微信云开发提供了多个基础能力，以下对各个主要能力介绍。

**数据库**

云开发提供了一个 JSON 数据库，顾名思义，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（相当于关系型数据中的表），集合可看做一个 JSON 数组，数组中的每个对象就是一条记录，记录的格式是 JSON 对象。

在图书信息中，我们用 title, author 来记录图书标题和作者，用 characters 数组来记录书中的主要人物，用 publishInfo 来记录图书的出版信息。在其中我们可以看到，字段既可以是字符串或数字，还可以是对象或数组，就是一个 JSON 对象。

每条记录都有一个 \_id 字段用以唯一标志一条记录、一个 \_openid 字段用以标志记录的创建者，即小程序的用户。需要特别注意的是，在管理端（控制台和云函数）中创建的不会有 \_openid 字段，因为这是属于管理员创建的记录。开发者可以自定义 \_id，但不可自定义和修改 \_openid 。\_openid 是在文档创建时由系统根据小程序用户默认创建的，开发者可使用其来标识和定位文档。

数据库 API 分为小程序端和服务端两部分，小程序端 API 拥有严格的调用权限控制，开发者可在小程序内直接调用 API 进行非敏感数据的操作。对于有更高安全要求的数据，可在云函数内通过服务端 API 进行操作。云函数的环境是与客户端完全隔离的，在云函数上可以私密且安全的操作数据库。

数据库 API 包含增删改查的能力，使用 API 操作数据库只需三步：获取数据库引用、构造查询/更新条件、发出请求。以下是一个在小程序中查询数据库的发表于美国的图书记录的例子：

**存储**

云开发提供了一块存储空间，提供了上传文件到云端、带权限管理的云端下载能力，开发者可以在小程序端和云函数端通过 API 使用云存储功能。

在小程序端可以分别调用 wx.cloud.uploadFile 和 wx.cloud.downloadFile 完成上传和下载云文件操作。下面简单的几行代码，即可实现在小程序内让用户选择一张图片，然后上传到云端管理的功能：

// 让用户选择一张图片

wx.chooseImage({

success: chooseResult => {

// 将图片上传至云存储空间

wx.cloud.uploadFile({

// 指定上传到的云路径

cloudPath: 'my-photo.png',

// 指定要上传的文件的小程序临时文件路径

filePath: chooseResult.tempFilePaths[0],

// 成功回调

success: res => {

console.log('上传成功', res)

},

})

},

})

上传完成后可在控制台中看到刚上传的图片。

更多的存储 API 和管理，可以参考[存储指引](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/storage.html)章节。

**云函数**

云函数是一段运行在云端的代码，无需管理服务器，在开发工具内编写、一键上传部署即可运行后端代码。

小程序内提供了专门用于云函数调用的 API。开发者可以在云函数内使用 [wx-server-sdk](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/functions/wx-server-sdk.html) 提供的 [getWXContext](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/reference-sdk-api/utils/Cloud.getWXContext.html) 方法获取到每次调用的上下文（appid、openid 等），无需维护复杂的鉴权机制，即可获取天然可信任的用户登录态（openid）。

比如我们如下定义一个云函数，命名为 add ，功能是将传入的两个参数 a 和 b 相加：

// index.js 是入口文件，云函数被调用时会执行该文件导出的 main 方法

// event 包含了调用端（小程序端）调用该函数时传过来的参数，同时还包含了可以通过 getWXContext 方法获取的用户登录态 `openId` 和小程序 `appId` 信息

const cloud = require('wx-server-sdk')

exports.main = async (event, context) => {

let { userInfo, a, b} = event

let { OPENID, APPID } = cloud.getWXContext() // 这里获取到的 openId 和 appId 是可信的

let sum = a + b

return {

OPENID,

APPID,

sum

}

}

在开发者工具中上传部署云函数后，我们在小程序中可以这么调用：

除了部署云函数进行调用外，我们还支持[云函数本地调试](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/functions/local-debug.html)，可以不用部署云函数即可测试

wx.cloud.callFunction({

// 需调用的云函数名

name: 'add',

// 传给云函数的参数

data: {

a: 12,

b: 19,

},

// 成功回调

complete: console.log

})

// 当然 promise 方式也是支持的

wx.cloud.callFunction({

name: 'add',

data: {

a: 12,

b: 19

}

}).then(console.log)

云函数的请求参数 data 有大小限制（100K）

如需在云函数中操作数据库、管理云文件、调用其他云函数等操作，可使用官方提供的 npm 包 [wx-server-sdk](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/reference-server-api/) 进行操作。

更多的云函数管理和 API，可以参考[云函数指引](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/functions.html)章节。

**云调用**

云调用是云开发提供的基于云函数使用小程序开放接口的能力，支持在云函数调用服务端开放接口，如发送模板消息、获取小程序码等操作都可以在云函数中完成，详情可见具体[开发指引](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/openapi/openapi.html)。

**HTTP API**

云开发资源也可以通过 HTTP 接口访问，即在小程序外访问，接口见[HTTP API 文档](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/reference-http-api/)。

通过这个章节，我们已经了解了云开发是什么，提供了哪些能力，能做什么，接下来跟着我们一起进入[开发指引](https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/)的章节，看看如何上手开发吧！