++){

var txt = returnTxt[i];

this.ctx.fillStyle = this.randomColor(50,160); //随机生成字体颜色

this.ctx.font = this.randomNum(25,30)+'px SimHei'; //随机生成字体大小

var x = 10+i\*20;

var y = this.randomNum(25,45);

var deg = this.randomNum(-15, 15);

//修改坐标原点和旋转角度

this.ctx.translate(x,y);

this.ctx.rotate(deg\*Math.PI/180);

this.ctx.fillText(txt, 10,0);

//恢复坐标原点和旋转角度

this.ctx.rotate(-deg\*Math.PI/180);

this.ctx.translate(-x,-y);

}

/\*\*绘制干扰线\*\*/

for(var i=0; i<8; i++){

this.ctx.strokeStyle = this.randomColor(40,180);

this.ctx.beginPath();

this.ctx.moveTo( this.randomNum(0,this.width), this.randomNum(0,this.height) );

this.ctx.lineTo( this.randomNum(0,this.width), this.randomNum(0,this.height) );

this.ctx.stroke();

}

/\*\*绘制干扰点\*\*/

for(var i=0; i<100; i++){

this.ctx.fillStyle = this.randomColor(0,255);

this.ctx.beginPath();

this.ctx.arc(this.randomNum(0,this.width),this.randomNum(0,this.height), 1, 0, 2\*Math.PI);

this.ctx.fill();

}

return;

},

changVer:function(){

this.getVer( this.drawPic);

}

},

components: {}

};

</script>

<!-- 这是样式 -->

<style>

.verCanvas{

margin-left:5px;

position:relative !important;

}

.verCanvas:hover{

cursor:pointer;

}

</style>

* + - 1. 在其他组件中引用验证码组件：

这里以一个空的组件为例示范在其他组件中的引用过程，：

<template>

<div class="containner">

<vercavas></vercavas>

Template中只能包含一个子代div标签，这个div标签内当然可以嵌套其他div或任何标签，我们的验证码引用标签就放在这里，<vercavas>标签的名字与 js中import时自己取的名字要对应

</div>

</template>

<script>

import vercavas from './verification.vue';

在这里引入组件

export default {

name: 'login',

data () {

return {

msg: ""

}

},

mounted:function(){

},

destoryed:function(){

},

computed: { },

methods:{ },

components: {vercavas}

引入组件时自己取的名字也要作为一个属性写在components对象里，这是告诉VUE应用这个组件的意思

* 1. 参考资料

无

* 1. 扩展练习

教师自行补充