

云开发入门实战

如果你想免费、快速的开发出一个完整的项目，用小程序的云开发可能是最好的选择。小程序的云开发所用到的主要是前端开发的知识。

注册微信小程序

小程序的注册非常方便，打开小程序注册页面，按照要求填入个人的信息，验证邮箱和手机号，扫描二维码绑定你的微信号即可，3分钟左右的时间即可搞定。

注册页面：[小程序注册页面](#)

注册小程序时不能使用注册过微信公众号、微信开放平台的邮箱哦，也就是需要你使用一个其他邮箱才行。

当我们注册成功后，就可以自动登入到小程序的后台管理页面啦，如果你不小心关掉了后台页面，也可以点击小程序后台管理登录页进行登录。

后台管理页：[小程序后台管理登录页](#)

小程序和微信公众号的登录页都是同一个页面，他们会根据你的不同的注册邮箱来进行跳转。

进入到小程序的后台管理页后，点击左侧菜单的**开发**进入设置页，然后再点击**开发设置**，在开发者ID里就可以看到**AppID(小程序ID)**，这个待会我们有用。

注意小程序的ID（AppID）不是你注册的邮箱和用户名，你需要到后台查看才行哦~

必备工具与云开发文档

大家可以根据自己的电脑操作系统来下载相应的版本，注意要选择稳定版 Stable Build 的开发者工具。

开发者工具： [小程序开发者工具下载地址](#)

和学习任何编程一样，官方技术文档都是最值得阅读的参考资料。技术文档大家先只需要**花五分钟左右的时间**了解大致的结构即可，**先按照我们的教学步骤学完**之后再来看也不迟哦。

技术文档： [云开发官方文档](#)

由于小程序的云开发在不断新增功能，更新非常频繁，所以要确保自己的开发者工具是最新的哦（不然会报很多奇奇怪怪的错误），比如你之前下载过要先同步到最新才行~

体验云开发模板小程序

安装完开发者工具之后，我们使用微信扫码登录开发者工具，然后使用开发者工具新建一个小程序的项目：

- **项目名称：** 这个可以根据自己的需要任意填写； **目录：** 大家可以先在电脑上新建一个空文件夹，然后选择它；
- **AppID：** 就是之前我们找到的AppID(小程序ID)（也可以下拉选择AppID）
- **开发模式** 为小程序（默认），
- **后端服务** 选择小程序·云开发
- 点击 **新建** 确认之后就能在开发者工具的模拟器里看到**云开发QuickStart小程序**，在编辑器里看到这个小程序的源代码。

接下来，我们点击开发者工具的工具栏里的预览图标，就会弹出一个二维码，使用微信扫码这个二维码就能在手机里看到这个小程序啦。

如果你没有使用微信登录开发者工具，以及你的微信不是该小程序的开发者是没法预览的哦。

在手机里（或模拟器）操作这个小程序，把小程序里的每个按键都点一遍，看看会有什么反应。我们会发现很多地方都会显示“调用失败”等，这非常正常，我们接下来会通过一系列的操作让小程序不报错。

开通云开发服务

点击微信开发者工具的“云开发”图标，在弹出框里点击“开通”，同意协议后，会弹出**创建环境**的对话框。这时会要求你输入**环境名称**和**环境ID**，以及当前云开发的基础环境配额（基础配额免费，而且足够你使用哦）。

按照对话框提示的要求填写完之后，点击创建，会初始化环境，环境初始化成功后会自动弹出云开发控制台，这样我们的云开发服务就开通啦。大家可以花**两分钟左右**的时间熟悉一下云开发控制台的界面。



找到云开发的环境ID

点击云开发控制台窗口里的**设置**图标，在**环境变量**的标签页找到**环境名称**和**环境ID**。

当云开发服务开通后，我们可以在小程序源代码cloudfunctions文件夹名看到你的环境名称。如果在cloudfunctions文件夹名显示的不是环境名称，而是“**未指定环境**”，可以鼠标右键该文件夹，选择“**更多设置**”，然后再点击“**设置**”小图标，选择环境并确定。

指定小程序的云开发环境

在开发者工具中打开源代码文件夹**miniprogram**里的app.js，找到如下代码：

```
wx.cloud.init({  
  // 此处请填入环境 ID，环境 ID 可打开云控制台查看  
  env: 'my-env-id',  
  traceUser: true,  
})  
在env: 'my-env-id'处改成你的环境ID，如env: 'learn-snoop'
```

下载Nodejs

NodeJS是在服务端运行JavaScript的运行环境，云开发所使用的服务端环境就是NodeJS。**npm**是Node包管理器，通过npm，我们可以非常方便的安装云开发所需要的依赖包。

下载地址：[Nodejs下载地址](#)

大家可以根据电脑的操作系统下载相应的NodeJS安装包并安装（**安装时不要修改安装目录，啥也别管直接next安装即可**）。打开电脑终端（Windows电脑为**cmd命令提示符**，Mac电脑为**终端Terminal**），然后逐行输入并按Enter执行下面的代码：

```
node --version  
npm --version
```

如果显示为v10.15.0以及6.10.0（可能你的版本号会有所不同），表示你的Nodejs环境已经安装成功。

学编程要仔细，一个字母，一个单词，一个标点符号都不要出错哦。注意输上面的命令时node、npm的后面有一个空格。

部署并上传云函数 cloudfuntions文件夹图标里有朵小云，表示这就是**云函数根目录**。展开cloudfunctions，我们可以看到里面有login、openapi等文件夹，这些就是**云函数目录**。

使用鼠标右键其中的一个云函数目录比如login，在右键菜单中选择**在终端中打开**，打开后在终端中输入以下代码并按Enter回车执行：

```
npm install
```

如果显示“npm不是内部或外部命令”，你需要关闭微信开发者工具启动的终端，而是重新打开一个终端窗口，并在里面输入`cd /D 你的云函数目录`进入云函数目录，比如`cd /D C:\download\tcb-project\cloudfunctions\login`进入login的云函数目录，然后再来执行`npm install`命令。

这时候会下载云函数的依赖模块，下载完成后，再右键login云函数目录，点击“**创建并部署：所有文件**”，这时会把本地的云函数上传到云端，上传成功后在login云函数目录图标会变成一朵小云。

在开发者工具的工具栏上点击“**云开发**”图标会打开**云开发控制台**，在云开发控制台点击**云函数**图标，就能在云函数列表里看到我们上传好的“login”云函数啦。

接下来我们按照这样的流程把**其他所有云函数**（如openapi）都部署都上传，也就是：

- 右键云函数目录，选择在终端中打开，输入`npm install`命令下载依赖文件；
- 然后再右键云函数目录，点击“**创建并部署：所有文件**”

预览体验小程序

当我们把云函数都部署上传成功后，我们可以在模拟器以及手机（需要重新点击预览图标并扫描二维码）里再来体验这个小程序。

点击获取openid

openid是小程序用户的唯一标识，也就是每一个小程序用户都有一个唯一的openid。点击“点击获取openid”，在用户管理指引页面如果显示“用户id获取成功”以及一串字母+数字，那么表示你login云函数部署并上传成功啦。**如果获取openid失败，你则需要解决login云函数的部署上传，才能进行下面的步骤哦。**

上传图片

使用模拟器或手机端点击**云开发QuickStart小程序**的上传图片按钮，选择一张图片并打开，如果在文件存储指引页面显示上传成功和文件的路径以及图片的缩略图，说明你的图片就上传到云开发服务器里啦。

点击云开发控制台的**存储**图标，就可以进入到存储管理页查看到你之前上传的图片啦，点击该图片名称可以看到这张图片的一些信息，如：文件大小、格式、上传者的OpenID以及**存储位置**、**下载地址**和**File ID**。复制**下载地址**链接，z在浏览器粘贴该地址就能查看到这张图片啦。

值得注意的是由于QuickStart小程序将“上传图片”这个按钮上传的所有图片都命名为**my-image**，所以上传同一格式的图片就会被覆盖掉，也就是无论你上传多少张相同格式的图片，只会在后台里看到最后更新的那张图片。以后我们会教大家怎么修改代码，让图片不会被覆盖。

小任务：在存储管理页，新建一个文件夹（也就是新建一个二级目录），文件夹名称比如为“tcb”，并上传一张图片到该文件夹下，然后点击名称查看该图片的存储位置、下载地址和FileID，了解一下它们和没有二级目录的图片有什么不同。

体验云调用

重新点击开发者工具的**预览**图标，然后用手机微信扫描二维码，在**手机端**点击云开发QuickStart的云调用，就可以发送**模板消息**和**获取小程序码**。

发送模板消息，只能在手机微信里预览测试哦，使用微信开发者工具是发送不了模板消息的哦

点击**发送模板消息**，你的微信就会收到一则服务通知，该通知是由你的小程序发出的**购买成功**通知。这就是微信的模板消息啦，很多微信公众号、小程序都会有这样的功能，使用小程序云开发，我们也可以轻松定制自己的服务通知（以后会教大家如何定制）。

点击**获取小程序码**，如果显示调用失败，说明你的**openapi云函数**没有部署成功，需要你先部署成功才行哦。调用成功，就能获取到你的小程序码啦，这个小程序码也会保存到云开发的存储里。

新建云函数

鼠标右键cloudfunctions云函数根目录，在弹出的窗口选择**新建Node.js云函数**，然后输入**sum**，按Enter确认后，微信开发者工具会在本地（你的电脑）创建出**sum云函数目录**，同时也会在线上环境中创建出对应的云函数。

打开**sum云函数目录**下的**index.js**，添加**sum:event.a+event.b**,到return函数里，如下所示。

```
return {  
  sum:event.a+event.b,  
  event,  
  openid: wxContext.OPENID,  
  appid: wxContext.APPID,  
  unionid: wxContext.UNIONID,  
}
```

添加完成之后记得**保存代码**哦，文件修改没有保存会在标签页有一个小的绿点。可以使用快捷键（同时按）Ctrl和S来保存（Mac电脑为Command和S）。

按照上面部署并上传云函数的方法，右键sum云函数目录，选择**在终端中打开**，然后在打开的终端输入以下代码下载依赖（其实没有必要执行这一步，因为云端已经部署好了依赖，只是让大家更深刻了解这个流程）：

```
npm install
```

和之前一样，下载完依赖之后，右键选择**“创建并部署：所有文件”**。上传后，在模拟器点击云开发QuickStart小程序的**快速新建云函数**，进入**云函数指引页面**，点击**测试云函数**，即可输出结果，如果在输出结果里看到**“sum”:3**就表示你的sum函数调用成功。

显示的调用结果比期望输出要复杂很多，这是因为return函数返回的数据太多啦，不过只要有看到**“sum”:3**就表示调用成功。

点击微信开发者工具的云开发图标，打开云开发控制台，点击数据库图标进入到数据库管理页，点击集合名称右侧的+号图标，就可以创建一个数据集合了。

云数据库与Excel、MySQL的对应理解

我们可以结合Excel以及MySQL（之前没有接触过MySQL也没有关系，只看与Excel的对应就行）来理解云开发的数据库。

云数据库	MySQL数据库	Excel文件
数据库database	数据库 database	工作簿
集合 collection	表 table	工作表
字段field	数据列column	数据表的每一列
记录 record/doc	记录row	数据表除开第一行的每一行

创建books的集合

比如我们现在来创建一个books的集合（相当于创建一张Excel表），用来存放图书馆里面书籍的信息，比如这样一本书：

属性	值	对象
书名title	JavaScript权威指南(第6版)	
作者author	弗兰纳根(David Flanagan)	
标准书号isbn	9787111376613	
出版信息publishInfo	出版社press	机械工业出版社
出版信息publishInfo	出版年份year	2012

然后选择该集合，给该集合里添加记录（类似于填写Excel含字段的第一行和其中一行关于书的信息记录），依次添加字段：

- 字段名：title，类型：string，值：JavaScript权威指南(第6版)
- 字段名：author，类型：string，值：弗兰纳根(David Flanagan)
- 字段名：isbn，类型：number，值：9787111376613
- 字段名：publishInfo，类型：object 然后我们再在publishInfo的下面（二级）添加字段press，类型为string，值为：机械工业出版社；year，类型为number，值为：2012 确定之后就可以在数据库管理页查看到该记录了。

可能有些人对数据类型会比较陌生，大家可以阅读一下技术文档：数据类型官方文档，也可以把数据类型大致理解为Excel单元格的数据类型。

在开发者工具的编辑器里展开**miniprogram**文件夹，打开**pages**文件下**databaseGuide**里的**databaseGuide.js**文件，在这里找到**onAdd: function (){}、onQuery: function (){}、onCounterInc: function (){}、onCounterDec: function (){}、onRemove: function (){}分别选中绿色的代码块，然后同时按快捷键Ctrl和/（Mac电脑的快捷键为Command和/），就可以批量取消代码的注释。**

//是前端编程语言JavaScript的单行注释，位于//这一行的代码都不会执行，我们使用快捷键就是批量取消这些代码的注释，让整段代码生效。之所以显示为绿色，是微信开发者工具为了让我们看得更清晰而做的语法高亮。

前端操作数据库的页面逻辑 以上的函数是在小程序的前端页面来操作数据库，点击开发者工具模拟器云开发QuickStart里的前端操作数据库，

- 在第1步（数据库指引有标注），我们会获取到用户的openid，也就是说没有获取到openid是没法通过小程序的前端来操作数据库的哦
- 第2步，需要我们在云开发控制台里的数据库管理页创建一个counters的集合（不需添加数据）；
- 第3步，点击按钮页面的按钮“新增记录”（按钮就在这个页面的第4条与第5条之间，看起来不是那么明显），这时会调用**onAdd**方法，往counters集合新增一个记录（之前手动添加有木有很辛苦？），我们可以去云开发控制台的数据库管理页查看一下**counters**集合是不是有了一条记录；大家可以多点击几下新增记录按钮，看数据库又有什么变化。也就是小程序前端页面通过**onAdd**方法，在数据库新增了记录。
- 第4步，点击按钮查询记录，这时调用的是**onQuery**方法就能在小程序里获取到第3步我们存储到数据库里的数据啦
- 第5步，点击计数器按钮**+**号和**-**号，可以更新count的值，点击**+**号按钮会调用**onCounterInc**方法，而点击**-**号**onCounterDec**方法，比如我们点击加号到7，再去数据库管理页查看最新的一条记录（也就是最后一条），它的count由原来的1更新到了7（先点刷新按钮），我们再点击**-**号按钮到5，再来数据库管理页查看有什么变化变化（先点刷新按钮）
- 第6步，点击删除记录按钮，会调用**onRemove**方法，这时会删掉数据库里最新的记录（也就是第5步里的那一条记录）。通过实战我们了解到，**databaseGuide.js**文件里的**onAdd**、**onQuery**、

`onCounterInc`、`onCounterDec`、`onRemove`可以实现小程序的前端页面来操作数据库。

这些函数大家可以结合`databaseGuide.js`文件和云开发技术文档关于数据库的内容来理解。

- `onAdd`：在集合中新增记录的`Collection.add`
- `onQuery`：获取根据查询条件筛选后的集合数据`Query.get`
- `onCounterInc`、`onCounterDec`：更新集合记录的`Document.update`
- `onRemove`：删除集合记录的`Document.remove`

结合实战案例**通过查阅技术文档的方式**来理解函数的逻辑，这是学编程非常重要的学习方法。所有编程语言都会涉及到非常多的函数、模块、库等等，这些不是教材或者技术博文所能概括的，而技术文档相当于收录这些函数的词典，多翻阅技术文档有助于我们的技术进阶。

云开发与AI

无需懂后端代码，就可以借助云开发以及腾讯的AI服务进行图像识别、人脸识别、手势识别、语音识别等，下面我们就来借助于人脸特征分析与检测扩展能力来实现人脸特征分析（颜值打分、年龄、性别等），借助云函数给照片添加滤镜，以及借助云存储和数据库来生成相册。

安装Git

下载地址：[Git下载地址](#)

大家可以根据电脑的操作系统下载相应的Git安装包并安装（**安装时不要修改安装目录，啥也别管直接next安装即可**）。打开电脑终端（Windows电脑为**cmd命令提示符**，Mac电脑为**终端Terminal**），然后逐行输入并按Enter执行下面的代码：

```
git --version
```

如果显示的形式为git version 2.20.1（版本可能有所不同，但是没有关系），就表示你的Git安装好啦。

一键部署AI

登录腾讯云控制台

首先我们使用小程序的账号登录[腾讯云控制台](#)，选择**其他登录方式**下的**微信公众号**，然后点击**前往公众号授权**，然后使用手机扫描二维码，就会进入一个账号的选择页面，选择你的小程序账号确认即可。

腾讯云控制台：[腾讯云控制台](#)

小程序、小游戏、微信公众号等都是同一个账号体系，所以这里是前往公众号授权，不要被公众号授权这个名词给误导了哦~

开通 AI 人脸特征检测与分析

点击腾讯云控制台顶部工具栏的**云产品**，在列表中找到云开发（在最后一列的**移动服务**下面），也可以直接点击[云开发控制台](#)，就会看到**环境**和**拓展能力解决方案**两个标签。



- 在**环境**标签页，可以看到我们之前使用微信开发者工具创建的环境；
- 点击**拓展能力解决方案**标签，如果AI人脸特征检测与分析在待开通的分类下点击**立即开通**，开通之后，点击**授权以部署**，然后点击**一键部署**，将AI人脸特征检测与分析的云函数部署到小程序的环境里。再点一下**一键部署**，如果看到部署状态显示为已部署，说明云函数已经部署成功啦。

之前我们使用开发者工具部署云函数时，都是需要先使用npm下载依赖包，然后点击部署上传，但是有了一键部署，不仅云函数、依赖包都自动在云端部署好了，而且还可以免鉴权调用AI服务（之前我们使用云数据库、云存储都是需要有openid等的鉴权），是不是非常方便？

下载任务源代码

在电脑上新建一个文件夹，比如AICamera，然后打开终端命令行（Windows电脑为**cmd命令提示符**，Mac电脑为**终端Terminal**），先cd进该文件夹，

Mac电脑 比如你把AICamera文件夹放在Mac的下载downloads里面，我们需要输入命令之后按Enter：

```
cd downloads/AICamera
```

Windows电脑 我们需要先找到AICamera文件夹路径，比如C:\download\AICamera，需要输入命令按Enter:

```
cd /d C:\download\AICamera
```

通常情况下cd命令即可切换路径，但是部分windows电脑直接使用cd命令无法切换，可以使用上面的**cd /d**命令，cd是change directory的缩写，这里**cd+空格+文件夹的路径**，表示进入该文件夹。当然你也可以直接输入cd和一个空格之后把文件夹用鼠标拖到终端窗口。

然后再输入如下`git clone`命令并按Enter执行，这样我们就把放在Github上面的开源项目源代码下载到AICamera文件夹里啦：

```
git clone https://github.com/TencentCloudBase/tcb-Demo-AICamera.git
```

打开你之前建好的**AICamera**文件夹，就可以在文件夹里看到使用Git下载好的**tcb-Demo-AICamera**文件夹，在该文件夹下，有两个文件夹分别为**init**（放置着项目的完整代码）和**intact**（此次活动实战代码）

导入AI项目

点击开发者工具工具栏**项目-导入项目**，**项目名称**可以任意填写比如“AI人脸识别”，项目路径为你之前建好的AICamera文件夹下tcb-Demo-AICamera里面的**intact文件夹**

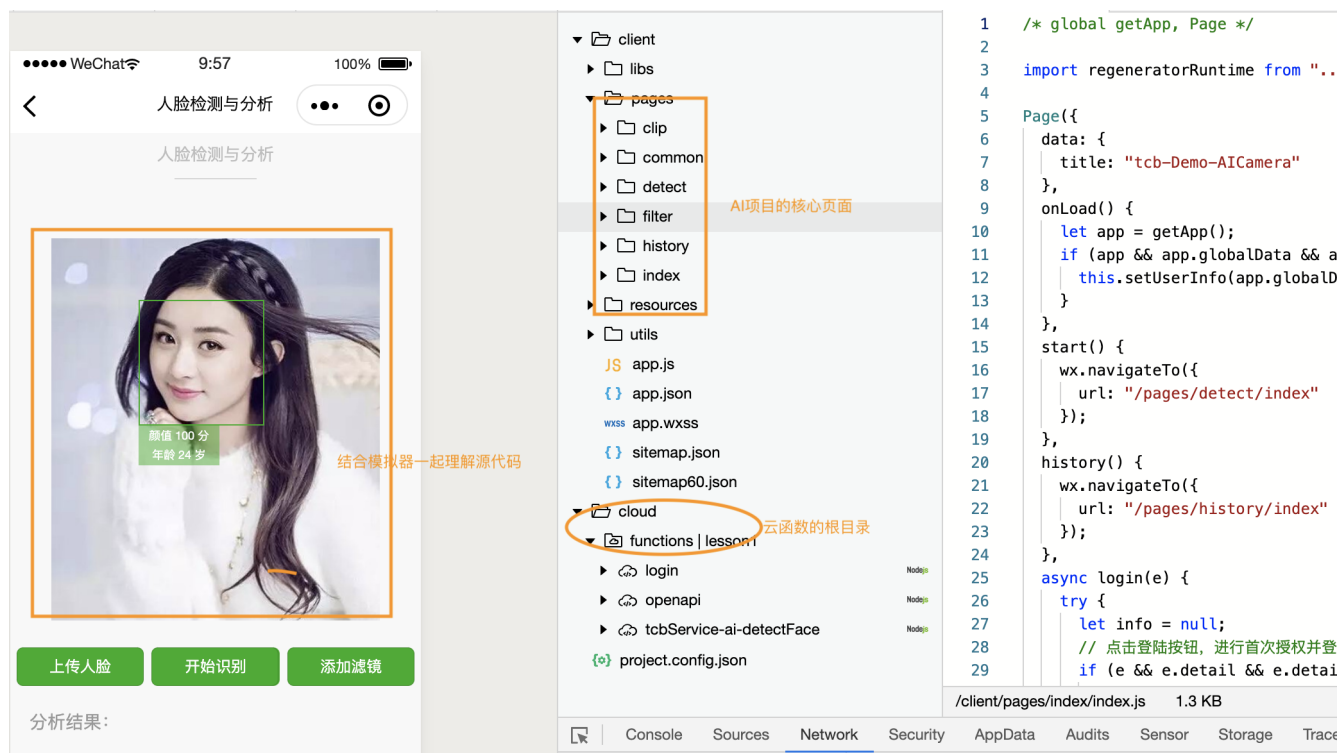
init文件夹内为这个AI项目的完整代码哦，只需完成“**导入AI项目**”这部分的任务，这个小程序就可以使用手机预览并愉快玩耍啦~不过大家还是先导入**intact**文件夹，后面会为大家讲解核心的函数是怎么写的。

创建云函数路径

使用开发者工具打开project.config.json，这个文件是当前小程序项目的配置文件，在这个文件里我们可以找到如下代码：

```
// 指定小程序源码的目录(需为相对路径)
"miniprogramRoot": "client/",
// 指定存放云函数的目录(需为相对路径)
"cloudfunctionRoot": "cloud/functions/",
// 看下配置的appid是不是你自己的，不是就需要修改一下哦
"appid": "wx7124afdb64d578f5",
```

可以看到**miniprogramRoot**配置项指定的小程序源码的目录为**client**，我们之前的云开发QuickStart小程序源码却是在miniprogram文件夹里，所以源代码到底在哪是由**miniprogramRoot**配置项控制的。



在**cloudfunctionRoot**配置项我们可以看到云函数的目录被指定为cloud/functions，我们可以在当前小程序（也就是现在这个小程序）的根目录下新建一个**cloud**文件夹，再在cloud文件夹下建一个**functions**文件夹，建完之后看functions文件夹有什么变化（会有特殊的小图标）。

鼠标右键functions文件夹（也就是**云函数根目录**），选择**同步云函数列表**，就能在本地看到**tcbService-ai-detectFace**云函数。

之前我们在云开发控制台使用一键部署的方式把AI 人脸特征检测与分析部署到了小程序的云端环境之中，当时部署的AI服务就是tcbService-ai-detectFace云函数。

修改环境ID

还记得怎么找你的环境ID么，打开client文件夹里的app.js，找到如下代码，填写一下你自己的环境ID。

```
wx.cloud.init({
  traceUser: true,
  envId: "learn-snoop"
});
```

添加pictures集合

AI小程序会使用到数据库，打开云开发控制台，在数据库管理页新建一个pictures集合，然后将pictures集合的权限设置为所有用户可读，仅创建者可读写。

了解小程序的页面逻辑

当我们想学习一个小程序的源代码时，首先就应该梳理清楚这个小程序的页面逻辑，主要是梳理client文件夹下pages的主要功能：

index页面

这是这个小程序的首页，使用开发者工具编辑器打开pages/index/index.wxml（**也要在模拟器注意页面的显示哦，对比着学**），我们可以看到登录的button，这个会获取用户的信息；当登录成功之后，会显示开始和相册两个button，点击开始会执行start函数，跳转到detect页面，点击相册会执行history函数，跳转到**history**页面；

detect页面

这是人脸检测与分析页面（也是核心功能页），打开pages/detect/index.wxml，可以看到有三个button，点击button按钮时，会执行bindtap里的函数，这些函数我们都可以找到在pages/detect/index.js文件里找到。

- ****上传人脸：****点击按钮会执行handleUploadTap函数，这个函数会调用上传图片uploadImage函数模块；
- ****开始识别：****点击按钮会执行handleRecognizeTap函数，这个函数会调用人脸检测与分析 tcbService-ai-detectFace云函数；
- ****添加滤镜：****点击按钮会执行handleFilterTap函数，这个函数除了会获取detect页面的照片数据，还去跳转到filter页面

filter页面

这是给照片添加滤镜的页面（也是核心功能页），打开pages/filter/index.wxml，可以看到有4个button，同样点击button会执行bindtap里的函数，而这些函数也可以在pages/filter/index.js文件里找到。

- ****还原：****点击按钮会执行handleOriginTap函数，这个函数会调用原始照片，照片就还原啦；
- ****怀旧：****点击按钮会执行handleOldTap函数，这个函数会对照片进行像素处理，使照片达到怀旧的效果；

- **毛玻璃**：点击按钮会执行handleSmoothTap函数，这个函数会对照片进行Canvas处理；
- **裁剪**：点击按钮会执行handleClipTap函数，这个函数会对照片进行智能裁剪，裁剪成功后会跳转到clip页面； clip页面

这个页面主要是展示裁剪之后的页面，并把裁剪后的照片保存到云端，打开pages/clip/index.wxml，可以看到有一个**保存云端**的按钮，点击按钮会执行handleSaveTap函数，会把照片存储的FileID与创建时间添加到**pictures**集合中。

history页面

这个页面主要是获取**pictures**集合中的照片。

上传图片的云函数

使用开发者工具的编辑器打开pages/detect/index.js文件，在第2行有一段代码：

```
import { uploadImage } from "../../utils/index";
```

也就是导入**client/utils/index/index.js**里，将一些公共的代码uploadImage上传图片抽离成了一个单独的js文件，作为一个模块，而在pages/detect/index.js里则导入了这个模块，这就是我们常说的模块化。

如果你对JavaScript不太了解，对代码的解释也是晕乎乎的，也没有关系，不需要你懂太多，只需要你了解这么模块、函数引入的逻辑以及uploadImage模块的功能就可以啦。

使用微信开发者工具打开**client/utils/index.js**将下面的代码复制粘贴到第11行处，添加了之后，上传人脸的按钮就可以使用啦，点击上传人脸，选择一张照片，就会把该照片上传到云存储里。

```
wx.cloud.uploadFile({
  cloudPath,
  filePath: fileName,
  success: res => {
    resolve(res);
  },
  fail: () => {
```



```
        wx.hideLoading();
        reject();
    }
});
```

这段代码主要是使用wx.cloud.uploadFile将本地资源上传到云存储空间。大家可以对照着这个函数的技术文档来深入理解。

技术文档：[wx.cloud.uploadFile函数](#)

调用AI分析与人脸检测云函数

使用微信开发者工具打开pages/detect/index.js，将下面的代码复制粘贴到第100行处，也就是粘贴到callFunction()函数的try{}内：

```
let { result } = await wx.cloud.callFunction({
  name: "tcbService-ai-detectFace",
  data: {
    FileID: this.data.fileID
  }
});
wx.hideLoading();

if (!result.code && result.data) {
  this.setData(
    {
      faceRects: this.getFaceRects(result.data)
    },
    () => {
      this.triggerEvent("finish", result.data);
    }
  );
} else {
  throw result;
}
```

这段函数主要是调用tcbService-ai-detectFace云函数，大家也可以结合小程序调用云函数的技术文档来理解。

技术文档：[调用云函数wx.cloud.callFunction](#)

滤镜处理

怀旧风格滤镜

使用微信开发者工具打开**pages/filter/index.js**将下面代码粘贴到第111行处，也就是添加到handleOldTap函数内：

```
for (let i = 0; i < data.length; i += 4) {  
  let r = originImageData.data[i];  
  let g = originImageData.data[i + 1];  
  let b = originImageData.data[i + 2];  
  data[i] = 0.393 * r + 0.769 * g + 0.189 * b;  
  data[i + 1] = 0.349 * r + 0.686 * g + 0.168 * b;  
  data[i + 2] = 0.272 * r + 0.534 * g + 0.131 * b;  
  data[i + 3] = originImageData.data[i + 3];  
}
```

整个怀旧风格的代码逻辑是先使用wx.canvasGetImageData函数获取到图像的像素点数据，获取到的是一个一维数组，每四项表示一个像素点的 rgba，然后将每个像素点的RGB值先分离出来，再按照下面的算式分别重新计算出RGB值然后做为当前点的RGB值（该算式是一个比较固定的怀旧风格算法）

```
R = 0.393 * r + 0.769 * g + 0.189 * b;  
G = 0.349 * r + 0.686 * g + 0.168 * b;  
B = 0.272 * r + 0.534 * g + 0.131 * b;
```

图片的毛玻璃滤镜

在**pages/filter/index.js**的第3行，我们可以看到这样一行代码：

```
import { smoothX, smoothY, kernel } from "../util";
```

和前面import的原理一样，这里是导入**pages/filter/util.js**里的smoothX, smoothY, kernel 等函数，使用开发者工具打开**pages/filter/util.js**将下面代码粘贴到第52行处，也就是SmoothY函数内：

```
let totalLength = data.length;  
let radius = kernel.length >> 1;  
let channels = [0, 0, 0, 0];  
let weight = 0;  
for (let index = 0; index < kernel.length; index++) {  
  let cur = pointIndex - (radius - index) * width * channels.length;  
  if (cur < 0) {
```

```

    continue;
  } else if (cur > totalLength) {
    break;
  } else {
    channels.forEach((channel, channelIndex) => {
      channels[channelIndex] += data[cur + channelIndex] * kernel[index];
    });
    weight += kernel[index];
  }
}
return channels.map(channel => {
  return channel / weight;
});

```

这段代码是对图像进行纵向高斯模糊。关于高斯模糊的算法，可以看[知乎关于高斯模糊算法的回答](#)。

再将下面代码粘贴到 `pages/filter/index.js` 的 `handleSmoothTap` 函数内（也就是放到对应的注释里，注释标有 `do something`）：

```

// 通过 smoothX 函数对原有像素进行横向扫描，存储处理后像素于 smoothData
for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {
    let pointIndex = (x + y * width) * 4;
    let channels = smoothX(
      pointIndex,
      gKernel,
      y * width * 4,
      ((1 + y) * width - 1) * 4,
      originImageData.data
    );
    channels.forEach((channel, index) => {
      smoothData[pointIndex + index] = channel;
    });
  }
}

// 对纵向像素进行处理，并实现 smoothY 函数
for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {
    let pointIndex = (x + y * width) * 4;
    let channels = smoothY(pointIndex, gKernel, width, smoothData);
    channels.forEach((channel, index) => {
      smoothData[pointIndex + index] = channel;
    });
  }
}

// 将平滑后的人脸像素输出到画布
wx.canvasPutImageData({
  canvasId: "canvas",

```

```

    x: 0,
    y: 0,
    width: width,
    height: height,
    data: smoothData,
    success: data => {
      resultImageData = { width, height, data: smoothData };
    },
    fail: e => {
      console.log(e);
    },
    complete: () => {
      // 绘制完成
      this.pending(false);
    }
  });

```

除了使用高斯模糊算法对图片的像素进行横轴、纵向处理，在将平滑后的人脸像素输出到画布使用了wx.canvasPutImageData函数，我们可以查看技术文档了解该函数的更多细节与详细用法。

技术文档：[wx.canvasPutImageData函数](#)

保存图片

使用微信开发者工具打开pages/clip/index.js将下面代码粘贴到第155行处：

```

let { fileId } = await uploadImage(tempFilePath);
let db = wx.cloud.database();
let collection = db.collection("pictures");
if (!collection) {
  throw {
    message: "需创建集合 pictures"
  };
}
await collection.add({
  data: {
    origin: this.data.fileID,
    output: fileId,
    createTime: new Date().getTime()
  }
});

```

这段代码是将照片的fileID、创建时间保存到pictures集合之中，主要用到的是给云数据库指定结合新增记录的知识。

技术文档：[Collection.add](#)

获取图片列表

使用微信开发者工具打开[pages/history/index.js](#)将下面代码粘贴到第19行处。

```
let { data } = await collection
  .orderBy("createTime", "desc")
  .limit(20)
  .get();
this.setData({
  list: data.reduce((list, { origin, output }) => {
    list.push(origin);
    list.push(output);
    return list;
  }, [])
});
```

这段代码的意思是按照照片的创建时间进行降序排序，选前20张。这里用到的是云数据库的排序与筛选知识。

云数据库的排序：[Collection.orderBy / Query.orderBy](#)

云数据库的筛选：[Collection.get / Query.get](#)