

云开发 CloudBase

扩展能力

产品文档



腾讯云

【版权声明】

©2013-2020 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系95716。

文档目录

扩展能力

概述

能力

 Hello World

 图像处理

 图像安全审核

 图像盲水印

 图像标签

 CMS内容管理系统

 短信验证码登录

 MySQL

 Redis

其他

 音视频

 智能图像

扩展能力

概述

最近更新时间：2020-07-03 16:29:14

云开发扩展能力是云开发团队为开发者提供的一站式云端服务，旨在降低开发者使用云服务的门槛，助力开发者快速开发应用。

计费模式

扩展能力会基于云开发和其他云上资源构建，当您使用云开发扩展时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

安装方式

使用云开发控制台

1. 打开 [云开发扩展控制台](#)；
2. 选择希望安装的扩展；
3. 单击【安装】，进行扩展程序的安装；
4. 等待扩展程序安装完成即可使用；

现有扩展能力

名称	介绍
Hello World	基于云开发快速搭建 web 网站和小程序，轻松实现多端同步，帮助开发者轻松开启云上开发之旅
图像处理	图像处理提供多种图像处理功能，包含智能裁剪、无损压缩、水印、格式转换等，您可通过扩展 SDK 轻松管理文件
图像安全审核	图像安全审核提供鉴黄、鉴政、鉴暴恐等多种类型的敏感内容审核服务，有效识别违禁图片，规避违规风险
图像盲水印	盲水印功能将水印图以不可见的形式添加到原图信息中，不会对原图质量产生太大影响。在图片被盗取后，您可对疑似被盗取的资源进行盲水印提取，验证图片归属

名称	介绍
图像标签	图像标签对云存储中存量数据的图片标签识别，返回图片中置信度较高的主题标签，帮助开发者分析图像
CMS内容管理系统	方便开发人员和内容运营者，随时随地管理小程序/Web 等多端云开发内容数据，支持 PC/移动端浏览器访问，支持文本、富文本、图片、文件、关联类型等多种内容类型的可视化编辑

能力

Hello World

最近更新时间：2020-07-03 16:30:30

基于云开发快速搭建 web 网站和小程序，轻松实现多端同步，帮助开发者轻松开启云上开发之旅。

扩展工作模式

当您使用这个扩展时：

1. 云函数渲染 web 页面。
2. web 页面和小程序监听页面内容输入变更并同步。

前置要求

已经开通云开发。

安装扩展

注意：

微信小程序开发者请使用【其他登录方式】-【微信公众号登录】登录，再选择关联的小程序账户登录；QQ 小程序开发者可直接通过QQ小程序开发者IDE【云开发】按钮登录，也可以通过关联的腾讯云账户登录。

您可以通过 [云开发控制台](#)，来安装和管理扩展。

使用扩展

1. 进入 [云开发控制台](#)，单击需安装扩展的环境（下文以 `custom_env` 指代对应的环境ID），单击左侧菜单栏【环境设置】页进入环境设置页：

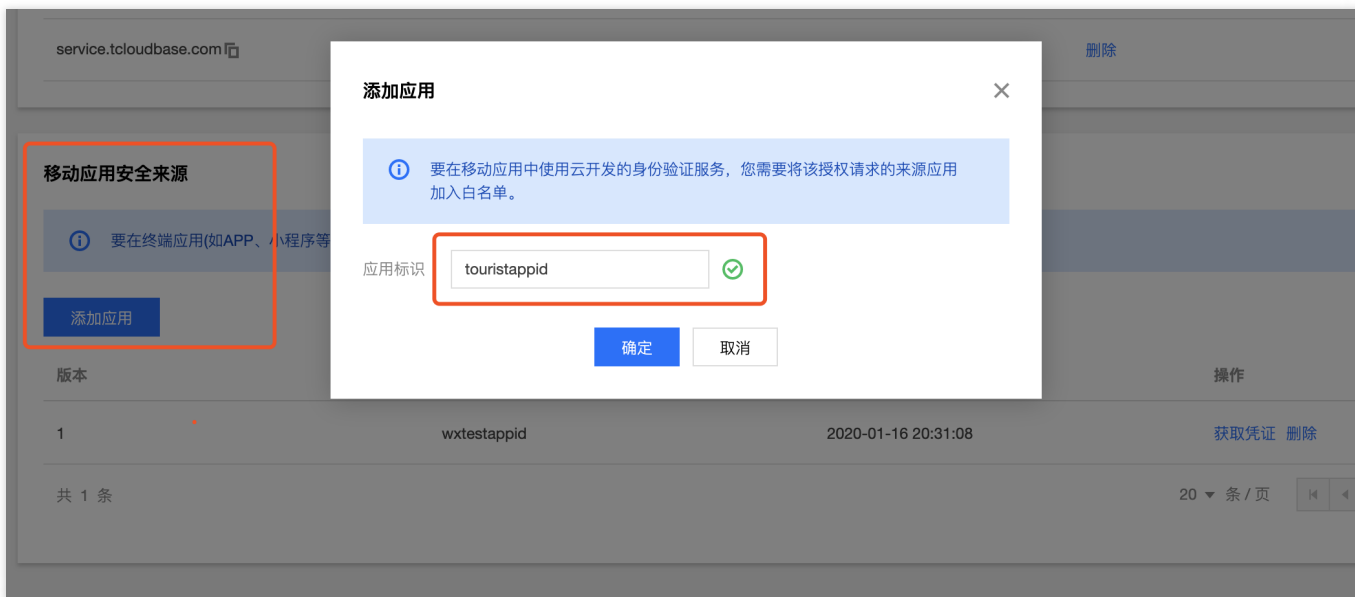
- 1.1 单击【登录方式】，打开【匿名登录】开关，允许应用中可以使用匿名登录的方式访问云开发；



- 1.2 单击【安全配置】，单击**WEB 安全域名**下的【添加域名】，添加云函数 HTTP 触发当前环境默认域名：`${custom_env}.service.tcloudbase.com`（该域名是云开发为开发者分配的 HTTP 触发云函数的域名，详情可以参考 [HTTP 触发](#)），允许在该域名的页面下调用 web 云开发；



- 1.3 访问【安全配置】，然后在【移动应用安全来源】中注册 `touristappid`，添加成功后获取凭证信息备用；



2.打开 `custom_env` 环境下的【云数据库】，找到 `tcb_hello_world` 集合进入详情，进入【权限设置】tab 页，点击【切换到安全规则】，输入：

```
{
  "read": true,
  "write": true
}
```

3.web 网站：使用浏览器可访问 `https://${custom_env}.service.tcloudbase.com/tcb_hello_world` ；

4.微信小程序：访问 [Hello World](#) 获取代码片段，一键导入代码到微信开发者IDE中（不要配置 `AppID`）；在项目 `app.js` 中更新如下配置：

```
const cloud = tcb.init({
  env: "${custom_env}", // 当前环境的ID
  appSign: "touristappid",
  appSecret: {
    appAccessKeyId: "移动应用安全来源 版本", // 步骤1.3中获取的移动应用安全来源版本
    appAccessKey: "移动应用安全来源 凭证" // 步骤1.3中获取的移动应用安全来源凭证
  }
});
```

5.在 web 或小程序中拖动数字，即可在另一端看到相同的变化。

注意：

该扩展为示例 Demo，请按照使用指引来完成配置。注册安全来源时要以 `touristappid` 来注册；在小程序中体验时，不需要配置 `AppID`，填入 `touristappid` 对应的版本及凭证信息即可。

其他

扩展配置信息

您可以配置以下参数：

- 环境 ID：选择要部署的环境，在哪个环境下使用。

计费

此扩展使用其他云开发或其他腾讯云服务，可能会产生相关费用：

- 云函数（[产品定价](#) 及 [使用明细](#)）。
- 云数据库（[产品定价](#) 及 [使用明细](#)）。

当您使用云开发扩展时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

创建的资源

- **Type:** Cloud Function
Description: 借用云开发 http service 能力，渲染 web 页面，方便开发者体验云开发能力。
- **Type:** Cloud DB
Description: 存储用户操作数据，借用实时数据库能力，监听数据变更，实现数据多端同步。

权限授予

主账户

该扩展能力使用云开发自有资源即可完成，无需再授予其他权限。

子账户

如果想让子账户也能使用该扩展，需要为子账户授予如下权限才能使用：

- **策略:** QcloudAccessForTCBRole
描述: 云开发（TCB）对云资源的访问权限。

图像处理

最近更新时间：2020-07-03 16:30:59

图像处理提供多种图像处理功能，包含智能裁剪、无损压缩、水印、格式转换等，您可通过我们提供的 SDK 轻松管理文件。

功能特性

对于有大量用户上传的应用，图片处理可以为用户提供便捷的裁剪、压缩、水印等服务来适配各个业务场景。

类型	介绍
缩放	等比缩放、设定目标宽高缩放等多种方式
裁剪	普通裁剪、缩放裁剪、内切圆、人脸智能裁剪
旋转	普通旋转、自适应旋转
格式转换	jpg、bmp、gif、png、webp、jpeg格式转换，gif 格式优化，渐进显示功能
质量变换	针对 JPG 和 WEBP 图片进行质量变换
高斯模糊	对图片进行模糊处理
锐化	对图片进行锐化处理
图片水印	提供图片水印处理功能
文字水印	提供实时文字水印处理功能
获取图片基本信息	查询图片基本信息，包括格式、长、宽等
获取图片 EXIF	查询图片 EXIF 信息，如照片的拍摄参数、缩略图等
获取图片主色调	获取图片主色调信息
去除元信息	去除图片元信息，减小图像体积
快速缩略模板	快速实现图片格式转换、缩略、剪裁等功能，生成缩略图
管道操作符	对图片按顺序进行多种处理

注意：

您不仅在云函数中可以使用该能力，也可以在客户端（web、微信小程序、QQ小程序）使用，文件读写权限策略与云存储一致，减去您额外的权限管理工作。

前置要求

已经开通云开发。

安装程序

注意：

微信小程序开发者请使用【其他登录方式】-【微信公众号登录】登录，再选择关联的小程序账户登录；QQ小程序开发者可直接通过QQ小程序开发者IDE【云开发】按钮登录，也可以通过关联的腾讯云账户登录。

您可以通过 [云开发控制台](#)，来安装和管理该能力。

使用程序

注意：

如果您在 web 网站使用该能力，请先在 [云开发控制台](#) 将网站域名添加为当前环境的安全域名。

获取图片时处理

如果您的图像已经上传到云存储中，可通过在 url 后添加参数的方式，对图像进行处理，自动获取处理后的结果。假设缩放图片宽高为原图50%，示例如下：

```
https://${您的云开发文件访问域名, 格式如: xxx.tcb.qcloud.la}/sample.jpeg?imageMogr2/thumbnail/!50p
```

注意：

图像访问域名请填写云开发分配的域名，格式如：xxx.tcb.qcloud.la。

支持的图像处理特性及参数拼接格式如下所示：

类型	参数拼接指引
缩放	参数拼接规则
裁剪	参数拼接规则
旋转	参数拼接规则
格式转换	参数拼接规则
质量变换	参数拼接规则
高斯模糊	参数拼接规则
锐化	参数拼接规则
图片水印	参数拼接规则
文字水印	参数拼接规则
获取图片基本信息	参数拼接规则
获取图片 EXIF	参数拼接规则
获取图片主色调	参数拼接规则
去除元信息	参数拼接规则
快速缩略模板	参数拼接规则
管道操作符	参数拼接规则

持久化图像处理

持久化图像处理只支持有结果图输出的处理操作，如获取图片基本信息、获取图片 EXIF、获取图片主色调等，请通过url拼接参数获取。

1. 安装 SDK 到项目

web端及云函数中使用：

```
npm install --save @cloudbase/extension-ci@latest
```

微信小程序中使用：

```
npm install --save @cloudbase/extension-ci-wxmp@latest
```

微信小程序中npm使用请参考 [npm 支持](#)。

2. 调用 SDK

调用参数

名称	类型	是否必须	说明
action	String	是	操作类型，传：ImageProcess
cloudPath	String	是	文件的绝对路径，与tcb.uploadFile中一致
fileContent	Uint8Array或Buffer	否	文件内容，有值，表示上传时处理图像；为空，则处理已经上传的图像
operations	Object	否	图像处理参数

operations节点内容

名称	类型	是否必须	说明
rules	Array.<Rule对象>	是	处理样式

Rule节点内容

名称	类型	是否必须	说明
fileid	String	是	处理结果的文件路径，如以 '/' 开头，则存入指定文件夹中，否则，存入原图文件存储的同目录
rule	String	是	处理样式参数，与下载时处理图像在url拼接的参数一致

返回内容

参数名称	类型	描述
UploadResult	Object	原图信息

UploadResult 节点内容：

参数名称	类型	描述
ProcessResults	Object	图片处理结果

ProcessResults 节点内容：

节点名称	类型	描述
Object	Object	每一个图片处理结果

Object 节点内容：

节点名称	类型	描述
Key	String	文件名
Location	String	图片路径
Format	String	图片格式
Width	Int	图片宽度
Height	Int	图片高度
Size	Int	图片大小
Quality	Int	图片质量

调用示例

web端使用

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-js-sdk");
const readFile = async function(file) {
  let reader = new FileReader();
  let res = await new Promise(resolve => {
    reader.onload = function(e) {
      let arrayBuffer = new Uint8Array(e.target.result);
      resolve(arrayBuffer);
    };
    reader.readAsArrayBuffer(file);
  });
};
```

```
return res;
};
let file = document.getElementById("selectFile").files[0];
let fileContent = await readFile(file);
```

云函数中使用

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-admin-node");
let fileContent = imageBuffer; // Uint8Array|Buffer格式图像内容
```

您可以根据需要选择在【客户端】或【云函数】中使用，再按照如下代码调用：

```
tcb.init({
  env: "您的环境ID"
});
tcb.registerExtension(extCi);

async function process() {
  try {
    const opts = {
      rules: [
        {
          // 处理结果的文件路径，如以'/'开头，则存入指定文件夹中，否则，存入原图文件存储的同目录
          fileId: "/image_process/demo.png",
          rule: "imageView2/format/png" // 处理样式参数，与下载时处理图像在url拼接的参数一致
        }
      ]
    };
    const res = await tcb.invokeExtension("CloudInfinite", {
      action: "ImageProcess",
      cloudPath: "demo.jpeg", // 图像在云存储中的路径，与tcb.uploadFile中一致
      fileContent, // 该字段可选，文件内容：Uint8Array|Buffer。有值，表示上传时处理图像；为空，则处理已经上传的图像
      operations: opts
    });
    console.log(JSON.stringify(res.data, null, 4));
  } catch (err) {
    console.log(JSON.stringify(err, null, 4));
  }
}
```

微信小程序客户端中使用

注意：

微信小程序中关联的云开发环境需要与该程序安装的环境一致；取值参数及返回与在云函数、web端使用一致。

```
const extCi = require("@cloudbase/extension-ci-wxmp");
const readFile = async function() {
  let res = await new Promise(resolve=>{
    wx.chooseImage({
      success: function(res) {
        let filePath = res.tempFilePaths[0]
        let fm = wx.getFileSystemManager()
        fm.readFile({
          filePath,
          success(res){
            resolve(res)
          }
        })
      }
    })
  })
  return res
}
let fileContent = await readFile();
async function demo() {
  const opts = {
    rules: [
      {
        // 处理结果的文件路径，如以'/' 开头，则存入指定文件夹中，否则，存入原图文件存储的同目录
        fileid: "/image_process/demo.png",
        rule: "imageView2/format/png", // 处理样式参数，与下载时处理图像在url拼接的参数一致
      }
    ]
  };
  extCi.invoke({
    action: "ImageProcess",
    cloudPath: "demo.jpeg", // 图像在云存储中的路径，与tcb.uploadFile中一致
    operations: opts,
    fileContent // 可选参数
  }).then(res => {
    console.log(JSON.stringify(res.data, null, 4));
  }).catch(err => {
    console.log(JSON.stringify(err, null, 4));
  });
}
```



```
})  
}
```

QQ小程序客户端中使用

请参考 [QQ小程序扩展能力使用指引](#)。

其他

程序配置信息

您可以配置以下参数：

- 环境 ID：选择要部署的环境，在哪个环境下使用。

计费

此能力使用云开发或其他腾讯云服务，可能会产生相关费用：

- 数据万象（[产品定价](#) 及 [使用明细](#)）。
- 云存储（[产品定价](#) 及 [使用明细](#)）。
- 云函数（[产品定价](#) 及 [使用明细](#)）。

当您使用该能力时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

启用的 API 和创建的资源

- **Type:** 数据万象
Description: 为开发者提供图片、视频等多种数据的智能处理服务。
- **Type:** Cloud Storage
Description: 存储图片，并通过CDN提升图片加载速度。
- **Type:** Cloud Function
Description: 检测图像处理参数并为图像处理生成签名，保证操作的合法性。

权限授予

主账户

角色名称	授权策略名	角色类型	描述
------	-------	------	----

角色名称	授权策略名	角色类型	描述
CI_QCSRole	QcloudAccessForCIRole、 QcloudCOSDataFullControl	服务角色	数据万象（CI）会对您腾讯云资源进行访问操作，含对对象存储（COS）数据的读取、修改、删除、列出等
TCB_QcsRole	QcloudCIFullAccess	服务角色	云开发（TCB）会对您的数据万象资源进行操作，方便您在该能力中使用该服务

子账户

如果想让子账户也能使用该能力，需要为子账户授予如下权限才能使用：

- 策略:** QcloudAccessForTCBRole
描述: 云开发（TCB）对云资源的访问权限。
- 策略:** QcloudCIFullAccess
描述: 数据万象全读写访问权限。

图像安全审核

最近更新时间：2020-07-03 16:31:04

图像安全审核提供鉴黄、鉴政、鉴暴恐等多种类型的敏感内容审核服务，有效识别违禁图片，规避违规风险。

功能特性

审核类型	描述
涉黄识别	检测色情、性感内容，有效降低人工审核成本，适用于各类社交平台、即时通讯应用，帮助平台规避涉黄风险
涉暴恐识别	识别图片中的恐怖分子、邪教组织、枪支弹药、血腥暴力等恐怖主义内容，帮助平台规避法律风险
涉政识别	通过人脸识别、图片泛化等技术检测图片涉政内容，帮助平台及时响应法律法规要求，规避舆论风险
广告识别	通过广告引导类图片的识别，帮助平台屏蔽广告信息

注意：

您不仅在云函数中可以使用该能力，也可以在客户端（web、微信小程序、QQ小程序）使用，文件读写权限策略与云存储一致，减去您额外的权限管理工作。

前置要求

已经开通云开发。

安装程序

注意：

微信小程序开发者请使用【其他登录方式】-【微信公众号登录】登录，再选择关联的小程序账户登录；QQ小程序开发者可直接通过QQ小程序开发者IDE【云开发】按钮登录，也可以通过关联的腾讯云账户登录。

您可以通过 [云开发控制台](#)，来安装和管理该能力。

使用程序

注意：

如果您在 web 网站使用该能力，请先在 [云开发控制台](#) 将网站域名添加为当前环境的安全域名。

1. 安装 SDK 到项目

web端及云函数中使用：

```
npm install --save @cloudbase/extension-ci@latest
```

微信小程序中使用：

```
npm install --save @cloudbase/extension-ci-wxmp@latest
```

微信小程序中npm使用请参考 [npm 支持](#)。

2. 调用 SDK

调用参数

名称	类型	是否必须	说明
action	String	是	操作类型，传：DetectType
cloudPath	String	是	文件的绝对路径，与tcb.uploadFile中一致
operations	Object	是	处理参数。

operations节点内容

名称	类型	是否必须	说明
type	String	是	审核类型支持：porn（涉黄识别）、terrorist（涉暴恐识别）、politics（涉政识别）、ads（广告识别），可选择多种识别类型，如porn,ads

返回内容

参数名称	类型	描述
RecognitionResult	Object	内容识别结果

RecognitionResult 节点内容：

参数名称	类型	描述
PornInfo	Object	鉴黄审核信息
TerroristInfo	Object	鉴暴恐审核信息
PoliticsInfo	Object	鉴政审核信息
AdsInfo	Object	广告审核信息

审核信息 (PornInfo、TerroristInfo、PoliticsInfo、AdsInfo) 中包含如下内容：

参数名称	类型	描述
Code	Int	错误码，0为正确，其他数字对应相应错误。详情请参见 错误码
HitFlag	Int	是否命中：0(未命中)，1(命中)，2(疑似)
Score	Int	审核分值：0 - 60分(正常)，60 - 90分(疑似敏感)，90 - 100分(确定敏感)
Label	String	识别出的图片标签

web端使用

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-js-sdk");
```

云函数中使用

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-admin-node");
```

您可以根据需要选择在【web端】或【云函数】中使用，再按照如下代码调用：

```
tcb.init({
  env: "您的环境ID"
```

```
});

tcb.registerExtension(extCi);

async function demo() {
try {
const opts = {
  type: "porn"
}
const res = await tcb.invokeExtension('CloudInfinite',{
  action:'DetectType',
  cloudPath: "ab.png", // 图像在云存储中的路径, 与tcb.uploadFile中一致
  operations: opts
})
console.log(JSON.stringify(res.data, null, 4));
} catch (err) {
console.log(JSON.stringify(err, null, 4));
}
}
```

微信小程序客户端中使用

注意：

微信小程序中关联的云开发环境需要与该能力安装的环境一致；取值参数及返回与在云函数、web端使用一致。

```
const extCi = require("@cloudbase/extension-ci-wxmp");
async function demo() {
const opts = {
  type: "porn"
}
extCi.invoke({
  action:'DetectType',
  cloudPath: "ab.png", // 图像在云存储中的路径, 与tcb.uploadFile中一致
  operations: opts
}).then(res => {
console.log(JSON.stringify(res.data, null, 4));
}).catch(err => {
console.log(JSON.stringify(err, null, 4));
})
}
```

QQ小程序客户端中使用

请参考 [QQ小程序扩展能力使用指引](#)。

其他

程序配置信息

您可以配置以下参数：

- 环境 ID：选择要部署的环境，在哪个环境下使用。

计费

此能力使用云开发或其他腾讯云服务，可能会产生相关费用：

- 数据万象（[产品定价](#) 及 [使用明细](#)）。
- 云存储（[产品定价](#) 及 [使用明细](#)）。
- 云函数（[产品定价](#) 及 [使用明细](#)）。

当您使用该能力时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

启用的 API 和创建的资源

- **Type:** 数据万象
Description: 为开发者提供图片、视频等多种数据的智能处理服务。
- **Type:** Cloud Storage
Description: 存储图片，并通过 CDN 提升图片加载速度。
- **Type:** Cloud Function
Description: 检测图像处理参数并为图像处理生成签名，保证操作的合法性。

权限授予

主账户

角色名称	授权策略名	角色类型	描述
CI_QCSRole	QcloudAccessForCIRole、 QcloudCOSDataFullControl	服务角色	数据万象（CI）会对您腾讯云资源进行访问操作，含对对象存储（COS）数据的读取、修改、删除、列出等

角色名称	授权策略名	角色类型	描述
TCB_QcsRole	QcloudCIFullAccess	服务角色	云开发（TCB）会对您的数据万象资源进行操作，方便您该能力中使用该服务

子账户

如果想让子账户也能使用该能力，需要为子账户授予如下权限才能使用：

- **策略:** QcloudTCBFullAccess
描述: 云开发全读写访问权限。
- **策略:** QcloudCIFullAccess
描述: 数据万象全读写访问权限。

图像盲水印

最近更新时间：2020-07-03 16:31:24

图像盲水印将水印信息以不可见的形式添加到原图信息中，您可对疑似被盗取的资源进行盲水印提取，验证图片归属。

功能特性

开通后，每个账户拥有三千次免费额度，超出后进行计费。为您提供半盲、全盲和文字盲水印三种类型，您可根据需要进行选择：

水印类型	特性	适用场景
半盲水印 (type1)	抗攻击性强，但提取水印需原图	小图 (640x640以下) 使用
全盲水印 (type2)	提取方便，提取水印仅需水印图	无需对比原图 批量添加，批量校验
文字水印 (type3)	可直接将文字信息添加至图片中	终端信息添加

适用场景

鉴权追责

您可对图片资源增加半盲水印，在发现恶意攻击方盗取您的资源后将疑似被盗取图取回，并与相应原图进行盲水印提取操作，若能够得到有效水印图即可证明资源归属。

上传查重

为解决部分用户使用其他用户资源重复上传相同信息的问题（如房产图、汽车图、商品图等），您可在用户上传图片资源前先进行全盲水印提取，若提取到水印图信息则证明该图片来自之前已有资源，并进行相应操作（如提醒用户请勿重复上传资源）；若不存在全盲水印则添加全盲水印，保护图片资源不被其他用户下载后重复上传。

资源防泄露

对于内部分享的图片资源，您可通过文字盲水印将访问方的信息在请求图片时添加至图片中，当资源泄露后可通过流传出的资源图提取出盲水印，进而得到泄露方信息。

注意：

您不仅在云函数中可以使用该能力，也可以在客户端（web、微信小程序、QQ小程序）使用，文件读写权限策略与云存储一致，减去您额外的权限管理工作。

前置要求

已经开通云开发。

安装程序

注意：

微信小程序开发者请使用【其他登录方式】-【微信公众号登录】登录，再选择关联的小程序账户登录；QQ小程序开发者可直接通过QQ小程序开发者IDE【云开发】按钮登录，也可以通过关联的腾讯云账户登录。

您可以通过 [云开发控制台](#)，来安装和管理该能力。

使用程序

注意：

如果您在 web 网站使用该能力，请先在 [云开发控制台](#) 将网站域名添加为当前环境的安全域名。其他要求如下：

- 使用盲水印功能，水印图的宽高不得超过原图的1/8。
- 为保证盲水印效果，水印图请选用黑底白色图片。
- 数据万象为每个账户提供每月3000张的免费体验额度，超出后将正常计费。未使用额度不会累积至下一月。
- 文字盲水印当前支持数字[0 - 9]及英文大小写[A - Z,a - z]。
- 盲水印可抵抗裁剪、涂抹、变色等多种图片盗取攻击，防盗效果与原图大小及攻击程度相关。

下载时添加水印

图片下载时添加盲水印与添加普通水印操作相同，只需在图片访问链接后使用 watermark 参数即可。相关内容如下：

```
watermark/3/type/<type>/image/<imageUrl>/text/<text>  
// 例如：http://xxxxx.tcb.qcloud.la/demo.jpeg?watermark/3/type/3/text/dGVuY2VudCBjbG91ZA==
```

水印参数取值如：

参数	类型	必选	描述
type	Int	是	盲水印类型，有效值：1 半盲；2 全盲；3 文字
image	String	否	水印图片地址，需要经过 URL 安全的 Base64 编码。当 type 为1或2时必填，type 为3时无效。须同时满足 3 个条件： 1. 水印图片存储在当前环境的云存储中； 2. 将您的文件访问域名中 tcb.qcloud.la 替换为 picsh.myqcloud.com 即可，如：examplebucket-1250000000.picsh.myqcloud.com； 3. URL 必须以 http:// 开始。
text	String	是	盲水印文字，需要经过 URL 安全的 Base64 编码。当 type 为3时必填，type 为1或2时无效。

上传图片并添加水印及提取水印

1. 安装 SDK 到项目

web端及云函数中使用：

```
npm install --save @cloudbase/extension-ci@latest
```

微信小程序中使用：

```
npm install --save @cloudbase/extension-ci-wxmp@latest
```

微信小程序中npm使用请参考 [npm 支持](#)。

2. 调用 SDK

调用参数

名称	类型	是否必须	说明
action	String	是	操作类型，传：WaterMark
cloudPath	String	是	文件的绝对路径，与 tcb.uploadFile 中一致
fileContent	Uint8Array或 Buffer	否	文件内容，有值，表示上传时处理图像；为空，则处理已经上传的图像
operations	Object	是	盲水印处理参数

operations节点内容

名称	类型	是否必须	说明
rules	Array.<Rule对象>	是	处理样式

Rule对象内容

名称	类型	是否必须	说明
fileid	String	是	处理结果的文件路径，如以"/"开头，则存入指定文件夹中，否则，存入原图文件存储的同目录
rule	Object	是	处理参数

添加水印处理参数 (rule) 取值如下：

参数	类型	必选	描述
mode	Int	是	操作类型，固定为3，添加水印
type	Int	是	盲水印类型，有效值：1 半盲；2 全盲；3 文字，必须跟添加盲水印时的 type 类型一致
image	String	否	盲水印图片地址，当前环境云存储中的路径
text	String	否	盲水印文字，当 type 为3时必填，type 为1或2时无效。

提取水印处理参数 (rule) 取值如下：

参数	类型	必选	描述
mode	Int	是	操作类型，固定为4，提取水印
type	Int	是	盲水印类型，有效值：1 半盲；2 全盲；3 文字，必须跟添加盲水印时的 type 类型一致
image	String	否	图片地址，当 type 为1或2时必填，type为3时无效。 type 为1时，为未带盲水印的原图图片地址；type 为2时，为水印图地址(黑底白色图片的地址)
text	String	否	盲水印文字，当 type 为3时必填，type 为1或2时无效

返回内容

参数名称	类型	描述
UploadResult	Object	原图信息

UploadResult 节点内容：

参数名称	类型	描述
ProcessResults	Object	图片处理结果

ProcessResults 节点内容：

节点名称	类型	描述
Object	Object	每一个图片处理结果

Object 节点内容：

节点名称	类型	描述
Key	String	文件名
Location	String	图片路径
Format	String	图片格式
Width	Int	图片宽度
Height	Int	图片高度
Size	Int	图片大小
Quality	Int	图片质量
WatermarkStatus	Int	提取盲水印且当 type 为2时返回该字段，表示提取到全盲水印的可信度。具体为0-100的数字，75分以上表示确定有盲水印，60-75表示疑似有盲水印，60以下可认为未提取到盲水印

调用示例

web端使用：

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-js-sdk");
```

```
const readFile = async function(file) {
  let reader = new FileReader();
  let res = await new Promise(resolve => {
    reader.onload = function(e) {
      let arrayBuffer = new Uint8Array(e.target.result);
      resolve(arrayBuffer);
    };
    reader.readAsArrayBuffer(file);
  });
  return res;
};
let file = document.getElementById("selectFile").files[0];
let fileContent = await readFile(file);
```

云函数中使用：

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-admin-node");
let fileContent = imageBuffer; // Uint8Array|Buffer格式图像内容
```

您可以根据需要选择在【客户端】或【云函数】中使用，再按照如下代码调用：

```
tcb.init({
  env: "您的环境ID"
});
tcb.registerExtension(extCi);

async function addWaterMark() {
  try {
    const opts = {
      rules: [
        {
          // 处理结果的文件路径，如以'/'开头，则存入指定文件夹中，否则，存入原图文件存储的同目录
          fileid: "/watermark/demo.jpeg",
          rule: { // 添加水印处理规则
            mode: 3,
            type: 3,
            text: 'tcb' // 支持数字[0 - 9]及英文大小写[A - Z,a - z]
          }
        }
      ]
    };
    const res = await tcb.invokeExtension("CloudInfinite", {
      action: 'WaterMark',
```

```
cloudPath: "demo.jpeg", // 图像在云存储中的路径, 与tcb.uploadFile中一致
fileContent, // 该字段可选, 文件内容: Uint8Array|Buffer。有值, 表示上传时处理图像; 为空, 则处理已经
上传的图像
operations:opts
});
console.log(JSON.stringify(res.data, null, 4));
} catch (err) {
console.log(JSON.stringify(err, null, 4));
}
}

async function getWaterMark() {
try {
const opts = {rules: [
{
fileid: "/watermark/demo.jpeg",
rule: { // 提取水印处理规则
mode: 4,
type: 1,
image: '/tcb.jpeg'
}
}
]}
const res = await tcb.invokeExtension("CloudInfinite", {
action: 'WaterMark',
cloudPath: "demo.jpeg", // 存储图像的绝对路径, 与tcb.uploadFile中一致
operations:opts
});
console.log(JSON.stringify(res.data, null, 4));
} catch (err) {
console.log(JSON.stringify(err, null, 4));
}
}
```

微信小程序客户端中使用：

注意：

微信小程序中关联的云开发环境需要与该能力安装的环境一致；取值参数及返回与在云函数、web端使用一致。

```
const extCi = require("@cloudbase/extension-ci-wxmp");
const readFile = async function() {
```

```
let res = await new Promise(resolve=>{
  wx.chooseImage({
    success: function(res) {
      let filePath = res.tempFilePaths[0]
      let fm = wx.getFileSystemManager()
      fm.readFile({
        filePath,
        success(res){
          resolve(res)
        }
      })
    }
  })
})
return res
}
let fileContent = await readFile();
async function demo() {
  const opts = {
    rules:
    [
      {
        // 处理结果的文件路径，如以'/' 开头，则存入指定文件夹中，否则，存入原图文件存储的同目录
        fileid: "/watermark/demo.jpeg",
        rule: { // 添加水印处理规则
          mode: 3,
          type: 3,
          text: 'tcb' // 支持数字[0 - 9]及英文大小写[A - Z,a - z]
        }
      }
    ]
  }
  extCi.invoke({
    action:'WaterMark',
    cloudPath: "demo.jpeg", // 图像在云存储中的路径，与tcb.uploadFile中一致
    fileContent, // 可选参数
    operations:opts
  }).then(res => {
    console.log(JSON.stringify(res.data, null, 4));
  }).catch(err => {
    console.log(JSON.stringify(err, null, 4));
  })
}
```


QQ小程序客户端中使用：

请参考 [QQ小程序扩展能力使用指引](#)。

其他

程序配置信息

您可以配置以下参数：

- 环境 ID：选择要部署的环境，在哪个环境下使用。

计费

此能力使用云开发或其他腾讯云服务，可能会产生相关费用：

- 数据万象（[产品定价](#) 及 [使用明细](#)）。
- 云存储（[产品定价](#) 及 [使用明细](#)）。
- 云函数（[产品定价](#) 及 [使用明细](#)）。

当您使用该能力时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

启用的 API 和创建的资源

- **Type:** 数据万象
Description: 为开发者提供图片、视频等多种数据的智能处理服务。
- **Type:** Cloud Storage
Description: 存储图片，并通过CDN提升图片加载速度。
- **Type:** Cloud Function
Description: 检测图像处理参数并为图像处理生成签名，保证操作的合法性。

权限授予

主账户

角色名称	授权策略名	角色类型	描述
CI_QCSRole	QcloudAccessForCIRole、 QcloudCOSDataFullControl	服务角色	数据万象（CI）会对您腾讯云资源进行访问操作，含对对象存储（COS）数据的读取、修改、删除、列出等

角色名称	授权策略名	角色类型	描述
TCB_QcsRole	QcloudCIFullAccess	服务角色	云开发（TCB）会对您的数据万象资源进行操作，方便您在该能力中使用该服务

子账户

如果想让子账户也能使用该能力，需要为子账户授予如下权限才能使用：

- **策略:** QcloudAccessForTCBRole
描述: 云开发（TCB）对云资源的访问权限。
- **策略:** QcloudCIFullAccess
描述: 数据万象全读写访问权限。

图像标签

最近更新时间：2020-07-03 16:31:15

图像标签对云存储中存量数据的图片标签识别，返回图片中置信度较高的主题标签，帮助开发者分析图像。

功能特性

识别图片中的场景、物品、人物等信息，例如自然风光（山、海、天空、夕阳等）、人造环境（建筑、游乐场、会议室等）、人像（男性、女性、自拍、合照等）、物品（美食、穿着、日常用品等）、动物（猫、狗、鸟类、哺乳动物、海洋生物等各种宠物或野生动物）。

适用场景

可以用于相册分类、信息流内容推荐、广告推荐、图库分类检索、视频内容理解、拍照识图等各种场景。

注意：

您不仅在云函数中可以使用该能力，也可以在客户端（web、微信小程序、QQ小程序）使用，文件读写权限策略与云存储一致，减去您额外的权限管理工作。

前置要求

已经开通云开发。

安装程序

注意：

微信小程序开发者请使用【其他登录方式】-【微信公众号登录】登录，再选择关联的小程序账户登录；QQ小程序开发者可直接通过QQ小程序开发者IDE【云开发】按钮登录，也可以通过关联的腾讯云账户登录。您可以通过 [云开发控制台](#)，来安装和管理该能力。

使用程序

注意：

如果您在 web 网站使用该能力，请先在 [云开发控制台](#) 将网站域名添加为当前环境的安全域名。

1. 安装 SDK 到项目

web端及云函数中使用：

```
npm install --save @cloudbase/extension-ci@latest
```

微信小程序中使用：

```
npm install --save @cloudbase/extension-ci-wxmp@latest
```

微信小程序中npm使用请参考 [npm 支持](#)。

2. 调用 SDK

调用参数

名称	类型	是否必须	说明
action	String	是	操作类型，传：DetectLabel
cloudPath	String	是	文件的绝对路径，与 tcb.uploadFile 中一致

返回内容

参数名称	类型	描述
RecognitionResult	Object	图片标签识别结果

RecognitionResult 节点内容：

参数名称	类型	描述
Labels	Object	标签信息

标签信息中包含如下内容：

参数名称	类型	描述
Confidence	Int	该标签的置信度分数，分数越高则该标签准确度越高
Name	String	识别出的图片标签

调用示例

web端中使用

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-js-sdk");
```

云函数中使用

```
const extCi = require("@cloudbase/extension-ci");
const tcb = require("tcb-admin-node");
```

您可以根据需要选择在【web端】或【云函数】中使用，再按照如下代码调用：

```
tcb.init({
  env: "您的环境ID"
});

tcb.registerExtension(extCi);

async function demo() {
  try {
    const res = await tcb.invokeExtension("CloudInfinite", {
      action: "DetectLabel",
      cloudPath: "demo.png" // 图像在云存储中的路径，与tcb.uploadFile中一致
    });
    console.log(JSON.stringify(res.data, null, 4));
  } catch (err) {
    console.log(JSON.stringify(err, null, 4));
  }
}
```

微信小程序客户端中使用

注意：

微信小程序中关联的云开发环境需要与该程序安装的环境一致；取值参数及返回与在云函数、web端使用一致。

```
const extCi = require("@cloudbase/extension-ci-wxmp");
async function demo() {
  extCi.invoke({
    action: "DetectLabel",
    cloudPath: "demo.png" // 图像在云存储中的路径，与tcb.uploadFile中一致
  }).then(res => {
    console.log(JSON.stringify(res.data, null, 4));
  }).catch(err => {
    console.log(JSON.stringify(err, null, 4));
  })
}
```

QQ小程序客户端中使用

请参考 [QQ小程序扩展能力使用指引](#)。

其他

程序配置信息

您可以配置以下参数：

- 环境 ID：选择要部署的环境，在哪个环境下使用。

计费

此能力使用云开发或其他腾讯云服务，可能会产生相关费用：

- 数据万象（[产品定价](#) 及 [使用明细](#)）。
- 云存储（[产品定价](#) 及 [使用明细](#)）。
- 云函数（[产品定价](#) 及 [使用明细](#)）。

当您使用该能力时，只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

启用的API和创建的资源

- **Type:** 数据万象
Description: 为开发者提供图片、视频等多种数据的智能处理服务。

- **Type:** Cloud Storage
Description: 存储图片，并通过 CDN 提升图片加载速度。
- **Type:** Cloud Function
Description: 检测图像处理参数并为图像处理生成签名，保证操作的合法性。

权限授予

主账户

角色名称	授权策略名	角色类型	描述
CI_QCSRole	QcloudAccessForCIRole、 QcloudCOSDataFullControl	服务角色	数据万象（CI）会对您腾讯云资源进行访问操作，含对对象存储（COS）数据的读取、修改、删除、列出等
TCB_QcsRole	QcloudCIFullAccess	服务角色	云开发（TCB）会对您的数据万象资源进行操作，方便您在该能力中使用该服务

子账户

如果想让子账户也能使用该能力，需要为子账户授予如下权限才能使用：

- **策略:** QcloudAccessForTCBRole
描述: 云开发（TCB）对云资源的访问权限。
- **策略:** QcloudCIFullAccess
描述: 数据万象全读写访问权限。

CMS内容管理系统

最近更新时间：2020-07-03 16:30:19

云开发 CMS 内容管理系统是云开发 CloudBase 提供的一个扩展程序，可以在云开发控制台一键安装在自己的云开发环境中，让开发人员和内容运营者随时随地管理小程序 / Web 等多端产生的内容数据。无须编写代码即可使用，还提供了 PC / 移动端浏览器的访问支持，支持文本、富文本、Markdown、图片、文件、关联类型等多种类型的可视化编辑。

推荐使用场景

1. 适用于需要为小程序应用增加一个运营管理后台的业务

CMS 扩展非常适合小程序的商品管理、文章编辑和发布、运营活动配置、素材管理等数据和内容管理的场景。使用 CMS 扩展，可以省去手动线上修改数据库数据或者投入人力物力开发管理后台的麻烦，只需要安装之后进行一些简单的配置，就可以随时随地使用 CMS 内容管理系统来管理内容，同时还提供了管理员和运营者的两套身份体系的权限控制。

2. 适用于快速开发内容型的网站应用、小程序应用的场景

CMS 扩展还可以用来配合开发网站应用和小程序应用，提升开发效率。使用 CMS 扩展可以解决内容和数据的管理和生产问题，并且省去一部分后端开发工作，直接可以结合前端应用框架读取云开发数据库数据进行渲染。例如基于 CMS 可以快速开发博客、企业官网等内容型的网站、小程序应用。

功能特性

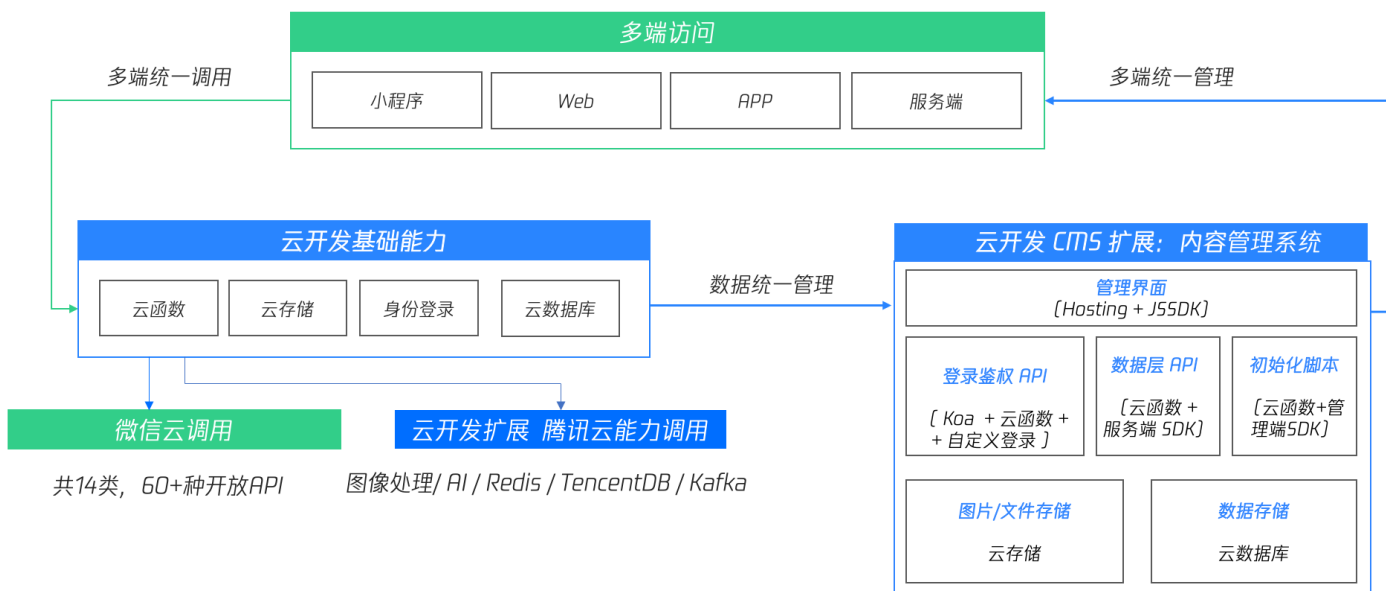
特性	介绍
免开发	基于后台建模配置生成内容管理界面，无须编写代码
多端适配	支持 PC/移动端访问和管理内容
功能丰富	支持文本、富文本、图片、文件 等多种类型内容的可视化编辑，并且支持内容关联
权限控制	系统基于管理员/运营者两种身份角色的访问控制
外部系统集成	支持 Webhook 接口，可以用于在运营修改修改内容后通知外部系统，如自动构建静态网站、发送通知等
数据源兼容	支持管理小程序/ Web / 移动端的云开发数据，支持管理已有数据集合，也可以在 CMS 后台创建新的内容和数据集合

特性	介绍
部署简单	可在云开发控制台扩展管理界面一键部署和升级

工作原理

说明：

使用 CMS 扩展时将在当前环境创建云函数、云数据库等资源，查询 [部署详情](#)。



视频教程

您也可以通过观看以下视频教程，部署云开发 CMS 内容管理系统。

[点击查看视频](#)

安装与使用

第一步：开通并创建云开发按量计费环境

开通云开发 CMS 扩展，需要开通并创建云开发按量计费环境。开发者可以根据下面的身份角色参考指引进行开通和创建。

- [微信小程序开发者](#)
- [腾讯云开发者](#)
- [QQ 小程序开发者](#)

微信小程序开发者

开通云开发环境

- 注册小程序账号：在微信公共平台进行申请并提交相应资料。 [前往注册](#)
- 下载微信开发者工具：简单高效地开发和调试小程序，并可预览和发布。 [前往下载](#)
- 开通云开发：在开发者工具中点击“云开发”按钮开通，开通后会自动开通环境。

切换按量计费

- 下载最新的 [Nightly Build 版本](#)的[微信开发者工具](#)，登录微信开发者工具 > 云开发控制台。
- 在【云开发控制台】>【设置】>【环境设置】>【支付方式】中点击切换【按量付费】即可。

注意：

目前仅支持从腾讯云账户付款，云开发环境切换【按量付费】模式前，请确保腾讯云账户中余额充足，否则将切换失败。



腾讯云开发者

开通并创建按量付费环境

- 注册腾讯云账号，并完成实名认证。
- 进入 [云开发控制台](#)，授权开通云开发。
- 单击进入控制台 [创建按量计费环境](#)（温馨提示，无资源消耗时，按量计费不产生任何费用）。

切换包年包月计费方式为按量计费

如果已经开通过云开发环境，需要在计费模式中切换计费方式为按量计费

登录腾讯云 [云开发控制台](#)。在云开发 CloudBase 控制台 > 环境 > [资源购买](#) 中的【计费模式】中单击【切换按量付

费】即可。



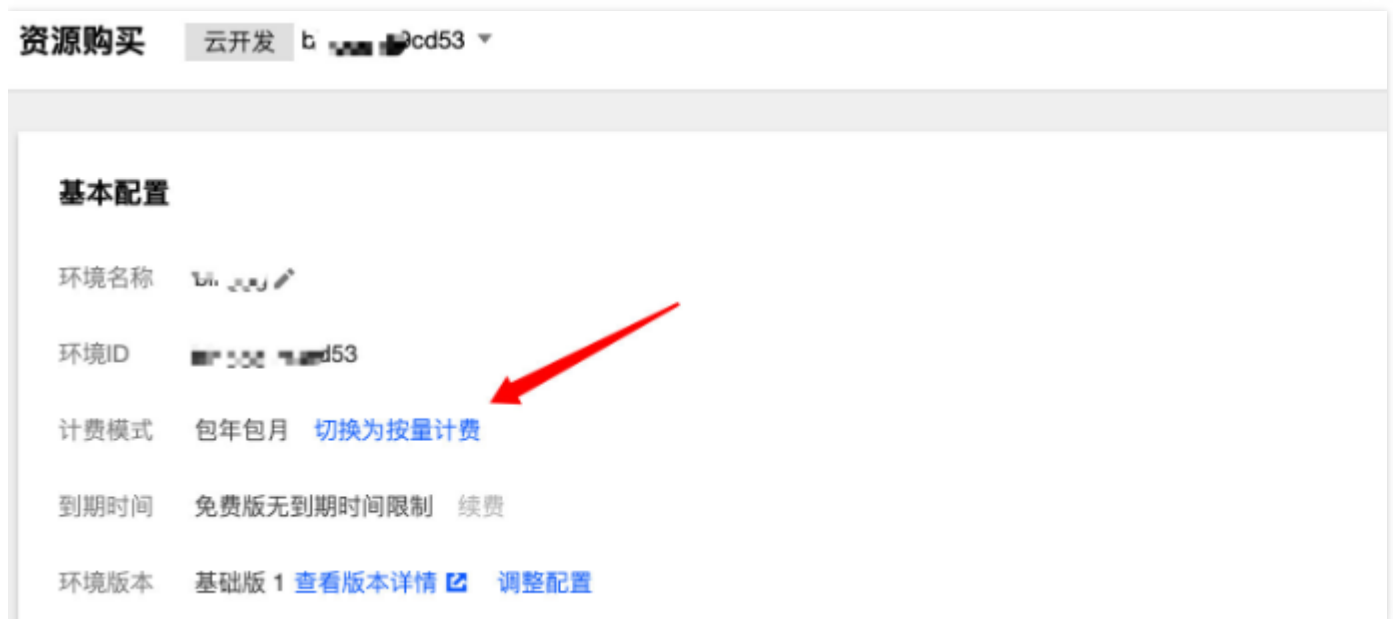
QQ 小程序开发者

开通环境

在 QQ 小程序开发者工具工具栏左侧，点击“云开发”按钮即可打开云控制台、根据提示开通云开发、创建云环境。

切换为按量计费

在云开发 CloudBase 控制台 > 环境 > [资源购买](#) 中的【计费模式】中点击【切换按量付费】即可。



第二步：在腾讯云控制台安装扩展

- 登录腾讯云 [云开发控制台](#)。

注意：

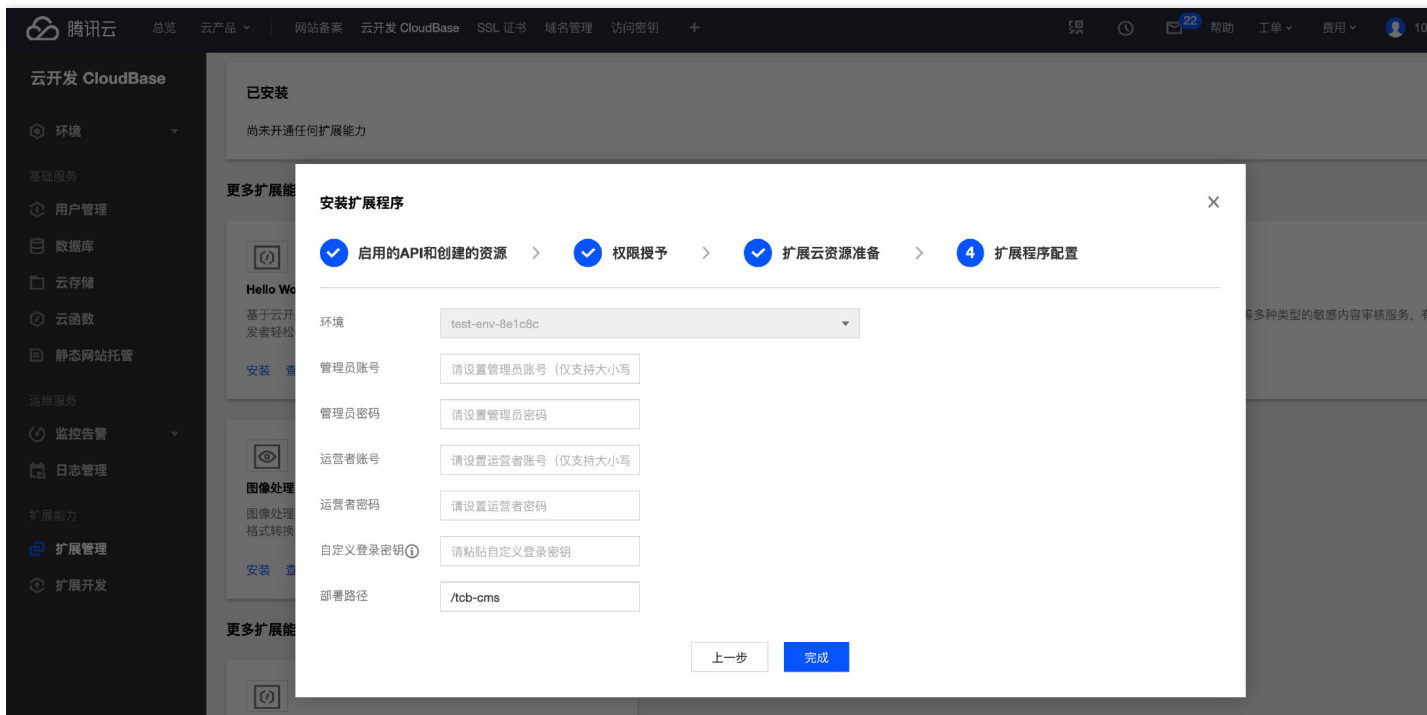
微信小程序开发者请使用【其他登录方式】-【微信公众号登录】登录，再选择关联的小程序账户登录。



注意：

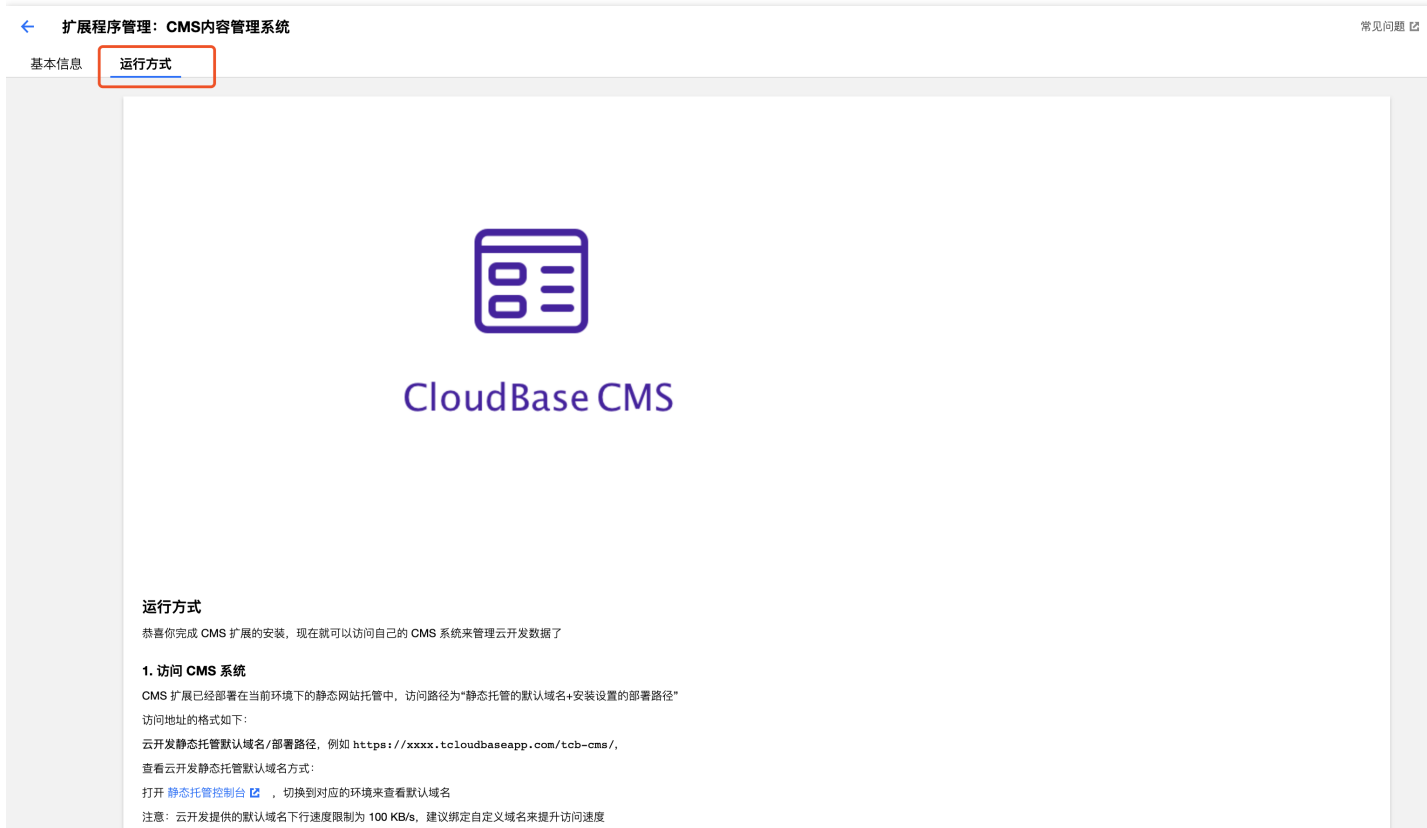
QQ 小程序开发者可直接通过 QQ 小程序开发者 IDE【云开发】按钮登录，也可以通过关联的腾讯云账户登录。

- 打开云开发控制台 > 扩展能力 > [扩展管理](#) 页面。
- 选择目标按量计费环境后，在更多扩展能力下单击【安装】【CMS 内容管理系统】扩展，并按提示完成扩展安装。
- 安装时需要进行资源的授权和扩展程序的配置，如管理员和运营者的账号密码配置等，同时需要提供自定义登录的密钥，可以点击自定义登录密钥旁边的小图标了解如何填写。



第三步：使用 CMS 内容管理系统

完成【CMS 内容管理系统】的安装以后，然后访问该扩展的管理页，可以在【扩展运行方式】Tab 查看使用指引，依照文档完成 CMS 的使用，下面简单介绍一下快速上手的步骤，更多细节可以参考运行方式。



访问 CMS 系统

CMS 扩展已经部署在当前环境下的静态网站托管中，访问路径为“静态托管的默认域名+安装设置的部署路径”。

访问地址的格式如下：

云开发静态托管默认域名/部署路径，例如 `https://xxxx.tcloudbaseapp.com/tcb-cms/`

账号登录

打开 CMS 系统后首先会提示需要登录，我们首先使用使用安装扩展时设置的管理员账号和密码进行登录。

内容建模

登录成功后，首先需要进行内容的建模设置，例如我们想为自己的博客应用（小程序/网站）来生成管理界面。

假设当前已有一个管理文章的数据库集合 `articles`，我们可以在 CMS 管理后台新建一个“文章”内容（如果新建内容的时候指定的集合名不存在，CMS 扩展会自动新建集合）来生成“文章”类型的内容管理界面。

假设数据库集合 `articles` 的结构如下：

字段名	类型	描述
<code>_id</code>	ID	文章唯一 id
<code>name</code>	String	文章标题
<code>cover</code>	String	封面图，这里存放云开发的存储的文件的 cloudID
<code>content</code>	String	文章内容，采用 markdown 格式
<code>author</code>	ID	作者的用户 id
<code>createTime</code>	DateTime	创建时间
<code>updateTime</code>	DateTime	更新时间
<code>tag</code>	String[]	标签，例如 <code>["serverless","cms"]</code>
<code>category</code>	String[]	分类，例如 <code>["前端","开发"]</code>

在“内容设置”中点击“新建”来创建“文章”类型时，可以对照上面的集合数据把字段类型和字段的限制进行配置，例如封面图可以直接选择“图片”字段类型，文章内容可以直接选择“Markdown”类型，这样在生成的管理界面里可以直接上传图片 and 通过编辑器编写文章，保存在数据库集合的时候，依然会保存为数据库支持的类型，图片会存储为云

存储的 CloudID, 内容会存储为字符串等。创建并保存之后会自动刷新生成“文章”的运营界面。

数据库集合名 *
articles
内容对应的数据库集合名, 可填写现有集合名, 如集合不存在则会自动创建

图标设置
选择图标
排序 (值越大越靠前)
9

描述信息
会展示在对应内容的管理页面顶部

字段列表

1	字段名称 * 文章标题 字段中文显示名称 最大长度 140	数据库字段名 * name 该字段在数据库中保存的名称 默认值	字段类型 * 字符串 最小长度 1 是否必填 <input checked="" type="checkbox"/>	字段说明 说明文字会展示在输入框下方
2	字段名称 * 封面图 字段中文显示名称 字段说明 说明文字会展示在输入框下方	数据库字段名 * cover 该字段在数据库中保存的名称 是否必填 <input type="checkbox"/>	字段类型 * 图片 默认值	封面图选择图片类型
3	字段名称 * 内容 字段中文显示名称 字段说明 说明文字会展示在输入框下方	数据库字段名 * content 该字段在数据库中保存的名称 是否必填 <input checked="" type="checkbox"/>	字段类型 * Markdown 默认值	正文选择 Markdown 类型

管理内容

可以使用运营者身份登录，对新创建的“文章”进行操作，我们可以新建一篇文章。

云开发 CMS 内容管理系统

运营 / 新建

文章标题 *
比块更快, Github Action + 云开发部署静态网站

封面图
将图片文件拖拽到这里, 或点击这里选择图片文件。

预览图片

编辑工具: H B I S ...

Github 为开源项目提供了用于静态页面展示的 Pages 服务。很多开发者都在上面托管了自己的静态网站和博客。不少开源项目的案例和文档页面也采用了这种方式。不过由于 Pages 的 CDN 节点大部分在国外，在国内的访问速度不是很理想，不少开发者希望能“提升网站在国内的访问速度和稳定性”，今天会介绍如何配合 Github Action 的持续集成服务和“云开发 Github Action 扩展”，自动部署到访问速度更快的云开发静态托管服务。

!Pages 国内测速数据[https://user-gold-cdn.xitu.io/2020/4/12/1716e89fec78f4d7?w=1886&h=1080&f=jpeg&s=201144]

云开发静态托管介绍

云开发静态托管是云开发提供的静态网站托管的能力，静态资源（HTML、CSS、JavaScript、字体等）的分发由腾讯云对象存储 COS 和拥有多个边缘节点的腾讯云 CDN 提供支持。可以通过腾讯云控制台、命令行工具以及下面提到的“云开发 Github Action”进行管理部署。云开发提供了免费的二级域名（未绑定自定义域名时下行速度有限制），同时支持免费绑定开发者自己的“自定义域名”。

首个环境可享受“1GB 容量”和每月“5GB 流量”的免费额度，对于访问量不是很大的个人博客应该足够了。如果流量大也没关系，计费是按照具体使用收费的按量付费方式，具体信息可以查看计费价格文档 [https://cloud.tencent.com/product/wh/pricing][https://cloud.tencent.com/product/wh/pricing]

云开发静态托管部署的网站，还可以使用云开发的提供的“一站式 Serverless 后端能力”，例如云函数、云数据库、云存储、身份服务等。比如可以在静态托管的个人博客上面使用云函数和云数据库实现“评论、留言板”功能等，或者可以把博客的内容管理从原来的静态文件部署变为“动态内容管理”等，拓展的用法非常的多，开发者可以继续深入探索。

如何通过 Github Action 一键部署到云开发

Desktop Tablet Mobile/Wechat

Github 为开源项目提供了用于静态页面展示的 Pages 服务。很多开发者都在上面托管了自己的静态网站和博客。不少开源项目的案例和文档页面也采用了这种方式。不过由于 Pages 的 CDN 节点大部分在国外，在国内的访问速度不是很理想，不少开发者希望能提升网站在国内的访问速度和稳定性，今天会介绍如何配合 Github Action 的持续集成服务和云开发 Github Action 扩展，自动部署到访问速度更快的云开发静态托管服务。

文章发布成功后，即可在文章列表中看到这篇文章。

云开发 CMS 内容管理系统

运营 / 文章

增加检索 + 新建 导出

<input type="checkbox"/>	文章标题	封面图	作者	创建时间	更新时间	标签	分类
<input type="checkbox"/>	比块更快, Github Action + 云开发部署静态网站		booker	2020/4/12	2020/5/5	CloudBase 云开发 静态网站 Github Action 博客	前端 工具

每页行数: 10 1-1/1

使用内容数据

采用 CMS 管理的内容，依然可以通过云开发各端 SDK 进行访问（需要注意的是在前端访问时，需要正确设置数据库的安全规则设置，例如设置为所有用户可读，仅创建者可写）。

例如，在上面的例子里，我们需要在云函数中获取文章的标签是 `CloudBase` 的最新 10 篇文章，可以采用以下代码来获取数据：

```
db.collection("articles")
  .where({ tag: "CloudBase" })
  .orderBy("createTime", "desc")
  .limit(10)
  .get();
```

获取到内容数据就可以在各种场景使用了，例如在小程序/ Web 中构建应用和网站。

```
db.collection('articles')
  .where({ tag: 'CloudBase'})
  .orderBy('createTime', 'desc')
  .limit(10)
  .get().then(console.log)
|> Promise {<pending>}
|> {data: Array(1), requestId: "69e9d4be554a4"}
  |> data: Array(1)
    |> 0:
      |> author: "982133855eb11f3e0015bb9a0d638a6c"
        |> category: (2) ["前端", "工具"]
          |> content: "# 比快更快, Github Action + 云开发部署静态网站"
            |> cover: "https://636d-cms-1c7821-1259727701.tcb.qcloud.la/uploads/1588666722784.png"
              |> createTime: "2020-04-12T00:00:00.000Z"
                |> id: "aa9f906d5eb121640017a67926512667"
                  |> name: "比快更快, Github Action + 云开发部署静态网站"
                    |> tag: (5) ["CloudBase", "云开发", "静态网站", "Github Action", "博客"]
                      |> updateTime: "2020-05-05T02:00:00.000Z"
                        |> _id: "aa9f906d5eb121640017a67926512667"
```

常见问题

在扩展管理界面找不到 CMS 扩展？

CMS 扩展会开通静态托管资源，静态托管目前仅支持在按量计费环境下开通，切换为按量计费后就可以安装 CMS 扩展

打开 CMS 时提示 404？

CMS 的安装路径是静态托管默认域名/tcb-cms, 可以看下地址是否正确

登录时报 “Invaild uid” 错误怎么处理？

管理员和运营者用户名只支持 4-32 位字符（大小写英文字母、数字），可以在扩展管理界面重新修改用户名进行修复

登录时报 “SYS_ERR 登录异常” 错误怎么处理？

自定义登录密钥填写有误。在填写自定义密钥时，请直接按照输入框左侧的图标提示，从登录设置页面复制粘贴，不需要修改。另外，请确保复制的是当前云开发环境的自定义密钥。

登录时 INVALID_CUSTOM_LOGIN_TICKET 报错？

在填写自定义密钥时，直接按照提示从登录设置页面复制粘贴就可以了，不需要修改，另外需要确保复制的是当前云开发环境的自定义密钥

使用管理员账号提示没有管理权限怎么办？

管理员和运营者账号不能相同，建议设置为不同的账号，可以在扩展管理界面重新修改用户名进行修复

登录报错，提示 No credentials found on headers or cookies 怎么处理？

用户打开了云函数的 HTTP 触发的访问鉴权功能，开启鉴权后，客户端需要在登录的情况下才能触发云函数。CMS 负责登录的 auth 云函数需要关闭访问鉴权才可以使用。

其他

扩展配置信息

您可以配置以下参数：

- 管理员账号：CMS 内容管理系统的管理员账号。
- 管理员密码：CMS 内容管理系统的管理员密码。
- 运营者账号：CMS 内容管理系统的运营者账号。
- 运营者密码：CMS 内容管理系统的运营者密码。
- 自定义登录密钥：云开发自定义登录密钥。
- 部署路径：CMS 内容管理系统的部署路径。

计费

此扩展使用其他云开发或其他腾讯云服务，可能会产生相关费用：

- 云开发（[产品定价](#) 及 [使用明细](#)）。
- 静态托管（[产品定价](#) 及 [使用明细](#)）。

当您使用云开发扩展时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

部署详情

- [云函数](#)

- tcb-ext-cms-auth
该函数提供登录鉴权功能，用户在 CMS 管理界面通过通过用户名和密码来进行登录时，会通过 HTTP 来请求该函数；
- tcb-ext-cms-api
提供 API 接口功能，所有对内容的操作和管理都会经过此函数调用，内容操作会根据用户权限来进行数据库操作
- tcb-ext-cms-init
提供初始化应用功能，安装扩展后，会通过该函数来进行静态资源的部署和密码的生成和设置，修改账号密码或者部署路径等扩展参数都会再次执行该函数来进行更新
- **云数据库**
 - tcb-ext-cms-contents 集合
CMS 系统内容配置数据，CMS 所有的系统内容类型配置、字段配置等信息都存储在该集合内；
 - tcb-ext-cms-users 集合
CMS 系统用户数据，存储 CMS 的用户信息，包括管理员和运营者的账号信息，包括角色信息，用户，加密存储的密码等；
 - tcb-ext-cms-webhooks 集合
CMS 系统 webhook 集合，存储 CMS 系统的回调接口配置，CMS 系统数据的变更可以通过回调来进行同步。
- **云存储**
存储图片、文件等 CMS 系统上传的文件。
- **静态网站托管**
CMS 系统前端界面，基于 React 开发，通过 TCB JS SDK 访问 CMS 的函数、数据库和存储等资源。

权限授予

主账户

该扩展能力使用云开发自有资源即可完成，无需再授予其他权限。

子账户

如果想让子账户也能使用该扩展，需要为子账户授予如下权限才能使用：

- **策略:** QcloudAccessForTCBRole。
描述: 云开发 (TCB) 对云资源的访问权限。

短信验证码登录

最近更新时间：2020-07-03 16:30:04

短信验证码登录支持在客户端使用短信验证码进行登录，封装了短信验证码的生成、发送、校验和定时清理逻辑，帮助开发者进行鉴权。登录成功后，用户身份将转化为正式用户。

功能特性

开通后，开发者可以在客户端调用扩展，给用户手机发送验证码。用户可以使用验证码进行正式登录。

适用场景

用于客户端用户的正式登录。

前置要求

已经开通云开发。如未开通可参考 [开通环境](#) 文档操作。

扩展配置信息

您可以通过以下配置参数：

- 环境 ID：选择要部署的环境，在哪个环境下使用。
- 签名：短信服务（SMS）的签名。
- 模板：短信模板。
- 有效期：验证码有效期，单位是"分钟"。
- APPID 和 AppKey：短信服务（SMS）的信息。

计费

此扩展使用云开发或其他腾讯云服务，可能会产生相关费用：

- 短信 SMS（[产品定价](#) 及 [使用明细](#)）。
- 云数据库（[产品定价](#) 及 [使用明细](#)）。
- 云函数（[产品定价](#) 及 [使用明细](#)）。

当您使用云开发扩展时，您只需要为您使用的云资源付费；云开发与云上其他资源分开计费，您可以在 [费用中心](#) 查看具体信息。

启用的 API 和创建的资源

- **Type:** 短信 SMS
Description: 为开发者提供短信发送服务。
- **Type:** Cloud DataBase
Description: 存储验证码、过期时间和对应的手机号。
- **Type:** Cloud Function
Description: 生成、校验和定时清理验证码，下发用于正式登录的 ticket。

权限授予

主账户

角色名称	授权策略名	角色类型	描述
TCB_QcsRole	QcloudSMSFullAccess	服务角色	云开发（TCB）会对您的短信资源进行操作，方便您在扩展能力中使用该服务

子账户

如果想让子账户也能使用该扩展，需要为子账户授予如下权限才能使用：

- **策略:** QcloudAccessForTCBRole
描述: 云开发（TCB）对云资源的访问权限。
- **策略:** QcloudSMSFullAccess
描述: 短信资源全读写访问权限。

安装扩展

您可以通过 [云开发控制台](#)，来安装和管理扩展。

使用扩展

注意：

如果您在 web 网站使用该扩展，请先在 [云开发控制台](#) 将网站域名添加为当前环境的安全域名。

1. 安装扩展 SDK 到项目

方法一：通过包管理器引入

```
npm install --save @cloudbase/extension-sms
```

方法二：通过CDN引入

```
<script src="//unpkg.com/@cloudbase/extension-sms/built/index.umd.js"></script>
```

2. 调用扩展 SDK

调用参数

名称	类型	是否必须	说明
action	String	是	操作类型，支持 Send 和 Login
phone	String	是	电话号码
app	Tcb	是	tcb 实例
smsCode	String	否	短信验证码，action 为 Login 时需要传入
customDomain	String	否	HTTP 触发的自定义域名

注意：

绑定 HTTP 触发的自定义域名之前，请先设置您的域名的 CNAME 记录值为

`${envId}.service.tcloudbase.com`，`envId` 是您的环境 Id。CNAME 记录不存在时会导致域名绑定失败！

返回内容

两种 action 都没有返回值。

调用示例

客户端使用：

说明：
通过 CDN 使用。

```
<script src="//imgcache.qq.com/qcloud/tcbjs/${version}/tcb.js" ></script>
<script src="//unpkg.com/@cloudbase/extension-sms/built/index.umd.js" ></script>
<script>
const app = tcb.init({
env: '您的环境ID'
});

tcb.registerExtension(window.extSms);

demo();

async function demo() {
try {
let phone = ""; // 输入用户手机号
// 发送短信验证码
await tcb.invokeExtension(extSms.name, {
action: 'Send',
app,
phone
});
let smsCode = ""; // 用户填写验证码
// 验证码登录
await tcb.invokeExtension(extSms.name, {
action: 'Login',
app,
phone,
smsCode,
});
console.log('登录成功，目前是正式用户');
} catch (err) {
console.log(JSON.stringify(err, null, 4));
}
}
</script>
```

说明：
通过包管理器使用。


```
const tcb = require('tcb-js-sdk')
const extSms = require('@cloudbase/extension-sms')

const app = tcb.init({
  env: '您的环境ID'
});

tcb.registerExtension(extSms);

demo();

async function demo() {
  try {
    let phone = ""; // 输入用户手机号
    // 发送短信验证码
    await tcb.invokeExtension(extSms.name, {
      action: 'Send',
      app,
      phone
    });
    let smsCode = ""; // 用户填写验证码
    // 验证码登录
    await tcb.invokeExtension(extSms.name, {
      action: 'Login',
      app,
      phone,
      smsCode
    });
    console.log('登录成功，目前是正式用户');
  } catch (err) {
    console.log(JSON.stringify(err, null, 4));
  }
}
```

注意：

调整短信发送频率设置，请前往 [短信服务\(SMS\)控制台](#)。

MySQL

最近更新时间：2020-07-03 16:30:42

云开发支持在云函数中使用 MySQL。

前置要求

已经开通云开发。

功能特性

提供基于开源数据库 MySQL 专业打造的高性能分布式数据存储服务，让用户能够在云中更轻松地设置、操作和扩展关系数据库。

适用场景

- 云开发的数据库满足不了业务的需求，需要使用到 MySQL。
- 已有的业务使用了 MySQL，业务迁移到云开发中，希望继续使用 MySQL。

安装程序

您可以通过 [云开发控制台](#)，来安装和管理该能力。

注意：

- 微信小程序开发者请使用【其他登录方式】>【微信公众号登录】登录，选择关联的小程序账户登录。
- QQ 小程序开发者可直接通过 QQ 小程序开发者 IDE【云开发】登录，也可以通过关联的腾讯云账户登录。

使用程序

如果您原有的业务代码中使用了 MySQL 连接池，在云函数中可能就不需要连接池了，每个请求都会分发到不同的云函数实例中，通常情况下，一个云函数有一个 MySQL 连接即可。

根据您的实际情况，推荐使用 [mysql](#) 或 [serverless-mysql](#)。

- 默认情况下，serverless-mysql 是 mysql 的一个封装，添加连接管理相关的功能，与 MySQL Server 只有一个连接；如果在云函数中需要使用连接池，推荐使用 mysql。
- 使用 serverless-mysql 的时候，请注意 MySQL 的 max_user_connections 和 max_connections 的配置。

调用示例

调用示例如下：

```
'use strict';
const mysql = require('serverless-mysql')({
  config: {
    host: process.env.HOST,
    port: process.env.PORT,
    database: process.env.DATABASE,
    // 需要填写真实的用户名与密码
    user: 'xxx',
    password: 'xxx'
  }
})
exports.main = async (event, context, callback) => {
  let res
  try {
    res = await mysql.query('SELECT * FROM mysql_test')
  } catch (e) {
    console.error(e)
  }
  return {
    res,
    code: 200
  }
}
```

其它

云函数的运行机制

云函数的运行机制，请参考 [深入理解云函数](#)。

私有网络

在云函数中，开发者如果需要访问腾讯云的云数据库等资源，推荐使用私有网络来确保数据及连接安全。关于私有网络、如何建立私有网络以及云函数加入私有网络的详细信息，请参考 [为云数据库 MySQL 创建私有网络](#) 中的相关章节。

云函数并发数与 MySQL 连接数

在 [云开发控制台-环境总览](#) 中可以看到当前环境所允许的云函数并发数的最大值，最大并发数也是云函数的最大实例数。

关于 MySQL 连接，推荐阅读 MySQL 官方博客的博文 [MySQL Connection Handling and Scaling](#)。

MySQL 连接数的相关参数配置，可以在 [云数据库 TencentDB](#) 控制台，【MySQL】>【数据库管理】>【参数设置】中配置。

在云函数中使用 MySQL，每个云函数实例与 MySQL Server 都会有连接，那么此云函数与 MySQL 的最大连接数是，单个实例的最大连接数 * 实际运行的最大并发数；在配置 MySQL 的 max_connections 的时候，此参数应该大于使用此数据库的所有云函数的最大连接数之和。

因此，在云函数中使用 MySQL，建议您将使用到同一个数据库（不仅仅是同一张表）的所有读写 MySQL 的代码集中到一个云函数中，这样做有两个好处：

- 云函数出现冷启动的概率比较低。
- MySQL 的最大连接数较小。

Redis

最近更新时间：2020-07-15 19:42:17

云开发支持在云函数内使用 Redis。

前置要求

已经开通云开发。

功能特性

提供兼容 Redis 协议的缓存数据库，具备高可用、高可靠、高弹性等特征。

适用场景

- 云开发的数据库满足不了业务的需求，需要使用到 Redis。
- 已有的业务使用了 Redis，业务迁移到云开发中，希望继续使用 Redis。

安装程序

您可以通过 [云开发控制台](#)，来安装和管理该能力。

注意：

- 微信小程序开发者请使用【其他登录方式】>【微信公众号登录】登录，选择关联的小程序账户登录。
- QQ 小程序开发者可直接通过 QQ 小程序开发者 IDE【云开发】登录，也可以通过关联的腾讯云账户登录。

使用程序

根据实际情况，推荐使用 [redis](#) 或 [ioredis](#)。

调用示例

调用示例如下：

```
'use strict';

const redis = require('redis')

let client = redis.createClient({
  host: process.env.HOST,
  port: process.env.PORT,
  // 需要填写真实的密码
  password: 'xxx'
})

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

  let res = await new Promise((resolve, reject) => {
    client.get('test', function (err, reply) {
      if (err) {
        resolve({
          err
        })
      }
      resolve({
        data: reply.toString()
      })
    })
  })

  return {
    res
  }
}
```

其它

云函数的运行机制

云函数的运行机制，请参考 [深入理解云函数](#)。

私有网络

在云函数中，开发者如果需要访问腾讯云的 Redis 等资源，推荐使用私有网络来确保数据及连接安全。关于私有网络、如何建立私有网络以及云函数加入私有网络的详细信息，请参考 [云数据库 Redis 配置网络](#) 中的相关章节。

云函数并发数与 Redis 连接数

在 [云开发控制台-环境总览](#) 中可以看到当前环境所允许的云函数并发数的最大值，最大并发数也是云函数的最大实例数。

关于 Redis 连接，推荐阅读 Redis 官方博客的博文 [Redis Clients Handling](#)。

Redis 连接数的配置参数 `maxclients`，默认是10000，如果需要修改请提交 [工单申请](#)。

云函数中使用 Redis 时，每个云函数实例与 Redis Server 都会有连接，那么此云函数与 Redis 的最大连接数是，单个实例的最大连接数 * 实际运行的最大并发数；在配置 Redis 的 `maxclients` 的时候，此参数应该大于使用此数据库的所有云函数的最大连接数之和。

因此，在云函数中使用 Redis，建议您将使用到同一个 Redis 实例的所有读写 Redis 的代码集中到一个云函数中，这样做有两个好处：

- 云函数出现冷启动的概率比较低。
- Redis 的最大连接数较小，减少连接数多内存的占用。

其他 音视频

最近更新时间：2020-06-29 16:30:32

本增值服务已整合腾讯云的 [实时音视频](#) 能力，通过云开发的云函数和数据库能力，简化配置拉取和房间管理等功能。

体验增值服务



DEMO 源码

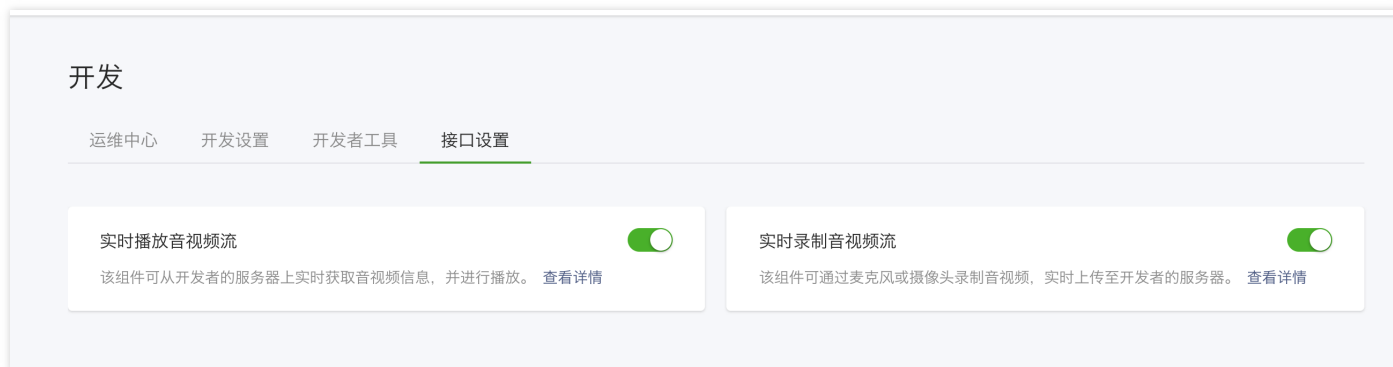
本章的案例 [代码 tcb-demo-video](#)。包含了小程序前端代码（`client` 目录下）和云函数代码（`cloud` 目录下），需要在微信开发者工具中打开整个项目。下面将详细介绍 DEMO 的接入流程。

DEMO 接入流程

1. **登录** 小程序的管理后台，选择【设置】>【基本设置】>【服务类目】中添加允许视频直播类的管理。



2. 在小程序管理后台选择【开发】>【接口设置】中，将**实时播放音视频流**和**实时录制音视频流**打开。



3. 在小程序管理后台选择【开发】>【开发设置】中，配置以下域名：

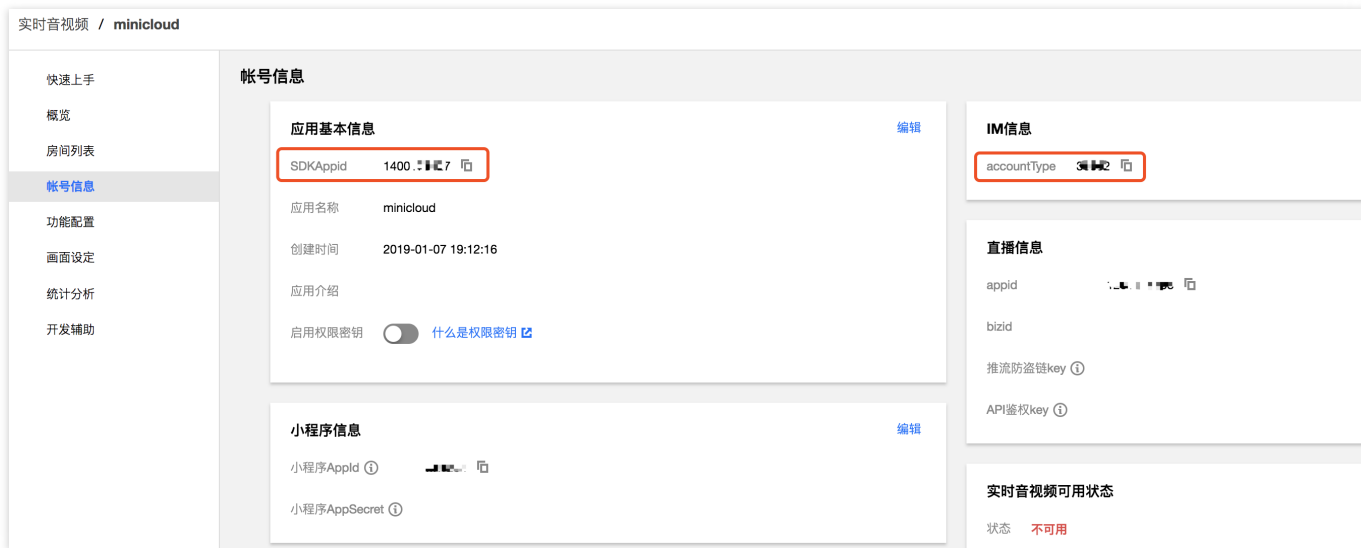
- <https://official.opensso.tencent-cloud.com>
- <https://yun.tim.qq.com>
- <https://room.qcloud.com>
- <https://webim.tim.qq.com>

服务器域名

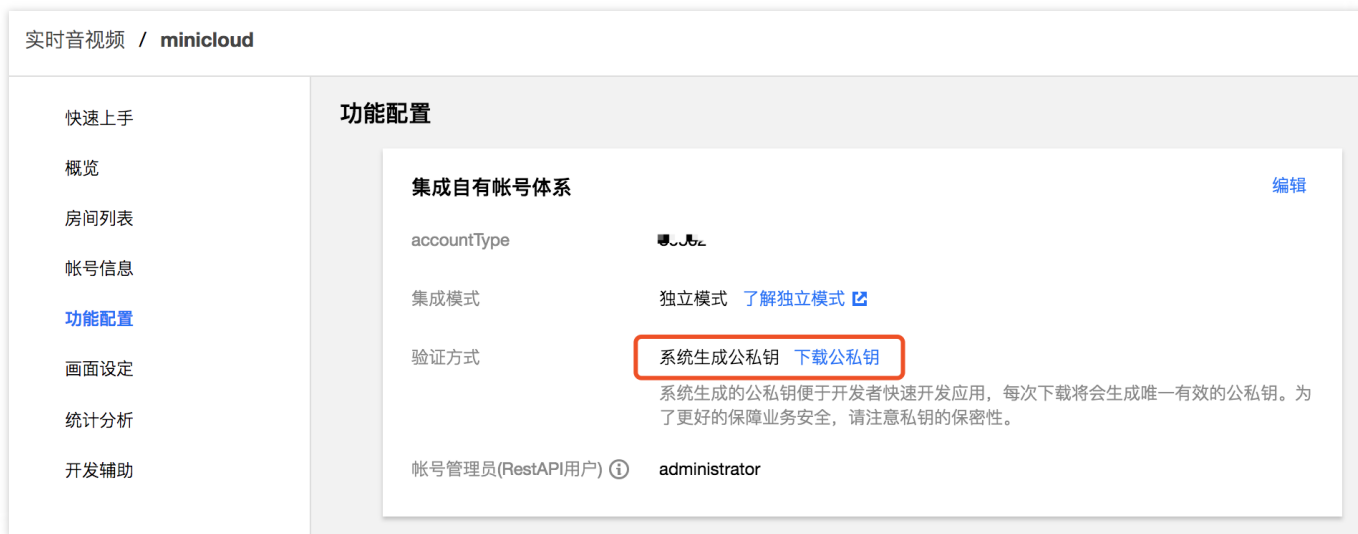
服务器配置	说明	操作
request合法域名	https://official.opensso.tencent-cloud.com https://yun.tim.qq.com https://room.qcloud.com https://webim.tim.qq.com	
socket合法域名	一个月内可申请5次修改 本月还可修改4次	修改
uploadFile合法域名		
downloadFile合法域名		

4. 通过 [微信公众号](#) 登录小程序对应的腾讯云账号（需要小程序管理员权限），然后在腾讯云官网单击 [实时音视频](#)，开通服务并购买体验包，进入控制台，并获取以下配置信息：

- SDKAppid 和 accountType



- 下载公私钥 (private_key) 文件

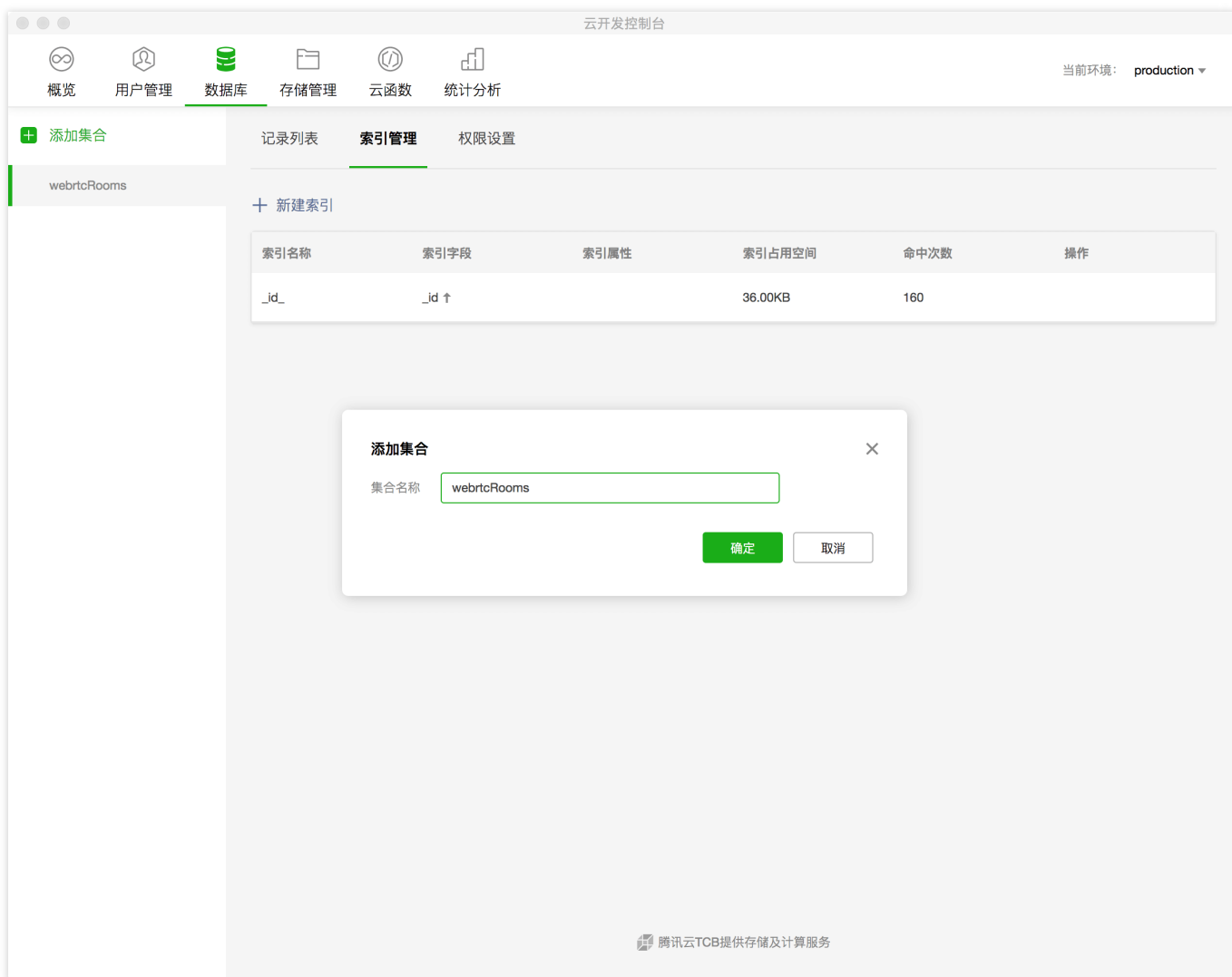


5. 请使用微信开发者工具打开 DEMO 源码，在根目录下的 project.config.json 文件，填写您的小程序 appid。

6. 请按以下步骤完成操作。

- 首先在云函数目录 cloud/functions 的函数 webtrcroom-enter-room 中，将 private_key 文件放到 config 目录下。
- 其次在每个云函数目录的 config 目录下，参照 example.js 文件，新建 index.js 文件，并配置好 sdkAppID 和 accountType。
- 再而上传部署所有的云函数。

- 最后，在云开发面板的数据库栏目中，创建 `webrtcRooms` 集合。



7. 打开微信开发者工具直接预览即可。

源码介绍

WebRTC 能力

WebRTC 的能力的体验，主要是围绕源码中 `client/pages/webrtc-room` 里面的 `join-room` 和 `room` 两个目录，一个是手动输入房间号进入房间，另一个是视频房间。 `cloud/functions/webrtcroom-enter-room/sign.js` 函数主要用于填写实时视频配置后，进行信息加密，将已加密的信息传递出来，小程序端可使用签名信息正常使用 WebRTC 视频通话能力。

其中 `cloud/functions/webrtcroom-enter-room/libs/sign.js` 是官方提供的 [签名逻辑文件](#)，而 `cloud/functions/webrtcroom-enter-room/sign.js`，调用其中的方法，然后将 `privateMapKey`、`userSig` 提供到小程序，小程序端才能正常将直播流唤起。

房间管理

房间管理主要通过云开发的云函数和数据库实现。 `webrtcRooms` 集合的数据格式如下：

```
"_id":XDMNvt7E7L4wMo3M
"createTime":Mon Jan 07 2019 16:28:46 GMT+0800 (CST)
"creator":al_hhF0-PfmY
"members":["al_hhF0-PfmY"]
"privateMapKey":
"roomId":111
"roomName":
```

数据包括有房间创建者和观众的 `openid` 房间 ID、房间名、房间创建时间，还有房间的权限位 `privateMapKey`。

云函数分别有**创建房间**（`webrtc-create-room`）、**进入房间**（`webrtc-enter-room`）、**退出房间**（`webrtc-quit-room`）、**获取房间信息**（`webrtc-get-room-info`）、**获取房间列表**（`webrtc-get-room-list`）云函数。

- `webrtc-create-room` 函数主要用于创建房间，这里用到了数据库的读写，先要判断房间是否存在，如果不存在，则创建房间。

注意：

该函数有一处逻辑，此处是通过循环的方式，去检查房间 ID，以防生成了重复的 ID。

```
// 循环检查数据，避免 generateRoomID 生成重复的roomId
while (await isRoomExist(roomInfo.roomID)) {
  roomInfo.roomID = generateRoomID()
}
```

- `webrtc-enter-room` 函数主要用于进入房间，如果房间存在，则将用户的 `openid` 写入房间观众字段，如果房间不存在，则调用 `webrtc-create-room` 进行房间创建。

-
- `webrtc-quit-room` 函数主要用于退出房间，如果房间还有观众，则将退出者的 `openid` 清除，如果没有观众了，则把房间数据清理掉。
 - `webrtc-get-room-info` 函数主要用于获取房间数据。
 - `webrtc-get-room-list` 函数主要用于获取房间列表数据。

智能图像

最近更新时间：2020-03-16 16:29:37

增值服务 AI 智能图像能力是借助了腾讯云 [人脸识别](#)、[人脸核身](#) 和 [人脸融合](#) 功能，通过云开发的云函数和存储，简化素材存储、配置拉取和服务调用 [image-node-sdk](#) 等功能。

体验增值服务



DEMO 源码

本章的案例代码，请参考 [代码 tcb-demo-ai](#)。

DEMO 接入流程

1. 请使用 [微信开发者工具](#) 打开 DEMO 源码，在根目录下的 `project.config.json` 文件，填写您的小程序 `appid`。
2. 通过此 [微信小程序第三方绑定流程](#) 登录小程序对应的腾讯云账号（需要小程序管理员权限），然后在 [云 API 密钥](#) 里获取 `SecretId` 和 `SecretKey`。
3. 在腾讯云的 [智能图像控制台](#)，开通相应的服务：
4. 本案例，前端页面（`client/pages/`）和云函数（`cloud/functions`）一一对应，如下：

功能	前端页面	云函数
人脸融合	face-fusion	FaceFusion

功能	前端页面	云函数
活体人脸检测	liveness-recognition	GetLiveCode 和 LivenessRecognition
身份信息认证	idcard-verification	IdCardVerification
人脸检测与分析	face-detect	DetectFace
通用印刷体识别	ocr-general	GeneralBasicOCR
身份证识别	ocr-idcard	IDCardOCR

注意：

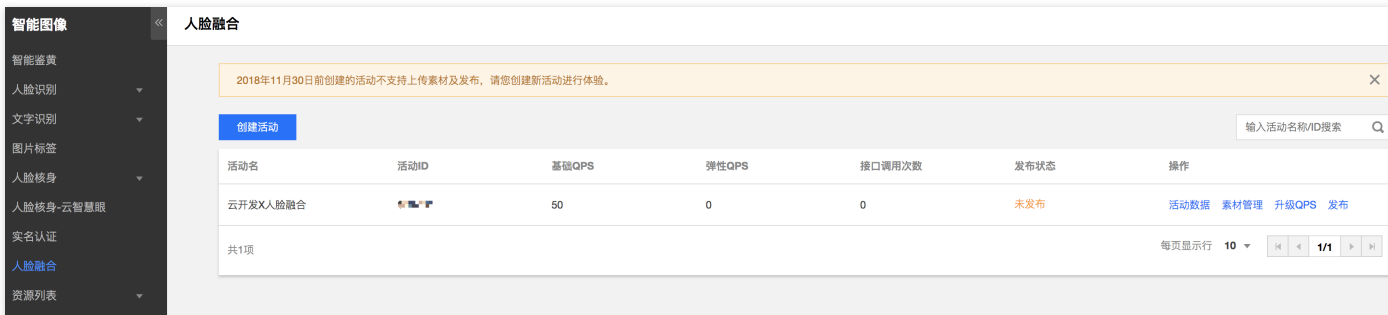
如果需要体验某个功能，需要在对应的云函数里参照 `config/example.js` 新建 `config/index.js`，并填入上面拿到的 `SecretID` 和 `SecretKey`，然后创建并部署云函数。

5. 如果是体验以下的功能，还需要做额外的准备工作。

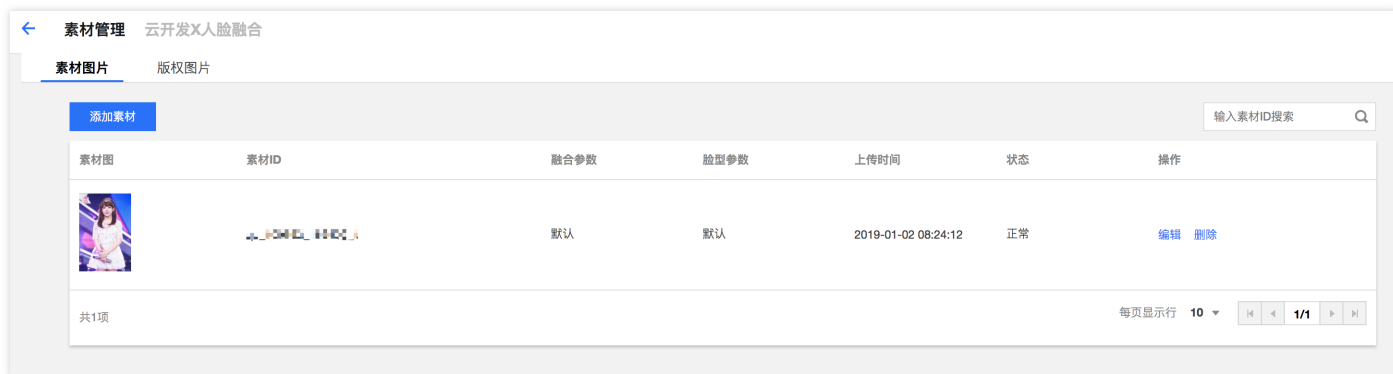
人脸融合

如果您想体验人脸融合功能，开通服务后，需要【创建活动】并【添加素材】，要获得以下配置：

- ProjectId (活动 ID)，您可在 [人脸融合控制台](#) > 【活动管理面板】中查看。



- ModelId (素材 ID), 您可在 [人脸融合控制台](#) > **【活动素材管理面板】** 中查看。



人脸识别

人脸识别已经升级至全新版本, 请访问新版 [人脸识别新控制台](#)。

注意：

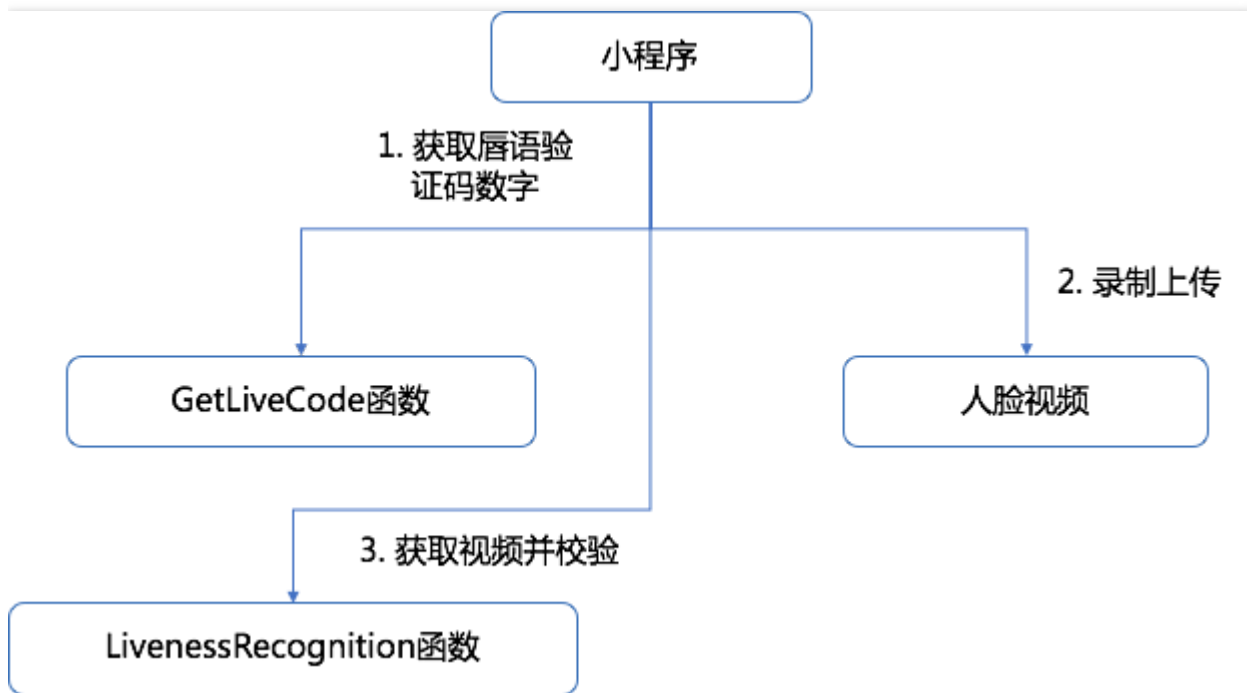
如果已登录旧版控制台, 您会在页面顶部看到相关提示, 单击使用可跳转全新版本。



源码介绍

活体人脸核身

本案例实现了该服务的一些基础能力。整个逻辑流程如下：



其中云函数 `LivenessRecognition` 的大体逻辑如下：

```
const {
  IdCard,
  Name,
  VideoFileID,
  LivenessType = 'SILENT',
  ValidateData
} = event

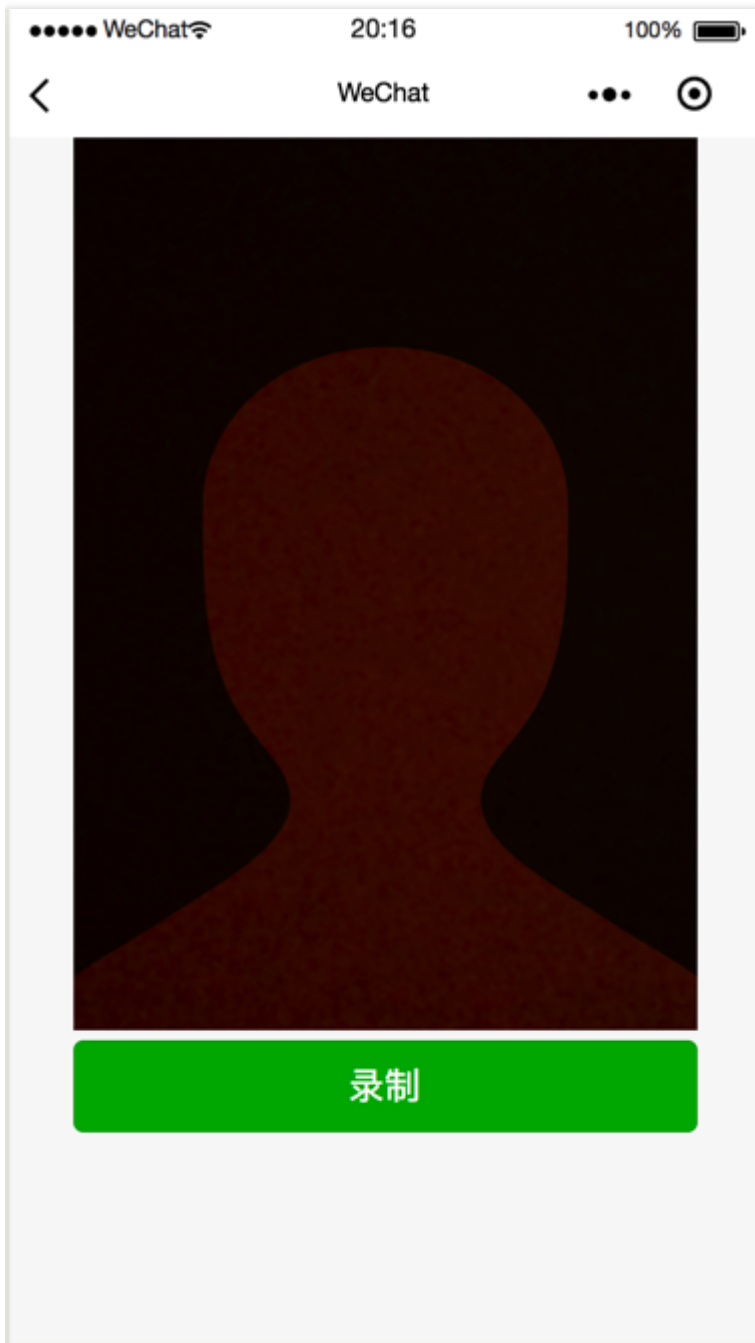
try {
  // 获取视频内容的字符串
  let fileContent = await tcbService.utils.getContent({
    fileID: VideoFileID
  })

  if (!fileContent) {
    return { code: 10002, message: 'fileContent is empty' }
  }

  const result = await tcbService.callService({
    service: 'ai',
    action: 'LivenessRecognition',
    data: {
      IdCard,
```

```
Name,  
VideoBase64: fileContent.toString('base64'), // 视频内容转 base64  
LivenessType,  
ValidateData  
},  
options: {  
  secretID: SecretID,  
  secretKey: SecretKey // 调用其它腾讯云账号的 AI 资源  
}  
})  
  
return result  
}  
catch (e) {  
return { code: 10001, message: e.message }  
}
```

在小程序端，需要有类似的遮罩，才能提高视频通过的概率。



在小程序端使用 `<cover-view>` 和 `<cover-image>` 可以让图片等元素，覆盖在 `<camera>`，`<video>` 等原生的组件上面，具体操作如下：

```
<camera
device-position="front"
flash="off"
binderror="error"
>
<cover-view class="camera-cover">
```

```
<cover-image  
class="camera-image"  
src="image path"  
>  
</cover-image>  
</cover-view>  
<cover-view  
class="number"  
wx-if="{{isRecording}}"  
>  
请念数字 : {{number}}  
</cover-view>  
</camera>
```