# 后端服务器开发过程中遇到的主要问题和对应的解决方案

## 1.内网穿透问题

开始时计划选择使用局域网与内网穿透两种方式实现联网功能，局域网实现较为容易，但是内网穿透过程中出现了一些问题。服务器使用的python的flask框架，内网穿透使用的是花生壳，一开始使用的tcp映射，将内网地址映射到公网上，但是考虑到以后服务器的可扩展性，改为使用http映射，但是由于未知的原因，无法创建http映射，且去网上查询资料无果，只能选择使用新的手机号注册了一个新账号，使用新用户免费的https映射

## 2.服务器端接收请求问题

小程序端发送post请求，但是服务器端无法接收到，使用浏览器发送请求则可以，通过查阅资料，发现flask默认不接收post请求，产生请求错误。通过查阅资料，将路由头部添加参数methods=['POST']以后，flask就可以接收post请求了

## 3.服务器返回请求问题

小程序端发送请求，但是显示返回类型错误，通过查阅资料，发现小程序端默认只能接收post格式的返回，直接返回字符串会产生返回错误，通过修改返回类型为post，并将参数封装在头部实现信息交流

## 4.图片文件保存问题

一开始的打算是小程序每请求一次，服务器端覆盖本地图片，但是在实际测试中发现open().write()方法无法自动覆盖同名文件，于是选择在每次请求的时候首先使用os将原图片文件删除，再保存的方法，实现图片文件的替换

## 5.图片文件下载问题

小程序通过将图片文件的下载地址放在post请求中发给服务器，服务器使用下载地址获取图片文件。但是在实际测试中发现无法获取图片文件，通过排查发现是下载链接的问题，通过request.get\_data()方法获取的下载链接其实是一个bytes类型的数据，通过查阅资料，使用decode()方法将bytes转换为String类型，而且由于该字符串包含的是一个类似字典类型的键值对，所以使用evol()函数将字符串转为字典，并通过key downloadURL获取了真正的下载地址

# 小程序开发过程中遇到的主要问题和对应的解决方案

## 微信小程序API的wx.uploadfile接口使用问题

在小程序的开发过程中，为了实现图片上传到服务器的功能，我们打算调用wx.uploadfile接口进行该功能的实现。按照官方给出的使用文档，我们输入了指定的参数，调用接口后，测试的服务器没有收到任何请求，小程序的控制台也没有任何报错，设定的成功后调用方法也没有执行。我们怀疑是在小程序调用接口时出现了错误，但不知道是什么错误。我们查阅众多资料后，并没有解决该BUG，但得到了解决该BUG的思路：进行error信息的打印，即不管请求成功与否，都打印res的值。打印出来后，发现需要的本地路径参数是要为一个String类型的参数，而我们传入的是Object类型的参数。完成字符串转换后，该接口调用正常执行了，但是在后端服务器端却出现了报错，经过查阅资料后得知该报错的原因是要对请求进行一个特殊的处理，但是在python语言版本中暂时没有找到对应的方法，于是我们就转换思路，引入一个中间介质（云端服务器），我们向云端服务器上传图片，然后让后端服务器去请求图片即可。

## 微信小程序自带的云端服务器的使用问题

转换思路后，我们查看了微信小程序自带的云端开发文档，得知每一个项目都可以申请免费额度的云端服务器（腾讯云服务）。于是我们决定使用该云端服务进行照片的存储。对于微信小程序开发自带的云端存储，我们使用了wx.cloud.uploadFile接口，将图片上传到了云端。但是后端服务器请求图片是需要向云服务器发请求的，即需要一个URL，而wx.cloud.uploadFile接口成功调用后仅会返回一个图片在云端的文件id（唯一标识符），没有其他的有用信息。在查询了小程序云端开发的使用手册后，我们得知有一个接口可以通过文件id来获取下载地址，即wx.cloud.getTempFileURL接口。我们调用该接口，拿到了下载地址，然后我们把下载地址放入到请求中作为参数，一同发送给后端服务器。后端服务器拿到了该参数，然后就去向云端服务器请求下载，得到了图片，可以进行处理。

## 3.云端服务器的配置问题

实现了传输图片的功能后，我们立马对传入的图片进行了检测。但是我们传入了有缺陷的图片后，检测结果为’有缺陷’，这是正确的。但是我们之后再多次传输图片进行测试时，无论传入的时有缺陷还是无缺陷的图片，其处理结果都是’有缺陷’。我们感到非常疑惑，于是将向云端服务器请求的图片下载到本地，从本地中打开进行查看。打开后发现，我们在一段时间内下载的图片，都是第一次传入的图片，也就是说，后续请求的图片都是最开始的图片，而我们上传的最新的图片并没有传入到后端。我们猜测：云端服务器的存储图片是异步的，也就是说我们上传图片给云端服务器，云端服务器收到后给我们返回一个’成功上传’的消息，但此时云端服务器还未对该图片进行存储，而是放在了队列中，等待前面的传输任务完成后才进行我们文件的存储。此时我们访问的资源就是之前还未更新的资源了，因此才出现了上述的现象。这是我们的初步讨论猜想，在对云开发环境配置的仔细查看中，我们发现了一个叫做’缓存设置’的配置选项，里面默认的缓存时间是2分钟。我们查询了相关资料，发现腾讯云服务器使用的是CDN缓存机制，也就是当有用户请求服务器资源时，会把资源放到缓存服务器缓存当中，并设置清楚时间为2分钟（默认），当2分钟内，有用户要访问同样的资源时，就会从缓存中直接返回给用户，不用再去数据库中进行查找。我们明白了出错原因，将缓存时间设置为0ms，也就是不缓存，来达到让后端服务器100%能拿到最新的资源。设置完毕后经过测试，后端服务器每次拿到的都是最新的用户上传的图片，满足了我们的功能要求。

至此，小程序端的Debug工作完成，实现了功能。