

# 15分钟打造 你自己的小程序

小程序从入门到精通（一）

「零基础开发自己的支付宝小程序  
(内附详细代码)

To Do 待办事项小程序、个人相册小程序、  
花呗支付小程序、运动步数小程序、  
云笔记小程序



阿里云开发者电子书系列



查看小程序云官方文档



阿里云开发者“藏经阁”  
海量免费电子书下载

# 目录

---

什么是小程序云 .....	2
开发前准备 .....	4
开发待办事项小程序 .....	21
搭建个人相册小程序 .....	27
快速搭建个人云笔记小程序 .....	32
教你开发获取运动步数小程序 .....	38
云上开发花呗支付小程序 .....	45

# 什么是小程序云

---

小程序云是阿里云面向小程序场景提供的一站式云服务，帮助开发者实现一云多端的业务战略。开发者可通过小程序云支撑各类小程序前端，在一朵云内实现统一的资源管理、统一的数据运营和统一的业务设计。

小程序云提供小程序Serverless和小程序云应用服务，帮助小程序开发者降低研发和运维成本。

## 小程序Serverless

阿里云小程序Serverless提供包括云函数、数据存储、文件存储等一整套后端服务。开发者通过API方式即可获取云函数、数据存储、文件存储、音视频、图像处理等服务，不需要关心服务器或底层运维设施，可以更专注于代码和业务本身。

Serverless是一种构建和管理基于微服务架构的完整流程。计算资源作为服务而不是服务器的概念出现，开发者不再需要考虑服务器的问题，可以更专注于代码和业务，更快速地开发软件。Serverless通常包含BaaS和FaaS服务：

- BaaS：后端即服务（Backend as a Service），向开发者提供后端基础功能，通过API方式进行调用。例如文件存储、数据库、消息等。
- FaaS：函数即服务（Function as a Service），允许开发者直接将程序部署到平台上，以函数作为最小单元，通过API方式进行调用。

阿里云小程序Serverless提供以下功能：

- 数据存储

数据存储服务是基于MongoDB托管在云端的数据库，数据以JSON格式存储。数据库中的每条记录都是一个JSON格式的对象。一个数据库可以有多个集合（相当于关系型数据库中的表）。

- 文件存储

文件存储服务支持文本、图片和其他由用户生成的内容存储到云端。开发者可以在小程序端和控制台使用云存储功能。图片上传成功后，系统会自动生成一个资源链接。开发者可以在小程序中使用该图片地址。

- 云函数

云函数服务支持使用Node.js进行开发。作为开发者，您可以将代码提交到云端运行，在客户端使用小程序提供的API进行调用。您还可以在云函数中直接通过API调用数据存储和文件存储的服务资源。

## 小程序云应用

小程序云应用是面向小程序应用场景，为开发者提供的一键构建后端应用运行环境、后端服务部署和运维监控等能力的一站式小程序部署服务。小程序云应用是小程序云的一个重要组成部分。小程序云是阿里云面向小程序场景提供的一站式云服务，帮助开发者实现一云多端的业务战略。开发者可通过小程序云支撑各类小程序前端，在一朵云内实现统一的资源管理、统一的数据运营和统一的业务设计。

小程序云服务提供以下功能：

- 资源编排

通过对云服务器（ECS）、云数据库RDS、负载均衡（SLB）和弹性公网IP（EIP）等资源进行编排，根据不同的业务情况提供不同的资源规格配置，自动搭建好符合业务场景的最优运行环境，低成本、高效率。

- 应用托管

为开发者提供主流应用框架的运行环境，可直接上传Spring Boot、Node.js和PHP部署包发布，开箱即用，极大地降低了开发者的开发成本。

- 一站式运维

支持开发者一站完成测试环境、生产环境的版本发布、扩容和资源监控等操作。无需运维即可高效进行线上服务的自动扩容和业务指标监控等。

# 开发前准备

在开始使用阿里云小程序Serverless服务开发支付宝小程序前，您需要确保完成以下准备工作。

## 步骤一：注册阿里云账号

1. 打开[阿里云官网](#)。
2. 单击页面右上角的**免费注册**。
3. 选择一种登录方式，完成阿里云账号注册。

## 步骤二：注册支付宝开发者账号

1. 打开[支付宝开放平台](#)。
2. 单击页面右上角的**立即入驻**。
3. 打开支付宝App，扫描页面上的二维码。
4. 补充用户信息，然后单击**确认加入**，完成开发者入驻。

## 步骤三：创建支付宝小程序应用

参考以下步骤，在蚂蚁金服开放平台创建支付宝小程序：

1. 使用支付宝账号登录[蚂蚁金服开放平台](#)，然后单击**开发者中心**。



2. 选择**小程序应用**，然后单击**创建应用**并根据引导创建一个小程序应用。



3. 在左侧导航栏单击**设置**。

#### 4. 单击开发设置，在开发信息下的接口加签方式区域配置支付宝公钥。

您可以通过RSA密钥生成工具一键生成小程序密钥。接口加签方式有普通公钥和证书公钥两种，主要用于签名和验证。其中私钥由开发者保存，公钥可对外公开，私钥用于签名，公钥用于验证签名。开发者需要保存自己的私钥，通过支付宝提供的公钥进行通信来保证信息传输的完整性以及发送者身份的真实性。详细信息，请参见[生成支付宝RSA密钥](#)。

 **说明：**

- 企业开发者若涉及资金类支出接口接入，必须使用公钥证书模式。
- 个人开发者不涉及到资金类接口，建议使用公钥方式进行加签。

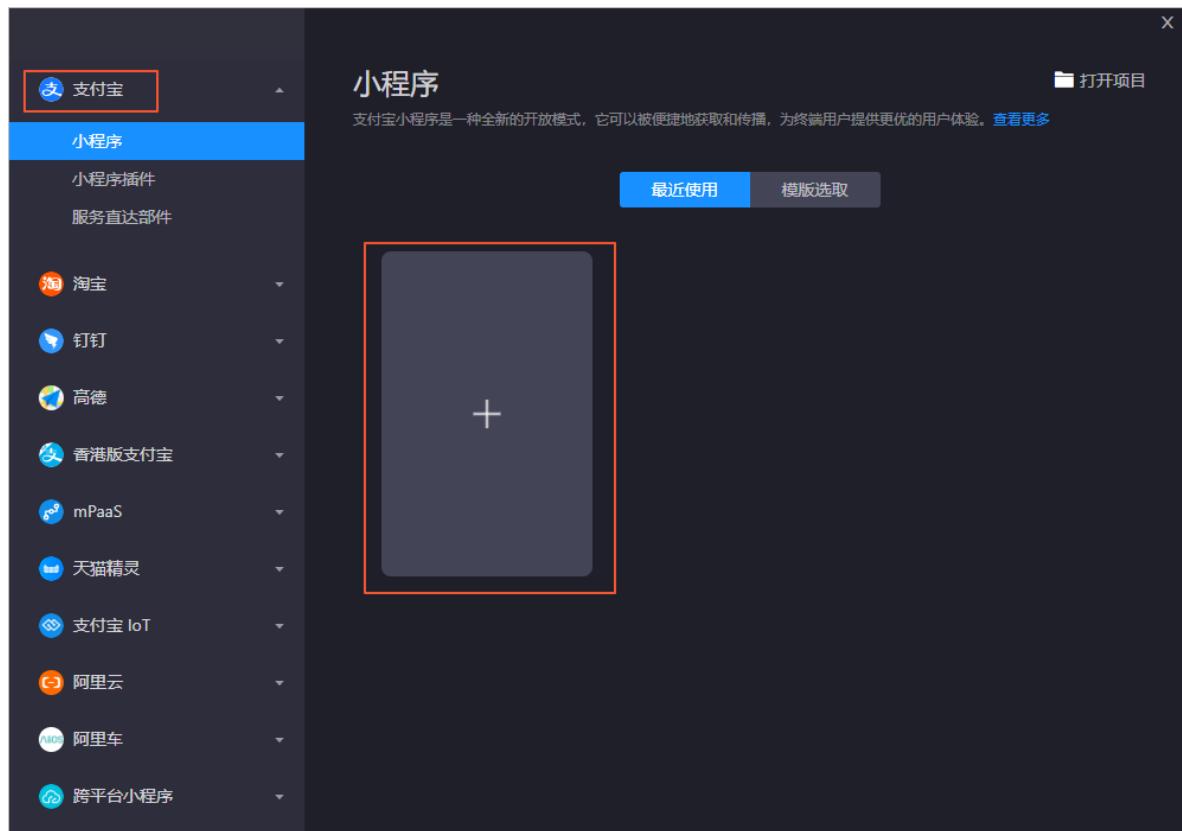


#### 步骤四：安装小程序开发者工具

下载并安装最新版本的小程序开发者工具（1.12以上版本）。详细信息，请参见[下载说明](#)。

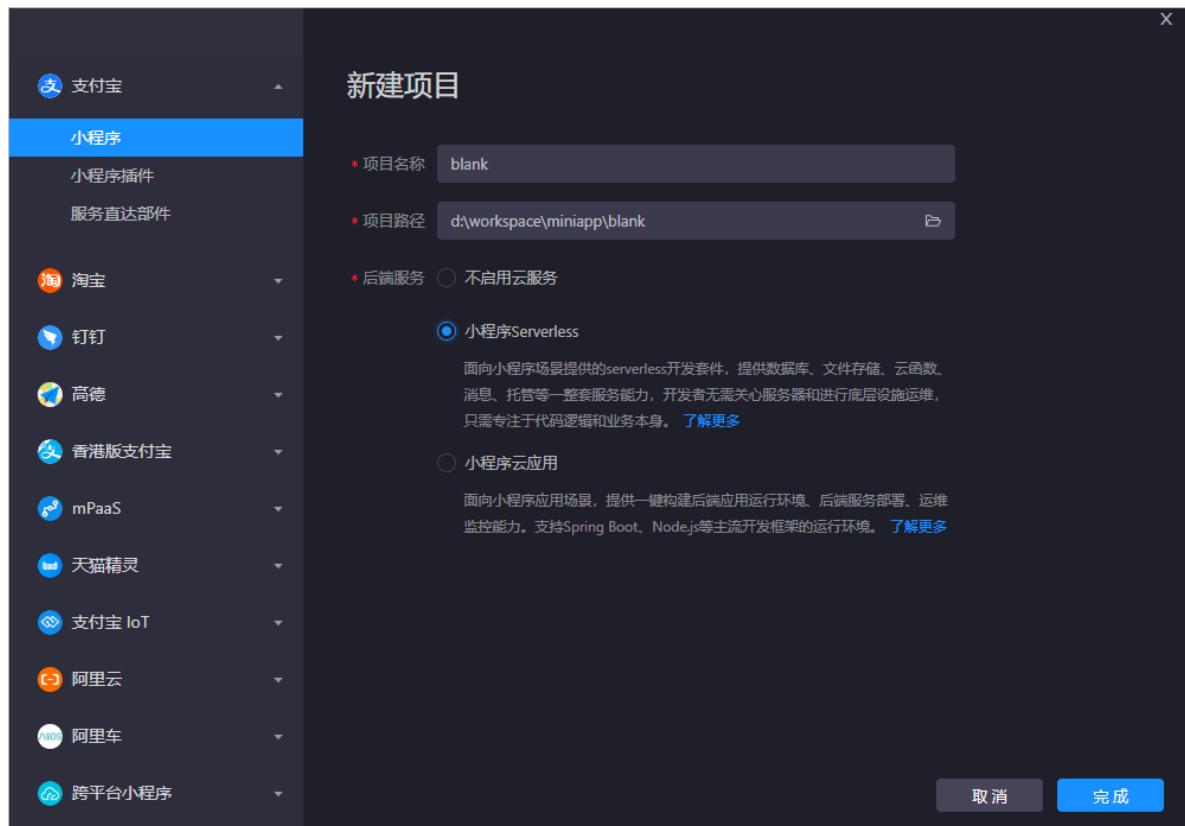
## 步骤五：新建小程序项目

1. 打开小程序开发者工具（IDE），单击新建项目，如下图所示。



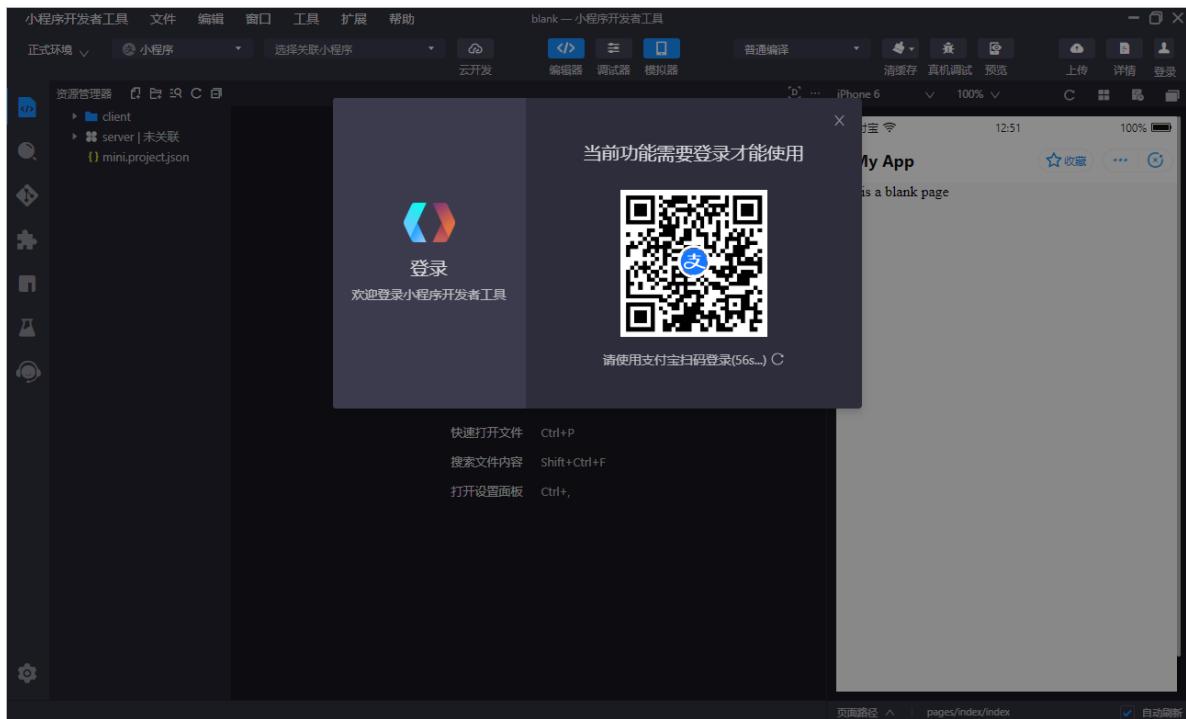
## 2. 填写项目信息。

- a) 填写项目名称，例如blank。
- b) 填写项目路径，例如d:\workspace\miniapp\blank。
- c) 后端服务选择**小程序Serverless**。
- d) 单击**完成**。

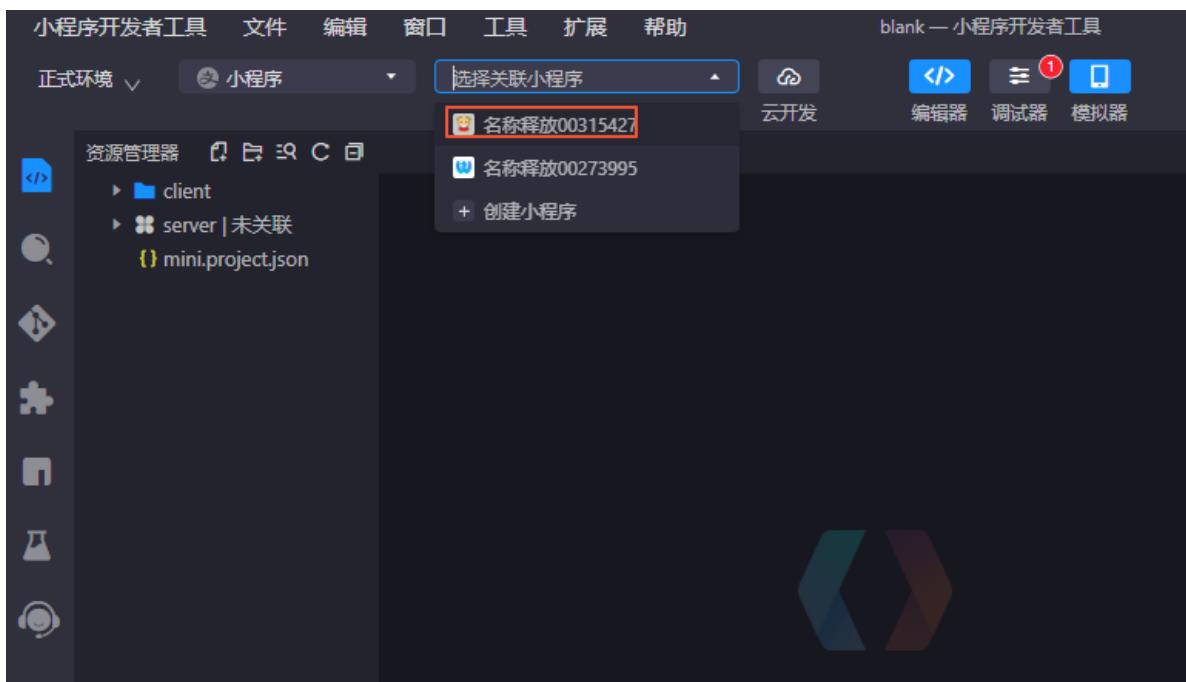


### 3. 登录IDE。

打开支付宝App，扫描页面上的二维码。

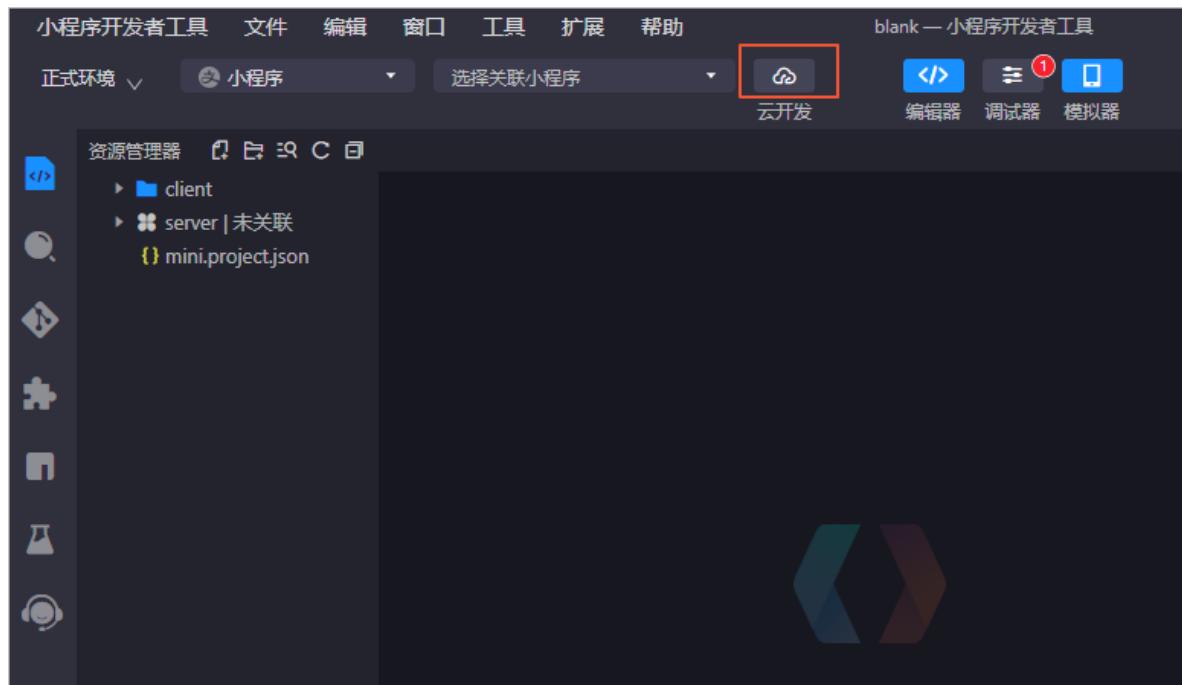


### 4. 选择关联小程序。



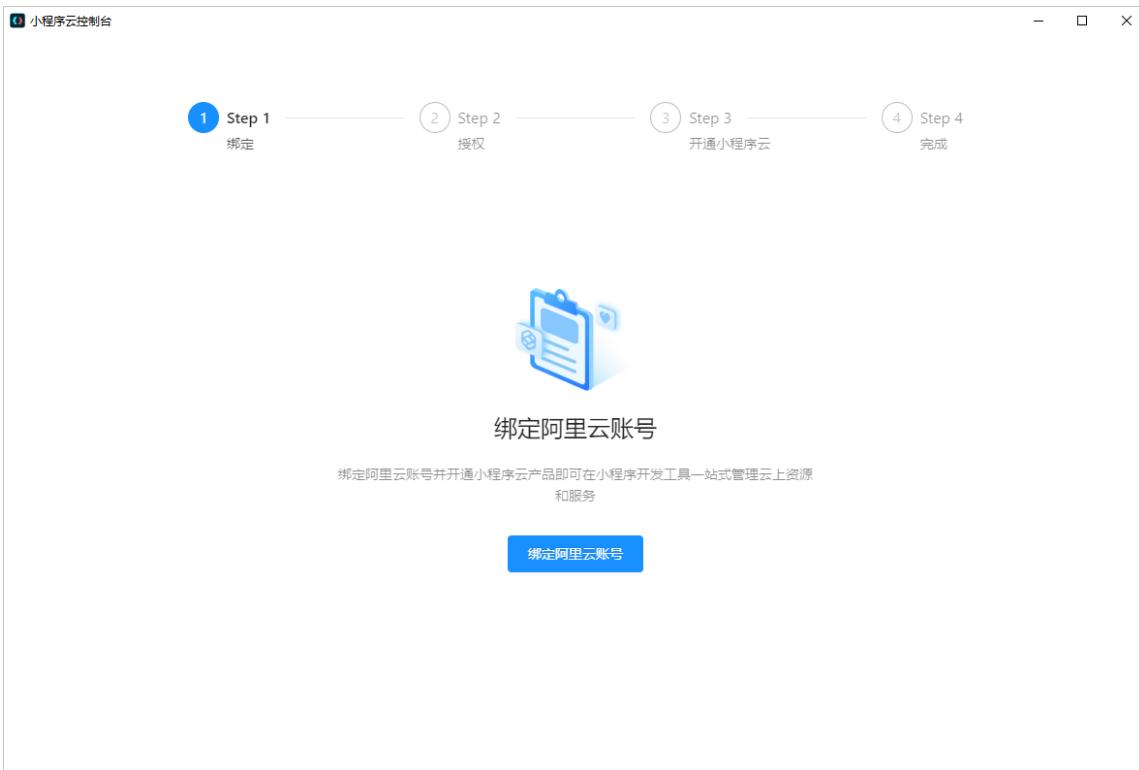
## 步骤六：绑定阿里云账号

1. 在IDE的工具栏中单击**云开发**按钮，打开云控制台。



## 2. 绑定阿里云账号。

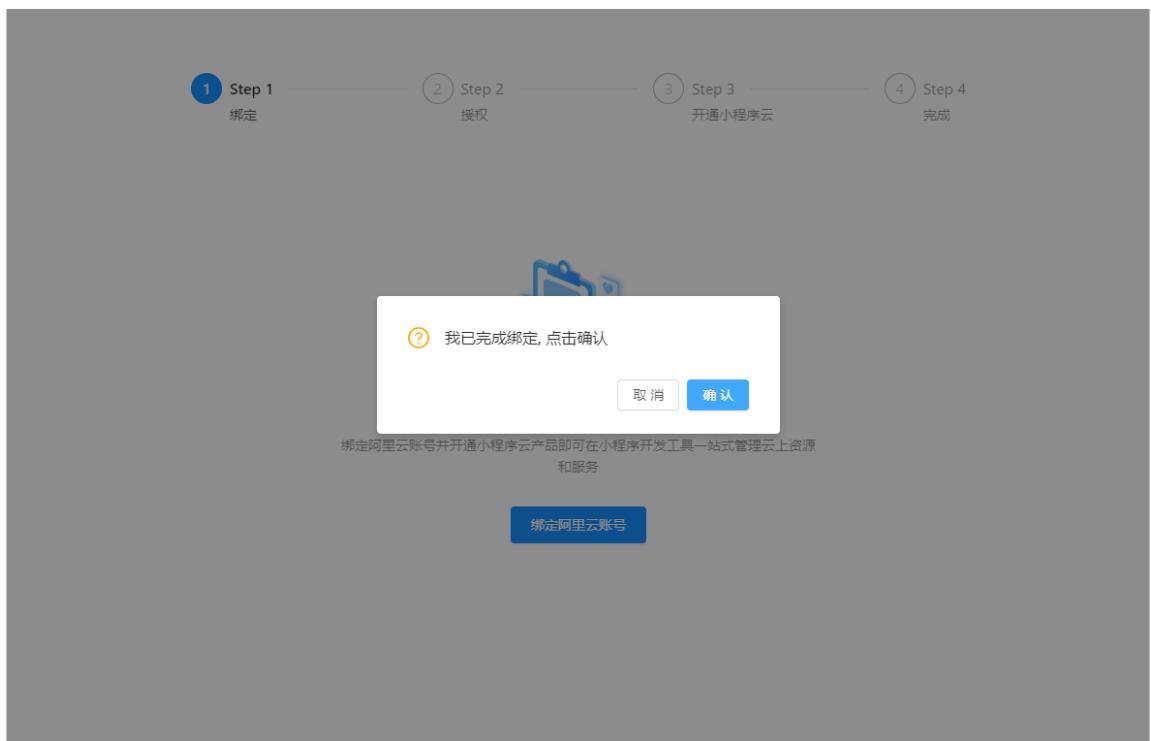
a) 单击**绑定阿里云账号**。



b) 服务授权。在弹出的服务授权页面中，单击《用户授权协议》，阅读后勾选**已阅读和同意**，并**单击授权**。



c) 确认绑定。完成服务授权后，在IDE的小程序云控制台窗口中单击**确认**。



### 3. 主账号授权。

a) 单击前往授权。



#### 主账号授权

小程序开发工具需要主账号授权后才能对云资源环境进行查询、部署等操作，请前往阿里云页面进行授权

[前往授权](#)

b) 在弹出的云资源访问授权页面，单击同意授权。

云资源访问授权

授权服务由访问控制（RAM）提供，将会为您自动创建所需的服务角色并为角色授权。

MPIDE 请求获取访问您云资源的权限

下方是系统创建的可供 MPIDE 使用的角色，授权后，MPIDE 拥有对您云资源相应的访问权限。

**AliyunMPIDETrustLoginRole** [收起详情](#)

描述：小程序开发工具(mini program IDE)使用此角色来访问在其小程序云产品中的资源。  
权限描述：小程序开发工具 (mini program IDE) 使用此角色来访问在其小程序云产品中的资源。

**AliyunMPIDEDefaultRole** [收起详情](#)

描述：小程序开发者工具(MPIDE)默认使用此角色来访问您在其他云产品中的资源  
权限描述：小程序开发者工具（MPIDE）默认使用此角色来访问您在其他云产品中的资源

[同意授权](#) [取消](#)

c) 确认授权。在IDE的小程序云控制台中单击确认。

Step 1 绑定 Step 2 授权 Step 3 开通小程序云 Step 4 完成

我已完成授权, 点击确认

取消 确认

小程序开发工具需要主账号授权后才能对云资源环境进行查询、部署等操作, 请前往阿里云页面进行授权

前往授权

#### 4. 开通小程序云服务。

- a) 单击立即开通。



- b) 在云产品开通页面，单击《小程序云服务协议》链接。



- c) 在阅读后，勾选我已阅读并同意《小程序云服务协议》复选框，然后单击立即开通。

- d) 单击前往RAM进行授权。

## 小程序云应用

! 使用小程序云应用服务需要创建默认角色

使用小程序云应用服务，请先前往RAM进行授权创建默认角色。

[前往RAM进行授权](#)

e) 单击同意授权完成访问授权。



f) 在IDE的小程序云控制台中单击确认。

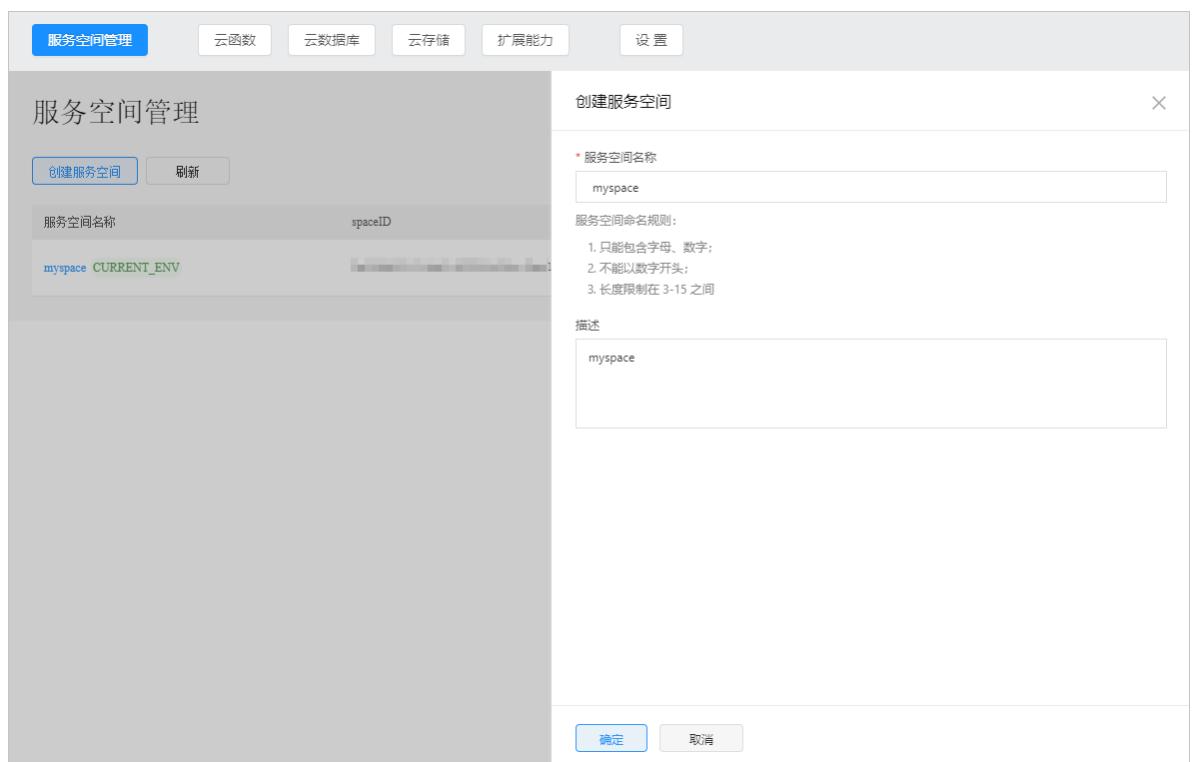


## 5. 单击进入控制台。

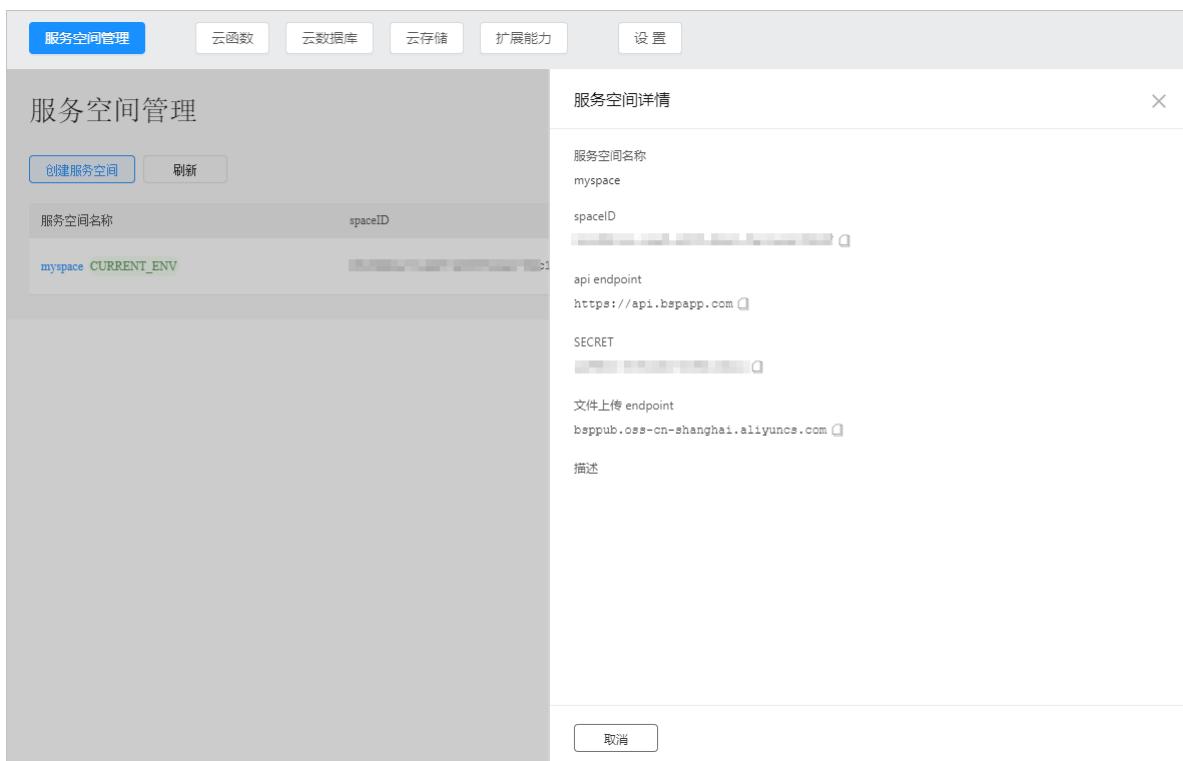


## 步骤七：创建小程序云服务空间

1. 在IDE的小程序云控制台，单击**创建服务空间**，输入服务空间名称，然后单击**确定**。



**2. 创建成功后，单击**详情**查看space ID、secret和endpoint信息。**



## 步骤八：添加支付宝小程序

1. 在IDE的小程序云控制台单击顶部的**设置**。
2. 单击**添加密钥**。

**3. 在添加支付宝密钥**页面填写以下信息，然后单击确定。

- 输入您的小程序应用ID，请登录到[支付宝小程序开发中心](#)查看。
- 输入您的私钥，即步骤三中创建的密钥对中的私钥。
- 输入支付宝小程序配置的公钥。参考以下步骤查看公钥：
  - a. 登录[蚂蚁金服开放平台](#)。
  - b. 在支付宝小程序菜单栏单击[设置](#)，然后单击[开发设置](#)。
  - c. 在[接口加签方式](#)区域，单击[查看](#)。



- d. 复制支付宝公钥信息，然后粘贴到阿里云支付宝小程序的配置中。

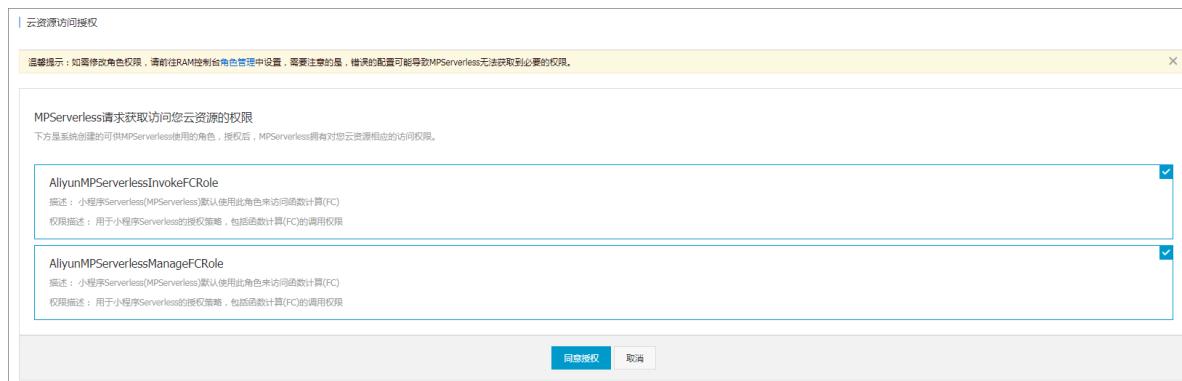
The screenshot shows the 'Signature Management' interface. At the top, there are three tabs: 'Signature Management' (selected), 'Signature Content Configuration' (with a checkmark icon), and 'Signature Configuration Completed' (with a blue circle icon). Below the tabs, there are two sections:

- Application Public Key:** A large area containing a long string of characters (public key) with several lines blurred out. To the right are two buttons: 'Replace Application Public Key' and 'Copy Public Key'.
- Alipay Public Key:** A smaller area containing a long string of characters (public key) with several lines blurred out. To the right is a button labeled 'Copy Public Key'.

At the bottom left is a link 'View Access Document'.

## 步骤九：开通扩展能力

1. 在IDE的小程序云控制台单击顶部的**扩展能力**。
2. 在**扩展能力**页面，单击**立即开通**。
3. 在弹出的对话框，单击前往RAM进行授权，然后单击同意授权允许小程序访问函数计算服务。



4. 返回**扩展能力**页面，查看开通状态。



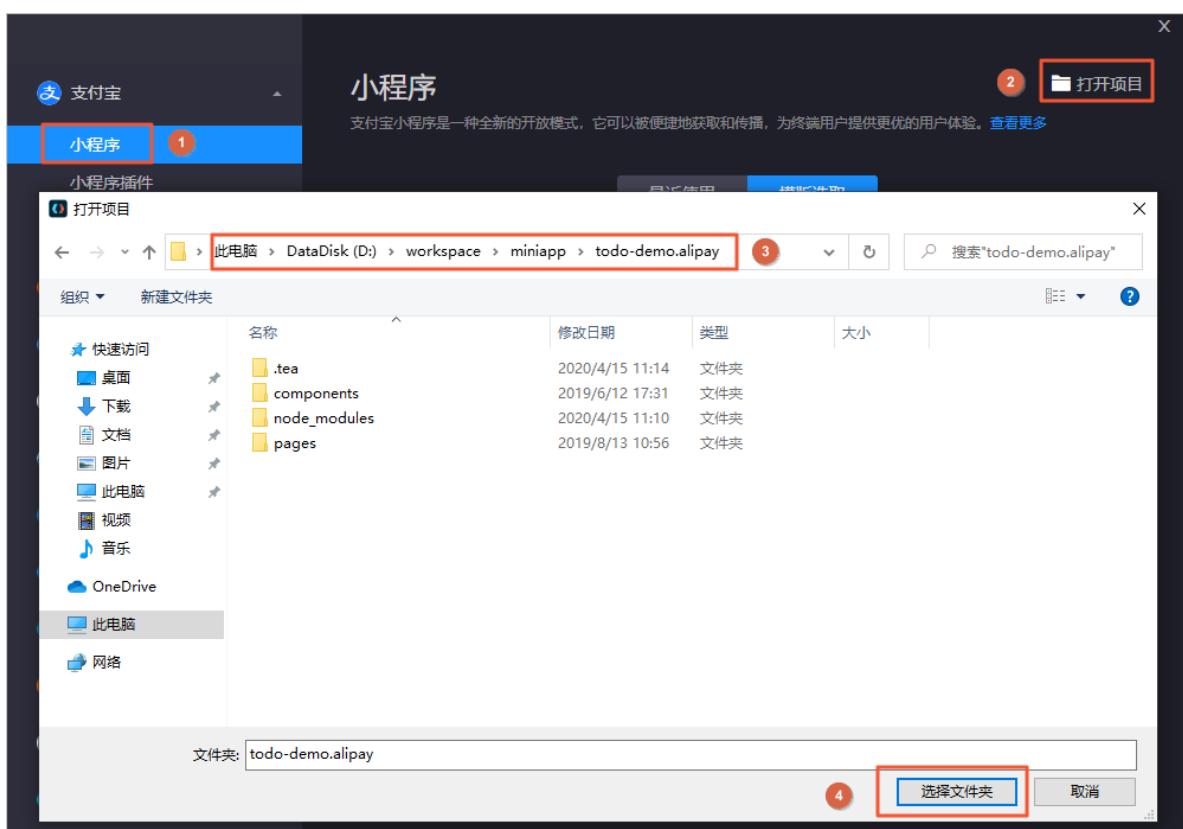
# 开发待办事项小程序

本教程介绍如何使用阿里云小程序Serverless服务开发一个待办事项的支付宝小程序。

## 步骤一：开发支付宝小程序

请参考以下步骤，开发支付宝小程序项目：

1. 单击[这里](#)下载小程序模板代码，并在本地解压。
2. 打开小程序开发者工具，然后选择支付宝 > 小程序。
3. 单击打开项目选择已解压的小程序项目。



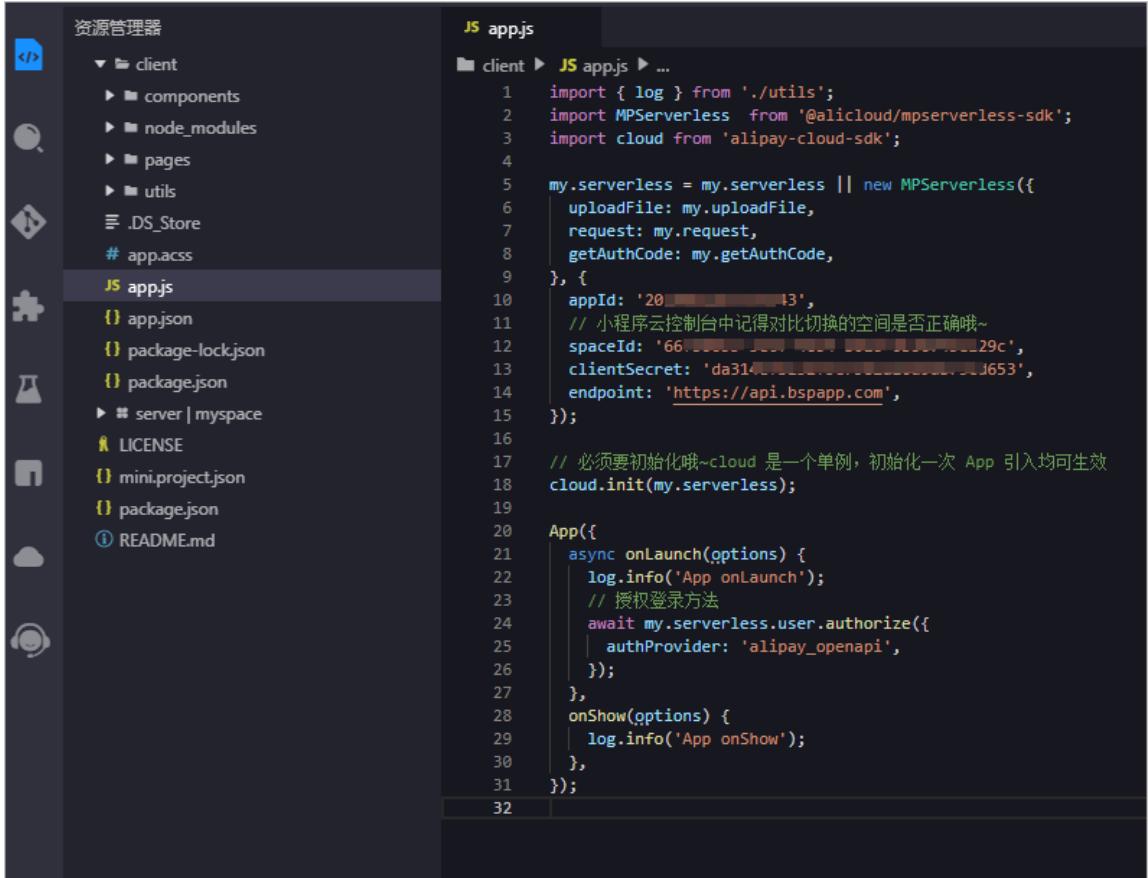
4. 在打开本地项目页面，单击打开。



5. 打开手机支付宝App扫描弹出的登录二维码。登录后，选择已创建的支付宝小程序应用完成关联，然后单击**确定**。
6. 右键单击左侧的server|未关联目录，然后**选择关联Serverless**。

## 7. 在项目文件列表中找到app.js文件，配置以下信息，然后保存。

- **appId**是小程序的ID。您可以在蚂蚁金服开放平台的支付宝小程序页面查看App ID。
- **spaceId、clientSecret和endpoint**在[小程序Serverless控制台](#)创建服务空间后可以获得。

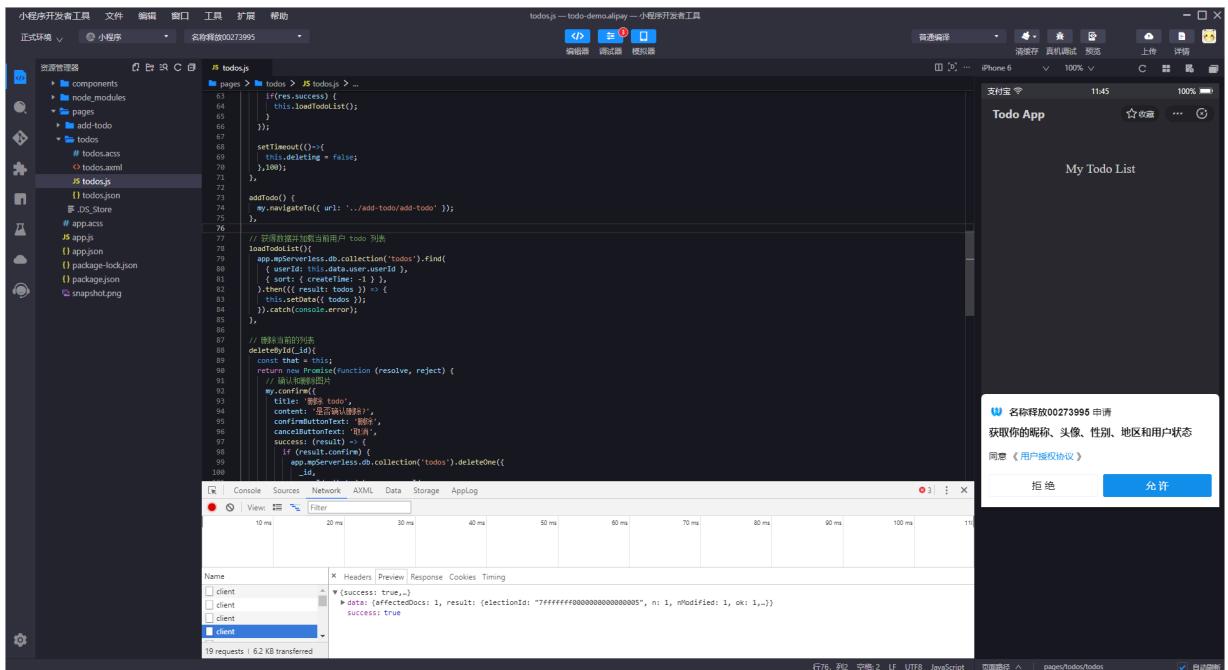


The screenshot shows a code editor interface with a sidebar containing a file tree. The tree includes 'client', 'components', 'node\_modules', 'pages', 'utils', '.DS\_Store', '# app.acss', 'JS app.js' (which is selected), '{} app.json', '{} package-lock.json', '{} package.json', 'server | myspace', 'LICENSE', '{} mini.project.json', '{} package.json', and 'README.md'. The main panel displays the contents of the 'JS app.js' file:

```
JS app.js
client > JS app.js > ...
1 import { log } from './utils';
2 import MPServerless from '@alicloud/mpserverless-sdk';
3 import cloud from 'alipay-cloud-sdk';
4
5 my.serverless = my.serverless || new MPServerless({
6   uploadfile: my.uploadFile,
7   request: my.request,
8   getAuthCode: my.getAuthCode,
9 }, {
10   appId: '20...13',
11   // 小程序云控制台中记得对比切换的空间是否正确哦~
12   spaceId: '66...29c',
13   clientSecret: 'da31...d653',
14   endpoint: 'https://api.bspapp.com',
15 });
16
17 // 必须要初始化哦~cloud 是一个单例，初始化一次 App 引入均可生效
18 cloud.init(my.serverless);
19
20 App({
21   async onLaunch(options) {
22     log.info('App onLaunch');
23     // 授权登录方法
24     await my.serverless.user.authorize({
25       authProvider: 'alipay_openapi',
26     });
27   },
28   onShow(options) {
29     log.info('App onShow');
30   },
31 });
32
```

## 步骤二：调试小程序

打开小程序模拟器，在弹出的服务授权框中仔细阅读《用户授权协议》后选择同意。



## 小程序Serverless代码详解

- SDK初始化

在使用小程序Serverless服务前，需要您在小程序中安装小程序Serverless客户端SDK并初始化。

小程序Serverless客户端SDK的更多信息请参见[安装客户端SDK2.2版本](#)。

```
//app.js
const MPSServerless = require('@alicloud/mpserverless-sdk');
const mpServerless = new MPSServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
},
{
  appId: "", // 小程序应用标识
  spaceId: "", // 服务空间标识
  clientSecret: "", // 服务空间 secret key
  endpoint: "" // 服务空间地址，从小程序Serverless控制台处获得
});

App({
  userInfo: null,
  mpServerless,
});
```

- 页面数据初始化

在页面渲染时进行用户授权并调用user.getInfo接口获取用户ID信息，通过用户ID向云数据库查询当前用户的待办事项列表。更多获取用户信息接口请参见[user.getInfo](#)。

```
async onShow() {
  if (!this.data.user.userId) {
    // 调用user.authorize进行用户授权
    await app.mpServerless.user.authorize({
      authProvider: 'alipay_openapi',
    });
}
```

```
// 调用user.getInfo获取用户信息
const res = await app.mpServerless.user.getInfo().catch(console.error);
if (res.success) {
  this.setData({
    user: {
      userId: res.result.user.userId,
    },
  });
  this.loadTodoList();
},
```

- 获取用户的待办事项列表

调用find方法查询todos数据库中指定用户ID的待办事项信息，并将返回结果按创建时间降序排列。更多数据库查询接口信息请参见[find](#)。

```
// 获得数据并加载当前用户 todo 列表
loadTodoList(){
  app.mpServerless.db.collection('todos').find(
    { userId: this.data.user.userId },
    { sort: { createTime: -1 } },
  ).then(({ result: todos }) => {
    this.setData({ todos });
  }).catch(console.error);
},
```

- 删除待办事项

根据列表ID和用户ID删除待办事项，更新删除接口信息请参见[deleteOne](#)。

```
// 删除当前的列表
deleteById(_id){
  const that = this;
  return new Promise(function (resolve, reject) {
    // 确认和删除图片
    my.confirm({
      title: '删除 todo',
      content: '是否确认删除?',
      confirmButtonText: '删除',
      cancelButtonText: '取消',
      success: (result) => {
        if (result.confirm) {
          app.mpServerless.db.collection('todos').deleteOne({
            _id,
            userId: that.data.user.userId,
          }).then(() => {
            resolve({ success: true });
          }).catch(err => {
            console.error(err);
            reject({ success: false });
          });
        }
      },
    });
});
```

```
},
```

- 完成待办事项

根据列表ID将待办事项的状态更新为已完成，并记录完成时间。更多数据更新接口信息请参见[updateOne](#)。

```
// 根据 id 改变当前 todo 状态
changeComplate(_id, completed){
  return new Promise(function (resolve, reject) {
    app.mpServerless.db.collection('todos').updateOne(
      {_id},
      {
        $set: {
          completed,
          completeTime: completed ? new Date() : false,
        }
      }
    ).then(() => {
      resolve({ success: true });
    }).catch(err => {
      console.error(err);
      reject({ success: false });
    });
  });
}
```

- 新增待办事项

调用insertOne方法向数据库todos中新增一条待办事项。更多新增数据接口信息请参见[insertOne](#)。

```
// 写入数据库 obj, 当前用户增加一条 todo
addTodo(){
  const that = this;
  return new Promise(function (resolve, reject) {
    app.mpServerless.user.getInfo().then((res) => {
      if (res.success) {
        app.mpServerless.db.collection('todos').insertOne({
          text: that.data.inputValue,
          iconUrl: that.data.iconUrl ? that.data.iconUrl : false,
          userId: res.result.user.userId,
          completed: false,
          createTime: new Date(),
          completeTime: false,
        }).then(() => {
          resolve({ success: true });
        }).catch(err => {
          console.error(err);
          reject({ success: false });
        });
      }
    }).catch(err => {
      console.error(err);
      reject({ success: false });
    });
  });
}
```

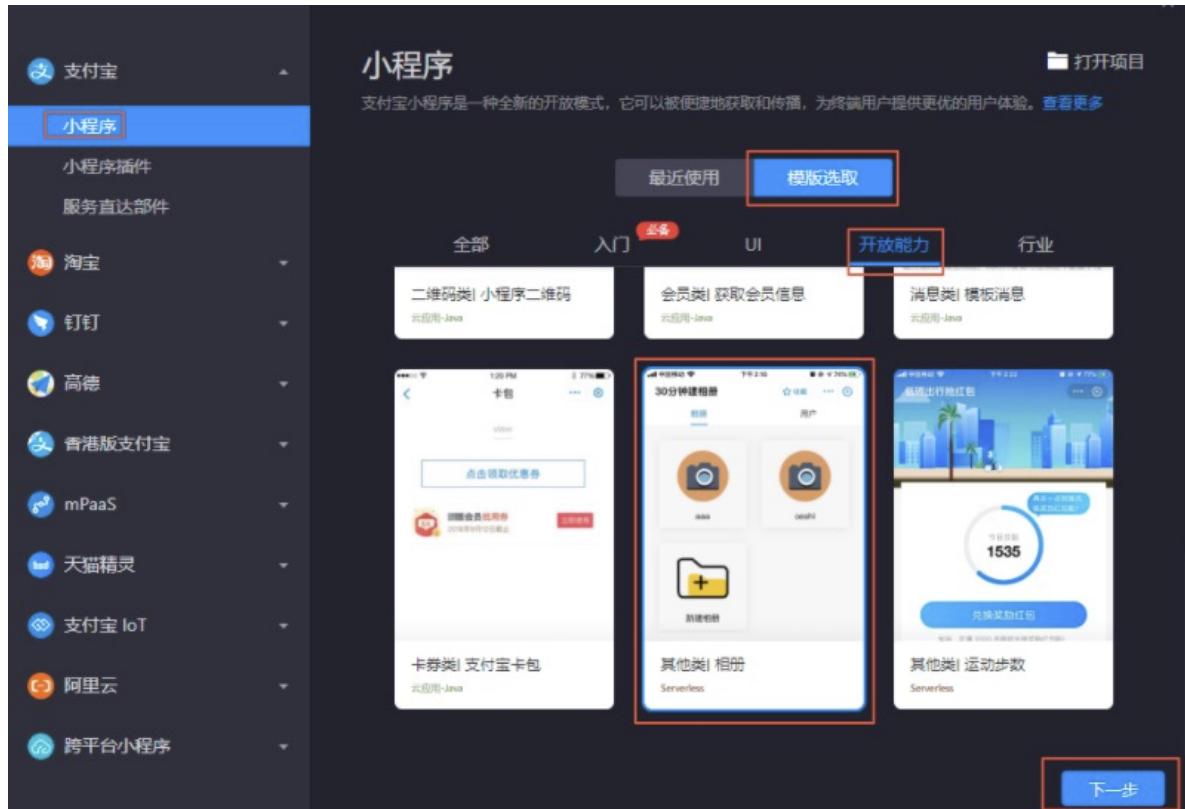
# 搭建个人相册小程序

本教程介绍如何使用小程序Serverless服务开发一款简单的个人相册小程序。图片存储，是所有应用开发里最常见的场景之一。借助小程序云开发能力，可以提升功能开发效率，提高数据隐私保护能力。

## 步骤一：开发支付宝小程序

请参考以下步骤，开发支付宝小程序项目：

1. 打开小程序开发者工具。
2. 在左侧导航栏选择**支付宝 > 小程序**。
3. 单击**模板选取**，然后单击**开放能力**页签，选择**个人相册**模板并单击**下一步**。

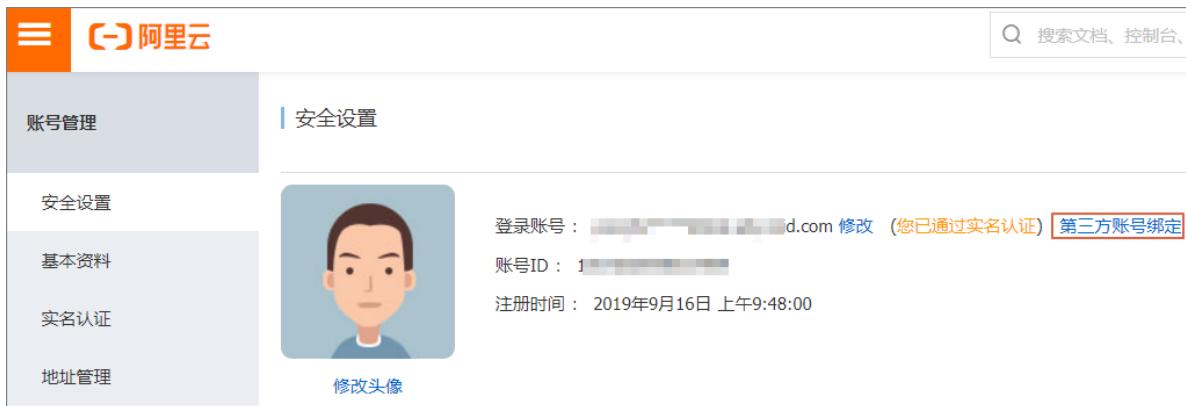


4. 设置项目名称和路径，然后单击**完成**。
5. 打开手机支付宝App扫描弹出的登录二维码。登录后，选择已创建的支付宝小程序应用完成关联，然后单击**确定**。
6. 当提示绑定阿里云账号时，单击**前往绑定阿里云账号**，然后在打开的验证页面选择一种账号验证方式并完成验证。或右键单击左侧的**server|未关联**目录，然后选择**关联Serverless**。



说明：

确保您要绑定的阿里云账号已关联支付宝账号。如果尚未关联，在[阿里云账号](#)页面，单击[第三方账号绑定](#)绑定支付宝。



## 7. 在项目文件列表中找到app.js文件，配置以下信息，然后保存。

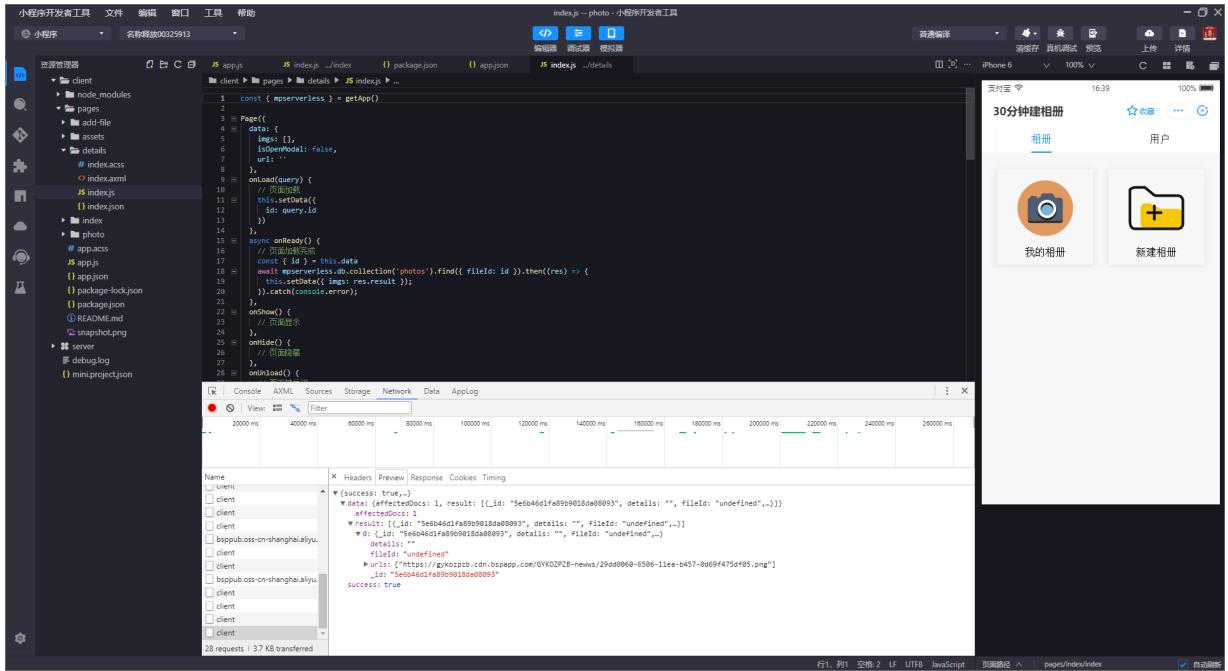
- **appId**是小程序的ID。您可以在蚂蚁金服开放平台的支付宝小程序页面查看App ID。
- **spaceId**、**clientSecret**和**endpoint**在[小程序Serverless控制台](#)创建服务空间后可以获得。

A screenshot of a code editor showing the 'app.js' file in a project structure. The left sidebar shows a file tree with 'client', 'components', 'node\_modules', 'pages', 'utils', '.DS\_Store', '# app.acss', 'JS app.js', 'app.json', 'package-lock.json', 'package.json', 'server | myspace', 'LICENSE', 'mini.project.json', 'package.json', and 'README.md'. The 'JS app.js' file is selected. The code editor shows the following content:

```
client > JS app.js > ...
1 import { log } from './utils';
2 import MPSServerless from '@alicloud/mpsserverless-sdk';
3 import cloud from 'alipay-cloud-sdk';
4
5 my.serverless = my.serverless || new MPSServerless({
6   uploadFile: my.uploadFile,
7   request: my.request,
8   getAuthCode: my.getAuthCode,
9 }, {
10   appId: '2013',
11   // 小程序云控制台中记得对比切换的空间是否正确哦~
12   spaceId: '6629c',
13   clientSecret: 'da314...J653',
14   endpoint: 'https://api.bspapp.com',
15 });
16
17 // 必须要初始化哦~cloud 是一个单例，初始化一次 App 引入均可生效
18 cloud.init(my.serverless);
19
20 App({
21   async onLaunch(options) {
22     log.info('App onLaunch');
23     // 授权登录方法
24     await my.serverless.user.authorize({
25       authProvider: 'alipay_openapi',
26     });
27   },
28   onShow(options) {
29     log.info('App onShow');
30   },
31 });
32
```

## 步骤二：调试小程序

打开小程序模拟器，在弹出的服务授权框中仔细阅读《用户授权协议》后选择同意。



## 小程序Serverless代码详解

- 初始化

在使用小程序Serverless服务前，需要您在小程序中安装小程序Serverless客户端SDK并初始化。

小程序Serverless客户端SDK的更多信息请参见[安装客户端SDK2.2版本](#)。

```
// client/app.js
import MPSServerless from '@alicloud/mpserverless-sdk';

const mpserverless = new MPSServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
}, {
  appId: '', // 小程序应用标识
  spaceId: '', // 服务空间标识
  clientSecret: '', // 服务空间 secret key
  endpoint: '' // 服务空间地址，从小程序Serverless控制台处获得
});
```

- 页面加载时添加授权

在调用小程序Serverless服务前，需要先调用authorize接口请求授权，支付宝小程序的授权请求参数authProvider应为alipay\_openapi，更多authorize接口信息请参见[authorize](#)。

```
// client/app.js
async onSubmit() {
  await mpserverless.user.authorize({
    authProvider: 'alipay_openapi',
    // authType: 'anonymous'
  })
}
```

```
},
```

- 全局使用serverless

在小程序Serverless客户端SDK初始化完成后，可以将实例化的mpserverless对象放入App全局对象中，这样SDK就可以全局使用了。

```
// client/app.js
App({
  mpserverless,
});
```

当需要使用SDK时只需从全局对象中获取mpserverless，就可以调用了。

以下示例实现了在其他文件中调用小程序Serverless的数据库服务，向files集合中插入了一条数据。

```
// 其他文件调用
const { mpserverless } = getApp()
// 添加文件数据
mpserverless.db.collection('files').insertOne( { name: '数据表名称', userId: '数据表ID' })
```

- 读取文件数据

以下代码实现了在页面加载完成后，调用find接口请求数据库，查询当前用户下的相册信息，更多find接口信息请参见[find](#)。

```
async onReady() {
  const result = await mpserverless.db.collection('files').find({ userId: '用户id' })
  this.setData({
    files: result.result || []
  })
},
```

- 选择本地图片

以下代码实现了调用小程序的my.chooseImage接口拍照或从本地相册中选择图片，然后调用file.uploadFile接口将图片上传至小程序Serverless的文件服务中。更多信息请参见[my.chooseImageuploadFile](#)。

```
attach() {
  // 选取照片
  my.chooseImage({
    chooseImage: 1,
    success: res => {
      const path = res.apFilePaths[0];
      const options = {
        filePath: path,
        headers: {
          contentDisposition: 'attachment',
        },
      };
      // 上传图片
      mpserverless.file.uploadFile(options).then((image) => {
        const { imgs } = this.data
      })
    }
  })
},
```

```
        imgs.push(image.fileUrl)
        this.setData({
            imgs,
        });
    }).catch(console.log);
},
});
},
```

- 将图片保存到数据库

以下代码实现了将包括图片的上传路径、图片描述等图片信息保存到云数据库photos集合中。

```
async submit() {
    const obj = { urls: '图片路径', details: '图片描述', fileId: '文件id' }
    await mpserverless.db.collection('photos').insertOne(obj)
    my.navigateBack()
},
```

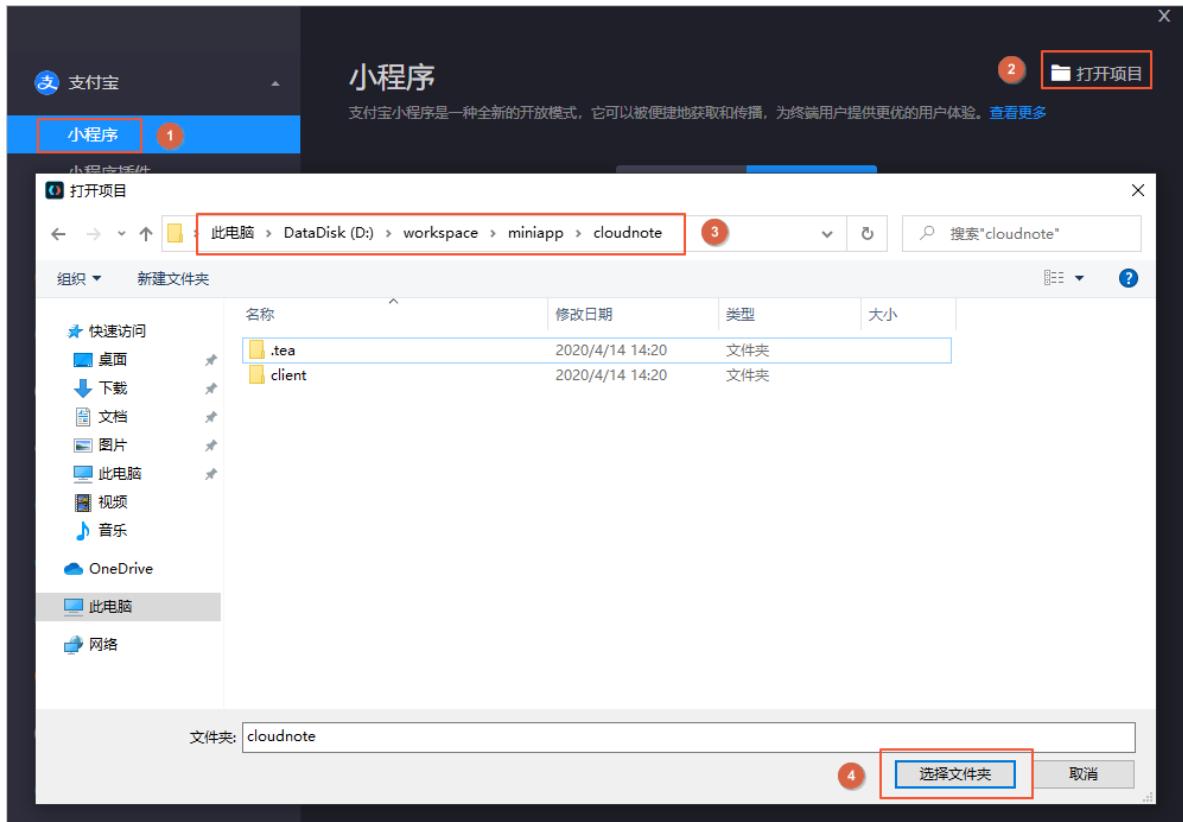
# 快速搭建个人云笔记小程序

本教程介绍如何使用小程序Serverless的数据存储服务开发一款简单的个人云笔记小程序。数据存储服务是基于MongoDB托管在云端的数据库，数据以JSON格式存储。作为开发者，您可以在客户端内直接操作数据，也可以在云函数中读写数据。

## 步骤一：开发支付宝小程序

请参考以下步骤，开发支付宝小程序项目：

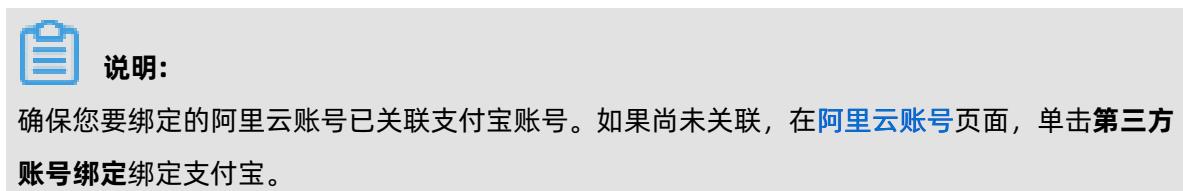
1. 单击[这里](#)下载小程序模板代码，并在本地解压。
2. 打开小程序开发者工具，然后选择支付宝 > 小程序。
3. 单击打开项目选择已解压的小程序项目。



4. 在打开本地项目页面，单击打开。

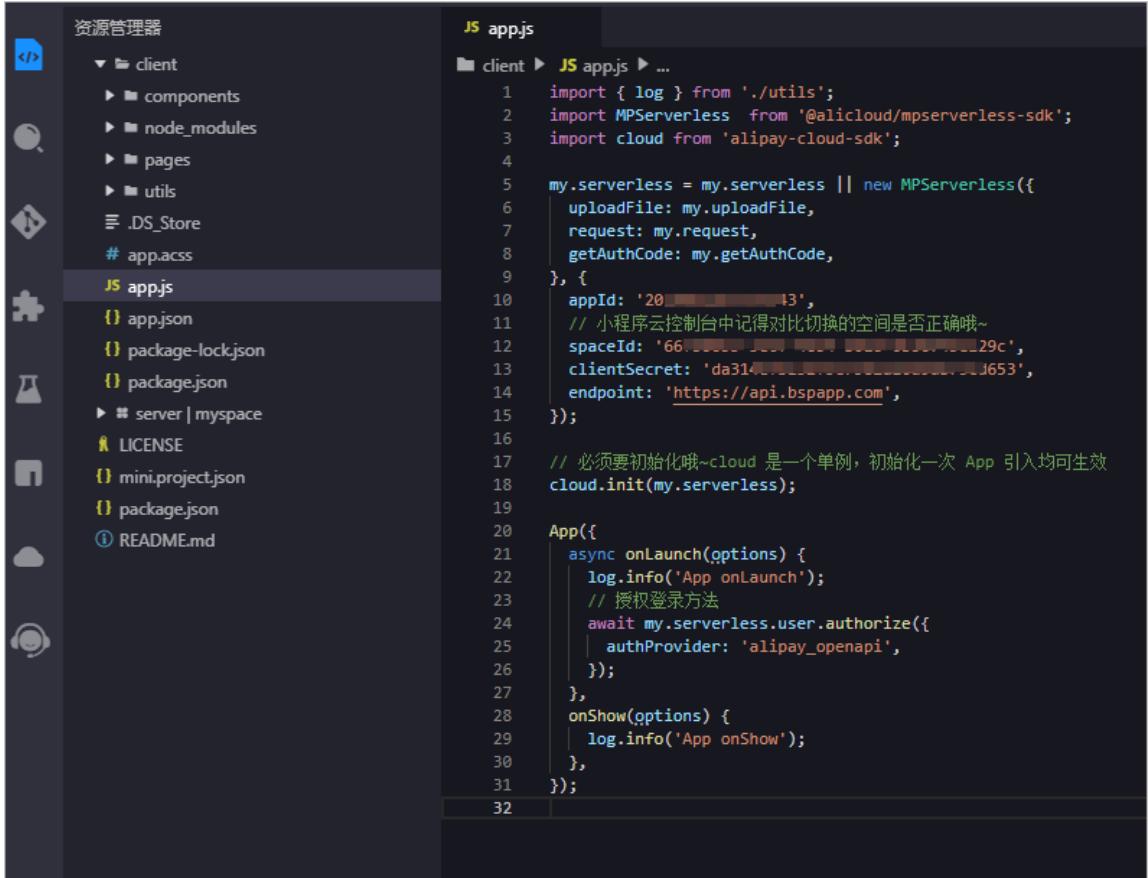


5. 打开手机支付宝App扫描弹出的登录二维码。登录后，选择已创建的支付宝小程序应用完成关联，然后单击**确定**。
6. 当提示绑定阿里云账号时，单击**前往绑定阿里云账号**，然后在打开的验证页面选择一种账号验证方式并完成验证。或右键单击左侧的**server|未关联目录**，然后**选择关联Serverless**。



## 7. 在项目文件列表中找到app.js文件，配置以下信息，然后保存。

- **appId**是小程序的ID。您可以在蚂蚁金服开放平台的支付宝小程序页面查看App ID。
- **spaceId、clientSecret和endpoint**在[小程序Serverless控制台](#)创建服务空间后可以获得。

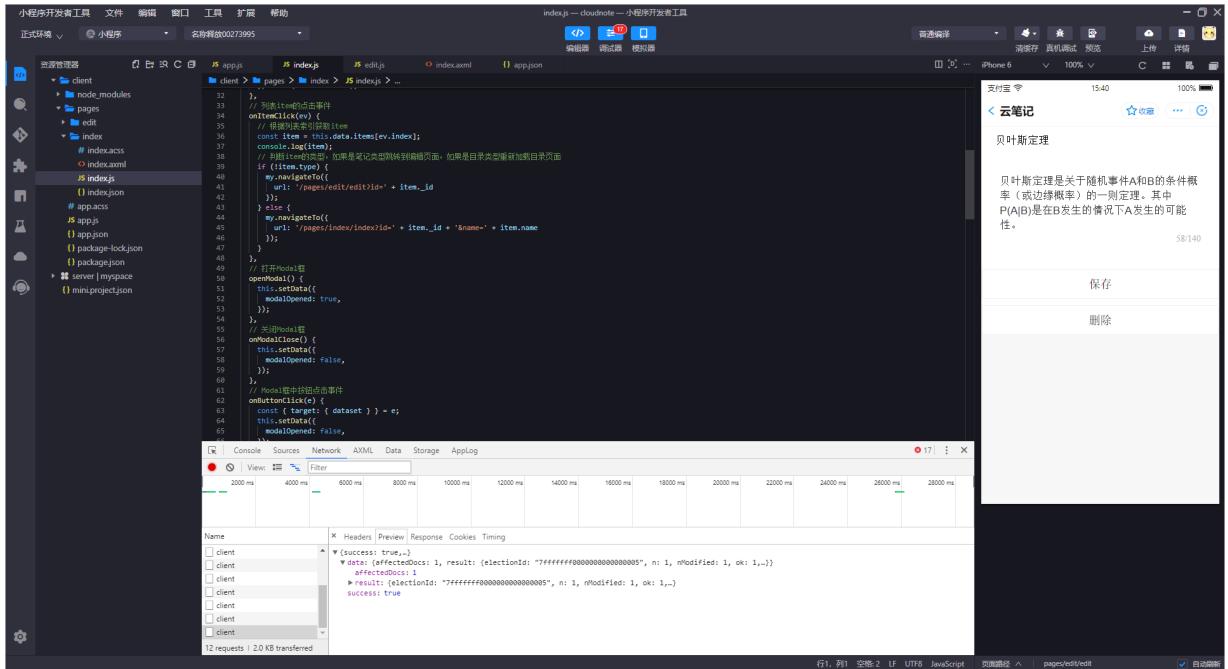


The screenshot shows a code editor with a sidebar containing a file tree. The tree includes 'client', 'components', 'node\_modules', 'pages', 'utils', '.DS\_Store', '# app.acss', 'JS app.js' (which is selected), '{} app.json', '{} package-lock.json', '{} package.json', 'server | myspace', 'LICENSE', '{} mini.project.json', '{} package.json', and 'README.md'. The main panel displays the contents of the 'JS app.js' file:

```
JS app.js
client > JS app.js > ...
1 import { log } from './utils';
2 import MPServerless from '@alicloud/mpserverless-sdk';
3 import cloud from 'alipay-cloud-sdk';
4
5 my.serverless = my.serverless || new MPServerless({
6   uploadfile: my.uploadFile,
7   request: my.request,
8   getAuthCode: my.getAuthCode,
9 }, {
10   appId: '20...13',
11   // 小程序云控制台中记得对比切换的空间是否正确哦~
12   spaceId: '66...29c',
13   clientSecret: 'da31...d653',
14   endpoint: 'https://api.bspapp.com',
15 });
16
17 // 必须要初始化哦~cloud 是一个单例，初始化一次 App 引入均可生效
18 cloud.init(my.serverless);
19
20 App({
21   async onLaunch(options) {
22     log.info('App onLaunch');
23     // 授权登录方法
24     await my.serverless.user.authorize({
25       authProvider: 'alipay_openapi',
26     });
27   },
28   onShow(options) {
29     log.info('App onShow');
30   },
31 });
32
```

## 步骤二：调试小程序

打开小程序模拟器，在弹出的服务授权框中仔细阅读《用户授权协议》后选择同意。



## 小程序Serverless代码详解

- 初始化

在使用小程序Serverless服务前，需要您在小程序中安装小程序Serverless客户端SDK并初始化。

小程序Serverless客户端SDK的更多信息请参见[安装客户端SDK2.2版本](#)。

```
// client/app.js
import MPSServerless from '@alicloud/mpserverless-sdk';

const mpserverless = new MPSServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
}, {
  appId: '', // 小程序应用标识
  spaceId: '', // 服务空间标识
  clientSecret: '', // 服务空间 secret key
  endpoint: '' // 服务空间地址，从小程序Serverless控制台处获得
});
```

- 第一次进入小程序时添加授权

在调用小程序Serverless服务前，需要先调用authorize接口请求授权，支付宝小程序的授权请求参数authProvider应为alipay\_openapi，更多authorize接口信息请参见[authorize](#)。

```
onLaunch(options) {
  // 授权
  mpserverless.user.authorize({
    authProvider: 'alipay_openapi',
    // authType: 'anonymous'
  });
}
```

```
 },
```

- 页面初始化数据

在页面显示时请求数据库note，过滤出parentId是当前目录的ID的节点。更多数据库查询接口信息请参见[find](#)。

```
onShow() {
  mpserverless.db.collection('note').find({ parentId: this.data.parentId }).then((res) =>
  {
    this.setData({ items: res.result });
    console.log(res);
  }).catch(console.error);
},
```

- 新增笔记

以下代码实现了在数据库note中新增笔记，插入成功后刷新列表页。更多insertOne接口信息请参见[insertOne](#)。

```
if (this.data.newNoteName != "") {
  mpserverless.db.collection('note').insertOne({
    name: this.data.newNoteName,
    type: Number(this.data.newNoteType),
    parentId: this.data.parentId,
    content: ""
  }).then(res => {
    console.log(res);
    if (res.success) {
      this.onShow();
    }
  }).catch(console.error);
};
```

- 更新笔记

以下代码实现了根据文档ID异步更新文档内容，更多updateOne接口信息请参见[updateOne](#)。

```
async updateData(currentData) {
  this.setData({
    note: currentData,
  });
  await mpserverless.db.collection('note').updateOne({
    _id: this.data.id
  },
  {
    $set: {
      parentId: currentData.parentId,
      type: 0,
      name: currentData.name,
      content: currentData.content
    }
  })
  .then(() => {})
  .catch(console.error);
};
```

```
 },
```

- 目录删除

以下代码实现了删除目录节点的功能，先查询该节点下是否有子节点，如果没有则执行删除，从而实现只能删除目录树的叶子节点。更多deleteOne接口请参见[deleteOne](#)。

```
// 执行删除操作，只允许删除叶子节点
mpserverless.db.collection('note').find({
  parentId: parentId
}).then((res) => {
  if (res.success && res.affectedDocs == 0) {
    mpserverless.db.collection('note').deleteOne({
      _id: parentId
    }).then((res) => {
      console.log(res);
      my.showToast({
        type: 'success',
        content: '删除成功',
        duration: 3000,
        success: () => {
          my.navigateBack();
        }
      });
    });
  } else {
    my.showToast({
      type: 'fail',
      content: '该目录下存在子节点，请先删除子节点',
      duration: 3000,
      success: () => { },
    });
  };
});
```

# 教你开发获取运动步数小程序

本教程介绍如何使用小程序Serverless服务开发一个支付宝运动小程序。借助运动数据功能，小程序在获得用户许可的情况下，可以获取用户最近30天内的运动步数，步数信息会在用户进入小程序时更新。



## 说明：

本场景的小程序Demo需要使用蚂蚁金服开放平台企业账号。单击[这里](#)进行申请。

## 步骤一：签约支付宝开放能力

参考以下步骤，在蚂蚁金服开放平台签约运动数据能力：

1. 使用支付宝开发者账号登录[蚂蚁金服开放平台](#)开发中心。
2. 在左侧导航栏单击**开发管理**。

功能名称	功能介绍	是否需要签约	状态
小程序个人开发者基础包	小程序个人开发者默认开放的基础功能包	否	未生效

3. 在小程序详情开发管理页面的能力列表中，单击右上角**添加能力**按钮，添加**运动数据**能力。

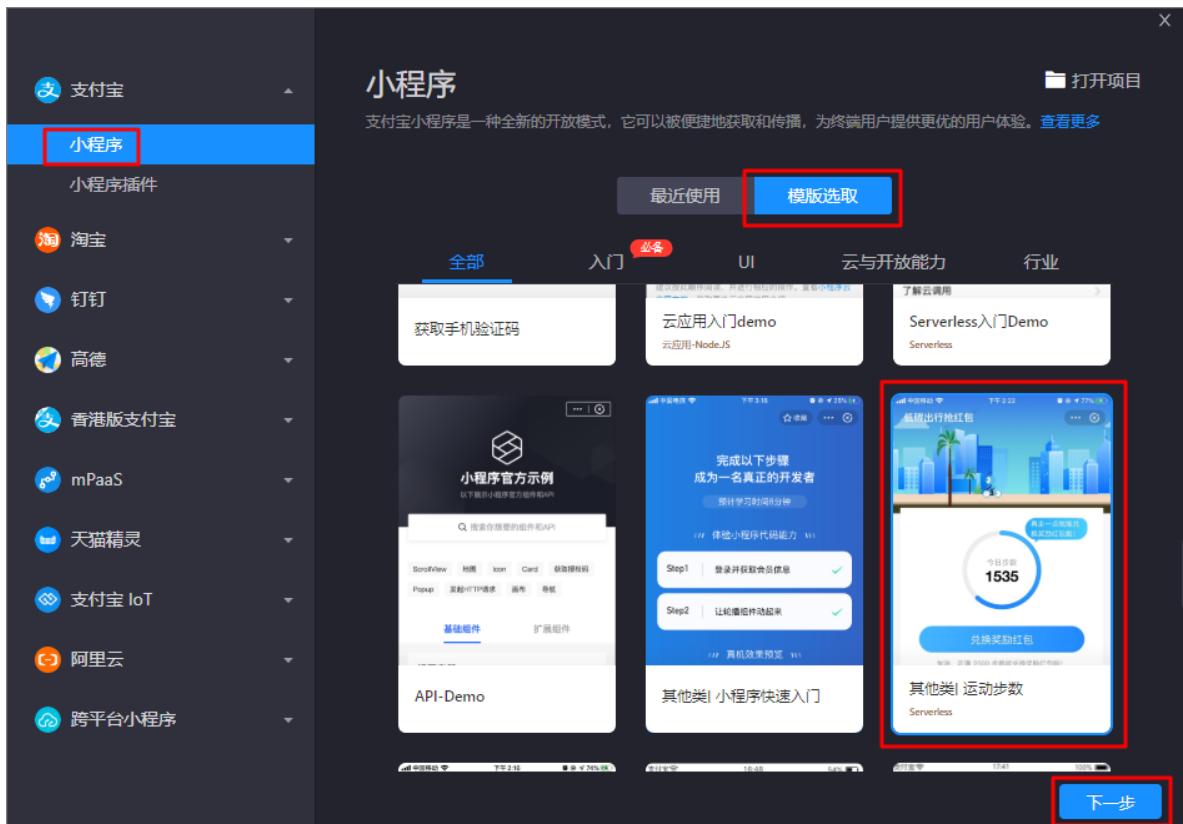
运动数据能力需要签约才能生效，在小程序上线后，单击功能列表右侧**立即签约**。签约完成后，需要1个工作日左右的审批时间（审批结果会以短信和邮件形式告知），审批成功后，功能状态会变为**已生效**。

功能名称	功能介绍	是否需要签约	状态	操作
小程序支付	用户可以在小程序内唤起支付宝收银台进行支付。	需签约	已生效	删除
身份验证	支付宝身份验证基于支付宝客户端的实人认证能力，采用人脸、指纹、证照、姓名和身份证号多因子认证技术快速得出认证结果；主要解决线上实人开户、帐号实名认证、帐号实人登录等场景中个人身份的识别问题。	需签约	已生效	删除
运动数据	借助运动数据功能，小程序可以获取用户最近30天内的运动步数，步数信息会在用户进入小程序时更新。	需签约	已生效	用户信息申请    删除

## 步骤二：开发支付宝小程序

请参考以下步骤，开发支付宝小程序项目：

1. 打开小程序开发者工具。
2. 在左侧导航栏选择**支付宝 > 小程序**。
3. 单击模板选取，然后单击全部页签，选择运动步数并单击下一步。

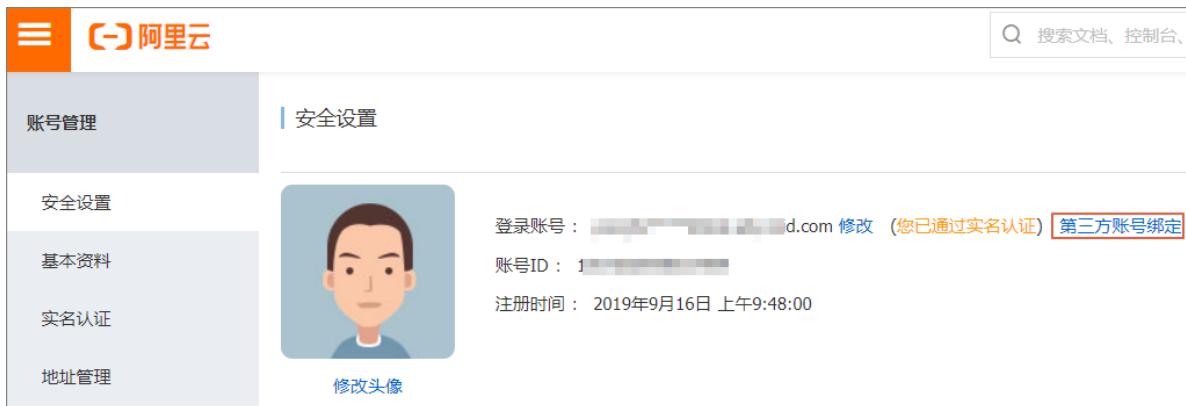


4. 设置项目名称和路径，然后单击完成。
5. 打开手机支付宝App扫描弹出的登录二维码。登录后，选择已创建的支付宝小程序应用完成关联，然后单击确定。
6. 当提示绑定阿里云账号时，单击前往绑定阿里云账号，然后在打开的验证页面选择一种账号验证方式并完成验证。或右键单击左侧的server|未关联目录，然后选择关联Serverless。



说明：

确保您要绑定的阿里云账号已关联支付宝账号。如果尚未关联，在[阿里云账号](#)页面，单击[第三方账号绑定](#)绑定支付宝。



7. 在项目文件列表中找到app.js文件，配置以下信息，然后保存。

- **appId**是小程序的ID。您可以在蚂蚁金服开放平台的支付宝小程序页面查看App ID。
- **spaceId**、**clientSecret**和**endpoint**在[小程序Serverless控制台](#)创建服务空间后可以获得。

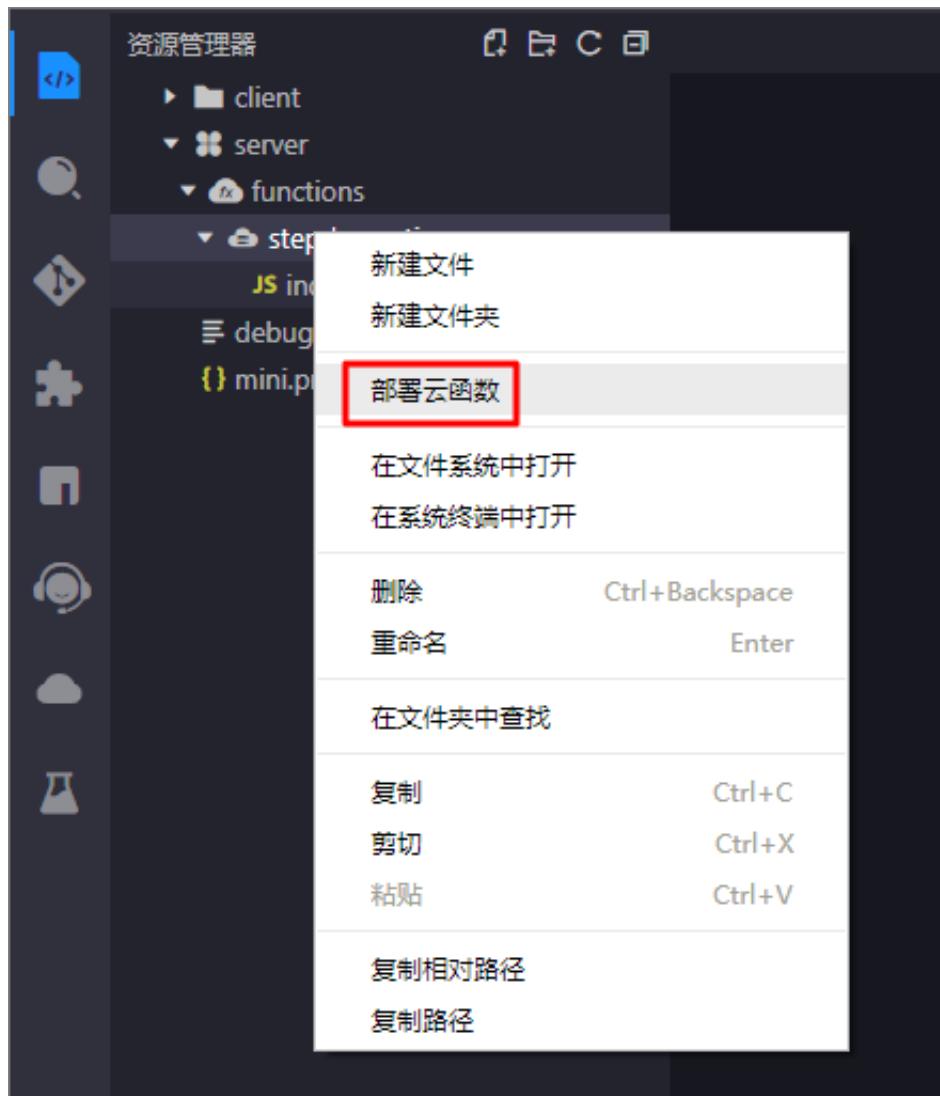
A screenshot of a code editor showing the 'app.js' file in a project structure. The left sidebar shows files like 'client', 'components', 'node\_modules', 'pages', 'utils', '.DS\_Store', '# app.acss', 'JS app.js', 'app.json', 'package-lock.json', 'package.json', 'server | myspace', 'LICENSE', 'mini.project.json', 'package.json', and 'README.md'. The right pane shows the content of 'app.js':

```
1 import { log } from './utils';
2 import MPSServerless from '@alicloud/mpsserverless-sdk';
3 import cloud from 'alipay-cloud-sdk';
4
5 my.serverless = my.serverless || new MPSServerless({
6   uploadFile: my.uploadFile,
7   request: my.request,
8   getAuthCode: my.getAuthCode,
9 }, {
10   appId: '20...13',
11   // 小程序云控制台中记得对比切换的空间是否正确哦~
12   spaceId: '66...29c',
13   clientSecret: 'da314...J653',
14   endpoint: 'https://api.bspapp.com',
15 });
16
17 // 必须要初始化哦~cloud 是一个单例，初始化一次 App 引入均可生效
18 cloud.init(my.serverless);
19
20 App({
21   async onLaunch(options) {
22     log.info('App onLaunch');
23     // 授权登录方法
24     await my.serverless.user.authorize({
25       authProvider: 'alipay_openapi',
26     });
27   },
28   onShow(options) {
29     log.info('App onShow');
30   },
31 });
32
```

8. 修改server/functions/stepdecryption/index.js文件下的aesSecret。

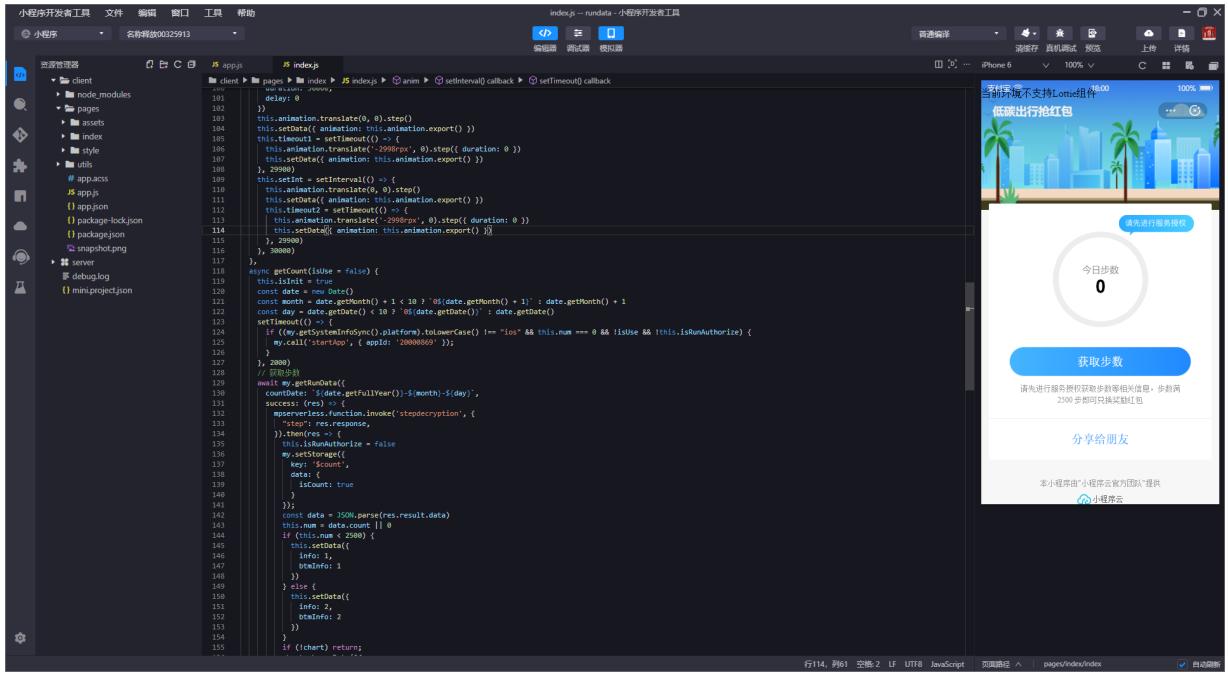
9. 选中server/functions/stepdecryption目录的代码作为云函数进行上传和部署。

- a) 右键单击functions目录，选择新建云函数，输入**stepdecryption**回车。
- b) 右键单击**stepdecryption**，选择**部署云函数**。



### 步骤三：调试小程序

打开小程序模拟器，在弹出的服务授权框中仔细阅读《用户授权协议》后选择同意。



## 小程序Serverless代码详解

- 初始化

在使用小程序Serverless服务前，需要您在小程序中安装小程序Serverless客户端SDK并初始化。

小程序Serverless客户端SDK的更多信息请参见[安装客户端SDK2.2版本](#)。

```
//client/app.js
import MPSServerless from '@alicloud/mpserverless-sdk';

const mpserverless = new MPSServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
},
{
  appId: '', // 小程序应用标识
  spaceId: '', // 服务空间标识
  clientSecret: '', // 服务空间 secret key
  endpoint: '' // 服务空间地址，从小程序Serverless控制台处获得
});
```

- 全局使用mpserverless和授权

在调用小程序Serverless服务前，需要先调用authorize接口请求授权，支付宝小程序的授权请求参数authProvider应为alipay\_openapi，更多信息请参见[authorize](#)。

以下代码将实例化的mpserverless对象放入全局对象中以便全局使用，并在第一次打开小程序时进行服务授权。

```
//client/app.js
App({
  mpserverless,
  async onLaunch(options) {
    // 第一次打开
```

```
        await mpserverless.user.authorize({
          authProvider: 'alipay_openapi',
          // authType: 'anonymous'
        })
      },
    });
  );
```

- 获取步数的加密数据

以下代码调用了my.getRunData支付宝开放能力API，获取用户最近30天内的运动步数，更多信息请参见[my.getRunData](#)。

```
await my.getRunData({
  countDate: `${date.getFullYear()}-${month}-${day}`,
  success: (res) => {
    console.log(res.response)
  },
  fail: (res) => {
  },
  complete: (res) => {
  },
});
});
```

- 调用Serverless云函数传入相关参数进行解密

以下代码调用了云函数stepdecryption将my.getRunData接口的返回结果进行解密。

```
mpserverless.function.invoke('stepdecryption', {
  "step": res.response,
})
```

- 云函数解密代码

以下为云函数stepdecryption代码，先将密文进行base64解码，然后使用aes-128-cbc算法解密，得到解密后的utf8明文。

```
'use strict';

const crypto = require('crypto');
const aesSecret = ""; // AES密钥

module.exports = async (ctx) => {
  const step = ctx.args.step;
  if (!step) {
    return {
      success: false,
      error: {
        code: 'InvalidParameter',
        message: '待解密部署不能为空'
      }
    }
  }
  try {
    ctx.logger.info('[args]', ctx.args);
    const crypted = Buffer.from(step, 'base64').toString('binary');
    const key = Buffer.from(aesSecret, 'base64');
    const iv = Buffer.alloc(16, 0);
    const decipher = crypto.createDecipheriv('aes-128-cbc', key, iv);
    let decoded = decipher.update(crypted, 'binary', 'utf8');
```

```
decoded += decipher.final('utf8');

return {
  success: true,
  data: decoded
}
} catch (e) {
  ctx.logger.error(e);
  return {
    success: false,
    error: {
      code: 'DecipheStepFail',
      message: e.message
    }
  }
}
```

# 云上开发花呗支付小程序

本文将介绍如何使用小程序Serverless云调用开发一个具有花呗支付能力的小程序。

 **说明:**

- 本场景的小程序Demo需要使用蚂蚁金服开放平台企业账号。单击[这里](#)进行申请。
- 本场景中的小程序Demo涉及到资金类接口，必须使用公钥证书方式进行接口加签。具体操作请参考[生成密钥](#)。

## 步骤一：签约支付宝开放能力

参考以下步骤，在蚂蚁金服开放平台签约小程序支付能力：

1. 使用支付宝开发者账号登录[蚂蚁金服开放平台](#)开发中心。
2. 在左侧导航栏单击**开发管理**。



The screenshot shows the 'Ability List' page in the Alipay Open Platform developer management center. On the left, there's a sidebar with links for 'Development Management', '小程序信息' (小程序 Information), and 'Grey Management'. The main area has a title 'Ability List' with a 'Add Ability' button and a note 'You have added 1 item'. Below is a table with columns: 'Function Name', 'Function Description', 'Is Signature Required', and 'Status'. One row is shown: '小程序个人开发者基础包' (Alipay Personal Developer Basic Package) with '否' (No) under '是否需要签约' and '未生效' (Not Effective) under '状态'.

3. 在小程序详情开发管理页面的能力列表中，单击右上角**添加能力**按钮，添加**小程序支付**能力。

花呗支付能力需要签约才能生效，在小程序上线后，单击功能列表右侧**立即签约**。签约完成后，需要1个工作日左右的审批时间（审批结果会以短信和邮件形式告知），审批成功后，功能状态会变为**已生效**。

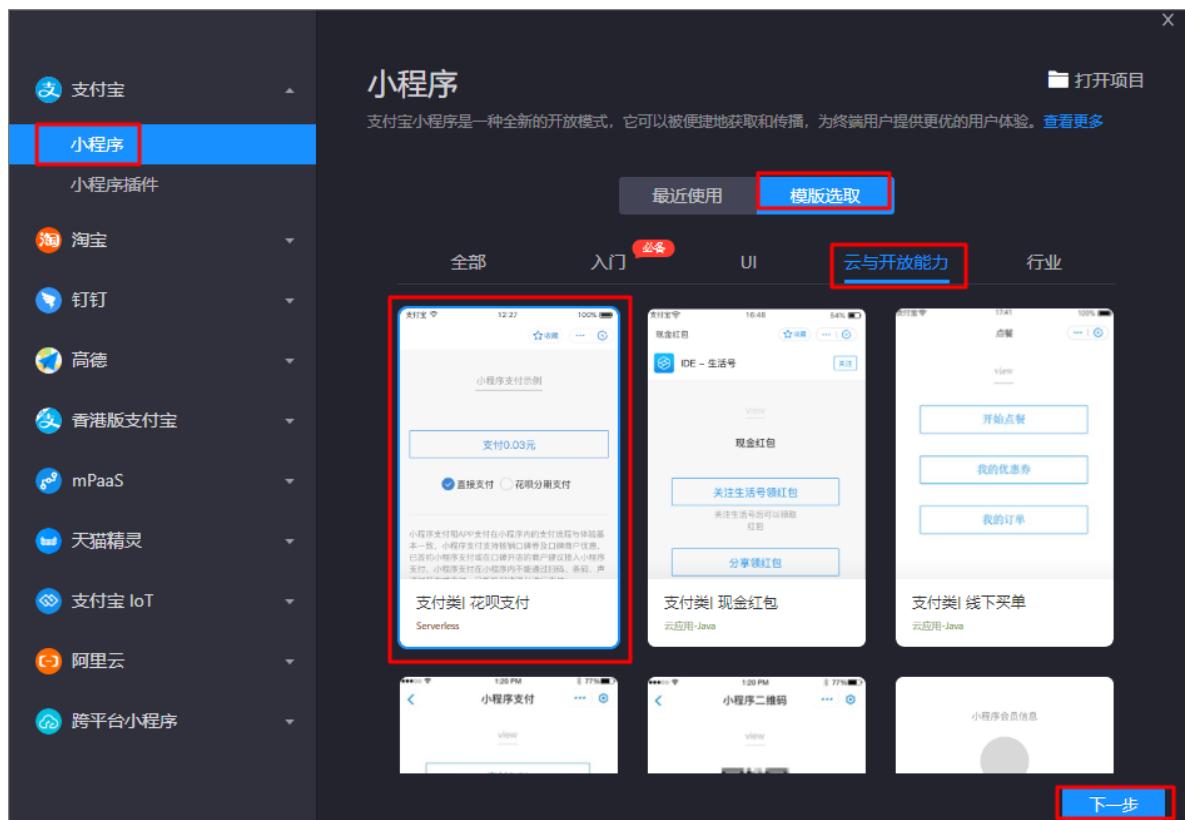


The screenshot shows the 'Ability List' page again, but now with the '小程序支付' (Alipay Mini Program Payment) row highlighted by a red border. This row contains the description 'User can trigger Alipay's cashier function within the mini program for payment.' and the status '已生效' (Effective). Other rows include '身份验证' (Identity Verification) which is also highlighted with a red border, showing its description and status.

## 步骤二：开发支付宝小程序

请参考以下步骤，开发支付宝小程序项目：

- 单击[这里](#)下载并安装小程序开发者工具。
- 打开小程序开发者工具。
- 在左侧导航栏选择**支付宝 > 小程序**。
- 单击**模板选取**，然后单击云与开放能力页签，选择**花呗支付**并单击**下一步**。



- 设置项目名称和路径，然后单击**完成**。
- 打开手机支付宝App扫描弹出的登录二维码。登录后，选择已创建的支付宝小程序应用完成关联，然后单击**确定**。
- 当提示绑定阿里云账号时，单击**前往绑定阿里云账号**，然后在打开的验证页面选择一种账号验证方式并完成验证。或右键单击左侧的**server|未关联**目录，然后选择**关联Serverless**。



**说明：**

确保您要绑定的阿里云账号已关联支付宝账号。如果尚未关联，在[阿里云账号](#)页面，单击[第三方账号绑定](#)绑定支付宝。



#### 8. 在项目文件列表中找到app.js文件，配置以下信息，然后保存。

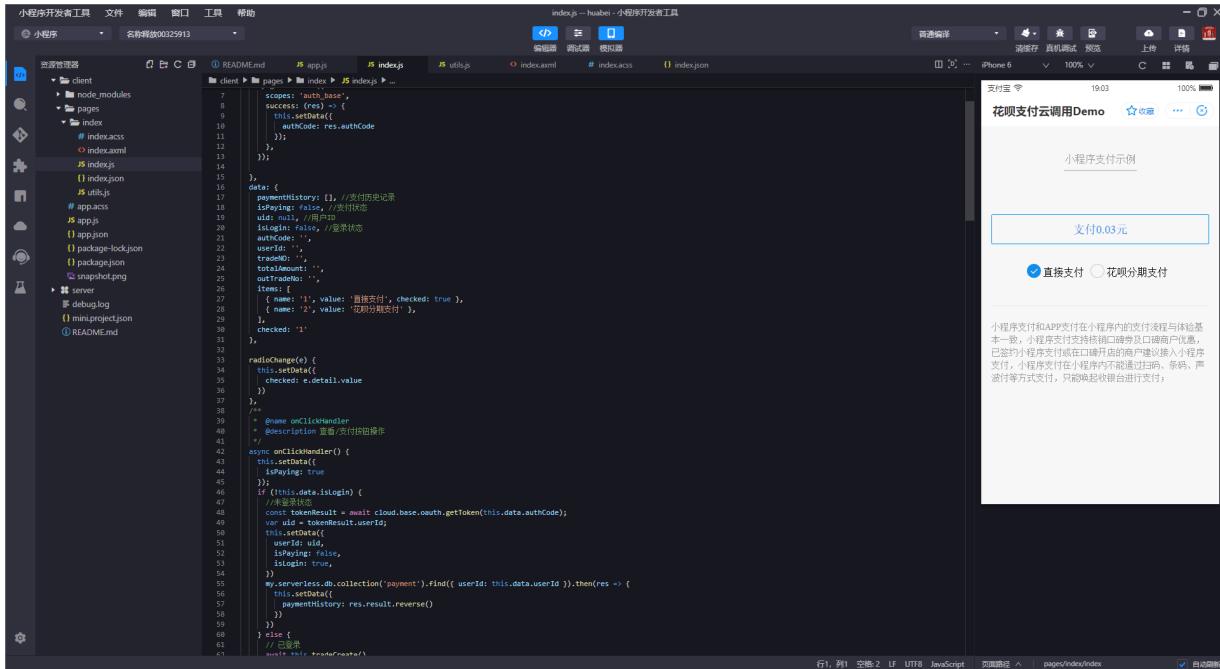
- **appId**是小程序的ID。您可以在蚂蚁金服开放平台的支付宝小程序页面查看App ID。
- **spaceId**、**clientSecret**和**endpoint**在[小程序Serverless控制台](#)创建服务空间后可以获得。

A screenshot of a code editor showing the 'app.js' file in a project structure. The left sidebar shows a file tree with 'client', 'components', 'node\_modules', 'pages', 'utils', '.DS\_Store', '# app.acss', 'JS app.js', 'app.json', 'package-lock.json', 'package.json', 'server | myspace', 'LICENSE', 'mini.project.json', 'package.json', and 'README.md'. The 'JS app.js' file is selected and shown in the main editor area. The code contains configuration for a Serverless application, including the App ID, Space ID, Client Secret, and Endpoint.

```
1 import { log } from './utils';
2 import MPSServerless from '@alicloud/mpsserverless-sdk';
3 import cloud from 'alipay-cloud-sdk';
4
5 my.serverless = my.serverless || new MPSServerless({
6   uploadFile: my.uploadFile,
7   request: my.request,
8   getAuthCode: my.getAuthCode,
9 }, {
10   appId: '2013',
11   // 小程序云控制台中记得对比切换的空间是否正确哦~
12   spaceId: '66129c',
13   clientSecret: 'da314...J653',
14   endpoint: 'https://api.bspapp.com',
15 });
16
17 // 必须要初始化哦~cloud 是一个单例，初始化一次 App 引入均可生效
18 cloud.init(my.serverless);
19
20 App({
21   async onLaunch(options) {
22     log.info('App onLaunch');
23     // 授权登录方法
24     await my.serverless.user.authorize({
25       authProvider: 'alipay_openapi',
26     });
27   },
28   onShow(options) {
29     log.info('App onShow');
30   },
31 });
32
```

#### 步骤三：调试小程序

打开小程序模拟器，在弹出的服务授权框中仔细阅读《用户授权协议》后选择同意。



## 小程序云Serverless代码详解

- 初始化Serverless SDK

在使用小程序Serverless服务前，需要您在小程序中安装小程序Serverless客户端SDK并初始化。

小程序Serverless客户端SDK的更多信息请参见[安装客户端SDK2.2版本](#)。

```
// client/app.js
import MPSServerless from '@alicloud/mpserverless-sdk';

const mpserverless = new MPSServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
},
{
  appId: '', // 小程序应用标识
  spaceId: '', // 服务空间标识
  clientSecret: '', // 服务空间 secret key
  endpoint: '' // 服务空间地址，从小程序Serverless控制台处获得
});
```

- 初始化云调用SDK并授权登录

要使用小程序云提供的扩展能力，您需要先在控制台开通云调用功能，然后在小程序中安装云调用SDK并进行初始化。

在调用小程序Serverless服务前，需要先调用authorize接口请求授权，支付宝小程序的授权请求参数authProvider应为alipay\_openapi，更多信息请参见[authorize](#)。

```
// client/app.js
import cloud from 'alipay-serverless-sdk';

cloud.init(my.serverless);
App({
  async onLaunch(options) {
```

```
    var res = await my.serverless.user.authorize({
      authProvider: 'alipay_openapi',
    });
  },
});
```

- 创建交易

以下代码调用payment.common.create和cloud.payment.huabei.create接口实现了创建直接付款订单和花呗分期付款交易能力，更多直接付款和花呗分期接口信息请参见[payment.common.create](#)和[payment.huabei.create](#)。

```
async tradeCreate() {
  let createRes
  if (this.data.checked === '1') {
    // 直接付款
    createRes = await cloud.payment.common.create('小程序Serverless支付测试', "demo" + new Date().getTime(), '0.03', this.data.userId);
  } else {
    // 花呗分期
    var huabeiConfig = new Object({
      hbFqNum: "3",
      hbFqSellerPercent: "100"
    });
    createRes = await cloud.payment.huabei.create('小程序Serverless支付测试', "demo" + new Date().getTime(), '0.03', this.data.userId, huabeiConfig);
  }
  this.setData({
    tradeNO: createRes.tradeNo,
    outTradeNo: createRes.outTradeNo
  })
  return createRes.tradeNo
},
```

- 付款

以下代码调用my.tradePay支付宝开放能力API唤起支付宝收银台进行支付，更多小程序支付信息请参见[小程序支付](#)。

```
async onRepeatPayHandler(data) {
  my.tradePay({
    tradeNO: this.data.tradeNO,
    success: async (res) => {
      if (res.resultCode === "9000") {
        my.alert({
          title: '支付成功',
          content: res.resultCode,
        });
        const value = data.currentTarget.dataset.item
        const { key } = value
        const { paymentHistory } = this.data
        paymentHistory.map(v => {
          if (v.key === key) {
            v.tradeStatus = 'TRADE_SUCCESS'
          }
        })
        my.serverless.db.collection('payment').updateOne({
          key
        }, {
```

```
        $set: {
          tradeStatus: 'TRADE_SUCCESS'
        })
      })
    this.setData({
      paymentHistory
    })
  },
},
});
```

- 退款

以下代码调用payment.common.refund云调用接口进行退款操作，更多信息请参见[payment.common.refund](#)。

```
async onRefundPayHandler(data) {
  const value = data.currentTarget.dataset.item
  const { outTradeNo, totalAmount, key } = value
  const rs = await cloud.payment.common.refund(outTradeNo, String(totalAmount))
  if (rs.code === "10000") {
    my.showToast({
      content: '退款成功',
    });
    const { paymentHistory } = this.data
    paymentHistory.map(v => {
      if (v.key === key) {
        v.tradeStatus = 'TRADE_CLOSED'
      }
    })
    this.setData({
      paymentHistory
    })
    my.serverless.db.collection('payment').updateOne({
      key
    }, {
      $set: {
        tradeStatus: 'TRADE_CLOSED'
      }
    })
  } else {
    my.showToast({
      content: '退款失败',
    });
  }
  console.log(rs)
},
```



查看小程序云官方文档



阿里云开发者“藏经阁”  
海量免费电子书下载