



链滴

【微信开发】上传用户语音 并用 ffmpeg 转码并分享

作者: [shirenuang](#)

原文链接: <https://ld246.com/article/1565839204374>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

版权声明：本文为博主原创文章，遵循CC 4.0 by-sa

版权协议，转载请附上原文出处链接和本声明。

本文链接: <http://blog.shiyi.online/articles/2019/08/15/1565839188597.html>

需求

临近圣诞节，运营组想了一个活动来拉新，活动的大概内容是这样的：

用户访问活动首页，点击【我想说】然后调用微信的 JSDK 来录音，录完音之后上传到微信的服务器，后 前端根据声音生成一张圣诞树形状的声波图，用户点击分享生成一张海报，其他用户扫描二维码可收听你分享的语音；然后别人参与活动的时候需要先关注公众号；

这个需求很简单吧，逻辑清晰的不行...好 开搞

查找微信开发者文档

[微信开发者JSDK文档](#)

开始录音接口

```
wx.startRecord();
```

停止录音接口

```
wx.stopRecord({
  success: function (res) {
    var localId = res.localId;
  }
});
```

监听录音自动停止接口

```
wx.onVoiceRecordEnd({
  // 录音时间超过一分钟没有停止的时候会执行 complete 回调
  complete: function (res) {
    var localId = res.localId;
  }
});
```

播放语音接口

```
wx.playVoice({
  localId: '' // 需要播放的音频的本地ID，由stopRecord接口获得
});
```

监听语音播放完毕接口

```
wx.onVoicePlayEnd({
  success: function (res) {
    var localId = res.localId; // 返回音频的本地ID
  }
});
```

上传语音接口

```
wx.uploadVoice({
  localId: '', // 需要上传的音频的本地ID, 由stopRecord接口获得
  isShowProgressTips: 1, // 默认为1, 显示进度提示
  success: function (res) {
    var serverId = res.serverId; // 返回音频的服务器端ID
  }
});
```

备注: 上传语音有效期3天, 可用微信多媒体接口下载语音到自己的服务器, 此处获得的 serverId 即 media_id, 参考文档. 目前多媒体文件下载接口的频率限制为10000次/天, 如需要调高频率, 请登录微信公众平台, 在开发 - 接口权限的列表中, 申请提高临时上限。

<https://mp.weixin.qq.com/cgi-bin/debug>

下载语音接口

```
wx.downloadVoice({
  serverId: '', // 需要下载的音频的服务器端ID, 由uploadVoice接口获得
  isShowProgressTips: 1, // 默认为1, 显示进度提示
  success: function (res) {
    var localId = res.localId; // 返回音频的本地ID
  }
});
```

<https://mp.weixin.qq.com/cgi-bin/debug>

看完这些文档, 很棒! 简单! 微信该提供的都提供了

那梳理一下流程应该是这样的

- 1、调用wx.startRecord();开始录音
- 2、调用wx.stopRecord();结束录音, 可以得到 localId(注意这个并不是录音存放在手机的路径)
- 3、调用wx.uploadVoice()上传录音,需要传入2拿到的localId; 微信会返回一个 serverId; (这个serverId 就是微信的素材文件的mediaId; 通过这个mediaId 可以从微信服务器下载文件的)
- 4、ok! 上传成功了 分享的时候把这个serverId分享出去!
- 5、被分享人拿到这个serverId之后调用wx.downloadVoice();这个接口需要4的serverId, 然后这个音素材就下载到了用户手机, 并且返回了一个 localId(注意这个并不是录音存放在手机的路径)
- 6、用户点击[播放语音] 则调用wx.playVoice()来播放语音, 需要5返回的localId;

完美! 一个流程下来虽然步骤多了一点, 但是并不是不可行啊,既然可行那就撸起袖子干吧!

但是仔细一看 就发现有很多问题了

方案不足地方

1、

上传语音接口

```
wx.uploadVoice({
  localId: '', // 需要上传的音频的本地ID, 由stopRecord接口获得
  isShowProgressTips: 1, // 默认为1, 显示进度提示
  success: function (res) {
    var serverId = res.serverId; // 返回音频的服务器端ID
  }
});
```

备注：上传语音有效期3天，可用微信多媒体接口下载语音到自己的服务器，此处获得的 serverId 即 media_id，参考文档。目前多媒体文件下载接口的频率限制为10000次/天，如需要调高频率，请登录微信公众平台，在开发 - 接口权限的列表中，申请提高临时上限。

通过这种方式上传的语音 属于 **【临时素材】**，只有3天的有效期！

那么就直接将上面想到的方案给否定掉了，因为你分享出去之后过了3天别人就听不到了！

2、上面的方案有个弊端就是，用户收听别人的语音都需要下载，每次都要下载是一件让用户很烦的一件事；

3、通过微信 **【下载临时素材】** 接口下载语音素材的格式是 amr；如果通过微信的 wx.playVoice() 播可能没什么问题，但是这个方案，但是这个方案已经不行了，因为你不能通过 wx.downloadVoice() 下载了（3天有效期一过就不行）；

提出新的方案

既然上面的方案不行，不足也很清楚，那么我们换种实现

主要思路：将文件下载到我们自己的服务器上，将 amr 格式转码成 mp3 等等 <audio 能够播放的格式，然后上传到阿里云 OSS 存储，开放外部访问，将 OSS 链接分享出去，直接播放这个链接就行了

详细步骤：

- 1、调用 wx.startRecord(); 开始录音
- 2、调用 wx.stopRecord(); 结束录音，可以得到 localId（注意这个并不是录音存放在手机的路径）
- 3、调用 wx.uploadVoice() 上传录音，需要传入 2 拿到的 localId；微信会返回一个 serverId；（这个 serverId 就是微信的素材文件的 mediaId；通过这个 mediaId 可以从微信服务器下载文件的）

（前面3个步骤不变）

4、前端上传完了之后，调用我的 **【下载临时素材到我们自己服务器并且转码成 MP3 并且上传到阿里云 OSS 并且返回访问链接】** 的接口

获取临时素材

公众号可以使用本接口获取临时素材（即下载临时的多媒体文件）。请注意，视频文件不支持https下载，调用该接口需http协议。

本接口即为原“下载多媒体文件”接口。

接口调用请求说明

```
http请求方式: GET,https调用
https://api.weixin.qq.com/cgi-bin/media/get?access_token=ACCESS_TOKEN&media_id=MEDIA_ID
请求示例（示例为通过curl命令获取多媒体文件）
curl -I -G "https://api.weixin.qq.com/cgi-bin/media/get?access_token=ACCESS_TOKEN&media_id=MEDIA_ID"
```

参数说明

参数	是否必须	说明
access_token	是	调用接口凭证
media_id	是	媒体文件ID

返回说明

https://mp.weixin.qq.com/debug/wxadon/dev/index.action?lang=zh_CN

需要3获取到的serverId(就是这里的media_id)作为参数;

5、前端拿到了4中的 语音链接 然后分享出去带上这个链接(链接太长可以自己做个对应)

6、被分享着点击播放 直接使用 元素 <audio 来播放这个链接

7、PS:千万不要在 微信开发者工具中 调试[上传语音]的接口; 它不支持的!

好! 流程这样就没有错了, 但是正在做的时候猜的坑也不少

最主要的就是 4 中 **【下载临时素材到我们自己服务器并且转码成MP3并且上传到阿里云OSS并且回访问链接】**

这个接口踩得坑真的不少,最主要的就是转码; 网上文章一艘没有几篇有价值的东西;

主要就是 调用 java 这个jar包去转码, 其实最终这个java 调用了一个叫做 ffmpeg 转码程序来转码的 然后这个程序不同环境还不一样的

我的是Mac 然后服务器又需要Linux版本的;

啰嗦我就不啰嗦了, 推荐一个解决方案,我就是用的这种

Amr转Mp3

这里我提供一下 Mac版的 ffmpeg下载 [ffmpeg-mac](#)

特别注意 下载下来的文件是 ffmpeg-mac

后缀 -mac是必须的, 因为这个作者是按照这个路径找的

```
public class changToMp3 {
    public static void main(String[] args) throws Exception {
```

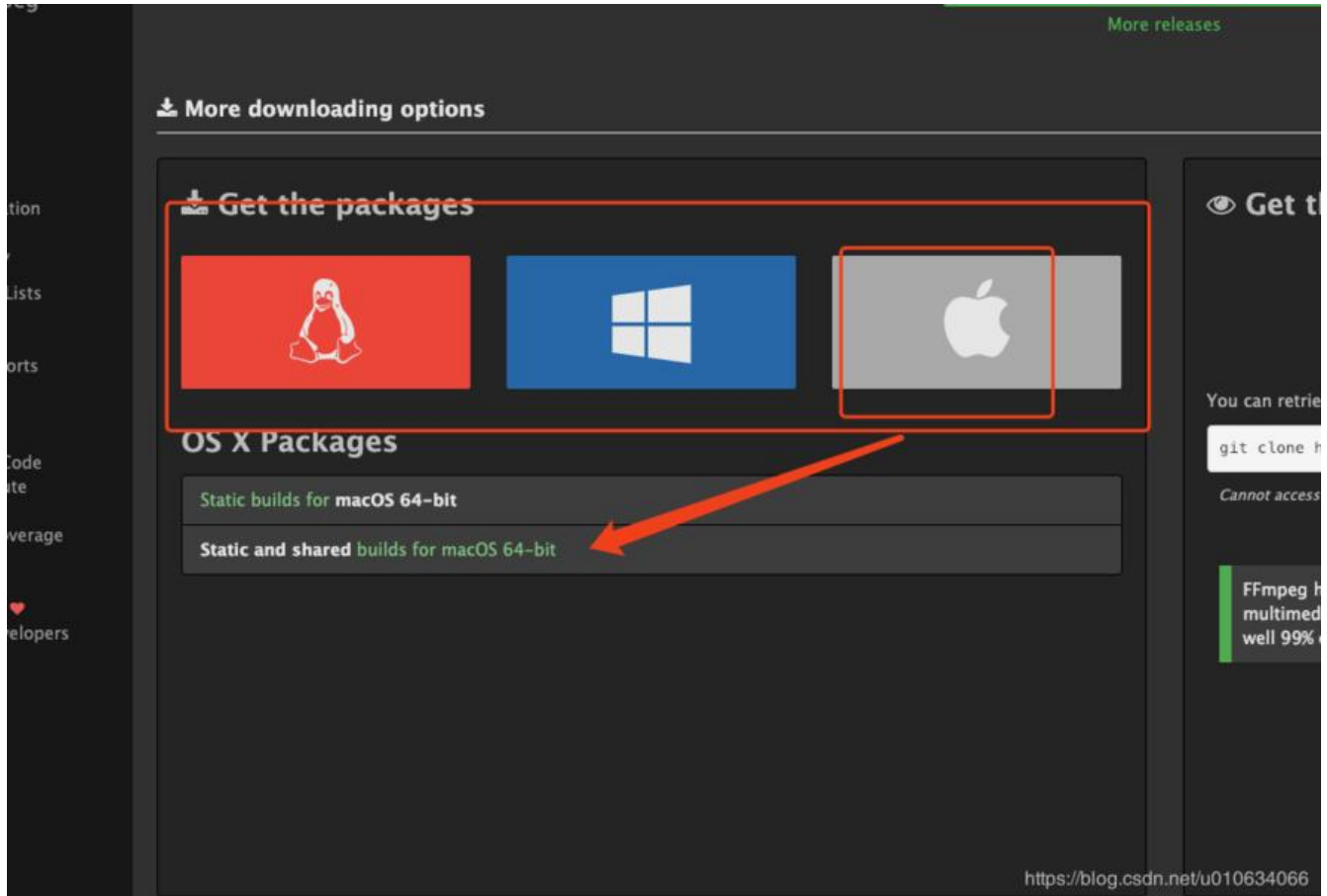
```

String path1 = "/Users/XXX/Desktop/voice/1111.amr";
String path2 = "/Users/XXX/Desktop/voice/1111.mp3";
System.setProperty("ffmpeg.home", "/Users/XXX/Desktop/voice/");
it.sauronsoftware.java.AudioUtils.amrToMp3(path1, path2);
}
}

```

再上传一个Linux版本的提供下载:[ffmpeg for linux](#)

如果还有windows版本的 或者上面我发的不支持的; 可以去官网自己下载 地址:[FFMPEG下载](#)



下载完解压之后 找到对应程序 放到某个路径 然后 代码设置的路径跟程序路径一直即可!

贴一下关键代码:

```

@Override
public RpcResult<PutResult> amr2mp3Upload2OSS(String gzhAppid, String mediaId,String
ossDirKey) {
    RpcResult result = new RpcResult();
    PutResult p = new PutResult();
    String filepath ;String name ;
    try {
        //下载临时文件到 服务器本地
        File file = WxMpConfiguration.getMpServices().get(gzhAppid).getMaterialService().me
iaDownload(mediaId);
        if(file == null){
            result.setFail("-1", "操作失败,mediaId 不存在或已过期");
            return result;
        }
    }
}

```

```

    }
    filepath = file.getPath();
    name = file.getName();
    name = name.replace(".amr", ".mp3");
} catch (WxErrorException e) {
    result.setFail(e.getError().getErrorCode() + "", e.getError().getErrMsg());
    return result;
}
//设置 使用转码软件的 程序地址 (要在对应的地址有这个执行脚本)
System.setProperty("ffmpeg.home", ffmpeg);
String targetPath = mp3_path+name;
//转码之后存到本地服务器
it.sauronsoftware.jave.AudioUtils.amrToMp3(filepath, targetPath);
PutResult putResult = mossClient.putObject(ossDirKey+ UUID.randomUUID(), new File(targetPath));
BeanUtils.copyProperties(putResult, p);
result.setResult(p);
result.setSuccess();
return result;
}

```

PS: 在调用 wx.uploadVoice() 的接口 千万不要在 微信开发者工具中测试,它不支持, 每次返回的都一样的数, 这个要在手机端测试

转码程序一定要放到目标服务器上,并且有可以执行的权限(X的权限)

微信开发者配置记得配置JS安全域名不然不能够调用接口

给微信开发者推荐一个开源项目

推荐:[班纳睿 / weixin-java-tools](#)

真的强, 基于这个开发特别方便! 强推!