

# 小程序Serverless 云开发

7天学会支付宝小程序开发  
云上开发小程序从入门到精通



- **5** 大核心技术+数据分析方法
- **7** 天云上一站式开发全掌握



阿里云开发者社区



更多独家电子书



阿里云开发者“藏经阁”  
海量免费电子书下载

加入钉群



获取更多资讯

加入钉群



与更多开发者交流

# I 目录

实现小程序云开通	4
如何操作云数据库?	12
如何使用云存储?	17
如何使用云函数	20
如何使用云调用?	27
教你一个云开发技巧	30
实现你的小程序统计分析	34

## 实现小程序云开通

### 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，您可以可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。

- 云调用

云调用是基于小程序 Serverless 的云函数来使用支付宝小程序开放接口的能力，可以方便的让开发者在小程序中直接调用支付宝的后端开放接口，不需要关注服务端的相关配置，极大的减少了接入的流程，进一步降低了支付宝小程序的开发门槛。

目前云调用涵盖了支付宝开放的基础能力、营销能力、会员能力、支付能力、安全能力，以及阿里云开放的短信能力。资金能力也已经开放使用。

## 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，您需要开通小程序 Serverless 服务并完成支付宝小程序开发者认证。

### 注册支付宝开发者账号

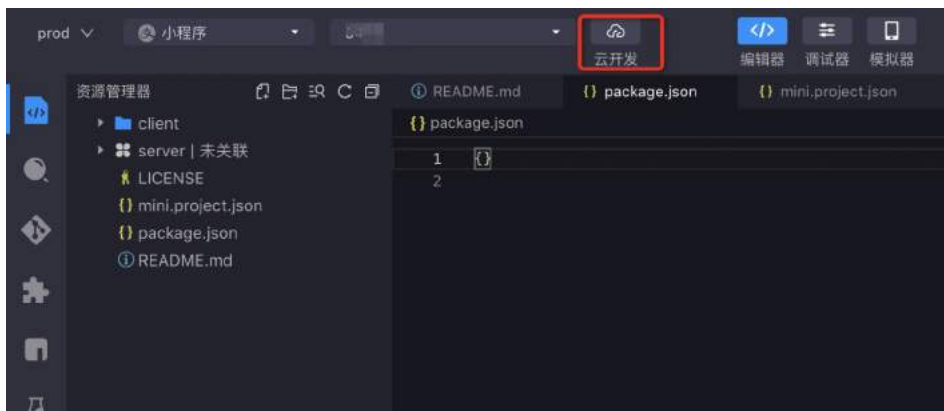
1. 打开[支付宝开放平台](#)。
2. 单击页面右上角的立即入驻。
3. 打开支付宝 App，扫描页面上的二维码。
4. 补充用户信息，然后单击确认加入，完成开发者入驻。

## 注册阿里云账号

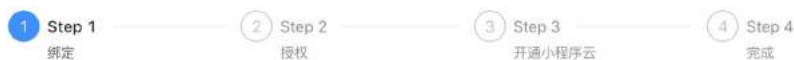
1. 打开[阿里云官网](#)。
2. 单击页面右上角的免费注册。
3. 选择一种登录方式，完成阿里云账号注册并完成实名认证。

## 开通小程序 Serverless 服务

1. 下载[小程序开发工具](#)。
2. 新建或打开小程序 Serverless 工程。
3. 打开小程序云控制台。



#### 4. 绑定阿里云账号。



#### 绑定阿里云账号

绑定阿里云账号并开通小程序云产品即可在小程序开发工具一站式管理云上资源和服务

绑定阿里云账号

#### 5. 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

## 5. 开通小程序云。



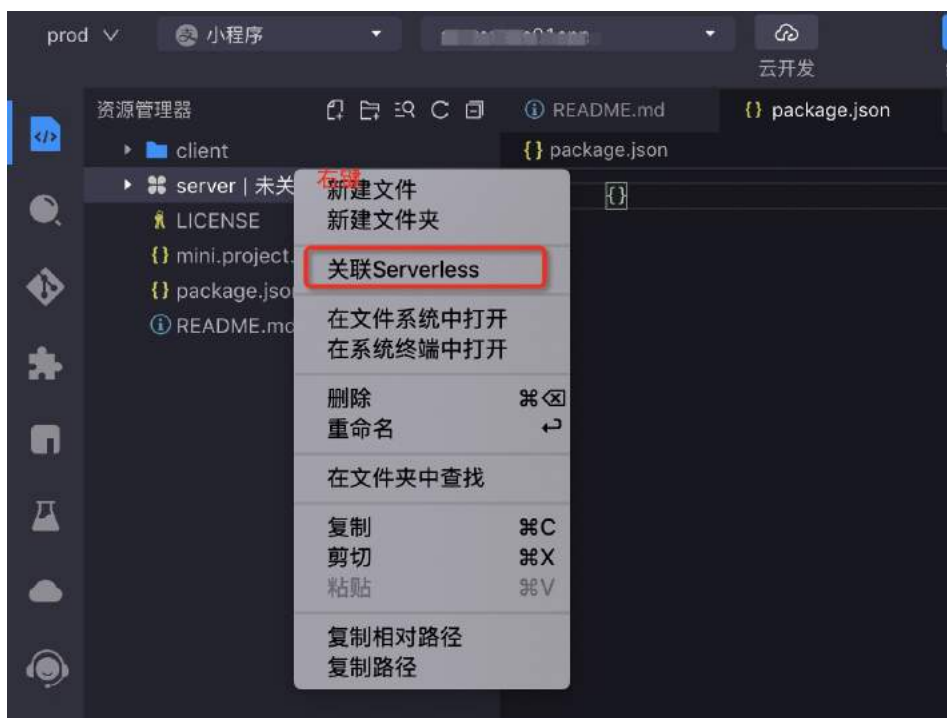
## 创建服务空间

## 1. 在云控制台新建服务空间。

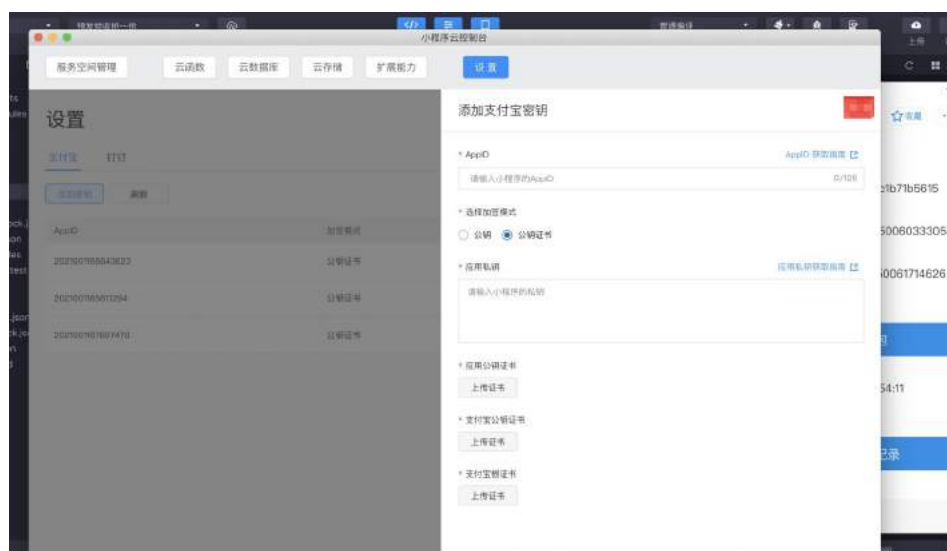
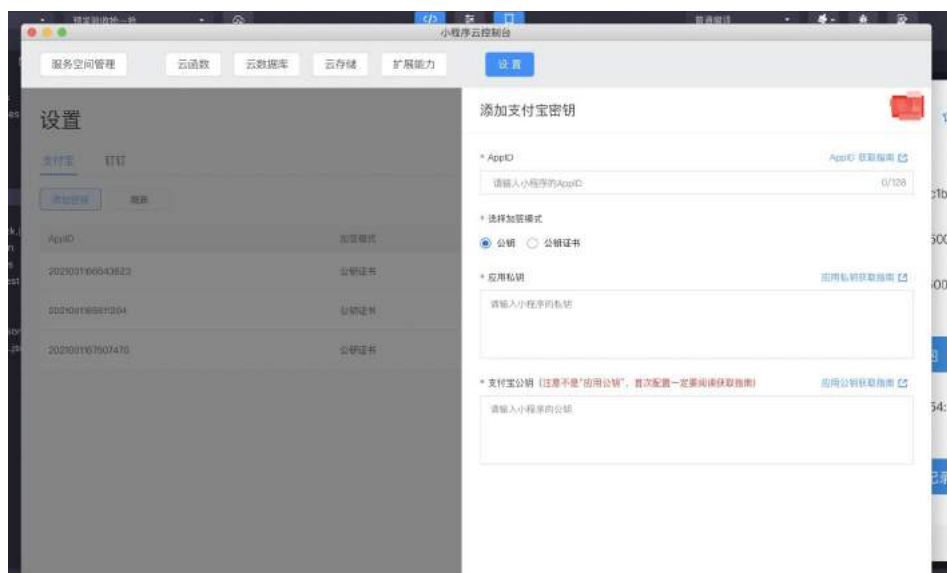




2. IDE 工程关联对应的服务空间。



3. 在云控制台的设置里设置支付宝小程序应用的 AppID 和密钥信息。最新的云控制台支持两种开放平台的加签模式：公钥模式和公钥证书模式。需要注意的是，如果要通过云调用使用**发放红包**，**转账**等资金类接口，就要使用公钥证书模式，将支付宝开放平台上应用对应的支付宝应用公钥证书，支付宝公钥证书以及支付宝根证书按指示上传。



## 项目配置

1. 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。



## 如何操作云数据库？


### 云数据库使用

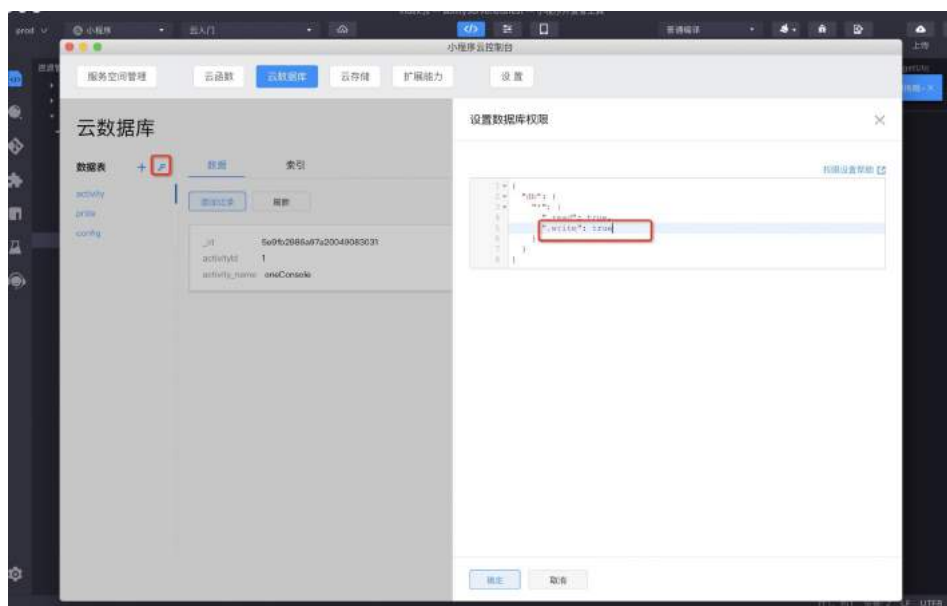
云数据库（数据存储服务）是基于 MongoDB 托管在云端的数据库，数据以 JSON 格式存储。作为开发者，您可以在客户端内直接操作数据，也可以在云函数中读写数据。

#### 步骤一：设置数据表权限

小程序 Serverless 提供了一套简明易懂的 JSON 语法用来控制用户对资源的访问，类似于身份验证体系里的 IAM 或者网络安全中的 ACL。您可以通过修改权限规则来控制数据表权限。

小程序 Serverless 会为每个新建的数据表自动提供一个默认权限规则。默认新建数据表无任何操作权限，您可以通过添加权限规则管理数据表的权限。


1. 打开小程序云控制台，在[云数据库](#)页面，单击权限图标 。
2. 在[设置数据库权限](#)页面，设置权限规则。然后直接修改安全规则。
3. 修改完成后，单击**确定**。



## 步骤二：数据库操作

### 创建数据表

小程序 Serverless 服务使用的是分布式文件存储数据库 MongoDB，以 JSON 格式存储数据。数据库中的每条记录都是一个 JSON 格式的对象，一个数据库可以包含多个集合（相当于关系型数据库中的表）。您可以在控制台上创建数据表。

1. 打开小程序云控制台，在[云数据库](#)页面，选择要关联的环境。
2. 单击添加图标，输入数据表名称，最后单击**确定**。

### 小程序端调用

1. 在小程序项目的根目录执行以下命令安装 SDK。

```
npm install @alicloud/mpserverless-sdk --save
```

2. 在 app.js 中初始化 SDK（全局只需初始化一次）。

```
// 1. 引入必要的 sdk
import MPServerless from '@alicloud/mpserverless-sdk';

// 2. 在 app.js 中对 sdk 进行初始化
// 2.1 初始化 MPServerless
my.serverless = my.serverless || new MPServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
}, {
  // 2.2 参数能在小程序云服务空间详情中获取
  appId: '',
  spaceId: '',
  clientSecret: '',
  endpoint: ''
});
```

### 3. 在小程序页面操作数据库

```
// 1. 添加一条记录
my.serverless.db.collection('users').insertOne({
  name: 'tom',
  age: 1
})
.then(res => {})
.catch(console.error);
// 2. 删除一条记录
my.serverless.db.collection('users')
.deleteOne({
  name: 'Tom'
})
.then((res) => {
  const hasDeleted = res.affectedDocs > 0;
})
.catch(console.error);
// 3. 更新集合中的一条记录
my.serverless.db.collection('users').updateOne({
  name: 'jerry'
}, {
  $set: {
    age: 10
  }
})
.then(res => {})
.catch(console.error);
// 4. 查询一条记录
// 调用 findOne 方法查询大于且最接近 18 岁的一条记录的 name。
```

```
// projection: { name: 1 } 表示只返回 name 字段, 其中 1 表示过滤, 0 表示排除。
// sort: { age: 1 } 表示将查询后的结果按升序排列, 其中 1 表示升序, -1 表示降序。
my.serverless.db.collection('users')
  .findOne({
    age: { $gt: 18 }
  }, {
    projection: { name: 1 },
    sort: { age: 1 }
  })
  .then(res => { })
  .catch(console.error);
```

## 云函数端调用

```
// 添加一条记录
module.exports = async function (ctx) {
  return await ctx.mpserverless.db.collection('users').insertOne({
    name: 'tom',
    age: 1
  });
};
```

## 设置数据库索引

使用数据库时, 对成为查询条件的字段设置索引可以有效提高查询效率, 更快的获取信息。缺省情况下会对 `_id` 字段默认创建一个非唯一索引。

1. 在数据库页面, 单击目标数据表, 然后单击**索引**页签进入索引设置页面。

2. 单击**添加索引**进入索引设置页面。

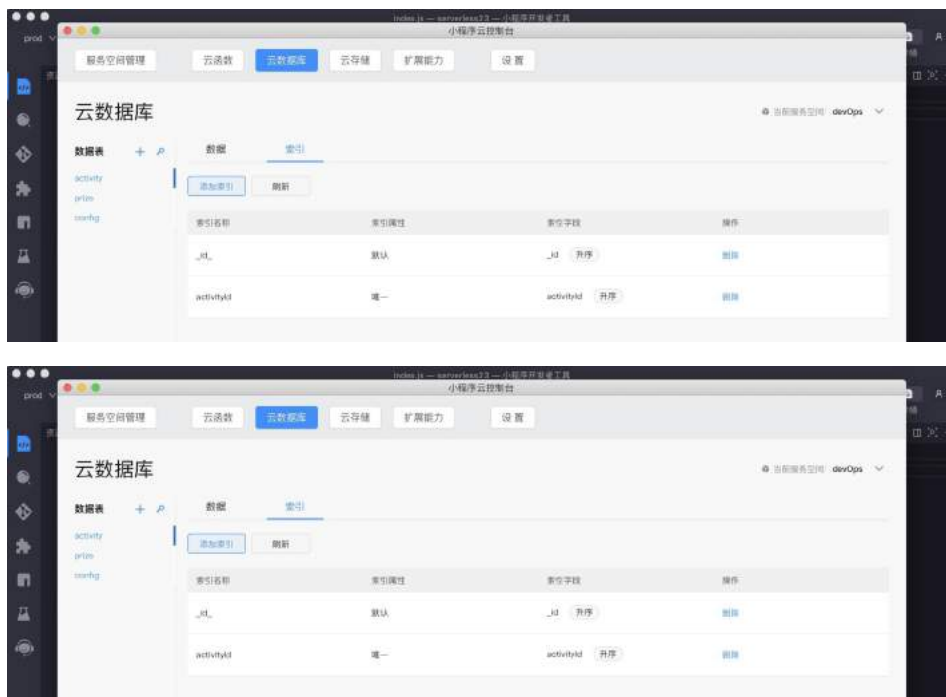
3. 填写索引名称, 设置索引属性、索引字段。

- 索引属性: 可以设置为唯一索引或非唯一索引。
  - 将字段设置为唯一索引后, 可以防止不同记录的被索引键上存储相同值。
- 索引字段: 支持单键索引和复合索引设置, 字段按升序或降序排列。
  - 单键索引是最常见的索引形式, 针对一个指定字段建立索引。对于单字段索引, 升序、降序的查询效果一样。

复合索引是单键索引的升级版, 针对多个字段联合创建索引, 先按照第一个字

段排列，第一个字段相同的记录按第二个字段排列，依次类推。

#### 4. 单击**保存**按钮，保存索引设置。



## 今日作业

在之前新建的服务空间里，设计云数据库的数据库表：以电商平台为场景，设计包含产品、用户、购物车和订单的数据库。订单是由用户购买产品产生的，订单就可以认为是产品和用户之间的关联关系产生的。在订单产生之前，产品和用户之间的关联关系是通过购物车来维持的。

- 用户表需要记录用户的登陆账号名，账号 id 等必要信息。
- 产品表需要记录商品的名称，商户 id，产品库存量，产品价格，产品图片 url 等必要信息。
- 购物车表需要记录用户 id，产品 id，购买产品的数量等必要信息。
- 订单表需要记录产品 id，总价，订单创建时间，订单状态，用户 id 等必要信息。



## 如何使用云存储?

### 云存储使用

mpserverless.file 对象提供 uploadFile 和 deleteFile 方法管理文件。上传的文件将通过 CDN 进行网络加速。

#### 步骤一：设置文件权限

小程序 Serverless 提供了一套简明易懂的 JSON 语法用来控制用户对资源的访问，类似于身份验证体系里的 IAM 或者网络安全中的 ACL。您可以通过修改权限规则来管理文件权限。

小程序 Serverless 会为每个新建的文件自动提供一个默认权限规则。默认规则规定所有文件都只有资源所有者可以进行写操作，所有人可读。您可以在控制台上修改文件权限。

1. 打开小程序云控制台，在云存储页面，单击权限图标。
2. 修改权限规则，然后单击**保存**。



上面的权限规则由多个子规则组成，每个子规则包含以下信息：

- 范围：用于定义规则生效范围。在数据存储的安全规则中，每一个规则的范围是一个文件（路径）。您可以通过文件路径指定一个特定的文件，也可以通过 JavaScript 的正则表达式指代一批文件或使用星号（\*）指代所有文件。
- 操作：用于定义生效范围内的特定操作。所有人都具有文件的读权限。.write 代表写操作，\* 代表所有操作。
- 策略：用于确定生效范围内的特定操作是否允许。默认没有标记为允许的操作，都是不允许的。策略是操作是否被允许的标记，可以是一个布尔值，或者是一个表达式。当策略冲突时，以最先出现的规则策略为准。如默认的权限规则策略表达式 `request.auth.userId == resource.auth.userId` 代表所有文件都只有资源所有者可以进行写操作，所有人可读。

## 步骤二：云存储操作

### 控制台操作

1. 打开小程序云控制台，在云存储页面，单击**上传文件**。
2. 单击**直接上传**或将文件拖拽至上传区域。

文件上传过程中，请勿刷新或关闭页面，否则上传任务会被中断且列表会被清空。



3. 上传成功后, 您可以单击[详情](#)查看图片下载地址。

## 小程序端调用

1. 在小程序项目的根目录执行以下命令安装 SDK。

```
npm install @alicloud/mpserverless-sdk --save
```

2. 文件上传或删除。

```
// 选择文件上传
my.chooseImage({
  chooseImage: 1,
  success: res => {
    const path = res.apFilePaths[0];
    const options = {
      filePath: path,
      headers: {
        contentDisposition: 'attachment',
      },
    },
  },
});

my.serverless.file.uploadFile(options).then(console.log).catch(console.error);
});

// 删除之前上传的文件
my.serverless.file.deleteFile('https://resource.bspapp.com/xxx-xx/4b82ded0-
0118-4de4-9f50-
ab13110a1ffb.jpg').then(res => {
  console.log(res);
}).catch(err => {
  console.error(err);
});
```

## 今日作业

实现在小程序中点击选择相册内图片上传, 上传成功后, 插入一条产品表的数据, 产品图片 url 字段使用上传成功后返回的 **fileUrl** 字段。

## 如何使用云函数

### 云函数开发

云函数 (FaaS) 是一段运行在云端的、轻量的、无关联的、并且可重用的代码。无需管理服务器，只需编写和上传代码，即可获得对应的数据结果。使用云函数可以使企业和开发者不需要担心服务器或底层运维设施，可以更专注代码和业务本身，也可以使代码进一步解耦，增加其重用性。

### 云函数示例

#### 云函数目录结构

在 Serverless 小程序工程的 `server/functions` 目录下创建云函数目录，其中 `index.js` 是云函数 `getImageList` 的入口文件。

```
└── server/  
    └── functions  
        └── getImageList  
            └── index.js
```

#### 云函数代码结构

以下代码示例展示了如何从数据库 `images` 里面查出特定用户上传的图片记录。云函数里可以直接通过 API 调用数据存储，文件存储的服务资源以及其他云函数。

- 当云函数被客户端或者其他云函数调用时，可以通过 `ctx.args` 获得调用传来的参数。
- 可以通过 `ctx.logger` 方法打印不同类型的日志信息，然后在云函数控制台中查看执行日志。
- 在云函数内通过 `ctx.env` 来获取环境参数，例如 `Spaceld`、调用来源、客户端源 IP 和客户端 `UserAgent` 等信息。`ctx.env` 是一个 object，它包含 `MP_`

SPACE\_ID (SpaceId), MP\_SOURCE (调用来源), MP\_USER\_AGENT (客户端 User-agent), MP\_CLIENT\_IP (客户端 IP)。

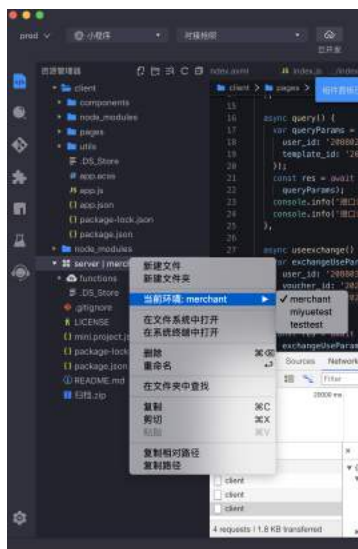
- 在云函数中，您可以直接调用同一环境（空间）的数据存储和文件存储服务以及获取当前请求用户的基本信息。ctx.mpserverless 在云函数封装了小程序的基础服务，API 使用方式和客户端一致。
- 通过 ctx.mpserverless.db 调用云数据库能力，如 ctx.mpserverless.db.collection('user').find({ uid: args.uid })。
- 通过 ctx.mpserverless.file 调用文件存储能力，如 ctx.mpserverless.file.deleteFile(args.filePathUrl)。
- 通过 ctx.mpserverless.user 获取当前访问用户基本信息，如 ctx.mpserverless.user.getInfo()。
- 通过 ctx.mpserverless.function 调用云函数能力，如 mpserverless.function.invoke('dataAnalytics', {range: 30}) 调用其他云函数。
- 通过 ctx.httpClient HTTP 请求方法，无需额外依赖，您就可以请求任何 HTTP 和 HTTPS 协议的 Web 服务。如 ctx.httpClient.request('https://www.alipay.com/x/notFound.htm')。

```
module.exports = async (ctx) => {
  ctx.logger.info('%s %s', ctx.env.MP_SPACE_ID, ctx.args.username);
  const images = await ctx.mpserverless.db.collection('images').find({ owner:
ctx.args.username });
  return { images };
};
```

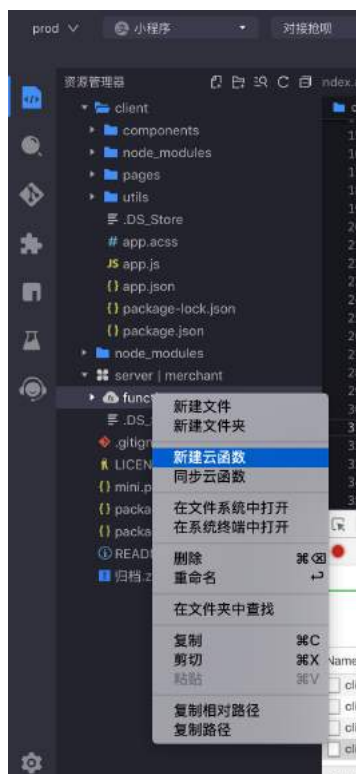
## 创建云函数

### 在代码工程创建云函数

1. 新建或打开小程序 Serverless 工程，点击 server 目录关联对应的云服务空间。



2. 在 functions 目录上右键选择新建云函数。



### 在云控制台创建云函数

打开小程序云控制台，在云函数页面，点击新建云函数。在新建云函数框里输入云函数的函数名。

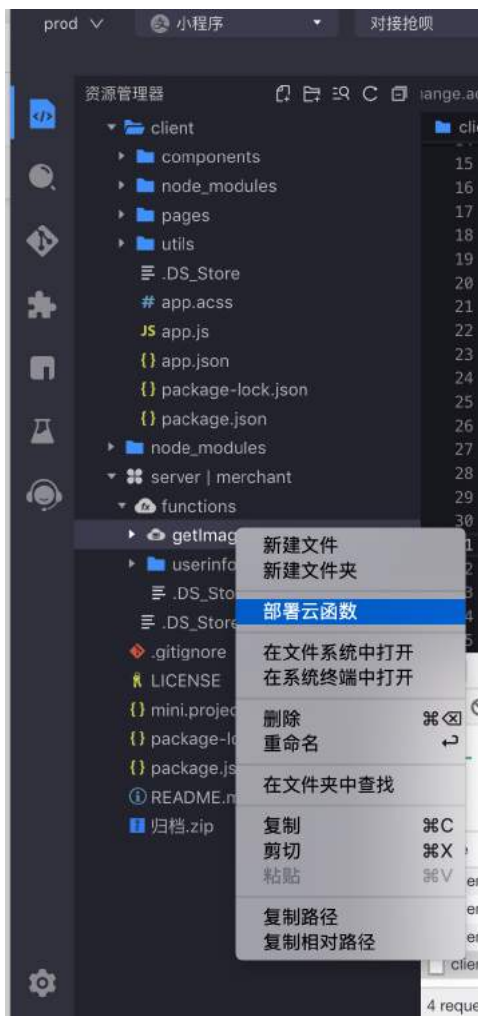


### 部署云函数

编写云函数代码后，您需要将云函数代码在代码工程里或者在云控制台上传。

### 在代码工程部署云函数

编写云函数代码后，右键点击已创建的云函数，选择部署云函数。



### 在云控制台部署云函数

1. 打开小程序云控制台，在云函数页面，单击已创建的函数名称链接。
2. 在**发布管理**页签，单击**上传 js 包**或**更新 js 包**，然后选择要上传的代码包。

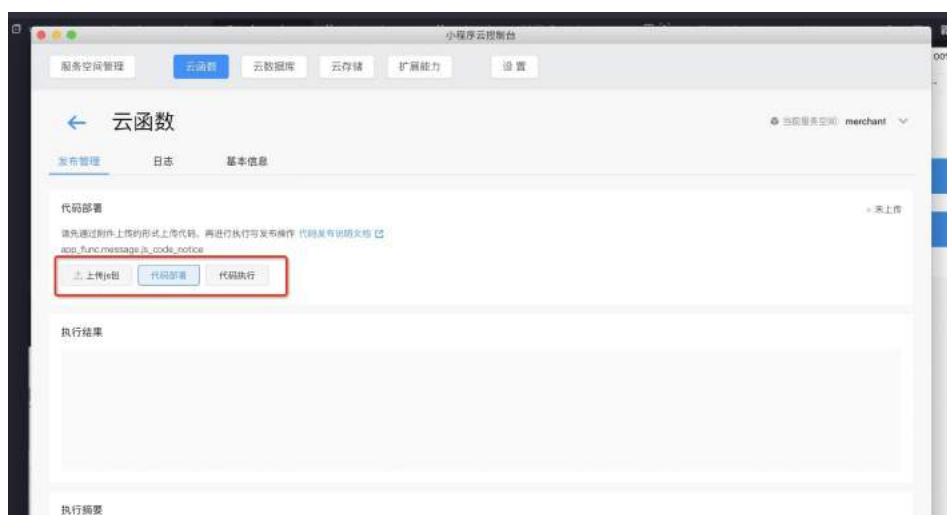
上传的代码包必须满足以下要求：

- 代码包的名称必须和在控制台上创建的函数名称一致。



- 代码包必须是 .zip 文件。
- 上传的代码包必须包含 index.js 文件。
- 如果引用了第三方包，代码包里必须包含 node\_modules。

2. 上传成功后，单击**代码部署**。



## 调用云函数

1. 在小程序项目的根目录执行以下命令安装 SDK。

```
npm install @alicloud/mpserverless-sdk --save
```

2. 在 app.js 中初始化 SDK (全局只需初始化一次)。

```
// 1. 引入必要的 sdk
import MPServerless from '@alicloud/mpserverless-sdk';

// 2. 在 app.js 中对 sdk 进行初始化
// 2.1 初始化 MPServerless
my.serverless = my.serverless || new MPServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
```

```
}, {  
  // 2.2 参数能在小程序云服务空间详情中获取  
  appId: '',  
  spaceId: '',  
  clientSecret: '',  
  endpoint: ''  
});
```

3. 在小程序页面您可以使用 `mpserverless.function.invoke` 方法调用上述定义的云函数。

```
// 调用云函数 getImageList  
my.serverless.function.invoke('getImageList', {  
  username: 'Vincent',  
}).then((res) => {  
  if (res.success && res.result) {  
    this.setData({ imageUrl: res.result.images });  
  }  
}).catch(console.error);
```

## 今日作业

编写一个云函数上传部署，在云函数里查询云数据库里产品表的所有数据。并在小程序页面端调用该云函数，并将返回的数据以列表形式显示在小程序页面。

## 如何使用云调用?

### 云调用使用

云调用是基于小程序 Serverless 的云函数来使用支付宝小程序开放接口的能力,可以方便的让开发者在小程序中直接调用支付宝的后端开放接口,不需要关注服务端的相关配置,极大的减少了接入的流程,进一步降低了支付宝小程序的开发门槛。

目前云调用涵盖了支付宝开放的基础能力、营销能力、会员能力、支付能力、安全能力,以及阿里云开放的短信能力。云调用的资金能力也将在近期开放。

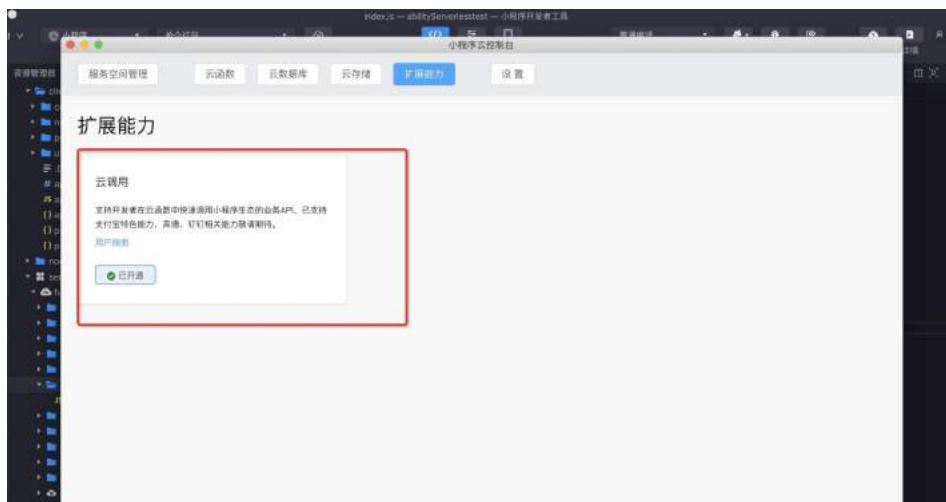
### 步骤一: 开通云调用功能

完成以下操作,在小程序云控制台开通云调用功能:

1. 打开小程序云控制台。
2. 在导航栏选择 扩展能力。
3. 在扩展能力页面,单击立即开通。
4. 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。



5. 返回扩展能力页面,查看开通状态。



## 步骤二：安装并初始化 SDK

1. 在小程序项目的根目录执行以下命令安装 SDK。

```
npm install @alicloud/mpserverless-sdk --save
npm install alipay-serverless-sdk --save
```

2. 在 app.js 中初始化云调用 SDK (全局只需初始化一次)。

```
// 1. 引入必要的 sdk
import MPServerless from '@alicloud/mpserverless-sdk';
import cloud from 'alipay-serverless-sdk';

// 2. 在 app.js 中对 sdk 进行初始化
// 2.1 初始化 MPServerless
my.serverless = my.serverless || new MPServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
}, {
  // 2.2 参数能在小程序云服务空间详情中获取
  appId: '',
  spaceId: '',
  clientSecret: '',
  endpoint: ''
});
```

```
// 2.3 初始化 alipay-serverless-sdk
cloud.init(my.serverless);
```

其中:

- appId 是小程序的 ID。您可以在支付宝小程序控制台获得。
- spacdId、clientSecret 和 endpoint 在小程序 Serverless 控制台创建服务空间后可以获得。

3. 在想要使用 SDK 的页面导入 alipay-serverless-sdk 即可使用。

```
// 比如在 page.js 导入 alipay-serverless-sdk
import cloud from 'alipay-serverless-sdk';

// 调用对应能力的 API
const res = await cloud.base.qrcode.create(urlParam, queryParam, describe);
```

更多云调用支持的开放能力见 <https://opendocs.alipay.com/mini/cloudservice/khf843>

同时我们提供了调用支付宝开放能力的通用云调用接口, 满足您的开发需要。

```
var exchangeUseParams = new Object({
  user_id: '2088xxxxxx',
  voucher_id: '2020132xxxxxx',
  out_biz_no: 'xxxxxx'
});
const res = await cloud.util.generic.execute('alipay.marketing.exchangevoucher.use',
  exchangeUseParams);
```

## 今日作业

在小程序端通过云调用的方式, 为你的小程序首页生成一个二维码。

## 教你一个云开发技巧

### 云开发技巧

在云开发技巧中，我们将向你介绍：

- 使用云函数 HTTP 触发器 + 云调用实现无需后端接入支付宝支付功能，并通过云函数接收支付回调功能；
- 使用云函数定时触发器实现每天凌晨定时调度任务。

### 使用云函数 HTTP 触发器实现支付回调

当开启某一函数的 HTTP 触发功能后，只要使用 HTTP 请求就可以访问到函数，开发者需要关注业务安全和资源安全。

- 业务安全：开发者需在云函数代码中做好权限控制和安全防护，避免未授权访问触发敏感操作。
- 资源安全：云函数开启了 HTTP 访问后，如果遇到大量恶意访问，消耗云函数资源，开发者可以将云函数的触发路径设置为空或关掉该 Space 的 HTTP 触发功能，停止 HTTP 触发支持。

### 开启云函数 HTTP 触发功能

- 新建用于处理支付回调逻辑的云函数，上传部署。
- 开启 Space 的 HTTP 触发功能调用接口 [UpdateHttpTriggerConfig](#) 接口来开启 Space 的 HTTP 触发功能，注意需要设置 EnableService 为 true。系统会为开启 HTTP 触发功能的 Space 分配一个默认域名 (DefaultEndpoint)，供 HTTP 触发使用。
- 开启云函数的 HTTP 触发功能调用接口 [UpdateFunction](#) 接口来为云函数设置 HTTP 触发路径 HttpTriggerPath，只有设置了触发路径的云函数才能使用该功能。
- HTTP 触发函数执行访问 `https://{DefaultEndpoint}{HttpTriggerPath}` 或

`http://${DefaultEndpoint}${HttpTriggerPath}`，即可触发对应的云函数执行。

- 介绍两种调用 [UpdateHttpTriggerConfig](https://help.aliyun.com/document_detail/158669.html) 和 [UpdateFunction](https://help.aliyun.com/document_detail/153370.html) 的方式，一种是在阿里云的 API 在线调试界面 [https://help.aliyun.com/document\\_detail/158669.html](https://help.aliyun.com/document_detail/158669.html) 以及 [https://help.aliyun.com/document\\_detail/153370.html](https://help.aliyun.com/document_detail/153370.html) 发起在线 API 调用。另外一种就是在代码里集成 `@alicloud/pop-core`，示例代码如下：

```
const Core = require('@alicloud/pop-core');
var client = new Core({
  accessKeyId: '你的 accessKeyId',
  accessKeySecret: '你的 accessSecret',
  endpoint: 'https://mpserverless.aliyuncs.com',
  apiVersion: '2019-06-15'
});
var params = {
  "RegionId": "cn-hangzhou",
  "EnableService": true,
  "SpaceId": "你的 spaceid"
}
var requestOptions = {
  method: 'POST'
};
client.request('UpdateHttpTriggerConfig', params, requestOptions).then((result) => {
  console.log(JSON.stringify(result));
}, (ex) => {
  console.log(ex);
})
```

## 云调用调用支付接口，并使用云函数的 HTTP 触发路径做为支付回调 Url 参数

1. 在小程序项目的根目录执行以下命令安装 SDK。

```
npm install @alicloud/mpserverless-sdk --save
```

2. 在 `app.js` 中初始化云调用 SDK（全局只需初始化一次），并将之前获得的用于处理支付回调逻辑的云函数的 HTTP 触发路径（暂时支持 http 协议格式路径）作为初始化云调用 SDK 的参数。

```
import MPServerless from '@alicloud/mpserverless-sdk';
import cloud from 'alipay-serverless-sdk';

my.serverless = my.serverless || new MPServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode,
}, {
  appId: '',
  // 小程序云控制台中记得对比切换的空间是否正确哦 ~
  spaceId: '',
  clientSecret: '',
  endpoint: 'https://api.bspapp.com'
});

// 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
cloud.init(my.serverless, {
  notify: 'http://xxxxxx.bspapp.com/callbackForPay' // 此处填写您获得的云函数 HTTP 触发路径
});
```

其中每次调用 [UpdateFunction](#) 都会触发云函数的 HTTP 触发路径发生变化，发生变化之后务必记得更改云调用 SDK 初始化里的 notify 参数。

### 3. 在小程序端使用云调用调用支付能力。

```
// 使用云调用 sdk 调用支付创建接口
cloud.payment.common.create('iPhoneX', '2020053101122008', '5100.50', '2088xxxxx');
```

## 使用云函数定时触发实现定时调度任务

1. 新建用于定时调度的云函数，代码编写完后上传部署。
2. 调用 UpdateFunction 接口，通过设置参数 TimingTriggerConfig 来为云函数设置定时触发配置。

TimingTriggerConfig 参数的值为一个 cron 表达式，例如 cron:0 0 \* \* \* \* 表示每个整点触发。配置规则仅支持设置 cron 表达式。

3. [UpdateFunction](#) 接口的调用方式同样有两种，一种是在阿里云的 API 在线



调试界面 [https://help.aliyun.com/document\\_detail/153370.html](https://help.aliyun.com/document_detail/153370.html) 发起在线 API 调用。另外一种就是在代码里集成 @alicloud/pop-core，示例代码如下：

```
const Core = require('@alicloud/pop-core');

var client = new Core({
  accessKeyId: '你的 accessKeyId',
  accessKeySecret: '你的 accessSecret',
  endpoint: 'https://mpserverless.aliyuncs.com',
  apiVersion: '2019-06-15'
});

var params = {
  "RegionId": "cn-hangzhou",
  "Name": "你的云函数名称",
  "SpaceId": "你的 spaceid",
  "TimingTriggerConfig": "你的 cron 表达式"
}

var requestOptions = {
  method: 'POST'
};

client.request('UpdateFunction', params, requestOptions).then((result) => {
  console.log(JSON.stringify(result));
}, (ex) => {
  console.log(ex);
})
```

## 今日作业

编写一个定时调度任务云函数，每周的周 1 到周 5 凌晨 1 点向云数据库的 daily 表中插入一条数据，数据内容如下：

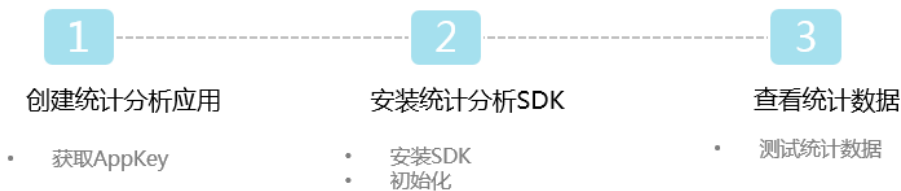
```
{
  "name": "xxxx 年 x 月 x 日定时调度", // 标示插入数据的年月日
  "createTime": 1590933848348 // 标示插入数据的时间戳
}
```

## 实现你的小程序统计分析

### 小程序统计分析

小程序上线之后，如果我们需要监控统计小程序的启动次数，用户新增，活跃用户等指标，我们可以先在控制台创建应用，然后在小程序项目中安装统计分析服务 SDK。

要接入支付宝小程序进行用户使用数据统计分析，您需要完成以下操作：



#### 步骤一：创建统计分析应用

1. 登录[小程序 WEB 云控制台](#)。
2. 选择服务空间。
3. 在左侧导航栏，单击**统计分析**，然后单击**创建应用**。



4. 根据提示配置应用信息，然后单击**确定**。

应用创建

\* 应用名称

支付宝ToDoApp

\* 平台选择

支付宝

\* 一级分类

实用工具

\* 二级分类

生活工具

描述

请输入应用的相关描述

确定

取消

5. 记录生成的 AppKey。

统计分析					
应用名称	平台	AppKey	应用描述	创建时间	操作
支付宝ToDoApp	支付宝	54887e457b9e99990002ad		2019-09-24 05:06:02	<a href="#">查看详情</a> <a href="#">删除</a>

## 步骤二：集成统计 SDK

1. 在小程序项目的根目录执行以下命令安装统计 SDK。

```
npm i @alicloud/mpserverless-analyse-alipaysdk -- save
```

2. 在支付宝小程序项目的 app.js 文件中添加如下代码。

```
// 1. 引入必要的 sdk
import MPServerless from '@alicloud/mpserverless-sdk';
```

```
import MPAnalyse from '@alicloud/mpserverless-analyse-alipaysdk';

// 2 初始化 alipay-serverless-sdk
const serverless = new MPServerless({
  uploadFile: my.uploadFile,
  request: my.request,
  getAuthCode: my.getAuthCode
}, {
  appId: '1234456789', // 小程序应用标识
  spaceId: 'db4dd657-7041-470a-90xxxxx', // 服务空间标识
  clientSecret: '6c3c86xxxx6', // 服务空间 secret key
  endpoint: 'https://endpoint' // 服务空间地址, 从小程序 Serverless 控制台处获得
});
// 3. 务必引入统计 sdk 的方法
App({
  onLaunch(options) {
    MPAnalyse.init('YOUR_APP_KEY', my, { // 务必填入已注册的统计分析 appKey, 不然将无法统计
      mpServerless: serverless, // 务必传入实例化 MPServerless 后的 serverless
      platformParam: options // 若不需要渠道分析功能, 可以暂不传入
    });
  },
  onShow() {
    MPAnalyse.resume(); // 请务必引入
  },
  onHide() {
    MPAnalyse.pause(); // 请务必引入
  }
});
```

### 步骤三：查看统计数据

1. 登录[小程序 WEB 云控制台](#)。
2. 在左侧导航栏，单击**统计分析**。
3. 单击目标统计分析应用的**查看数据**链接进入统计分析页面。





更多独家电子书



阿里云开发者“藏经阁”  
海量免费电子书下载

加入钉群



获取更多资讯

加入钉群



与更多开发者交流