

# 小程序云开发

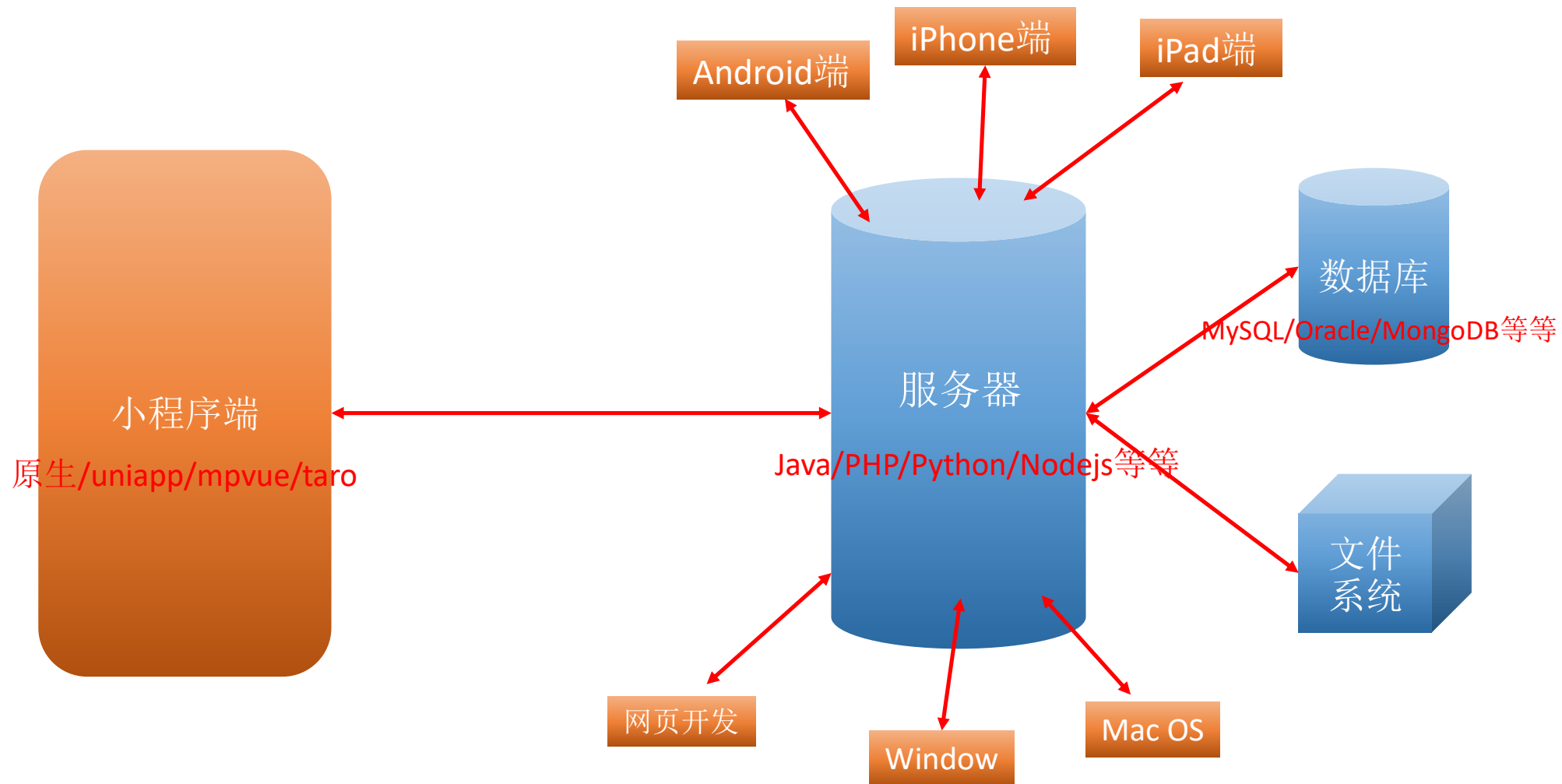
姓名：王红元

微博/公众号：coderwhy



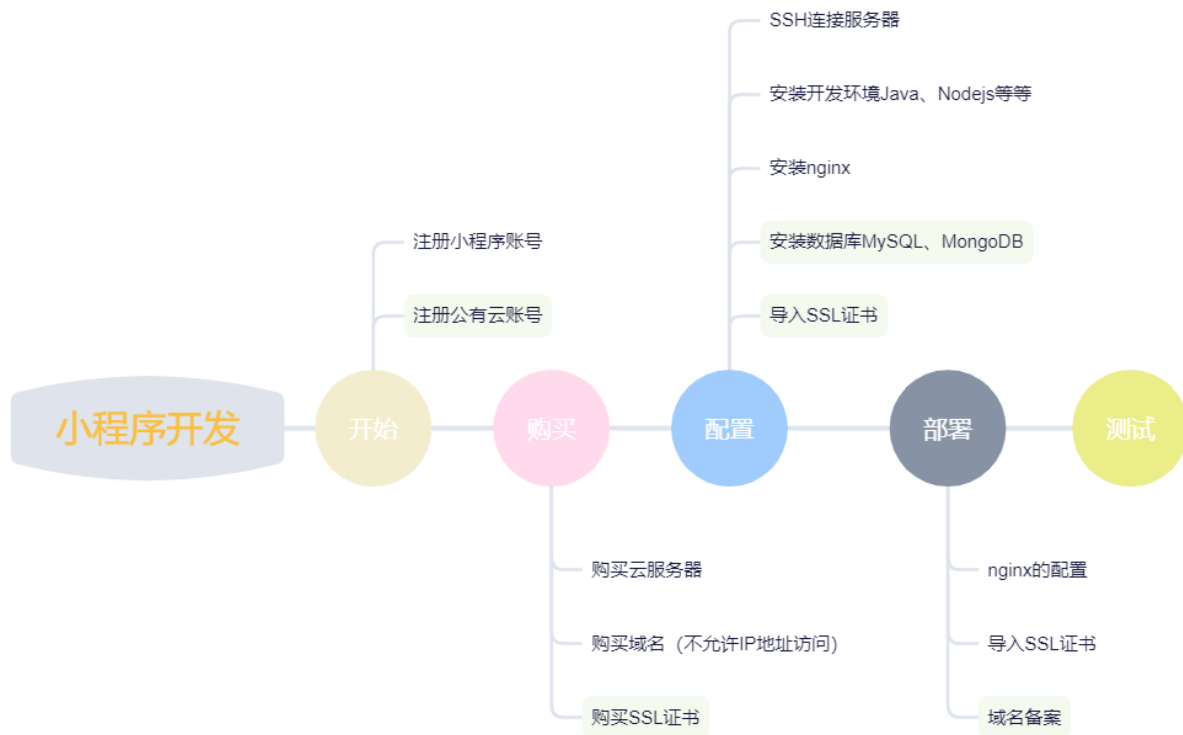
实力IT教育 [www.520it.com](http://www.520it.com)

# 完整的小程序项目



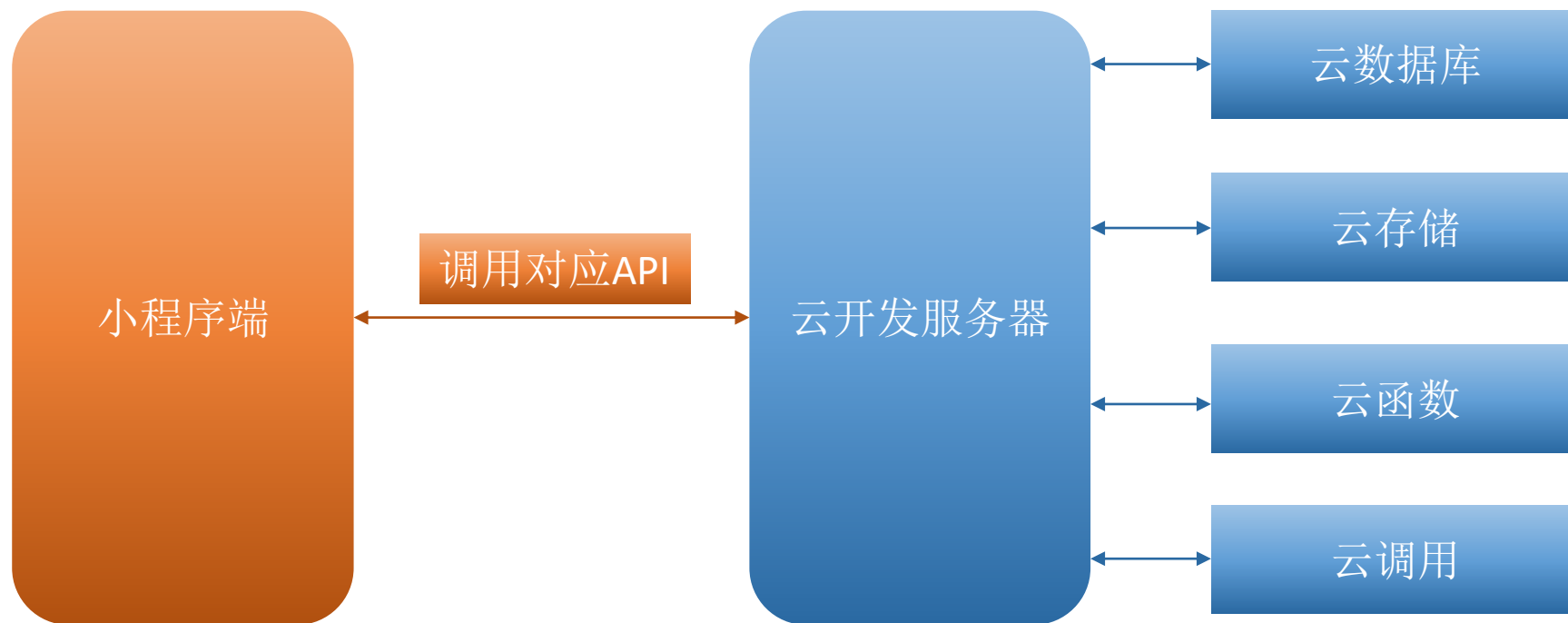
# 开发成本的考虑

- 但是，如果一个 小公司或个人 只是想开发一个小程序推广自己的产品或者实现某个想法了？
- 按照传统的开发模式，我们需要考虑哪些东西呢？
  - 成本角度：维护服务器成本，并且需要考虑并发量大后服务器的扩展。
  - 技术成本：对于单纯会前端的人来说，学习后端相关的技术，成本较高。

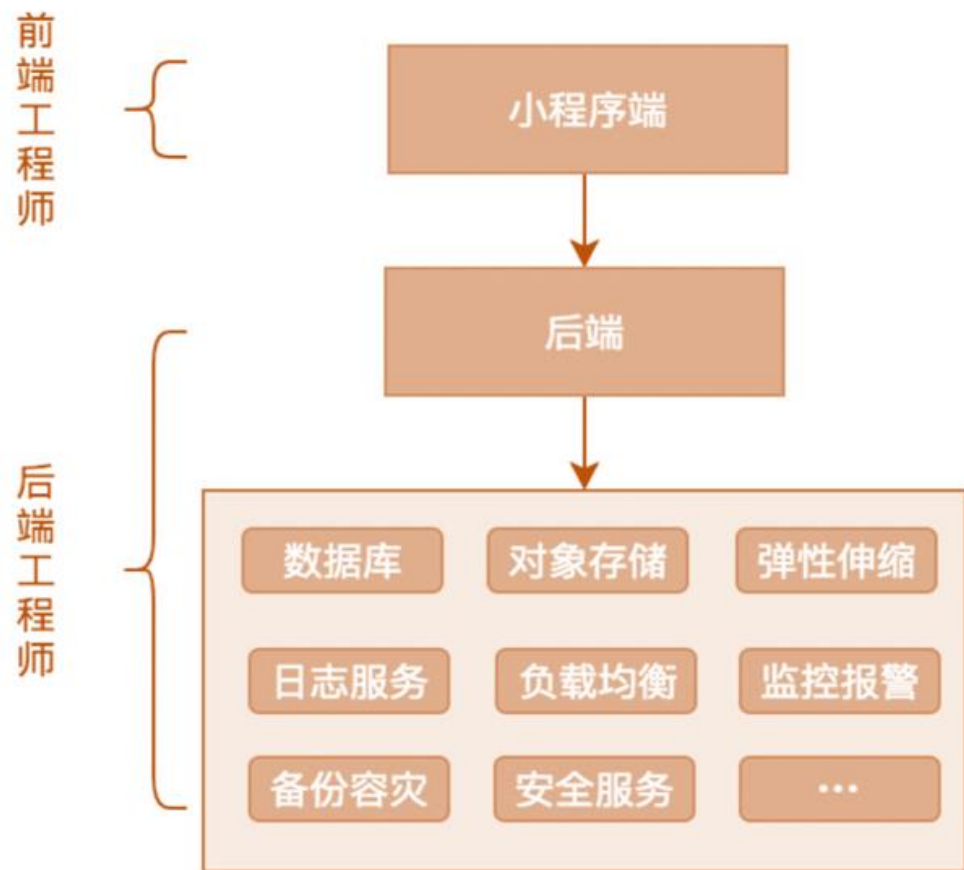


**是否有一种新的开发模式  
让开发者可以更多地专注业务逻辑**

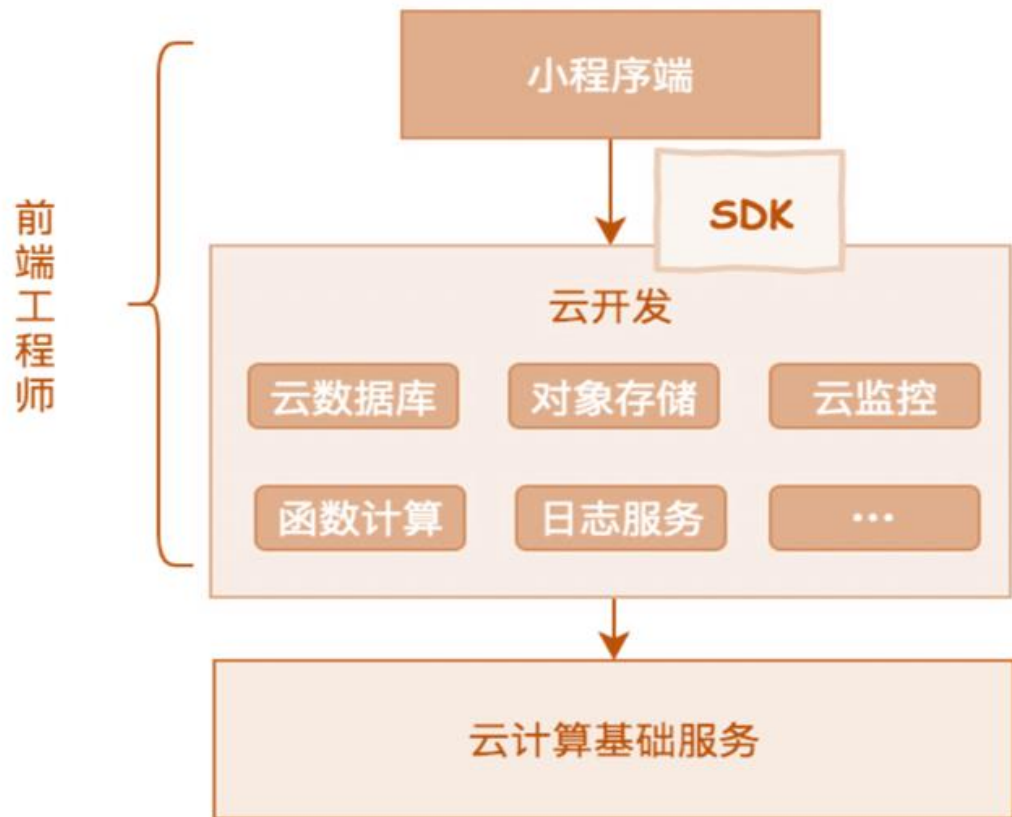
# 云开发的模式



# 云开发模式和传统模式对比

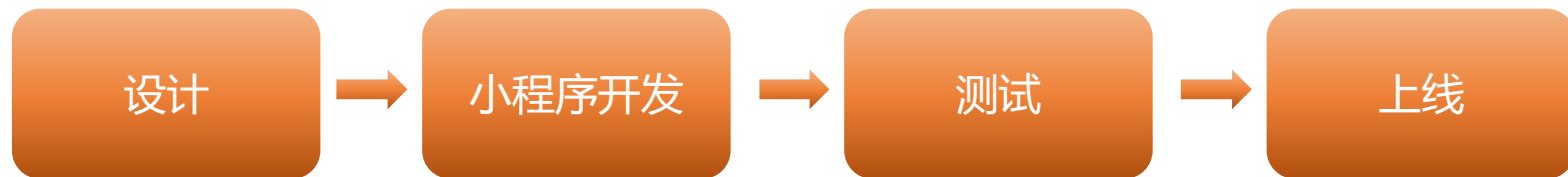
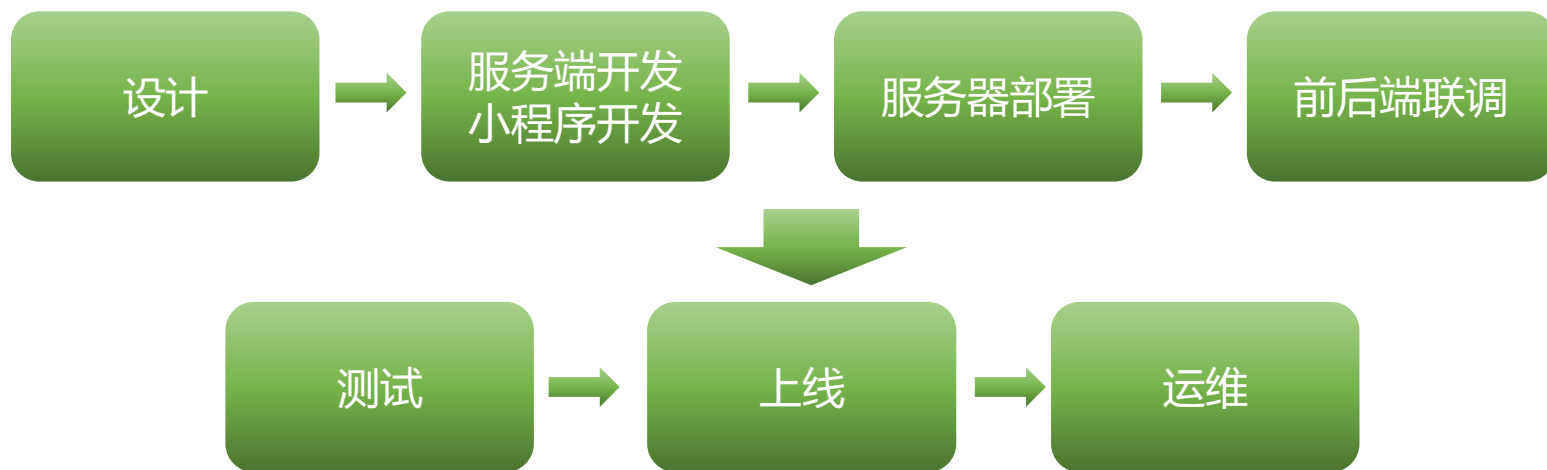


传统小程序开发



基于 Serverless 的小程序开发

# 项目流程对比



## ■ 云开发主要包含三大核心技术：

### ■ 云数据库：

- 提供在小程序端直接对数据库进行增删改查的能力；
- 数据库是类似于MongoDB的文档存储的数据库，操作非常方便；

### ■ 云存储：

- 可以在小程序端直接上传、下载、删除文件；
- 自带CDN，提高文件访问速度；
- 可以获取临时链接，支持在小程序外访问；

### ■ 云函数：

- 提供了在服务器代码的执行能力；
- 包含微信天然的私有鉴权；
- 更大权限的操作数据库等；
- 进行云调用、HTTP请求等操作；



## ■ 创建项目时，选中云开发

小程序项目

小程序

小游戏

代码片段

公众号网页项目

公众号网页



新建项目

导入项目

项目名称

LearnMiniCloud

目录

C:\Users\home\WeChatProjects\miniprogram-2

AppID

wx44a8878d2bb8c7d8

若无 AppID 可 [注册](#)

或使用 [测试号](#) ?

开发模式

小程序

后端服务

☒ 小程序·云开发

小程序·云开发为开发者提供数据库、存储和云函数等完整的云端支持。无需搭建服务器，使用平台提供的 API 进行核心业务开发，即可实现小程序快速上线和迭代。 [了解详情](#)

☐ 不使用云服务



注销 >

取消

新建

# 打开项目

项目 文件 编辑 工具 界面 设置 帮助 微信开发者工具

3. 点击可以进入云控制台

小程序模式 普通编译 编译 预览 真机调试 切后台 清缓存 上传 版本管理 详情

模拟器 编辑器 调试器 云开发

iPhone 5 100% WiFi 模拟操作

cloudfunctions | coderwhy  
miniprogram 2. 云函数和小程序目录  
README.md  
project.config.json

app.json

```
13 "pages/openapi/cloudid/cloudid",
14 "pages/im/im",
15 "pages/im/room/room"
16 ],
17 "window": {
18   "backgroundColor": "#F6F6F6",
19   "backgroundTextStyle": "light",
20   "navigationBarBackgroundColor": "#F6F6F6",
21   "navigationBarTitleText": "云开发 QuickStart",
22   "navigationBarTextStyle": "black"
23 },
24 "sitemapLocation": "sitemap.json",
25 "style": "v2"
26 }
```

点击获取 openid

上传图片

前端操作数据库

即时通信 Demo 1. 官方提供的示例程序

快速新建云函数

云调用

页面路径 pages/index/index 复制 预览 场景值 页面参数

Console Sources Network Security AppData Audits Sensor Storage Trace Wxml

top Filter Default levels 1 hidden

Thu Nov 21 2019 18:29:41 GMT+0800 (中国标准时间) sitemap 索引情况提示 VM107:4

根据 sitemap 的规则[0], 当前页面 [pages/openapi/serverapi/serverapi] 将被索引 VM32:1

Thu Nov 21 2019 18:29:46 GMT+0800 (中国标准时间) sitemap 索引情况提示 VM112:4

根据 sitemap 的规则[0], 当前页面 [pages/openapi/openapi] 将被索引 VM32:1

Thu Nov 21 2019 18:29:47 GMT+0800 (中国标准时间) sitemap 索引情况提示 VM113:4

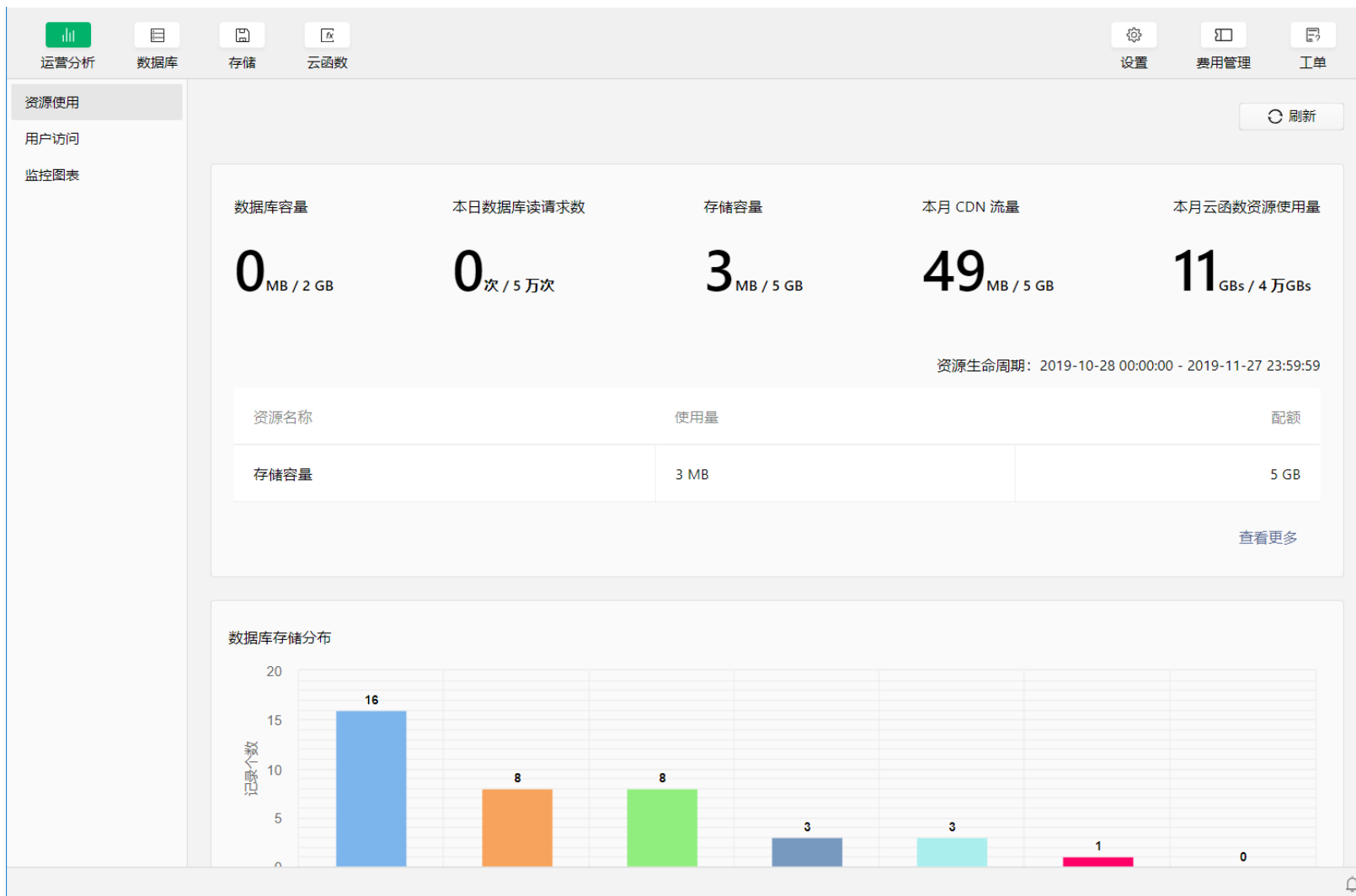
根据 sitemap 的规则[0], 当前页面 [pages/index/index] 将被索引 VM32:1

## ■ 云开发控制台

- 运营分析
- 数据库（云数据库）
- 存储（云存储）
- 云函数

## ■ 开通云开发

- 设置 – 环境名称 – 创建环境



# 重要概念 – 环境和配额

## ■ 环境：

- 一个环境对应一整套独立的云开发资源，包括数据库、存储空间、云函数等资源。
- 各个环境是相互独立的，用户开通云开发后即创建了一个环境，默认可拥有最多两个环境。
- 在实际开发中，建议每一个正式环境都搭配一个测试环境，所有功能先在测试环境测试完毕后再上到正式环境。

## ■ 配额：

- 默认有一定的免费配额（你懂的）；
- 后期可以根据自己的业务量选择对应的更高配额；
- <https://developers.weixin.qq.com/miniprogram/dev/wxcloud/billing/quota.html>

- 在小程序端开始使用云能力前，需先调用 `wx.cloud.init` 方法完成云能力初始化

字段	数据类型	必填	默认值	说明
env	string   object	是（否）	默认选中环境	后续 API 调用的默认环境配置，传入字符串形式的环境 ID 可以指定所有服务的默认环境，传入对象可以分别指定各个服务的默认环境，见下方详细定义
traceUser	boolean	否	false	是否在将用户访问记录到用户管理中，在控制台中可见

```
if (!wx.cloud) {
  console.error('请使用 2.2.3 或以上的基础库以使用云能力')
} else {
  wx.cloud.init({
    // env: 'my-env-id',
    traceUser: true,
  })
}
```

## ■ JSON数据库：

- 云开发提供了一个文档型数据库，类似于MongoDB，里面存放的是一条条JSON格式的对象；
- 一个数据库可以包含多个集合，一个集合中包含多个JSON对象；

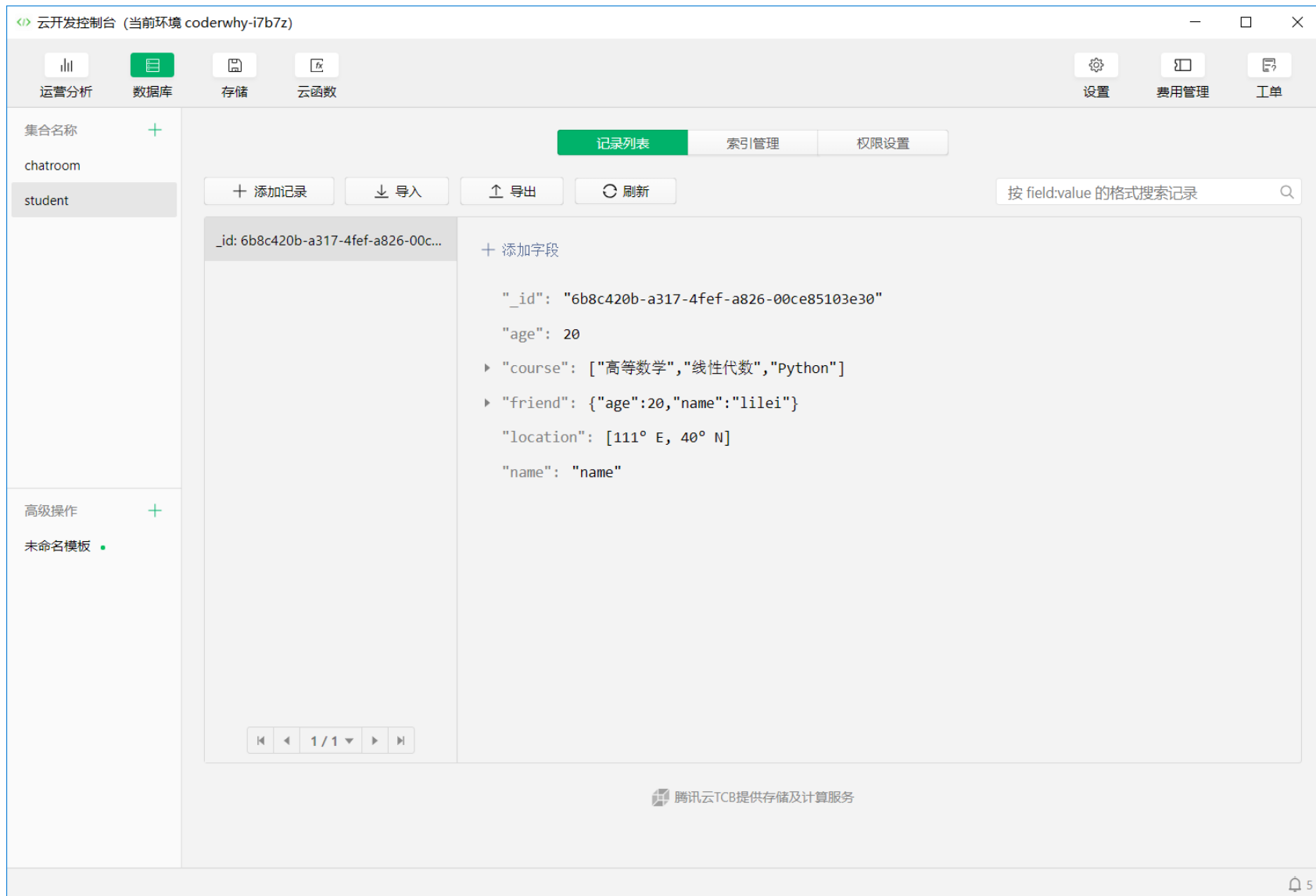
关系型	文档型
数据库 database	数据库 database
表 table	集合 collection
行 row	记录 record / doc
列 column	字段 field

- 提供方便的API调用：学习这些API即可；
- 提供了小程序端和服务端（云函数）中调用的区分；

# 操作数据库 – 控制台操作

## ■ 演练：

- 创建集合
- 创建一条数据
- 添加字段
- 导入一组数据



云开发控制台 (当前环境 coderwhy-i7b7z)

运营分析 数据库 存储 云函数 设置 费用管理 工单

集合名称 +

chatroom

student

高级操作 +

未命名模板

记录列表 索引管理 权限设置

+ 添加记录 导入 导出 刷新

按 field:value 的格式搜索记录

+ 添加字段

```
{
  "_id": "6b8c420b-a317-4fef-a826-00ce85103e30"
  "age": 20
  "course": ["高等数学", "线性代数", "Python"]
  "friend": {"age": 20, "name": "lilei"}
  "location": [111° E, 40° N]
  "name": "name"
}
```

腾讯云TCB提供存储及计算服务

# 小程序端操作数据库 - 界面搭建

项目 文件 编辑 工具 界面 设置 帮助 微信开发者工具

小程序模式 云数据库 编译 预览 真机调试 切后台 清缓存

模拟器 编辑器 调试器 云开发

iPhone 5 100% WiFi 模拟操作

添加数据  
删除数据  
修改数据  
查询数据

```
<!--miniprogram/pages/w-database/w-database.wxml-->
<button bindtap="addData">添加数据</button>

<button bindtap="removeData">删除数据</button>

<button bindtap="updateData">修改数据</button>

<button bindtap="queryData">查询数据</button>
```

w-database.js

```
2 Page({
3   addData: function() {
4
5   },
6
7   removeData: function() {
8
9   },
10
11  updateData: function() {
12
13  },
14
15  queryData: function() {
16
17  }
```

/miniprogram/pages/w-database/w-database.js 189 B 行 13, 列 5 JavaScript

AppData Audits Sensor Storage Trace Wxml

Default levels 1 hidden

Thu Nov 21 2019 19:21:14 GMT+0800 (中国标准时间) sitemap 索引情况提示 VM880:4

根据 sitemap 的规则[0], 当前页面 [pages/w-database/w-database] 将被索引 VM830:1

官方示例 云数据库 云存储 云函数

页面路径 pages/w-databas... 复制 预览 场景值 页面参数



# 添加数据 和 调用结果

## ■ 添加数据的调用过程：

- 1. 获取数据库对象
- 2. 获取操作的集合
- 3. 添加数据

## ■ 获取操作后的回调结果：

- 基于回调：传入success、fail、complete
- 基于Promise：使用then、catch、finally

```
success: res => {  
  console.log(res)  
},  
fail: err => {  
  console.err(err)  
},  
complete: res => {  
  console.log("完成添加数据", res)  
}
```

```
}).then(res => {  
  console.log(res)  
}).catch(err => {  
  console.err(err)  
}).finally(() => {  
  console.log("完成调用")  
})
```

```
addData: function() {  
  // 1. 获取数据库对象  
  const db = wx.cloud.database();  
  
  // 2. 获取要操作的集合  
  const collection = db.collection("student")  
  
  // 3. 在数据库中添加数据  
  collection.add({  
    data: {  
      name: "lilei",  
      age: 30,  
      course: ["计算机操作系统", "离散数学", "数据结构"],  
      isMarried: false,  
      girlFriend: {  
        name: "hanmeimei",  
        age: 25  
      },  
      location: db.Geo.Point(113.36199, 23.12463),  
      entrance: new Date("2011-11-11")  
    },  
  })  
}
```

## ■ 查询数据的方式:

□ **方式一:** 通过ID查询精确的某一条数据;

✓ 使用doc查询ID

□ **方式二:** 根据条件查询满足条件的数据;

✓ 使用where作为条件

□ **方式三:** 通过指令过滤数据;

✓ 使用db.command的指令

□ **方式四:** 通过正则表达式匹配符合的数据;

✓ 使用db.RegExp创建正则规则

□ **方式五:** 获取整个集合的数据 (小程序端一次性最多20条, 云函数中可以获取100条) ;

✓ 直接调用get

□ **方式六:** 过滤、分页、排序查询数据

✓ 使用field、skip、limit、orderBy

# 数据查询二

// 1.通过ID精准查找

```
collection.doc("6b8c420b-a317-4fef-a826-00ce85103e30").get().then(res => {  
  console.log(res);  
})
```

// 2.查找name是tom的数据

```
collection.where({  
  name: "tom"  
}).get().then(res => {[  
  console.log(res)  
]})
```

// 3.根据指令查询

```
const cmd = db.command;  
collection.where({  
  age: cmd.gt(20)  
}).get().then(res => {  
  console.log(res)  
})
```

// 4.使用正则匹配

```
collection.where({  
  name: db.RegExp({  
    regexp: "^a.+",  
    options: "i"  
  })  
}).get().then(res => {  
  console.log(res)  
})
```

// 5.获取整个集合

```
collection.get().then(res => {  
  console.log(res)  
})
```

// 6.综合查询

```
collection.field({  
  name: true,  
  age: true,  
  girlfriend: true  
}).skip(0).limit(3)  
.orderBy("age", "desc")  
.get().then(res => {  
  console.log(res)  
})
```

# 修改数据

## ■ 修改数据有两种方式：

- **update**: 更新（增加）某一个字段
- **set**: 使用新对象替换原来对象

```
// update更新(增加)某个字段
collection.doc("953037125dd684d901ec126678d91858")
  .update({
    data: {
      height: 1.98
    }
  }).then(res => {
    console.log(res)
  })
```

```
// 替换某条数据
collection.doc("953037125dd684d901ec126678d91858")
  .set({
    data: {
      height: 1.98
    }
  }).then(res => {
    console.log(res)
  })
```

## ■ 注意：在小程序中不能通过条件查询多条数据一起修改

## ■ 删除数据使用remove即可

```
removeData: function() {  
  collection.doc("953037125dd684d901ec126678d91858")  
    .remove().then(res => {  
      console.log(res)  
    })  
},
```

# 简单实现即时通信

## ■ 云数据库目前已经支持即时通信：

□ 云开发数据库支持实时推送变更数据的能力

□ 给定查询条件，每当数据库更新而导致查询条件对应的查询结果发生变更时，小程序可收到一个更新事件，其中可获取更新内容和更新后的查询结果快照。

```
listenWChatroom: function() {  
  const watch = db.collection("mychat").where({  
    groupid: "110"  
  }).watch({  
    onChange: function(snapshot) {  
      console.log(snapshot)  
    },  
    onError: function(err) {  
      console.log(err)  
    }  
  })  
},
```

```
sendMessage: function() {  
  db.collection("mychat").add({  
    data: {  
      groupid: "110",  
      message: "你好啊，李银河"  
    }  
  }).then(res => {  
    console.log(res)  
  })  
}
```



## ■ 云存储用于将文件存储到云端：

- 云存储提供高可用、高稳定、强安全的云端存储服务；
- 持任意数量和形式的非结构化数据存储，如视频和图片；
- 并在控制台进行可视化管理；

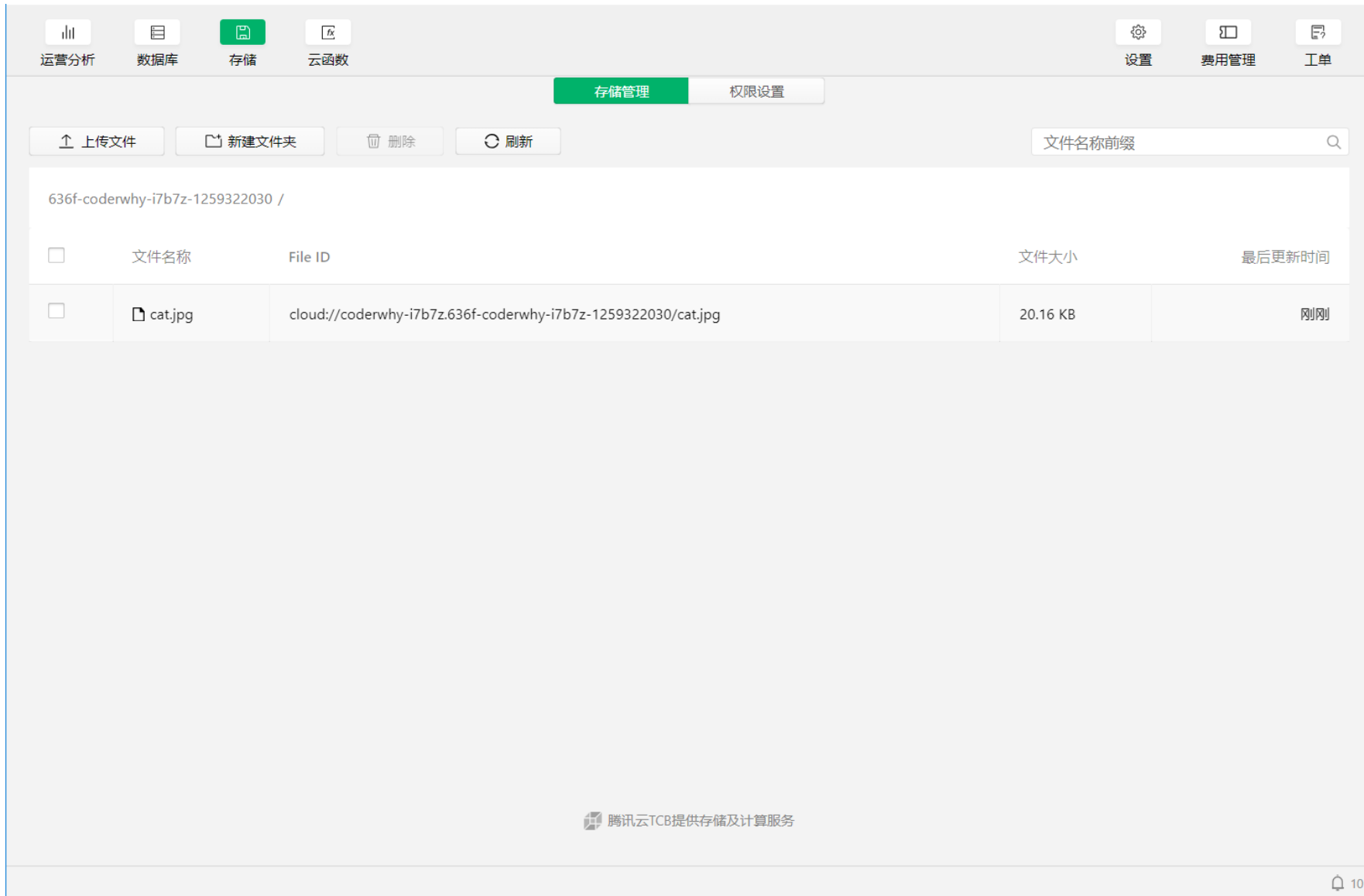
## ■ 云存储常见的操作：

- 上传文件到云存储中（图片、视频、音频等等都可以）
- 获取文件的临时链接（在外网可以访问）
- 下载文件到本地（本地文件缓存）
- 将云存储中的文件删除

# 云存储操作 – 在控制台操作

## ■ 演练:

- 上传文件到云存储中
- 获取fileID在项目中显示
- 获取URL在浏览器显示



The screenshot shows the Tencent Cloud console's storage management interface. At the top, there are navigation tabs: 运营分析 (Operation Analysis), 数据库 (Database), 存储 (Storage), and 云函数 (Cloud Functions). The 存储 (Storage) tab is selected. On the right, there are icons for 设置 (Settings), 费用管理 (Cost Management), and 工单 (Tickets). Below the navigation bar, there are two main tabs: 存储管理 (Storage Management) and 权限设置 (Permission Settings). The 存储管理 (Storage Management) tab is active. Under this tab, there are buttons for 上传文件 (Upload File), 新建文件夹 (New Folder), 删除 (Delete), and 刷新 (Refresh). A search bar labeled 文件名称前缀 (File Name Prefix) is on the right. The main content area shows a breadcrumb path: 636f-coderwhy-i7b7z-1259322030 /. Below this is a table with columns: 文件名称 (File Name), File ID, 文件大小 (File Size), and 最后更新时间 (Last Update Time). The table contains one entry: a file named cat.jpg with File ID cloud://coderwhy-i7b7z.636f-coderwhy-i7b7z-1259322030/cat.jpg, a size of 20.16 KB, and an update time of 刚刚 (Just now). At the bottom, there is a footer with the text 腾讯云TCB提供存储及计算服务 (Tencent Cloud TCB provides storage and computing services) and a page number 10.

文件名称	File ID	文件大小	最后更新时间
cat.jpg	cloud://coderwhy-i7b7z.636f-coderwhy-i7b7z-1259322030/cat.jpg	20.16 KB	刚刚



# 小程序端操作云存储 - 界面搭建

项目 文件 编辑 工具 界面 设置 帮助 微信开发者工具

小程序模式 云存储 编译 预览 真机调试 切后台 清缓存 上传 版本管理 详情

iPhone 5 100% WiFi 模拟操作

上传文件  
临时链接  
下载文件  
删除文件

```
<!--miniprogram/pages/w-storage/w-storage.wxml-->
<button bindtap="uploadFile">上传文件</button>

<button bindtap="getTempURL">临时链接</button>

<button bindtap="downloadFile">下载文件</button>

<button bindtap="deleteFile">删除文件</button>
```

官方示例 云数据库 云存储 云函数

页面路径 pages/w-storage/... 复制 预览 场景值 页面参数

# 云存储 - 上传文件

```
// 1.选择图片文件
wx.chooseImage({
  success: (res) => {
    // 2.获取filePath
    const filePath = res.tempFilePaths[0];

    // 3.上传文件
    const timestamp = new Date().getTime();
    const openid = "123"
    wx.cloud.uploadFile({
      cloudPath: `${timestamp}_${openid}.jpg`,
      filePath
    }).then(res => {
      console.log(res)
      this.setData({
        imgFileID: res.fileID
      })
    })
  },
})
```

```
// 2.选择视频文件
// 1.选择视频
wx.chooseVideo({
  success: function(res) {
    // 2.获取filePath
    const filePath = res.tempFilePath;

    // 3.上传视频文件
    wx.cloud.uploadFile({
      cloudPath: "123.mp4",
      filePath
    }).then(res => {
      // 4.拿到结果
      console.log(res)
    })
  }
})
```

## ■ 为什么要获取临时链接？

- 我们将文件上传到云存储后，可以通过fileID在小程序中直接访问；
- 但是，如果我们希望在小程序以外的地方访问（比如浏览器、手机端），那么fileID是不可以的；
- 这个时候，我们可以通过获取临时链接，该链接可以在小程序以外访问；

```
getTempURL: function() {  
  const fileID1 = "cloud://coderwhy-i7b7z.636f-coderwhy-i7b7z-1259322030/123.mp4"  
  wx.cloud.getTempFileURL({  
    fileList: [fileID1]  
  }).then(res => {  
    console.log(res)  
  })  
},
```

# 云存储 - 下载文件

- 如果文件是放在云存储中，那么必然需要有网络的情况下才能访问。
- 某些情况下，我们可能希望把某些重要的文件下载到本地，就可以使用云存储的文件下载了。

```
downloadFile: function() {  
  const fileID = "cloud://coderwhy-i7b7z.636f-coderwhy-i7b7z-1259322030/123.mp4";  
  wx.cloud.downloadFile({  
    fileID: "cloud://coderwhy-i7b7z.636f-coderwhy-i7b7z-1259322030/123.mp4"  
  }).then(res => {  
    console.log(res)  
  })  
},
```

# 云存储 - 删除文件

- 某些文件不再使用时，可以将其从云存储中删除掉，这样可以省略空间

```
deleteFile: function () {  
  const fileID = "cloud://coderwhy-i7b7z.636f-coderwhy-i7b7z-1259322030/123.mp4";  
  wx.cloud.deleteFile({  
    fileList: [fileID]  
  }).then(res => {  
    console.log(res)  
  })  
}
```

## ■ 云函数即在云端（服务器端）运行的函数：

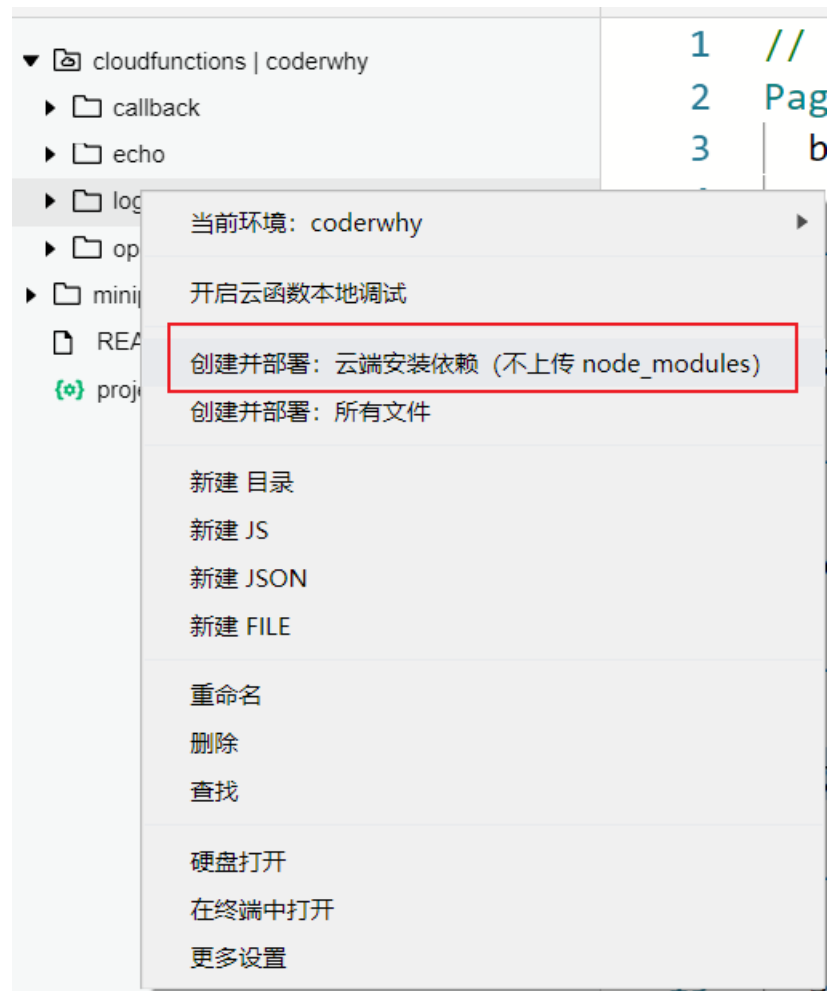
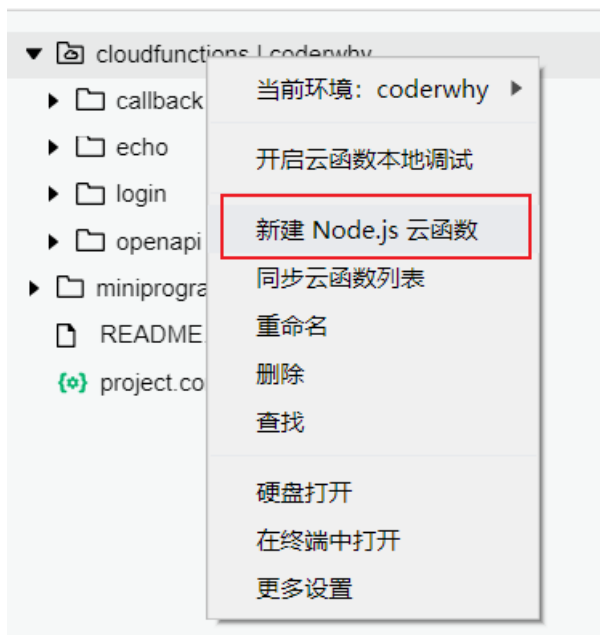
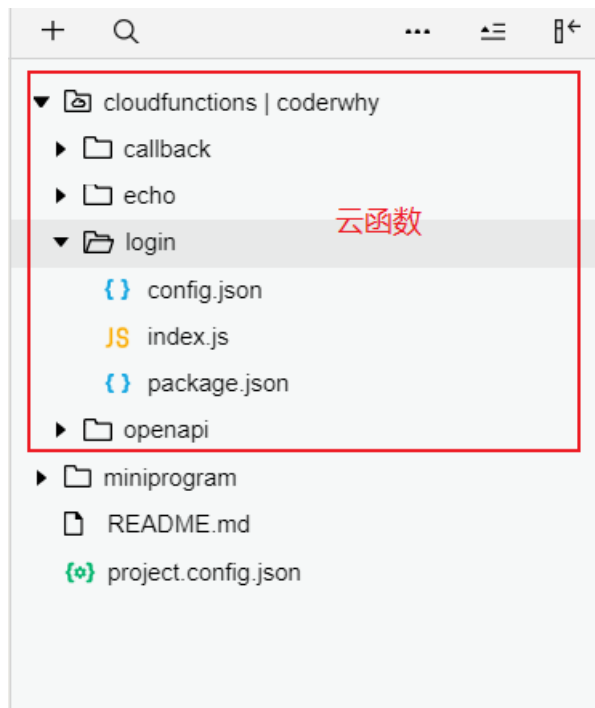
- 在物理设计上，一个云函数可由多个文件组成，占用一定量的 CPU 内存等计算资源；
- 各云函数完全独立；可分别部署在不同的地区；
- 开发者无需购买、搭建服务器，只需编写函数代码并部署到云端即可在小程序端调用；
- 同时云函数之间也可互相调用；

## ■ 云函数的编写方式：

- 一个云函数的写法与一个在本地定义的 JavaScript 方法无异，代码运行在云端 Node.js 中；（需要专门学习 Node.js 吗？）
- 当云函数被小程序端调用时，定义的代码会被放在 Node.js 运行环境中执行；
- 我们可以如在 Node.js 环境中使用 JavaScript 一样在云函数中进行网络请求等操作，而且我们还可以通过云函数后端 SDK 搭配使用多种服务，比如使用云函数 SDK 中提供的数据库和存储 API 进行数据库和存储的操作

- 云开发的云函数的独特优势在于与微信登录鉴权的无缝整合。当小程序端调用云函数时，云函数的传入参数中会被注入小程序端用户的 openid，开发者无需校验 openid 的正确性因为微信已经完成了这部分鉴权，开发者可以直接使用该 openid。

# 创建云函数过程



# 云函数的演练 - 界面搭建

项目 文件 编辑 工具 界面 设置 帮助 微信开发者工具

小程序模式 云函数 编译 预览 真机调试 切后台 清缓存

上传 版本管理 详情

iPhone 5 100% WiFi 模拟操作

云函数基本使用  
云函数登录状态  
云函数操作数据库  
获取小程序码  
发送模板消息

云函数基本使用  
云函数登录状态  
云函数操作数据库  
获取小程序码  
发送模板消息

```
<!--miniprogram/pages/w-cloudfunc/w-cloudfunc.wxml-->
<button bindtap="basicUsing">云函数基本使用</button>

<button bindtap="getOpenID">云函数登录状态</button>

<button bindtap="operateDatabase">云函数操作数据库</button>

<button bindtap="getMiniCode">获取小程序码</button>

<button bindtap="sendTemplateMessage">发送模板消息</button>
```

```
// miniprogram/pages/w-cloudfunc/w-cloudfunc.js
Page({
  basicUsing: function() {
  },
  getOpenID: function() {
  },
  operateDatabase: function() {
  },
  getMiniCode: function() {
  },
  sendTemplateMessage: function() {
  }
})
```

pages/w-cloudfun... 复制 预览 场景值 页面参数

/miniprogram/pages/w-cloudfunc/w-cloudfunc.js 243 B 行 4, 列 5 JavaScript



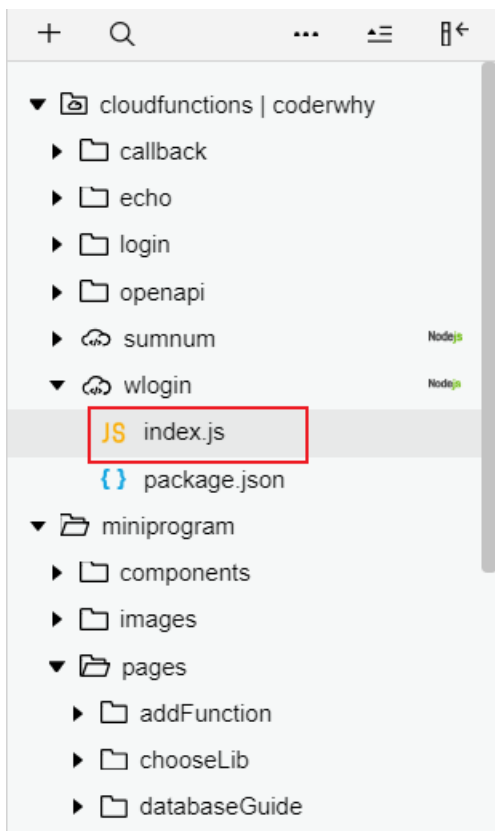
## ■ 案例：让云函数帮我们计算两个数字的和



```
1 // 云函数入口文件
2 const cloud = require('wx-server-sdk')
3
4 cloud.init()
5
6 // 云函数入口函数
7 exports.main = async (event, context) => {
8   return event.num1 + event.num2
9 }
```

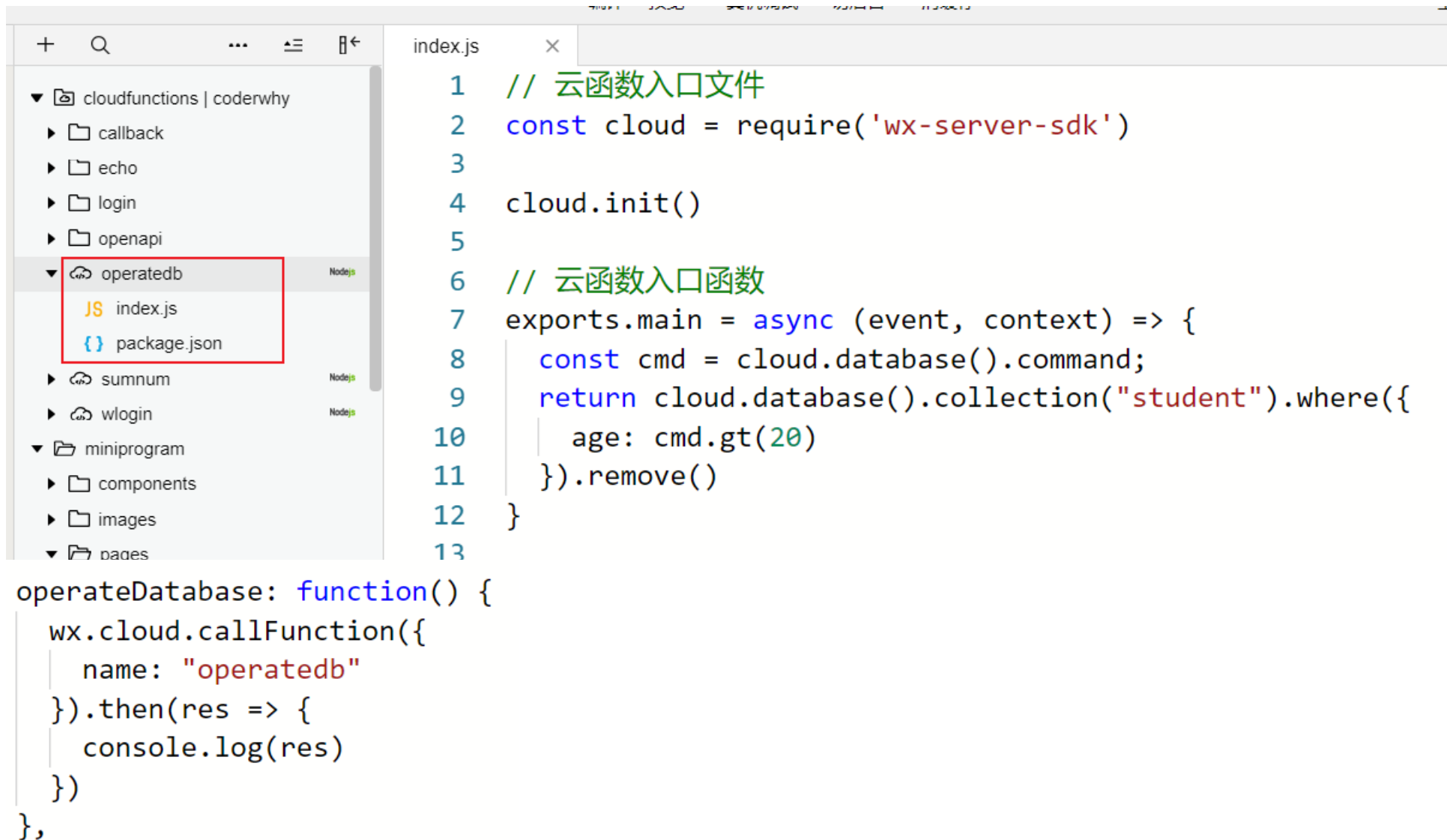
```
basicUsing: function() {
  wx.cloud.callFunction({
    name: "sumnum",
    data: {
      num1: 20,
      num2: 30
    }
  }).then(res => {
    console.log(res)
  })
},
```

# 云函数 – 获取openID



```
index.js
1 // 云函数入口文件
2 const cloud = require('wx-server-sdk')
3
4 cloud.init()
5
6 // 云函数入口函数
7 exports.main = async (event, context) => {
8   const wxContext = cloud.getWXContext()
9
10  return {
11    openid: wxContext.OPENID,
12    appid: wxContext.APPID,
13    unionid: wxContext.UNIONID,
14  }
15 }
```

```
getOpenID: function() {
  wx.cloud.callFunction({
    name: "wlogin"
  }).then(res => {
    console.log(res)
  })
},
```



The image shows a code editor interface. On the left is a file explorer with a tree view. The selected folder is 'operatedb', which contains 'index.js' and 'package.json'. The 'index.js' file is open in the editor. The code in 'index.js' is as follows:

```
1 // 云函数入口文件
2 const cloud = require('wx-server-sdk')
3
4 cloud.init()
5
6 // 云函数入口函数
7 exports.main = async (event, context) => {
8   const cmd = cloud.database().command;
9   return cloud.database().collection("student").where({
10     age: cmd.gt(20)
11   }).remove()
12 }
13
```

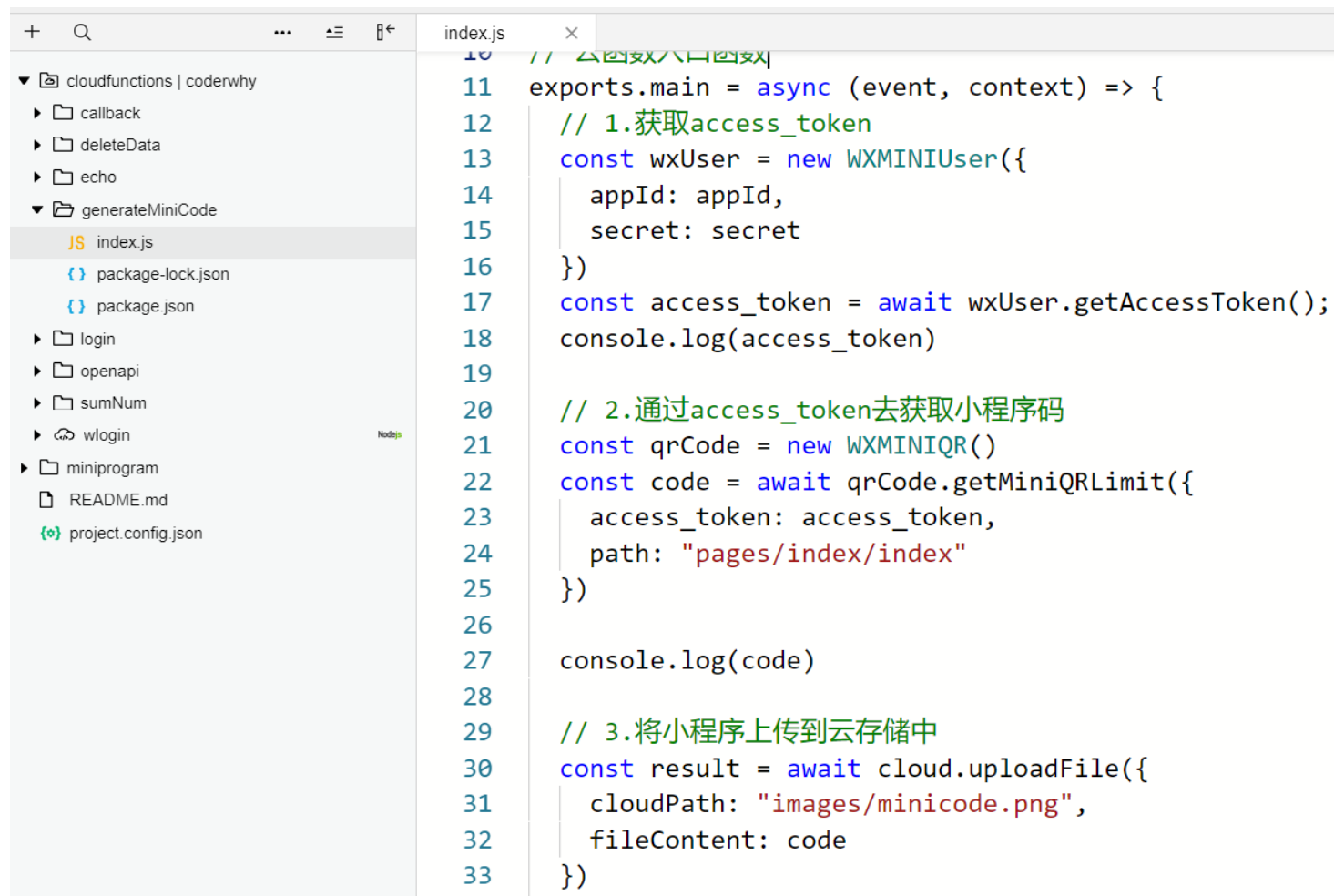
Below the editor, there is a code block showing the 'operateDatabase' function:

```
operateDatabase: function() {
  wx.cloud.callFunction({
    name: "operatedb"
  }).then(res => {
    console.log(res)
  })
},
```

## ■ 生成小程序码的逻辑:

- ❑ 1. 通过appId和secret换取access\_token
- ❑ 2. 通过access\_token请求小程序码
- ❑ 3. 将小程序码上传到云存储中
- ❑ 4. 将上传后的fileID返回给客户端保存和显示

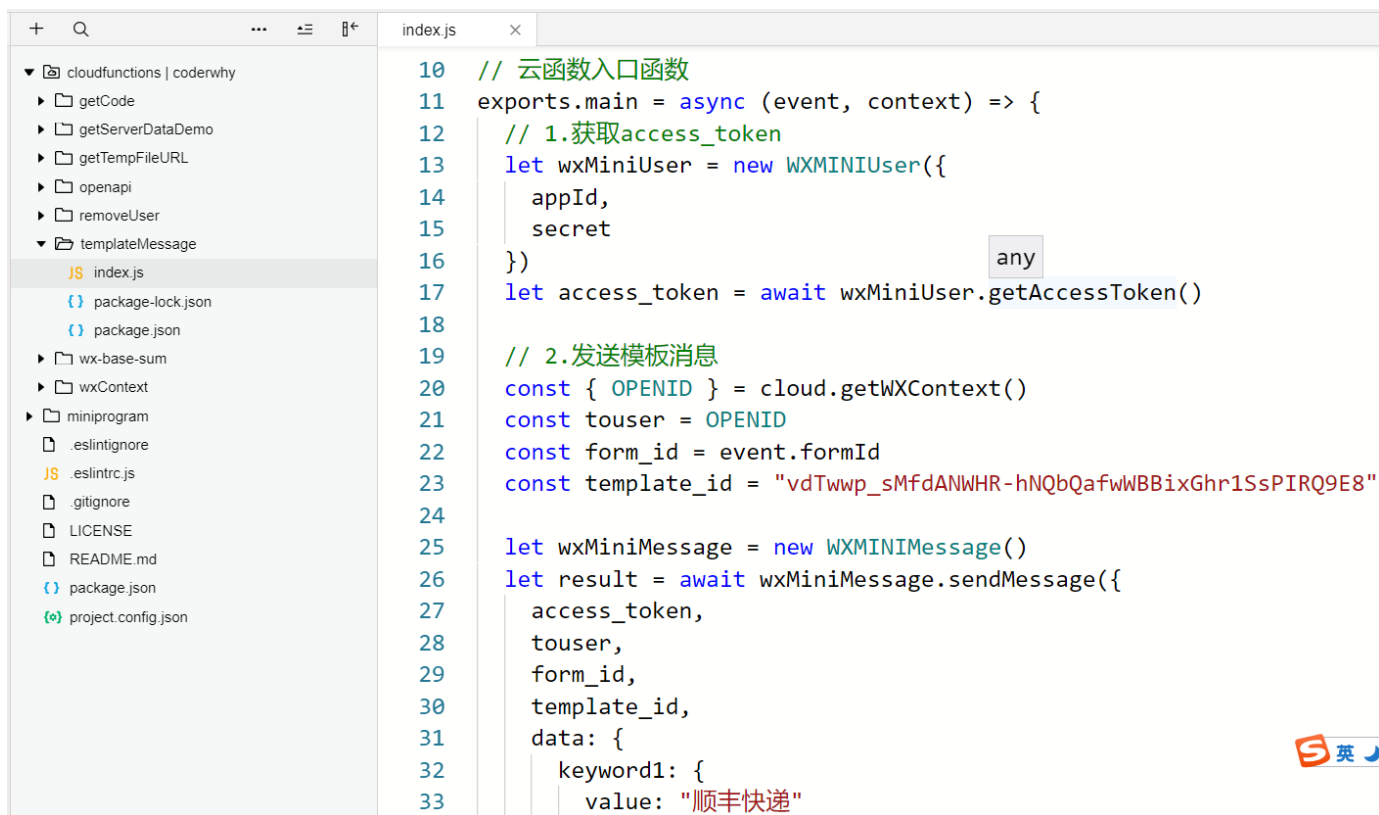
```
getMiniCode: function() {  
  wx.cloud.callFunction({  
    name: "generateMiniCode",  
    success: (res) => {  
      console.log(res.result.fileID)  
      this.setData({  
        miniCodeFileID: res.result.fileID  
      })  
    }  
  })  
}
```



```
index.js  
10 // 云函数入口函数  
11 exports.main = async (event, context) => {  
12   // 1. 获取access_token  
13   const wxUser = new WXMINIUser({  
14     appId: appId,  
15     secret: secret  
16   })  
17   const access_token = await wxUser.getAccessToken();  
18   console.log(access_token)  
19  
20   // 2. 通过access_token去获取小程序码  
21   const qrCode = new WXMINIQR()  
22   const code = await qrCode.getMiniQRLimit({  
23     access_token: access_token,  
24     path: "pages/index/index"  
25   })  
26  
27   console.log(code)  
28  
29   // 3. 将小程序上传到云存储中  
30   const result = await cloud.uploadFile({  
31     cloudPath: "images/minicode.png",  
32     fileContent: code  
33   })
```

## ■ 发送消息模板的逻辑：

- ❑ 1. 获取formid（必须在表单中才可以）
- ❑ 2. 使用appid、secret换取access\_token
- ❑ 3. 调用对应的发送模板消息方法，传入对应的参数



```
10 // 云函数入口函数
11 exports.main = async (event, context) => {
12   // 1.获取access_token
13   let wxMiniUser = new WXMLIUser({
14     appId,
15     secret
16   })
17   let access_token = await wxMiniUser.getAccessToken()
18
19   // 2.发送模板消息
20   const { OPENID } = cloud.getWXContext()
21   const touser = OPENID
22   const form_id = event.formId
23   const template_id = "vdTwwp_sMfdANWHR-hNQbQafwWBBixGhr1SsPIRQ9E8"
24
25   let wxMiniMessage = new WXMLIMessage()
26   let result = await wxMiniMessage.sendMessage({
27     access_token,
28     touser,
29     form_id,
30     template_id,
31     data: {
32       keyword1: {
33         value: "顺丰快递"
```