

# 语音合成 实践教学



腾讯云

## 【 版权声明 】

©2013–2025 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 4009100100或 95716。

## 文档目录

### 实践教程

结合 AI 语音合成和云开发快速上线一款实用工具小程序

# 实践教程

## 结合 AI 语音合成和云开发快速上线一款实用工具小程序

最近更新时间：2023-12-07 16:01:23

### 说明：

本文来自 [AI 专题用户实践征文](#)，仅供学习和参考。

## 名词介绍

语音合成（Text To Speech, TTS）满足将文本转化成拟人化语音的需求，打通人机交互闭环。提供多场景、多语言的音色选择，支持 SSML 标记语言，支持自定义音量、语速等参数，让发音更专业、更符合场景需求。语音合成广泛适用于智能客服、有声阅读、新闻播报、人机交互等业务场景，提升人机交互体验，提高语音类应用构建效率。

云开发（Tencent CloudBase, TCB）是腾讯云提供的云原生一体化开发环境和工具平台，为开发者提供高可用、自动弹性扩缩的后端云服务，包含计算、存储、托管等 serverless 化能力，可用于云端一体化开发多终端应用（小程序、公众号、Web 应用、Flutter 客户端等），帮助开发者统一构建和管理后端服务和云资源，避免了应用开发过程中繁琐的服务器搭建及 [运维](#)，开发者可以专注于业务逻辑的实现，开发门槛更低，效率更高。

## 开发背景

对于媒体从业人员来说，一款简单、易用的文本转语音软件是非常必要的，并且要随时随地无需下载和注册都能使用，所以结合小程序和云开发是最合适的。

## 开发工具

云开发 [CloudBase](#)

## 代码开发

### 第一步：用户输入框

```
// WXML
<textarea class="" type="text" value="{{txt}}" bindinput="setTxt" auto-
height="true"></textarea>
// JS
data: {
  txt: ""
```

```
    }  
    setTxt(e) {  
      this.setData({txt:e.detail.value})  
    },  
  },  
}
```

## 第二步：用户提交按钮

```
// WXML  
<view class="" bindtap="submit"></view>  
//js  
submit() {  
  wx.cloud.callFunction({  
    name: "submitTxt",  
    data: {  
      txt: this.data.txt  
    }  
  }).then(res=> {  
  
  })  
}
```

## 第三步：新建云函数

新建云函数，名字为上一步 submit() 中调用的 submitTxt。

```
const tencentcloud = require("tencentcloud-sdk-nodejs");  
  
const TtsClient = tencentcloud.tts.v20190823.Client;  
  
// 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey, 此处还请注意密钥对  
// 的保密  
// 密钥可前往https://console.cloud.tencent.com/cam/capi网站进行获取  
// 云开发中的云函数相当于服务端，可以明文填写您的密钥  
const clientConfig = {  
  credential: {  
    secretId: "XXXXXXXXXXXXXXXXXXXXXXXX",  
    secretKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",  
  },  
  region: "", // 地域，可以为空  
  profile: {  
    httpProfile: {  
      endpoint: "tts.tencentcloudapi.com",  
    },  
  },  
}
```

```
    },
  },
};
exports.main = async (event, context) => {
  var txt = event.txt
  const client = new TtsClient(clientConfig);
  // 此处仅提供必填参数，更多参数选择请参考官方文档。
  const params = {
    "Text": txt,
    "ModelType": 1, // 模型类型，1-默认模型。
    "VoiceType": 10510000 // 10510000-智逍遥，旁对白阅读风格男声
  };
  client.CreateTtsTask(params).then(
    (data) => {
      console.log(data);
      // {
      //   RequestId: 'dc7708bc-1b8c-412e-9033-...',
      //   Data: { TaskId: 'gz-2e20190c-8f20-4941-
ab7e-...' }
      // }
      var taskId = data.Data.TaskId;
      return taskId
    },
    (err) => {
      console.error("error", err);
      return "FAIL"
    }
  );
}
```

因为长文本的语音合成是需要一定时间的，所以回调是一个任务 ID，需要再根据这个 ID 查询任务状态，如果已完成，会有返回一个 URL。开发者可以根据需要，考虑是否把提交和查询合并在一起。这里采用的是分离的模式。

#### 📌 说明：

此处依赖可以不用本地安装，直接右键选择云端安装依赖即可。在云函数文件中 package.json 加入。

```
"dependencies": {
  "tencentcloud-sdk-nodejs": "^4.0.348", // 加入此项
  "wx-server-sdk": "~2.6.1"
}
```

## 第四步：查询任务状态

新建一个与上一步的环境相同的云函数。

```
const client = new TtsClient(clientConfig);
const params = {
  "TaskId": TaskId // 上一步得到的任务ID
};
client.DescribeTtsTaskStatus(params).then(
  (data) => {
    console.log(data);
    // {
    //   RequestId: 'effa415f-94b1-46c1-a5fa-.....',
    //   Data: {
    //     TaskId: 'gz-2e20190c-8f20-4941-ab7e-.....',
    //     Status: 2, //任务状态码, 0: 任务等待, 1: 任务执行中, 2: 任务成功, 3: 任务失败。
    //     StatusStr: 'success',
    //     ResultUrl: 'https://tts-async-audio-.....',
    //     ErrorMsg: ''
    //   }
    // }
    if(data.Data.Status === 2) return data.Data.ResultUrl
    return data.Data.StatusStr
  },
  (err) => {
    console.error("error", err);
    return "FAIL"
  }
);
```

前端得到第三步返回的任务 ID 后, 可提供该 ID 继续调用第四步的云函数, 获取任务结果。如果成功, 即可返回一段 MP3 的播放地址。

## 第五步：前端加入播放组件

```
<video
  id="myVideo"
  src={{ResultUrl}}
></video>
```

## 第六步：前端下载功能

```
wx.downloadFile({
  url: url, //之前获取的地址
  success (res) {
    // 只要服务器有响应数据，就会把响应内容写入文件并进入 success 回调，业务需要自行
    判断是否下载到了想要的内容
    if (res.statusCode === 200) {
      // TODO
    }
  }
})
```

至此，一款简单的语音合成小程序，已经完成了核心功能。您可以根据需要，美化前端 UI，并增加用户管理功能，结合云开发的数据库可以很好的管理用户的数据。也可以增加用户使用的逻辑，例如新用户可以直接使用 X 次，通过激励用户观看视频或者分享，可以获取使用次数等。