**云端数据查询**

**1、用户列表**

第一步，编辑index.vue页面，效果如下4-20所示。



图4-20 编辑index.vue页面

对应代码如下所示：

|  |
| --- |
| <template>  <view>  <view class="item">  <button type="default" @click="goto('add')" >添加用户</button>  **<button type="default" @click="goto('list')" >用户列表</button>**  </view>  </view>  </template>  <script>  export default {  data() {  return {  title: 'Hello'  }  },  onLoad() {  },  methods: {  goto(e) {  uni.navigateTo({  url: "../" + e + "/" + e  })  }  }  }  </script>  <style>  page{  margin: 10rpx;  }    **button{**  **margin-top: 30rpx;**  **}**    .title {  font-size: 36rpx;  color: #8f8f94;  }  </style> |

第二步，新建一个list.vue页面，如下图4-21所示。



图4-21 新建list.vue页面

第三步：创建云函数，如图4-22所示。



图4-22 创建getUser云函数

Index.js代码如下所示

|  |
| --- |
| 'use strict';  const db= uniCloud.database()  const userDb= db.collection('user')  exports.main = async (event, context) => {  let res= await userDb.where({}).get();  return res;  }; |

第四步：上传函数，如图4-23所示。

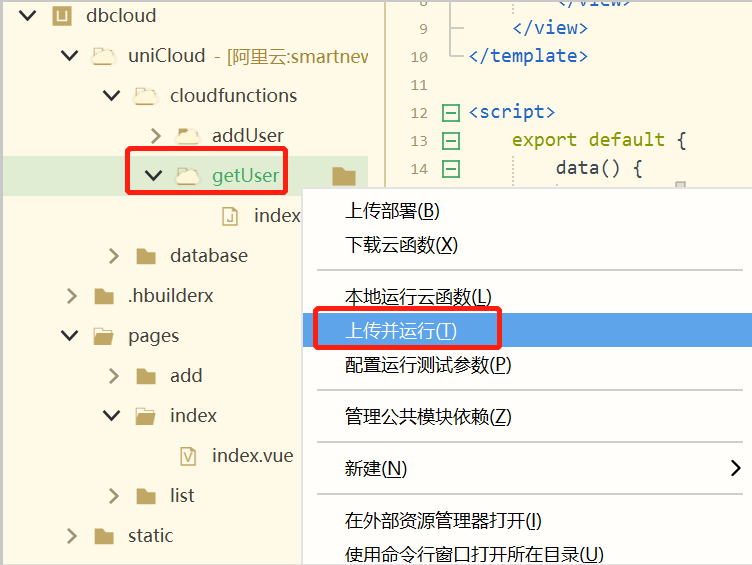


图4-23 上传云函数

第五步：编辑list.vue页面。

|  |
| --- |
| <template>  <view>  <view class="list" v-for="(item,i) in list" :key="i">  <view >  <view>序号：{{i+1}}</view>  <view>用户名：{{item.username}} </view>  <view>手机号：{{item.mobile}} </view>  <view>Email:{{item.email}}</view>  </view>    <view class="btn"><button>删除</button> <button>编辑</button> </view>  </view>  </view>  </template>  <script>  export default {  data() {  return {  list: []  }  },  onLoad() {  this.getData();  },  methods: {  getData() {  console.log("------------执行查询操作-----------");  uniCloud.callFunction({  name: "getUser",  success: (e) => {  console.log(e.result);  this.list = e.result.data;  }  })  }  }  }  </script>  <style>  page {  margin: 30rpx;  }  .list {  border-bottom: 1rpx solid #F1F1F1;  padding-bottom: 20rpx;  }  .btn{  display: flex;  }    </style> |

预览效果如下图4-24所示。



图4-24 预览效果