# 笔记07

## 发布上线

### 如果当前使用的是测试号开发的话，那么先把测试号换成自己真实的appid

### 将所有的网络请求的http改成https

### 把所有的域名在后台进行配置

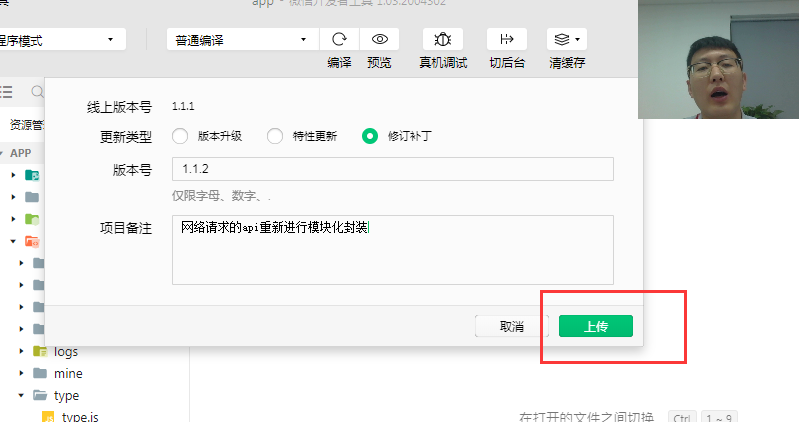


### 设置基本信息

正常填写主体信息即可，主要注意服务类目的问题



### 上传代码



### 提交审核



## 服务器相关内容(了解)

### 服务器/虚拟主机

在阿里云上购买，登录，使用ftp账号，进行连接远程服务器，进行项目部署

### 域名

购买域名

### 备案



### 域名绑定和解析

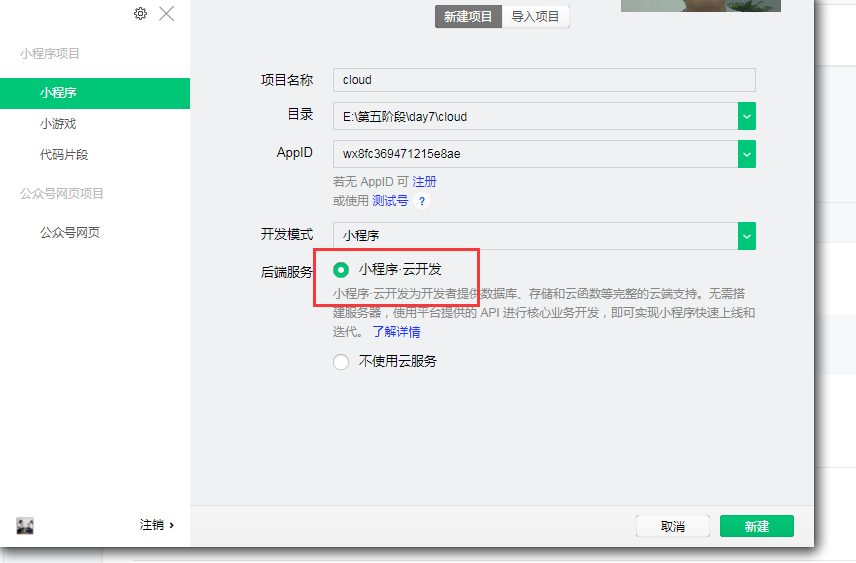
绑定：

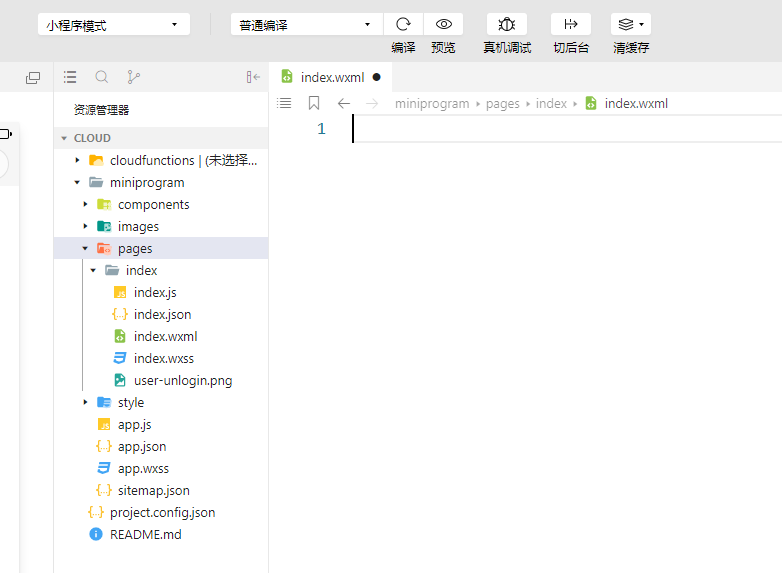


## 云开发

| **能力** | **作用** | **说明** |
| --- | --- | --- |
| 云函数 | 无需自建服务器 | 在云端运行的代码，微信私有协议天然鉴权，开发者只需编写自身业务逻辑代码 |
| 数据库 | 无需自建数据库 | 一个既可在小程序前端操作，也能在云函数中读写的 JSON 数据库 |
| 存储 | 无需自建存储和 CDN | 在小程序前端直接上传/下载云端文件，在云开发控制台可视化管理 |
| 云调用 | 原生微信服务集成 | 基于云函数免鉴权使用小程序开放接口的能力，包括服务端调用、获取开放数据等能力 |

### 1.创建一个云开发的项目

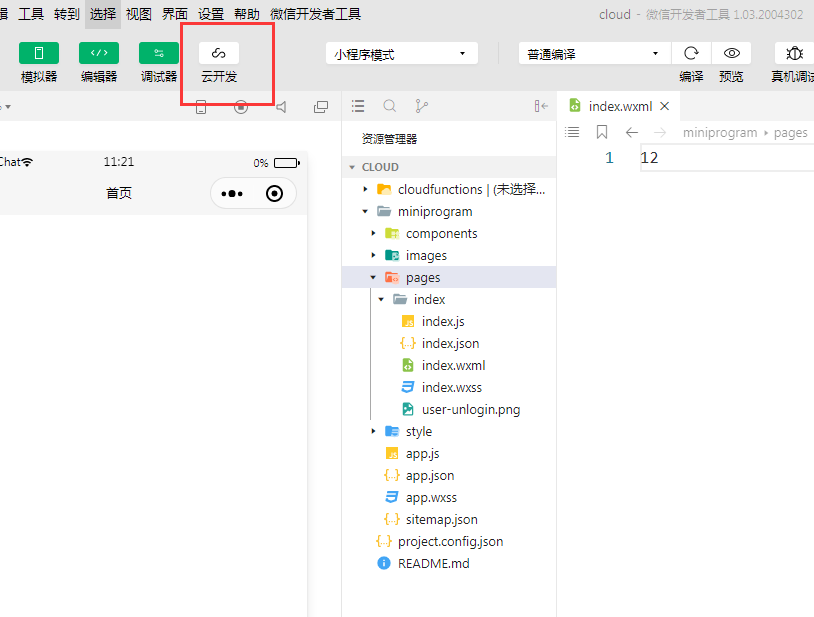




注意：

1. 使用云开发，必须使用费非测试账号
2. 使用的数据库是mongodb， 环境 nodejs

### 开通云能力

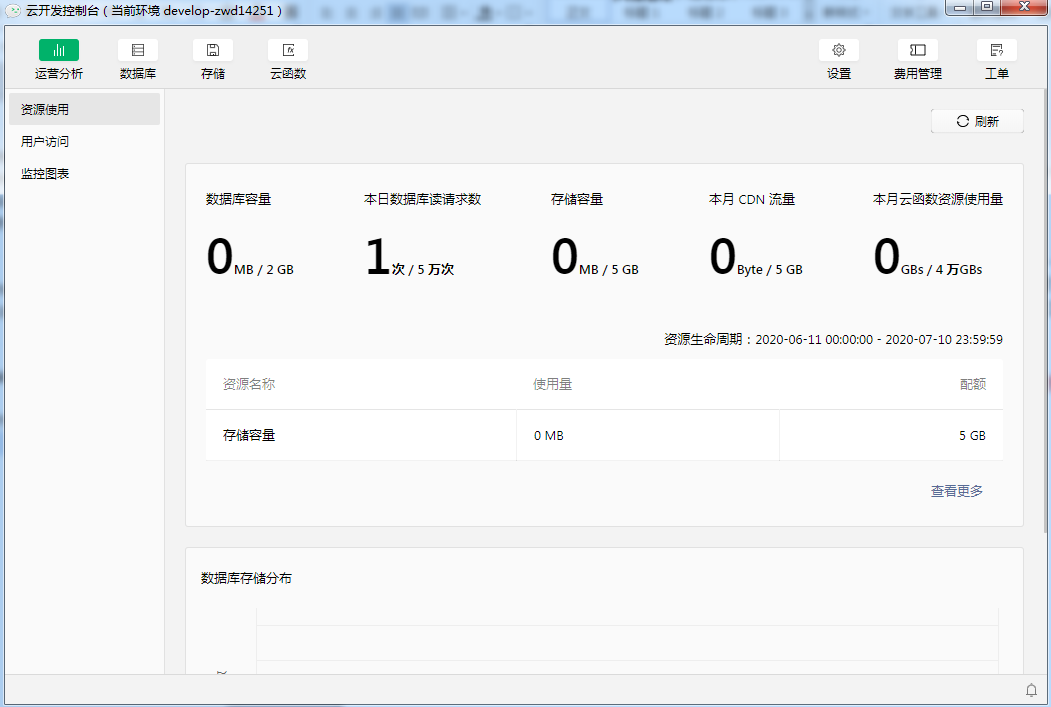


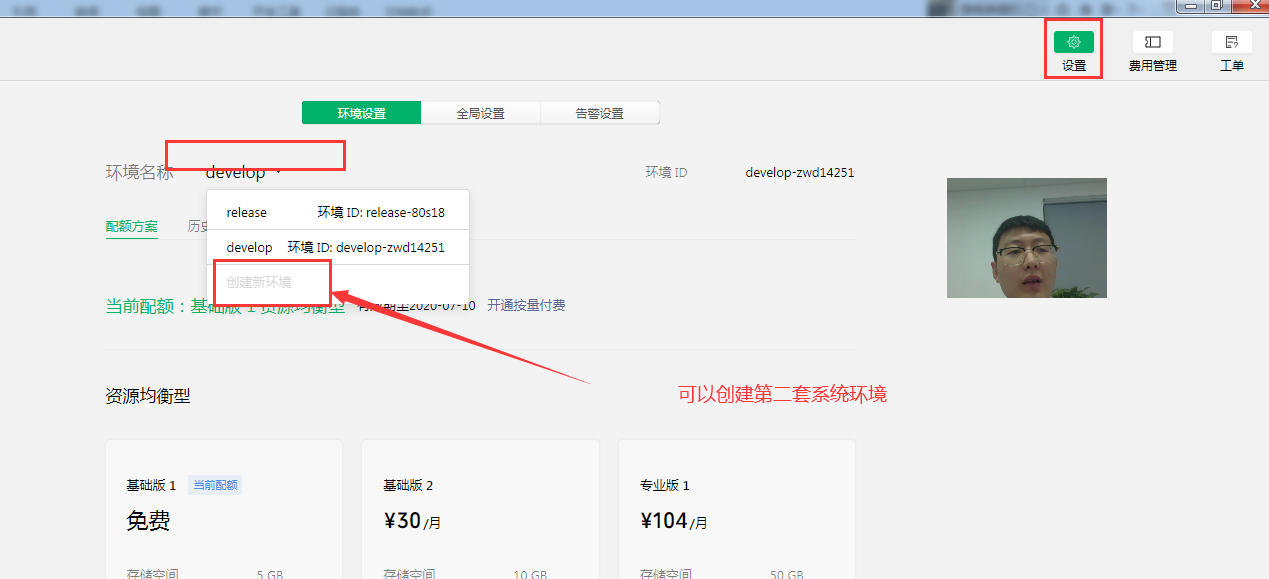




默认配额下可以创建两个环境，各个环境相互隔离，每个环境都包含独立的数据库实例、存储空间、云函数配置等资源。每个环境都有唯一的环境 ID 标识，初始创建的环境自动成为默认环境。

【开通云开发之后，送2套环境，互相独立的，每一套里面都有 数据库，存储，云函数】





### 基本操作

#### 云能力初始化



### 数据库基本操作

#### 进行数据库初始化操作

  const  db = wx.cloud.database(); // 获取数据库引用

#### 添加操作

 // 执行添加

      db.collection('user').add({

        // 插入的数据

        data:{

          username,password

        },

        success(res){

          // console.log(res)

          if(res.errMsg == "collection.add:ok"){

            wx.showToast({

              title: '注册成功',

            })

          }

        }

      })

#### 数据库查询操作

根据where条件进行查询 返回值是一个数组

  find(){

      // 查询所有

      let \_this = this;

      // eq  lt gt    指令对象

      const \_ = db.command;

      db.collection('user').where({

        // password:"123456"

        // age:\_.gt(17)

      }).get({

        success(res){

          // console.log(res)

          \_this.setData({

             users:res.data

          })

        }

      })

  },

根据id进行查询，返回值是一个对象

onLoad: function (options) {

      let id = options.id;

      let  \_this = this;

       db.collection('user').doc(id).get().then(res=>{

        //  console.log(res)

        \_this.setData({

          users:res.data

        })

       })

  },

#### 修改

1. 修改一条数据 ， 使用doc（id）

 // 执行修改

  doupdate(e){

      // 1.获取要修改的信息

      let  {username,password} = e.detail.value;

      //2. 执行修改操作

      db.collection('user').doc(this.data.id).update({

        data:{

          username,password

        },

        success(res){

            // console.log(res)

            if(res.errMsg == "document.update:ok"){

                wx.showToast({

                  title: '修改成功',

                })

                setTimeout(()=>{

                    wx.navigateBack({

                      delta:1,

                      complete: (res) => {},

                    })

                },1500)

            }

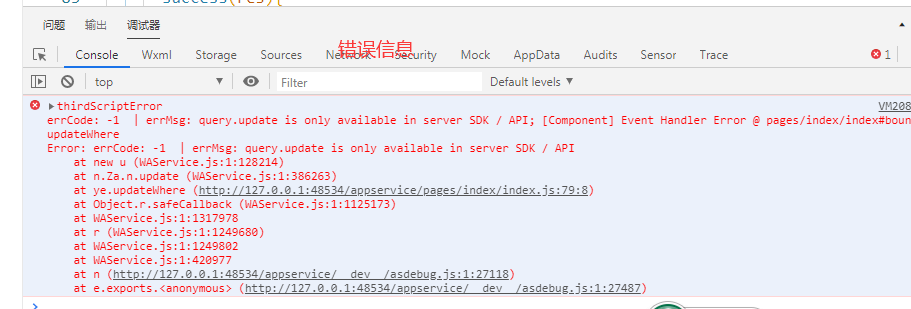
        }

      })

  }

1. 修改多个





**在云端修改多个数据；**

小程序端的调用

   let \_this = this;

    wx.cloud.callFunction({

      name:"updateWhere",

      data:{

        table:"user",

        where:{

          password:"123456"

        },

        info:{

          username:"admin"

        }

      },

      success(res){

        // console.log(res);

        if(res.result.errMsg == "collection.update:ok"){

            \_this.find()

        }

      }

    })

云函数的编写：

// 云函数入口文件

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

cloud.init({

  env:"develop-zwd14251"

})

// 在云端再去初始化数据库，获取数据库引用

const db = cloud.database();  // 小程序端  wx.cloud.database()

// 云函数入口函数

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

  //  event  table 集合     where   条件   data  要修改的数据

  // 修改多条数据

  return  db.collection(event.table).where(

      // 执行修改条件

    event.where

    // {

    //   password:"123456"

    // }

  ).update({

     //执行修改的数据

    data:event.info

  })

}

#### 删除

1. 删除一个

 // 删除一条数据

  del(e){

    let  id = e.target.dataset.id;

    let \_this = this;

    db.collection('user').doc(id).remove({

      success:(res)=>{

        // console.log(res)

        if(res.errMsg == "document.remove:ok"){

           \_this.find()

        }

      }

    })

  }

1. 删除多个

删除多个和修改多个一样，在小程序端都会报错，只能在云端进行删除

注意：

// 封装云函数调用

const   cloudfun = (name,data)=>{

  return  new Promise((resolve,reject)=>{

    wx.cloud.callFunction({

      name, // 云函数的名称

      data, // 集合的名称  条件

      success:resolve,

      fail:reject

    })

  })

}

module.exports  =  cloudfun;

// 根据条件删除的方法

  async delWhere() {

    let data = {

      table: "user",

      where: {

        password: "123456"

      }

    }

    const delResult = await cloudfun("delWhere", data)

    // console.log(delResult,'1234567890')

    if(delResult.result.errMsg == "collection.remove:ok"){

        this.find()

    }

  }

### 云函数的基本操作

#### 返回俩个数之和的云函数

* 在云函数的根目录里，右键，选择创建nodejs云函数选项即可
* 在云函数的index。Js中进行编写代码
* 右键选择当前云函数的文件目录，上传并部署：云端安装依赖选项
* 在小程序端调用云函数

 // 在小程序端如何调用云端云函数

    wx.cloud.callFunction({

      name:"total", //要调用的云函数的名称

      data:{}, //传递给云函数的参数

      success(res){

        console.log(res)

      }

    })

### 存储

将 视频，音频，文件上传到指定的存储位置，返回生成一个地址，将地址存储到指定的数据库的表中；

作业：

继续写小说阅读项目，

联系编写云开发项目 （用户管理）