**第二题**

function CountList2(list){

let obj={};

list.forEach((item)=>{

if(obj[item.lang]){

obj[item.lang]+=1;

}else{

obj[item.lang]=1;

}

})

return obj

}

console.log(CountList2(list));

**第三题**

第一种，vue项目中配置proxy

找到vue.config.js,配置

devServer:{

proxy:{

'/news':{

target:'http://www.qq.com'

}

}

}

第二种：使用node + express + http-proxy-middleware搭建一个proxy服务器。

前端代码：

var xhr = new XMLHttpRequest();

// 前端开关：浏览器是否读写cookie

xhr.withCredentials = true;

// 访问http-proxy-middleware代理服务器

xhr.open('get', 'http://www.domain1.com:3000/ ', true);

xhr.send();

中间件服务器代码：

var express = require('express');

var proxy = require('http-proxy-middleware');

var app = express();

app.use('/', proxy({

// 代理跨域目标接口

target: 'http://www.qq.com/news/recommended',

changeOrigin: true,

// 修改响应头信息，实现跨域并允许带cookie

onProxyRes: function(proxyRes, req, res) {

res.header('Access-Control-Allow-Origin', 'http://www.domain1.com');

res.header('Access-Control-Allow-Credentials', 'true');

},

// 修改响应信息中的cookie域名

cookieDomainRewrite: 'www.domain1.com' // 可以为false，表示不修改

}));

app.listen(3000);

console.log('Proxy server is listen at port 3000...');

**第五题**

定义滚动条事件，注意进行节流。

List.vue

|  |
| --- |
| <template>  <div  @scroll="handleScroll"  style="height: 700px; width: 300px; overflow-y: auto"  >  <div  v-for="(item, index) in list"  :key="index"  class="box"  :style="{ background: item.background }"  >  {{ index }}  </div>  <div v-show="list.length<50">加载中...</div>  </div>  </template>  <script>  import { reactive } from "vue";  export default {  name: "myList",  setup() {  let list = reactive([{ background: "rgb(233,32,38)" }]);  let timer = null;  function getList(num) {  for (let i = 0; i < num; i++) {  let c1 = Math.floor(Math.random() \* 256);  let c2 = Math.floor(Math.random() \* 256);  let c3 = Math.floor(Math.random() \* 256);  list.push({ background: "rgb(" + c1 + "," + c2 + "," + c3 + ")" });  }  }  getList(10);  const handleScroll = (e) => {  if (timer) return;  timer = setTimeout(() => {  timer = null;  const scrollTop = e.target.scrollTop;  const wrapperHeight = e.target.clientHeight;  const scrollHeight = e.target.scrollHeight;  if (scrollTop + wrapperHeight >= scrollHeight / 2 && list.length < 50) {  getList(10);  }  }, 2000);  }; return {  list,  handleScroll,  };  },  };  </script>  <style lang='less' scoped >  .box {  width: 100%;  height: 500px;  }  </style> |