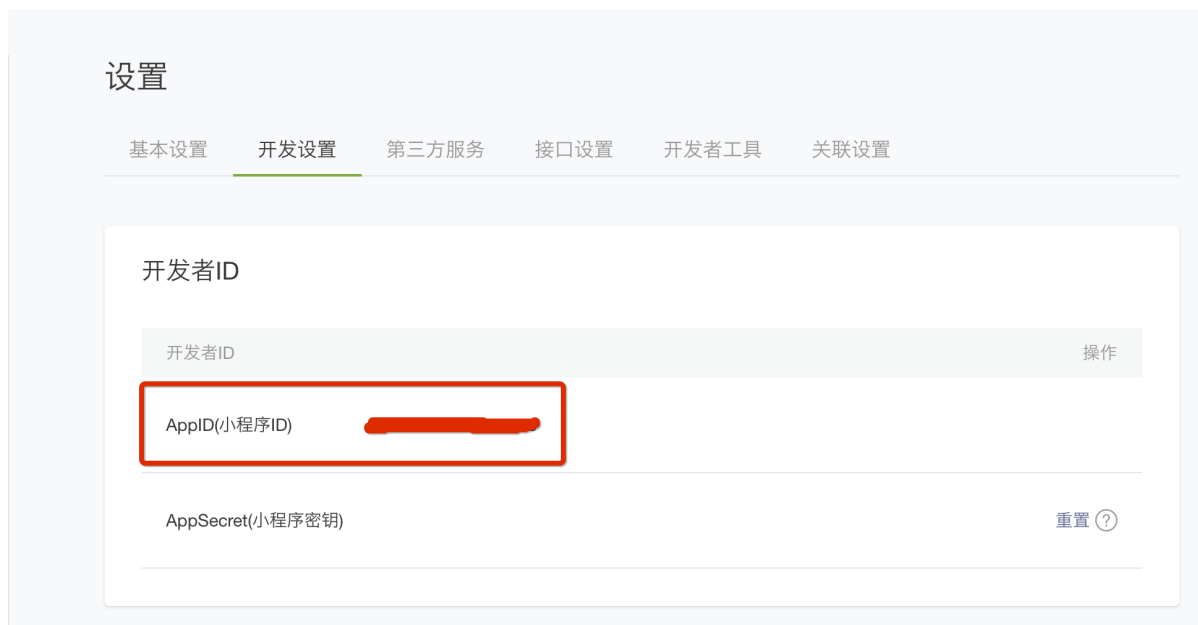


基于云开发制作电商小程序

准备工作

1. 已经申请小程序，获取小程序 AppID 和 Secret 在[小程序管理后台](#)中，【设置】 -> 【开发设置】 下可以获取微信小程序 AppID 和 Secret。



2. 微信支付商户号，获取商户号和商户密钥

在[微信支付商户管理平台](#)中，【账户中心】 -> 【商户信息】 下可以获取微信支付商户号。



在【账户中心】 -> 【API安全】 下可以设置商户密钥。



3. 微信开发者 IDE(下载)
4. 下载电商小程序代码包
5. 运行环境 Node8.9 或以上

效果预览



知识点

1. 学习如何用云开发控制台上传图片、录入商品数据。
2. 学习如何用云开发插入、读取数据。
3. 学习如何用 wx-js-utils 和云函数实现小程序微信支付逻辑。

任务一：创建小程序·云开发环境

任务目标: 创建小程序·云开发环境，用于后面存储信息和开发云函数。

开发者工具创建项目

打开微信开发者工具，创建一个新的小程序项目，项目目录选择电商小程序Demo的目录，AppID填写已经申请公测资格的小程序对应的AppID。

← 小程序项目管理



小程序项目

编辑、调试小程序

项目目录

选择项目目录



AppID

已申请小程序·云开发公测资格的 AppID

若无 AppID 可 [注册](#)

或使用测试号: [小程序](#) / [小游戏](#)

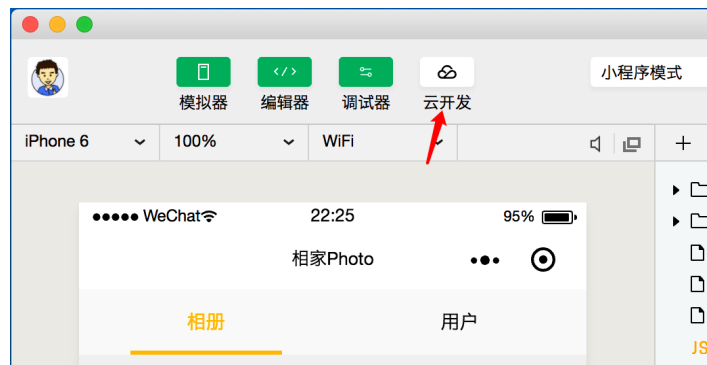
项目名称

任意填写项目名称

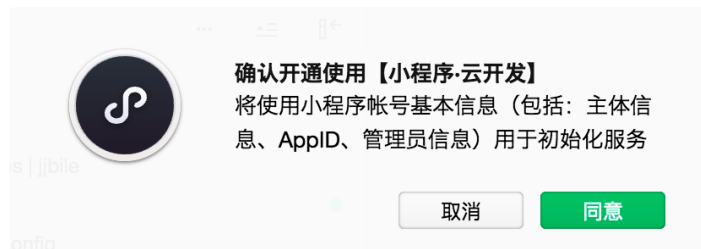
确定

开通云开发环境

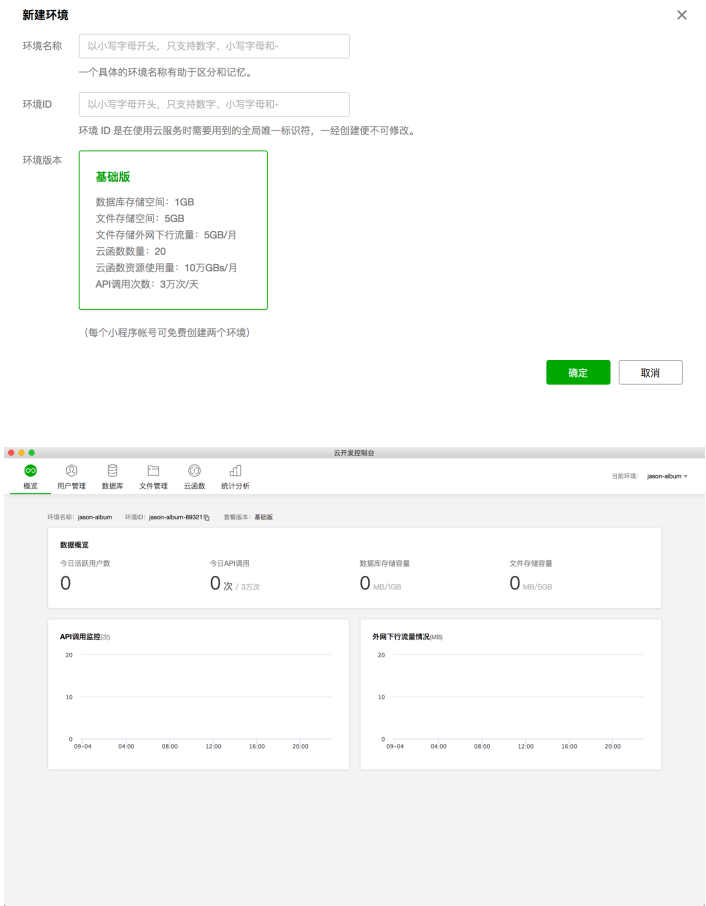
1. 点击开发者工具上的【云开发】按钮



2. 点击【同意】



3. 填写环境名称和环境ID，点击【确定】创建环境，即可进入云开发控制台。



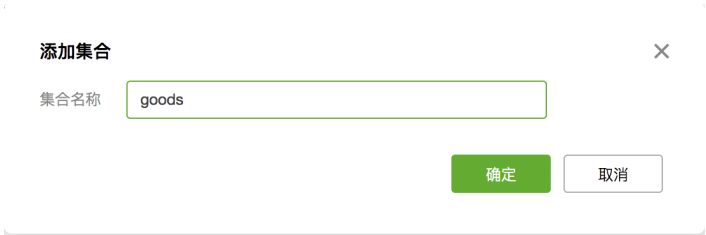
创建数据库

电商小程序会使用到云开发提供的数据库能力，数据库使用的是MongoDB，需要优先创建一个集合，方便之后使用。

- 1. 打开云开发控制台，点击菜单栏中的【数据库】，然后点击左侧边栏的【添加集合】按钮



- 2. 分别输入集合名称 "goods" 和 "order", 然后点击确定即可创建集合。



添加集合

×

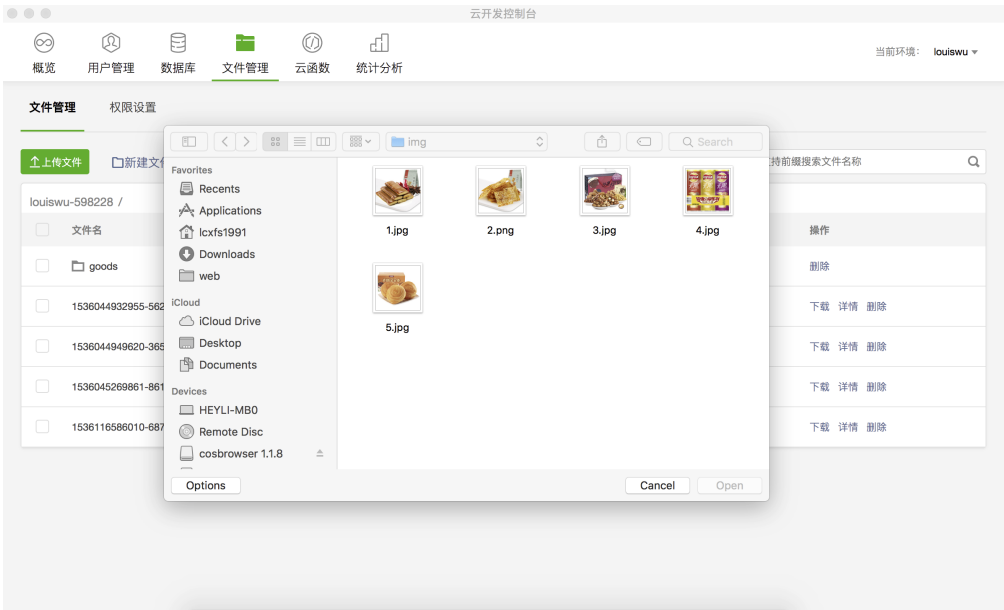
集合名称

确定

取消

录入商品数据

1. 在云开发控制台，点击菜单栏中的【文件管理】，然后点击左侧的【新建文件夹】，新建一个目录 goods，点击进入 goods 目录后，点击左侧的【上传文件】按钮，将小程序项目目录 ./img 中的商品图片，都选择进行上传。



上传完毕后，分别点击红框框选的复制按钮，将这些文件的 fileId 保存下来。

文件名	fileid	大小	更新时间	操作
<input type="checkbox"/> 1.jpg	cloud://[redacted]/goods/1.jpg	22.07KB	2018-09-04 16:27:54	下载 详情 删除
<input type="checkbox"/> 2.png	cloud://[redacted]/goods/2.png	74.97KB	2018-09-04 16:27:54	下载 详情 删除
<input type="checkbox"/> 3.jpg	cloud://[redacted]/goods/3.jpg	30.32KB	2018-09-04 16:27:54	下载 详情 删除
<input type="checkbox"/> 4.jpg	cloud://[redacted]/goods/4.jpg	25.82KB	2018-09-04 16:27:54	下载 详情 删除
<input type="checkbox"/> 5.jpg	cloud://[redacted]/goods/5.jpg	19.59KB	2018-09-04 16:27:54	下载 详情 删除

3. 返回【数据库】，在 goods collection 下，点击【添加记录】，按以下格式录入数据。

添加记录

集合名称 /goods

文档ID

将使用系统自动生成的ID 输入ID

字段		类型	值	
name	=	string	鸡蛋干	-
商品名称				
price	=	number	1	-
价格: 单位 (分)				
pic	=	string	cloud://louiswu-	-
fileid				
timestamp	=	date		-
创建日期				
2018-09-05				
16:06:57				

+

以下是数据例子

```
{
  "name": "鸡蛋干",
  "pic": "", //fileid
  "price": "1", // 价格: 单位 (分) ,
  "timestamp": "Tue Sep 04 2018 16:31:34 GMT+0800 (CST)" // Date 对象数据
}
{
  "name": "辣条",
  "pic": "", //fileid
  "price": "1", // 价格: 单位 (分) ,
  "timestamp": "Tue Sep 04 2018 16:31:34 GMT+0800 (CST)" // Date 对象数据
}
{
  "name": "坚果",
  "pic": "", //fileid
  "price": "1", // 价格: 单位 (分) ,
  "timestamp": "Tue Sep 04 2018 16:31:34 GMT+0800 (CST)" // Date 对象数据
}
{
```

```
    "name": "薯片",
    "pic": "", //fileid
    "price": "1", // 价格: 单位 (分) ,
    "timestamp": "Tue Sep 04 2018 16:31:34 GMT+0800 (CST)" // Date 对象数据
  }
  {
    "name": "包子",
    "pic": "", //fileid
    "price": "1", // 价格: 单位 (分) ,
    "timestamp": "Tue Sep 04 2018 16:31:34 GMT+0800 (CST)" // Date 对象数据
  }
}
```

任务二：读取数据与发起订单

读取数据

1. 将下面代码，输入到 `client/pages/list/index.js` 中的 `getGoodsList` 方法中，通过此方法，可以获取所有商品的数据。

```
const db = wx.cloud.database();
const result = await db.collection('goods').get();

let data = result.data || [];

this.setData({
  goods: data
});
```

发起订单

1. 将下面代码，输入到 `client/pages/list/index.js` 中的 `makeOrder` 方法中，通过此方法，可以调用云函数进行订单发起。

```
wx.showLoading({
  title: '正在下单',
});

let id = e.target.dataset.goodid;
const { result } = await wx.cloud.callFunction({
  name: 'pay',
  data: {
    type: 'unifiedorder',
    data: {
      goodId: id
    }
  }
});

const data = result.data;
```

```
wx.navigateTo({
  url: `/pages/result/index?id=${data.out_trade_no}`
});
```

2. 创建云函数 `pay`

在 `cloud/functions/pay` 目录下，运行以下命令，安装依赖。

```
npm i --production
```

2. 填写腾讯云、微信商户与微信小程序相关配置 新建 `cloud/functions/pay/config/index.js`，并填入腾讯云的 `AppId`, `SecretId`, `SecretKey`:

```
module.exports = {
  mpAppId: '', // 小程序 AppID
  envName: '', // 小程序云开发环境ID
  MCHID: '', // 商户号
  KEY: '', // 商户密钥
  TIMEOUT: 10000 // 毫秒
};
```

3. 将下面代码，输入到 `cloud/functions/pay/index.js` 中的 `switch` 代码块中，通过此 `case`，可以进行订单的发起。

```
case 'unifiedorder': {
  // 统一下单
  const { goodId } = data;

  let goods = await goodCollection.doc(goodId).get();

  if (!goods.data.length) {
    return new Res({
      code: 1,
      message: '找不到商品'
    });
  }

  let good = goods.data[0];

  const curTime = Date.now();
  const tradeNo = `${goodId}-${curTime}`;
  const body = good.name;
  const spbill_create_ip = ip.address() || '127.0.0.1';
  const notify_url = 'http://www.qq.com'; // '127.0.0.1';
  const total_fee = good.price;
  const time_stamp = '' + Math.ceil(Date.now() / 1000);
```



```
const out_trade_no = `${tradeNo}`;
const sign_type = WXPConstants.SIGN_TYPE_MD5;

let orderParam = {
  body,
  spbill_create_ip,
  notify_url,
  out_trade_no,
  total_fee,
  openid,
  trade_type: 'JSAPI',
  timeStamp: time_stamp,
};

const {
  return_code,
  ...restData
} = await pay.unifiedOrder(orderParam);

let order_id = null;

if (return_code === 'SUCCESS' && restData.result_code === 'SUCCESS') {
  const {
    prepay_id,
    nonce_str
  } = restData;

  // 下面是进行微信小程序支付的
  const sign = signMiniPay({
    mpAppId,
    KEY,
    nonce_str,
    prepay_id,
    time_stamp
  });

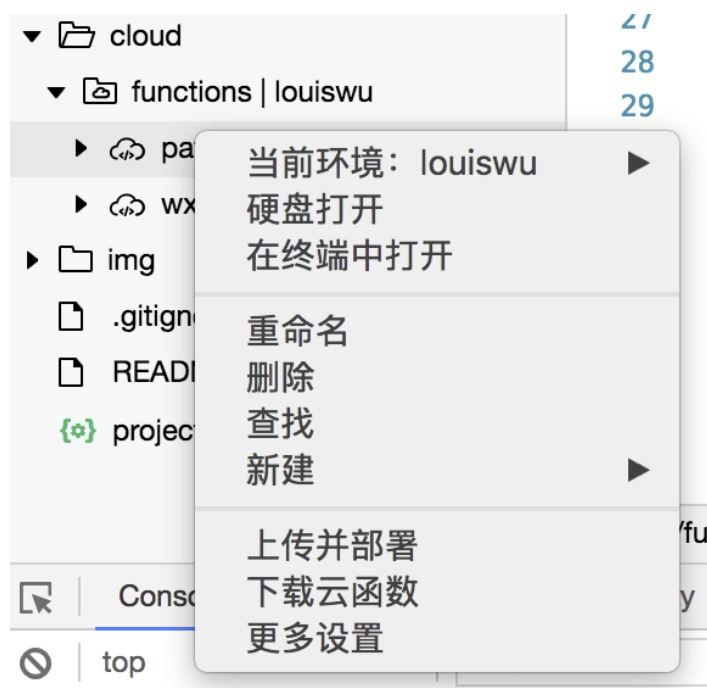
  let orderData = {
    out_trade_no,
    time_stamp,
    nonce_str,
    sign,
    sign_type,
    body,
    total_fee,
    prepay_id,
    sign,
    status: 0, // 0表示刚创建订单
    _openid: openid,
  };

  let order = await orderCollection.add(orderData);

  order_id = order.id;
}
```

```
return new Res({
  code: return_code === 'SUCCESS' ? 0 : 1,
  data: { out_trade_no, time_stamp, order_id, ...restData }
});
}
```

5. 上传云函数 在微信开发者工具中，右键点击云函数 `pay`，选取好环境后，上传云函数（如果是新建，会显示上传并创建）。



任务二：发起支付与模板消息通知

发起支付

1. 将下面代码，输入到 `client/pages/result/index.js` 中的 `pay` 方法中，通过此方法，可以调起微信支付。

```
let orderQuery = this.data.order;
let out_trade_no = this.data.out_trade_no;
let _this = this;

const {
  time_stamp,
  nonce_str,
  sign,
  sign_type,
  prepay_id,
  body,
  total_fee
} = orderQuery;

wx.requestPayment({
```

```
timeStamp: time_stamp,
nonceStr: nonce_str,
package: `prepay_id=${prepay_id}`,
signType: 'MD5',
paySign: sign,
async success(res) {
  wx.showLoading({
    title: '正在支付',
  });

  wx.showToast({
    title: '支付成功',
    icon: 'success',
    duration: 1500,
    async success() {
      _this.getOrder();

      await wx.cloud.callFunction({
        name: 'pay',
        data: {
          type: 'payorder',
          data: {
            body,
            prepay_id,
            out_trade_no,
            total_fee
          }
        }
      });
      wx.hideLoading();
    }
  });
},
fail: function (res) { }
})
```

2. 将下面代码，输入到 `cloud/functions/pay/index.js` 中的 `switch` 代码块中，通过此 `case`，可以进行微信支付。

```
case 'payorder': {
  const {
    out_trade_no,
    prepay_id,
    body,
    total_fee
  } = data;

  const { return_code, ...restData } = await pay.orderQuery({
    out_trade_no
  });
}
```

```

    if (restData.trade_state === 'SUCCESS') {
      let result = await orderCollection
        .where({ out_trade_no })
        .update({
          status: 1,
          trade_state: restData.trade_state,
          trade_state_desc: restData.trade_state_desc
        });

      let curDate = new Date();
      let time = `${curDate.getFullYear()}-${curDate.getMonth() +
        1}-${curDate.getDate()}
        ${curDate.getHours()}:${curDate.getMinutes()}:${curDate.getSeconds()}`;

      let messageResult = await app.callFunction({
        name: 'wxmessage',
        data: {
          formId: prepay_id,
          openId: userInfo.openId,
          appId: userInfo.appId,
          page: `/pages/result/index?id=${out_trade_no}`,
          data: {
            keyword1: {
              value: out_trade_no // 订单号
            },
            keyword2: {
              value: body // 物品名称
            },
            keyword3: {
              value: time // 支付时间
            },
            keyword4: {
              value: (total_fee / 100) + "元" // 支付金额
            }
          }
        }
      });
    }

    return new Res({
      code: return_code === 'SUCCESS' ? 0 : 1,
      data: restData
    });
  }
}

```

模板消息通知

1. 配置小程序相关信息 新建 `cloud/functions/wxmessage/config/index.js`，并填写小程序相关配置。

```
module.exports = {  
  secret: '', // 小程序 secret  
  templateId: '' // 模板 id  
};
```

模板 id 可以在小程序管理后台的【模板消息】->【我的模板】中获取，如果仍未添加，则可以在【模板库】中先添加 **支付成功通知** 模板。



2. 在 `cloud/functions/wxmessage` 目录下，运行以下命令，安装依赖。安装完毕后，在开发者工具中右键点击该目录，点击上传并创建云函数。

```
npm i --production
```

预览

实验完成，可以在微信开发者工具中，用手机微信扫一扫预览效果。