**一、uniCloud简介**

**1、什么是uniCloud**

uniCloud 是 DCloud 联合阿里云、腾讯云，为开发者提供的基于 serverless 模式和 js 编程的云开发平台。

**补充：**serverless模式：要说Serverless是什么，直译过来就是无服务器。根据 CNCF 的定义，Serverless 是指构建和运行不需要服务器管理的应用程序的概念。

2、**优点**

* 熟悉的js，轻松搞定前后台整体业务。
* 开发成本大幅下降。不用再雇佣php或java等服务器工程师，每年至少节省几十万。
* 只需专注于你的业务，其他什么服务器运维、弹性扩容、防DDoS攻击，全都不需要操心;
* 不需要购买备案域名。小程序和App可以免域名使用服务器。

**3、创建数据库**

第一步：登录云端（网址：https://unicloud.dcloud.net.cn/login），登录成功后，首先可以查看到已有服务器空间，如下图4-1所示。



图4-1 创建云端数据库空间

第二步：创建空间。点击右上角【创建服务空间】，进入下图4-2所示界面。

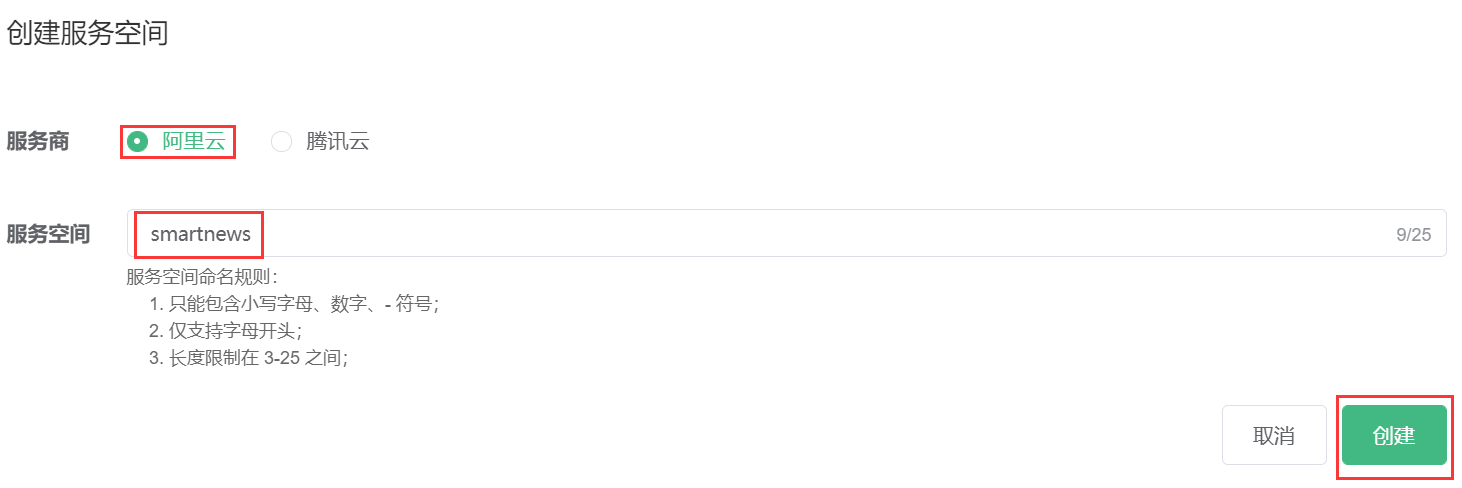


图4-2 选择空间服务商并命名

至此服务空间创建成功。

**4、创建数据表**

在创建表时我们需要了解如何编写DB Schema。

**{**

**"bsonType": "object",**

**"required": [],**

**"permission": {**

**"read": false,**

**"create": false,**

**"update": false,**

**"delete": false**

**},**

**"properties": {**

**"\_id": {**

**"description": "ID，系统自动生成"**

**}**

**}**

**}**

**第一步：创建schema。**在项目右键，选择创建database目录。如下图4-3所示。

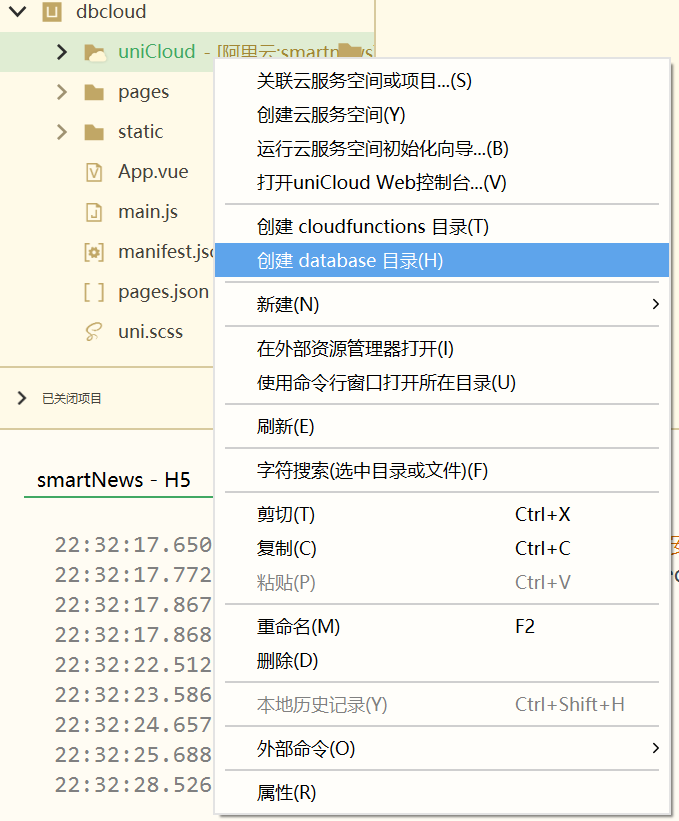


图4-3 创建database目录

在第一步创建的database目录右键选择新建数据集合schema，如图4-4所示。

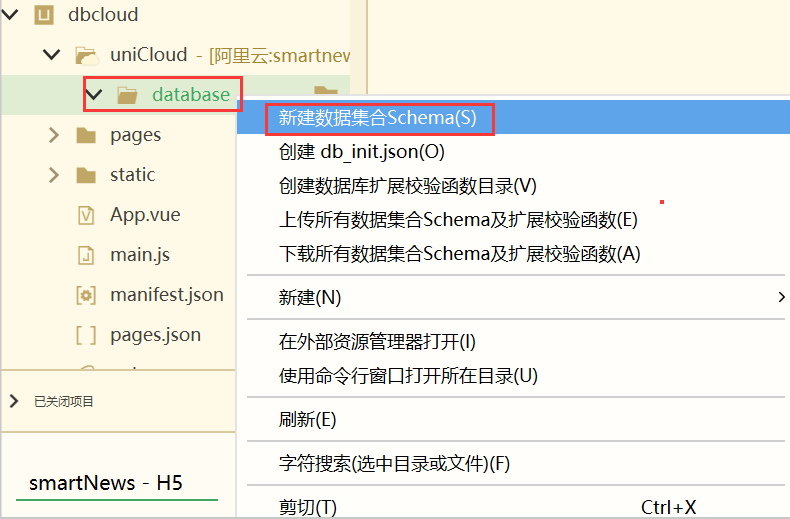


图4-4 新建数据集合

输入数据集合名称，如图4-5所示。

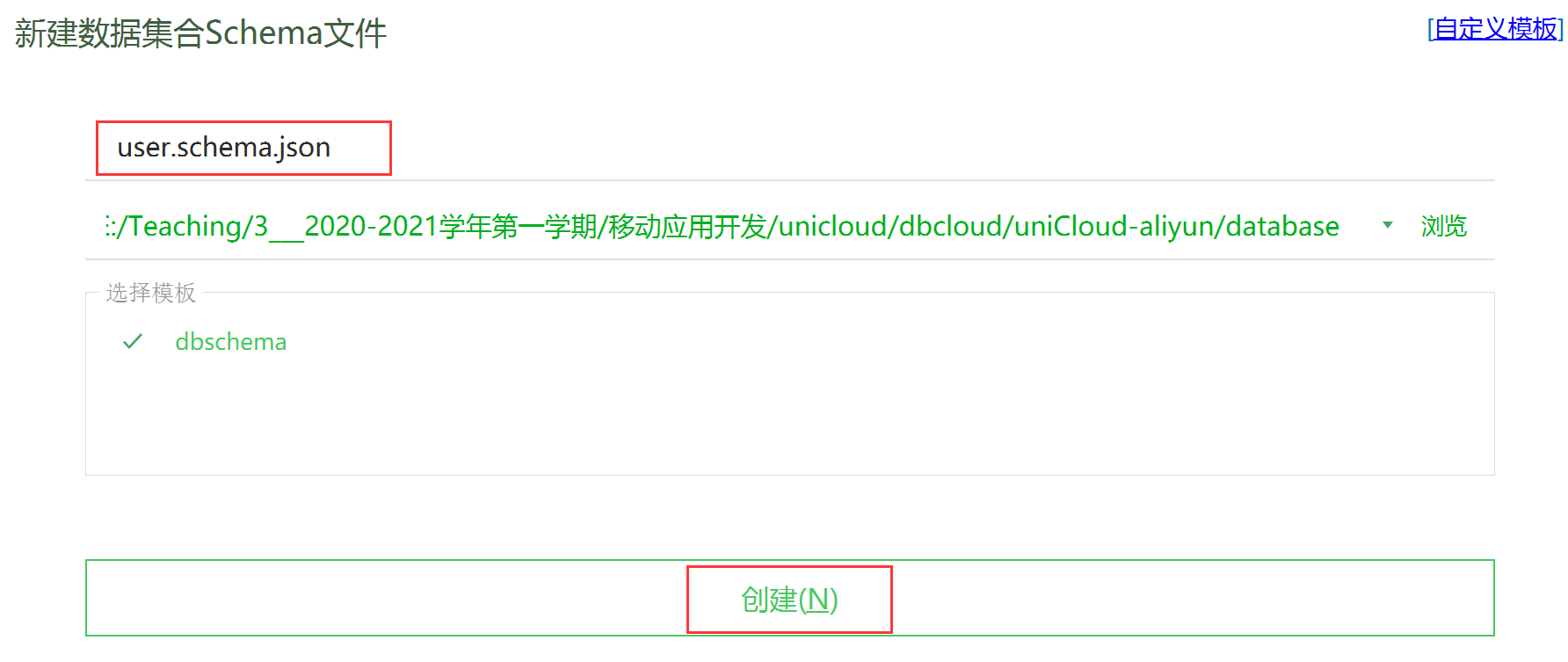


图4-5 创建user数据集合

创建成功后，文档默认结构如下图4-6所示。



图4-6 默认结构

**第二步：上传schema。**在单个schema文件右键可以只上传当前选中的schema,如图4-7所示。快捷键是【Ctrl+u】。（Ctrl+u是HBuilderX的通用快捷键，不管是发布App还是上传云函数、schema，都是Ctrl+u）

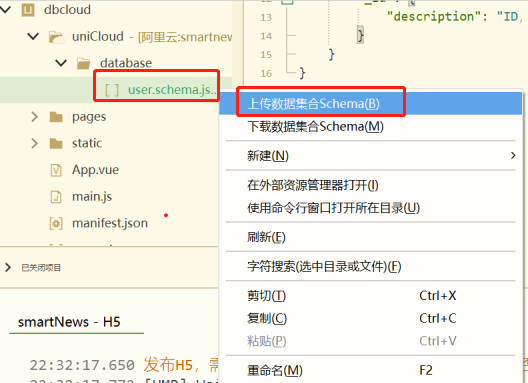


图4-7 上传数据集合

提示是否自动创建该表并上传，如图4-8所示。

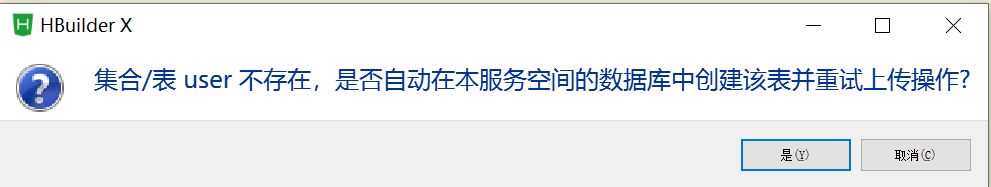


图4-8 上传提示

上传成功后，可以前往云端查看，如图4-9所示。

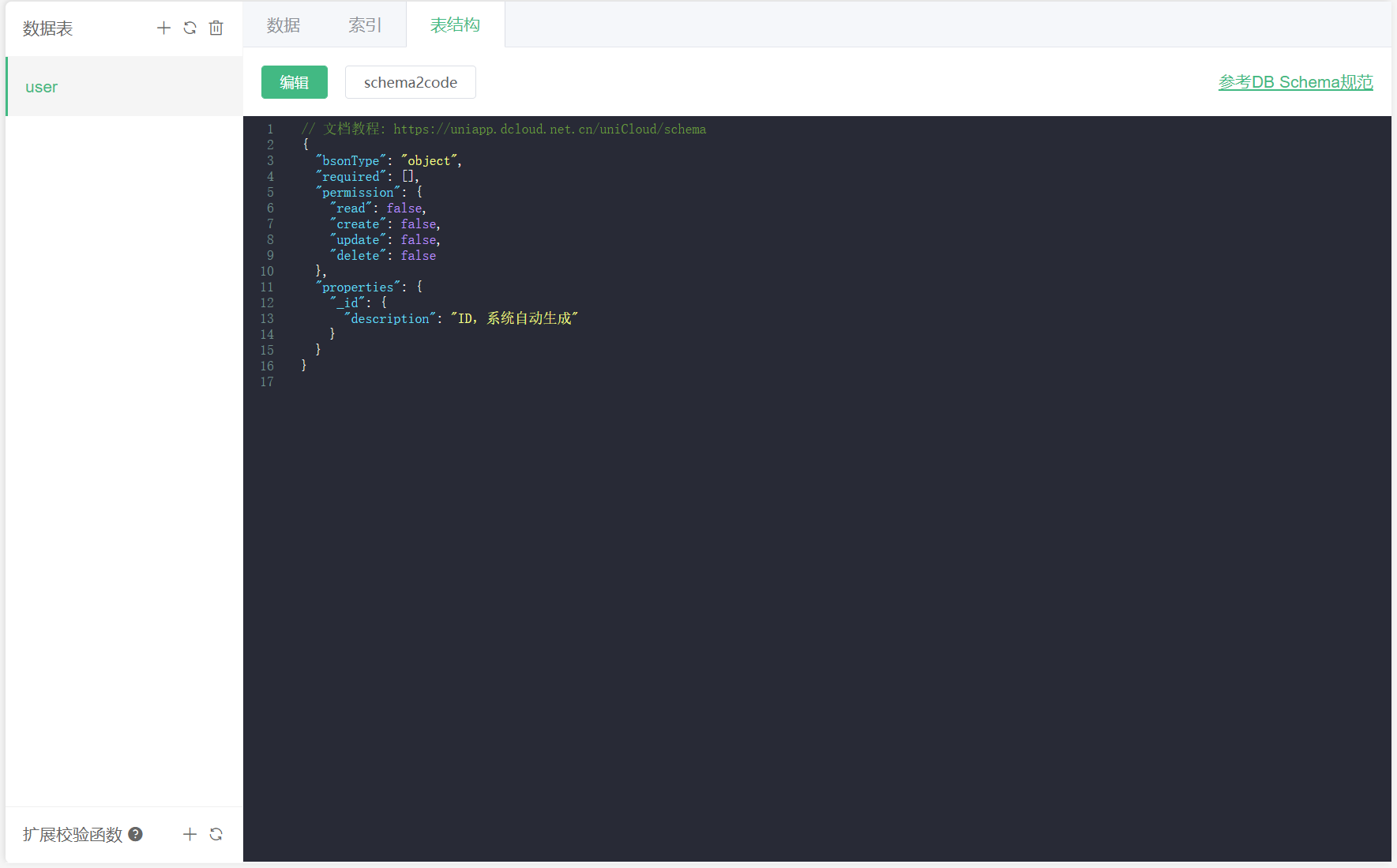


图4-9 云端查看上传结果

**5、添加数据**

第一步：编辑index.vue页面，效果如下图4-11所示.



图4-11 index.vue效果

对应代码如下所示：

|  |
| --- |
| <template>  <view>  <view class="item">  <button type="default" @click="goto('add')" >添加用户</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;  }  </style> |

第二步，新建一个add.vue页面，如下图4-12所示。



图4-12 新建add.vue页面

第三步：创建云函数,如图4-13至图4-16所示。

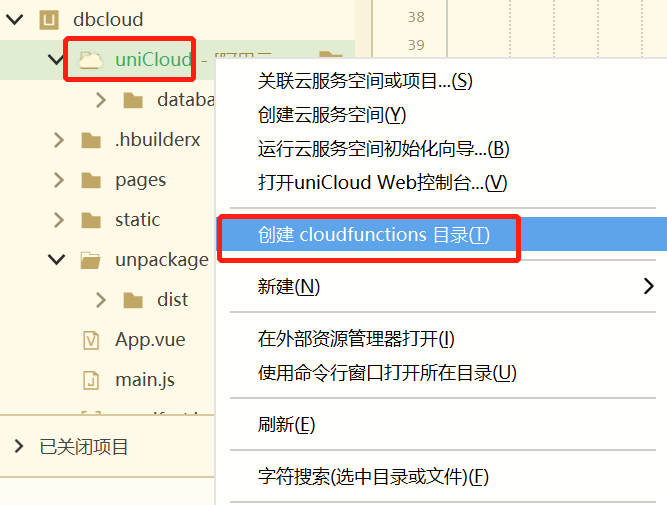


图4-13 新建云函数目录



图4-14 创建云函数



图4-15 创建addUser云函数

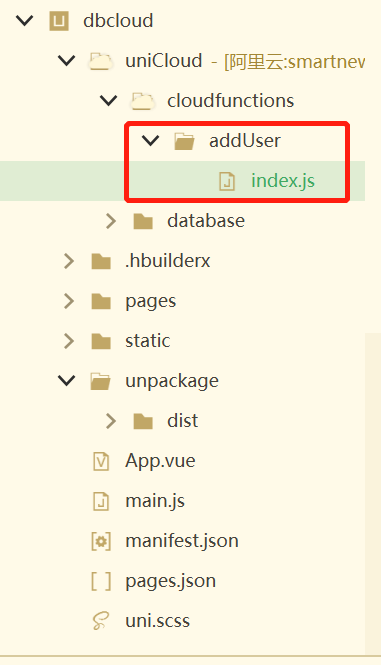


图4-16 云函数目录结构

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 addUserRes= await userDb.add(event);  return addUserRes;  }; |

第四步：上传函数，如图4-17所示。

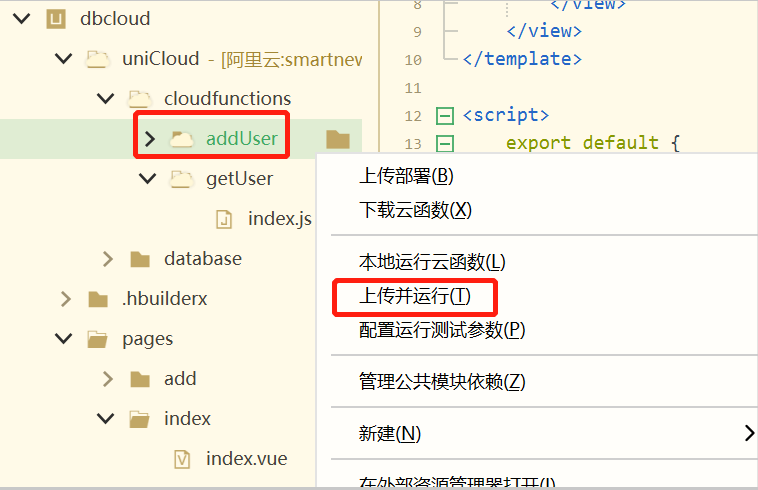


图4-17 上传云函数

第五步：编辑add.vue页面。

|  |
| --- |
| <template>  <view>  <view class="form">  <view class="content ">  <view class="forget-bg">  <view class="forget-card">  <view class="forget-input forget-margin-b">  <input type="text" placeholder="用户名" v-model="form.username" />  </view>  <view class="forget-input forget-margin-b">  <input type="text" placeholder="密码" v-model="form.password" />  </view>  <view class="forget-input forget-margin-b">  <input type="text" placeholder="昵称" v-model="form.nickname" />  </view>  <view class="forget-input forget-margin-b">  <input type="number" placeholder="手机号" v-model="form.mobile" />  </view>  <view class="forget-input forget-margin-b">  <input type="text" placeholder="Email" v-model="form.email" />  </view>  </view>  </view>  <view class="forget-btn">  <button class="landing" type="primary" @click="submit"> 添 加 </button>  </view>  </view>  </view>  </view>  </template>  <script>  export default {  data() {  return {  form: {  username: "",  password: "",  nickname: "",  mobile: "",  email: ""  }  }  },  methods: {  submit() {  console.log("------------执行添加操作-----------");  uniCloud.callFunction({  name: 'addUser', //云函数名称  data: this.form,  success: (e) => {  console.log(e.result); //打印返回数据  uni.showToast({  title: '添加成功',  duration: 2000  });  },  })  }  }  }  </script>  <style>  .verify-left {  width: calc(100% - 260upx);  }  .verify-right {  padding-left: 20upx;  }  .verify-btn {  height: 80upx;  line-height: 80upx;  font-size: 28upx;  width: 240upx;  border-radius: 8upx;  background: linear-gradient(left, #FF978D, #FFBB69);  }  .verify-left,  .verify-right {  float: left;  }  .landing {  height: 84upx;  line-height: 84upx;  border-radius: 44upx;  font-size: 32upx;  background: linear-gradient(left, #FF978D, #FFBB69);  }  .forget-btn {  padding: 10upx 20upx;  margin-top: 580upx;  }  .forget-input input {  background: #F2F5F6;  font-size: 28upx;  padding: 10upx 25upx;  height: 62upx;  line-height: 62upx;  border-radius: 8upx;  }  .forget-margin-b {  margin-bottom: 25upx;  }  .forget-input {  padding: 10upx 20upx;  overflow: auto;  }  .forget-card {  background: #fff;  border-radius: 12upx;  padding: 60upx 25upx;  box-shadow: 0 6upx 18upx rgba(0, 0, 0, 0.12);  position: relative;  margin-top: 120upx;  }  .forget-bg {  height: 260upx;  padding: 25upx;  background: linear-gradient(#FF978D, #FFBB69);  }  </style> |