### 1.创建小程序项目·开通云开发服务

### 1.1 创建QuickStart小程序项目

确保自己的开发者工具是最新的,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。

#### 1.2 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。



### 2. 配置项目的环境ID

### 2.1 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。

#### 2.2 指定小程序的云开发环境

```
<> databaseGuide.wxml
                                            2
                                                 App({
                                            3
                                                   onLaunch: function () {
    wxss databaseGuide.wxss
                                            4
 ▶ ☐ deployFunctions
                                            5
                                                      if (!wx.cloud) {
 ▶ 🗀 index
                                                       console.error('请使用 2.2.3 或以上的基础库以使用云能力')
                                            6
                                            7
                                                      } else {
 ▶ ☐ openapi
                                                        wx.cloud.init({
 ▶ 🗀 storageConsole
                                                           <del>// 此处请填入环境 ID,环境</del> ID 可打开云控制台查看
 ▶ 🗀 userConsole
                                                          env: 'lesson1-683345',
                                           10
▶ 🗀 style
                                                          traceUser: true,
                                           11
                                                       })
                                           12
  JS app.js
                                           13
  {} app.json
                                           14
 wxss app.wxss
                                                     this.globalData = {}
                                           16
  {} sitemap.json
                                           17
                                                 })
README.md
```

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。

# 3. 部署并上传云函数

# 3.1 下载云函数依赖环境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

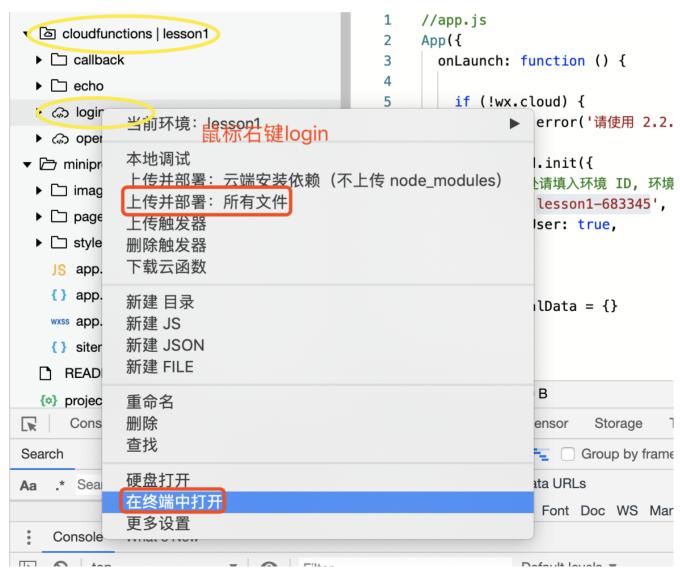
大家可以根据电脑的操作系统下载相应的NodeJS安装包并安装(**安装时不要修改安装 目录,啥也别管直接next安装即可**)。打开电脑终端(Windows电脑为**cmd命令提示符**,Mac电脑为**终端Terminal**),然后逐行输入并按Enter执行下面的代码:

```
[(base) lidongdeMacBook-Pro:~ lidong$ node --version
v10.15.0
[(base) lidongdeMacBook-Pro:~ lidong$ npm --version
6.10.1
```

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。

#### 3.2 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: 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。

#### 3.3 部署并上传其他云函数

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"

### 4. 下载并导入AI项目的源代码

- **4.1 下载AI项目并解压** 点击下载此次活动的AI项目源代码压缩包,并解压,可以看到 tcb-Demo-AlCamera-master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(**此次活动的实战代码**)。
- **4.2 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**

#### 5. 配置AI项目

#### 5.1 配置项目的环境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"
});
```

### 5.2 添加pictures集合并设置权限



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**。

# 6. 上传图片到云存储功能

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。

```
exholic indication abroadinade(litremanne) f

▼ Client

                                                2
                                                       return new Promise((resolve, reject) => {
 ▶ 🗀 libs
                                                3
                                                         const dotPosition = fileName.lastIndexOf(".");
                                                4
                                                         const extension = fileName.slice(dotPosition);
 ▼  pages
                                                5
                                                         const cloudPath = `${Date.now()}-${Math.floor(
   ▶ 🗀 clip
                                                           Math.random(0, 1) * 10000
                                                6
   ▶ 🗀 common
                                                7
                                                         )}${extension}`;
   ▶ 🗀 detect
                                                8
                                                9
                                                         wx.cloud.uploadFile({
   ▶ 🗀 filter
                                               10
                                                            cloudPath,
   ▶ 🗀 history
                                               11
                                                            filePath: fileName,
   ▶ 🗀 index
                                               12
                                                            success: res => {
                                               13
                                                             resolve(res);
 ▶ ☐ resources
                                               14
                                                           },

    □ utils

                                               15
                                                           fail: () => {
    JS index.js
                                                              wx.hideLoading();
                                               16
   JS app.js
                                               17
                                                              reject();
                                               18
   {} app.json
                                               19
                                                         });
   wxss app.wxss
                                               20
   {} sitemap.json
                                               21
                                                       });
                                               22
   / 1 citaman60 ican
```

```
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。

# 7. 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:

```
► LI IIUS
                                             96
                                                        try {
▼  pages
                                             97
                                                          let { result } = await wx.cloud.callFunction({
                                                            name: "tcbService-ai-detectFace",
                                             98
 ▶ 🗀 clip
                                             99
                                                            data: {
 ▶ 🗀 common
                                                             FileID: this.data.fileID
                                            100

▼ P→ detect

                                            101
                                                          });
     JS index.js
                                            102
                                            103
                                                          wx.hideLoading();
     { } index.ison
                                            104
     <> index wxml
                                                          if (!result.code && result.data) {
                                            105
     wxss index wxss
                                                            this.setData(
                                            106
                                            107
 ▶ 🗀 filter
                                            108
                                                                faceRects: this.getFaceRects(result.data)
 ▶ 🗀 history
                                            109
                                                              },
 ▶ 🗀 index
                                            110
                                                              () => {
▶ ☐ resources
                                            111
                                                                this.triggerEvent("finish", result.data);
                                            112
▶ □ utils
                                            113
                                                            );
  JS app.js
                                            114
                                                          } else {
  {} app.json
                                            115
                                                            throw result;
                                            116
  wxss app.wxss
                                            117
 sitemap.ison
```

```
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。

# 8. 照片的滤镜处理

### 8.1 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:

```
let { width, height } = originImageData;
                                          110
▼ P pages
                                         111
 ▶ 🗀 clip
                                         112
                                                     // 开始像素处理
 ▶ 🗀 common
                                         113
                                                     this.pending(true);
                                                     let data = new Uint8ClampedArray(width * height * 4);
                                         114
 ▶ 🗀 detect
                                         115
                                                     for (let i = 0; i < data.length; i += 4) {
 r 🗁 filter
                                          116
                                                       let r = originImageData.data[i];
    JS index.js
                                          117
                                                       let g = originImageData.data[i + 1];
    {} index.json
                                          118
                                                       let b = originImageData.data[i + 2];
                                         119
                                                       data[i] = 0.393 * r + 0.769 * g + 0.189 * b;
    <> index.wxml
                                                       data[i + 1] = 0.349 * r + 0.686 * g + 0.168 * b;
                                         120
    wxss index.wxss
                                                       data[i + 2] = 0.272 * r + 0.534 * g + 0.131 * b;
                                         121
    JS util.js
                                                       data[i + 3] = originImageData.data[i + 3];
                                         122
                                         123
 ▶ 🗀 history
                                         124
 ▶ [☐ index
                                         125
                                                     // 将平滑后的人脸像素输出到画布
▶ ☐ resources
                                         126
                                                     wx.canvasPutImageData({
▶ 🗀 utils
                                         127
                                                       canvasId: "canvas",
                                          120
```

```
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];
}</pre>
```

#### 8.2 图片的毛玻璃滤镜

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:

```
export function smoothY(pointIndex, kernel, width, data)
                                                                                                                                                              51
     ▶ 🗀 libs
                                                                                                                                                              52
                                                                                                                                                                                          let totalLength = data.length;
     ▼  pages
                                                                                                                                                              53
                                                                                                                                                                                          let radius = kernel.length >> 1;
                                                                                                                                                              54
                                                                                                                                                                                          let channels = [0, 0, 0, 0];
           ▶ 🗀 clip
                                                                                                                                                              55
                                                                                                                                                                                          let weight = 0;
           ▶ 🗀 common
                                                                                                                                                                                          for (let index = 0; index < kernel.length; index++) {</pre>
                                                                                                                                                              56
           ▶ 🗀 detect
                                                                                                                                                              57
                                                                                                                                                                                                 let cur = pointIndex - (radius - index) * width *
                                                                                                                                                                                   channels.length;

▼ 

| filter | Filtre | Filter | F
                                                                                                                                                              58
                                                                                                                                                                                                 if (cur < 0) {</pre>
                      JS index.js
                                                                                                                                                              59
                                                                                                                                                                                                        continue;
                      {} index.json
                                                                                                                                                                                                 } else if (cur > totalLength) {
                                                                                                                                                              60
                      <> index.wxml
                                                                                                                                                              61
                                                                                                                                                                                                        break;
                                                                                                                                                              62
                                                                                                                                                                                                 } else {
                      wxss index.wxss
                                                                                                                                                                                                        channels.forEach((channel, channelIndex) => {
                                                                                                                                                              63
                       JS util.js
                                                                                                                                                              64
                                                                                                                                                                                                               channels[channelIndex] += data[cur + channelIndex]
          ▶ 🗀 history
                                                                                                                                                                                   kernel[index];
           ▶ ☐ index
                                                                                                                                                              65
                                                                                                                                                                                                      });
                                                                                                                                                              66
                                                                                                                                                                                                        weight += kernel[index];
      ▶ ☐ resources
                                                                                                                                                              67
      ▶ □ utils
                                                                                                                                                              68
            JS app.js
                                                                                                                                                              69
                                                                                                                                                                                         return channels.map(channel => {
                                                                                                                                                              70
                                                                                                                                                                                                 return channel / weight;
            {} app.json
                                                                                                                                                              71
            wxss app.wxss
                                                                                                                                                              72
```

```
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++) {</pre>
  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。

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):

```
// 四部門啄糸紅田花柱,八天花 300000111 四数
▼  pages
                                          158
                                                     // do something
                                          159
 ▶ 🗀 clip
                                                     // 处理完成将平滑后的人脸像素输出到画布
                                          160
 ▶ 🗀 common
                                          161
                                                     // 类似 handleOldTap 最后操作
 ▶ 🗀 detect
                                          162
                                                    // do something
 ▼ 🗁 filter
                                          163
                                          164
                                                     for (let x = 0; x < width; x++) {
     JS index.js
                                                       for (let y = 0; y < height; y++) {
                                          165
    {} index.json
                                                         let pointIndex = (x + y * width) * 4;
                                          166
    <> index.wxml
                                          167
                                                         let chanels = smoothX(
                                          168
                                                          pointIndex,
    wxss index.wxss
                                          169
                                                           gKernel,
    JS util.js
                                                          y * width * 4,
                                          170
 ▶ 🗀 history
                                          171
                                                          ((1 + y) * width - 1) * 4,
 ▶ 🗀 index
                                                          originImageData.data
                                          172
                                          173
▶ ☐ resources
                                          174
                                                         chanels.forEach((chanel, index) => {
▶ □ utils
                                                           smoothData[pointIndex + index] = chane];
                                          175
  JS app.js
                                          176
                                                         });
                                          177
                                                      }
 {} app.json
                                          178
  wxss app.wxss
                                          179
                                                     for (let x = 0; x < width; x++) {
 {} sitemap.json
                                          180
                                                       for (let y = 0; y < height; y++) {
  sitemap60.json
                                          181
                                                         let pointIndex = (x + y * width) * 4;
                                                         let chanels = smoothV(nointIndex aKernet wid
```

```
for (let y = 0; y < height; y++) {
      let pointIndex = (x + y * width) * 4;
      let chanels = smoothX(
        pointIndex,
        gKernel,
        y * width * 4,
        ((1 + y) * width - 1) * 4,
        originImageData.data
      chanels.forEach((chanel, index) => {
        smoothData[pointIndex + index] = chanel;
      });
}
// 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 chanels = smoothY(pointIndex, gKernel, width, smoothData);
      chanels.forEach((chanel, index) => {
        smoothData[pointIndex + index] = chanel;
      });
    }
}
// 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

```
▼ □ pages
                                          153
                                                        // 将生成的文件上传到云端, 在 db 里存下处理前后 fileID, 记录
                                          154
                                                        // do something
 ▼ 🗁 clip
                                          155
                                                        let { fileID } = await uploadImage(tempFilePath);
     IS index.is
                                                        let db = wx.cloud.database();
                                          156
     {} index.json
                                          157
                                                        let collection = db.collection("pictures");
                                          158
                                                        if (!collection) {
     <> index.wxml
                                          159
                                                         throw {
     wxss index.wxss
                                                           message: "需创建集合 pictures"
                                          160
 ▶ 🗀 common
                                          161
                                                         };
 ▶ 🗀 detect
                                          162
                                                        await collection.add({
                                          163
 ▶ 🗀 filter
                                          164
                                                         data: {
 ▼ 🗁 history
                                          165
                                                            origin: this.data.fileID,
     JS index.js
                                          166
                                                            output: fileID,
     {} index.json
                                          167
                                                            createdTime: new Date().getTime()
                                          168
     <> index.wxml
                                          169
                                                        });
     wxss index.wxss
                                          170
     JS util.js
                                          171
                                                        wx.hideLoading();
                                                        wx.showToast({
 ▶ 🗀 index
                                          172
                                                         title: "保存成功",
                                          173
▶ 🗀 resources
```

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,
       createdTime: 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。

# 10. 获取云端存储的图片列表

```
let collection = db.collection("pictures");
                                             12
13
                                                        if (!collection) {
 ▶ 🗀 libs
                                             14
                                                         return;
 ▼  pages
                                             15
                                             16
                                                       try {
  ▶ 🗀 clip
                                             17
                                                          let { data } = await collection
  ▶ 🗀 common
                                             18
                                                            .orderBy("createdTime", "desc")
   ▶ 🗀 detect
                                             19
                                                            .limit(20)
  ▶ 🗀 filter
                                             20
                                                            .get();
                                             21
                                                          this.setData({
  ▼ 🗁 history
                                                            list: data.reduce((list, { origin, output }) => {
                                             22
      JS index.js
                                             23
                                                              list.push(origin);
       index.json
                                             24
                                                              list.push(output);
     打开history
index.wxml
                   下的index.js
                                             25
                                                              return list;
                                             26
                                                           }, [])
      wxss index.wxss
                                             27
                                                          }):
      JS util.js
                                             28
   ▶ 🗀 index
                                                        } catch (e) {
                                             29
 ▶ 🗀 resources
                                             30
                                                          console.log(e);
                                             31
                                                          wx.showToast({
 ▶ 🗀 utils
                                             32
                                                            title: "查询失败",
   JS app.js
                                                            icon: "none"
                                             33
   {} app.json
                                             34
                                                         });
                                             35
   wxss app.wxss
```

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

这段代码的意思是根据照片的创建时间进行降序排序,选前20张。