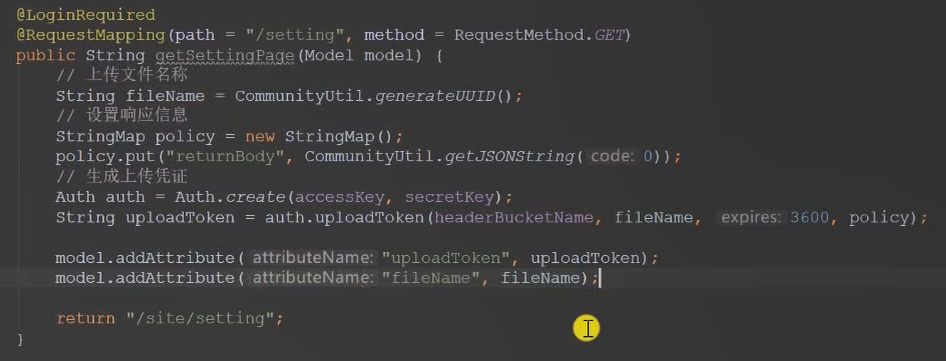


选用免费的七牛云服务器，从maven中导入七牛云的包

AK:标识用户身份（使用者才有权限往云服务器中存入数据），SK：为数据内容加密

将头像上传至云服务器（客户端上传）：需要将上传凭证传给表单，使用户在表单上点击提交时可以将相应的数据传给七牛云云服务器，其中设置响应信息可以设置为网页也可以设置为json字符串，一般选用字符串，因为这个是七牛云给返回的，如果选择返回网页的话，太乱了，会让客户感觉到，一会是我们自己的网站，一会又是七牛云的，不太好，响应信息的意思是你提交成功后，给你返回的提示



更新头像路径：在用户上传了新的头像后，需要将此用户的头像路径更改，更改的路径为你云服务器的路径＋文件名称

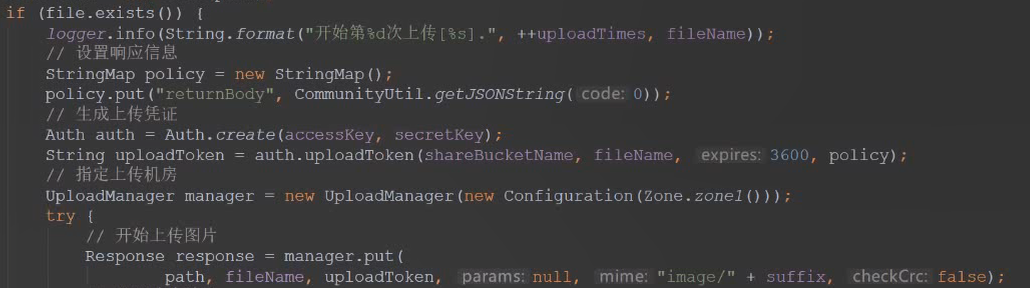


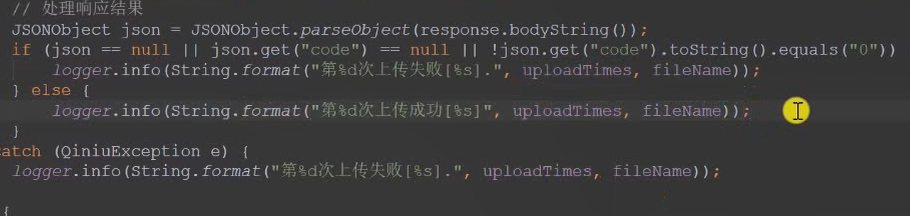
注意：前面只是说了给表单一个上传凭证和更新头像路径，那上传头像的操作在哪呢？这在后端是没有的，需要在前端的js（首先在提交表单时将头像上传至云服务器，之后调用controller层的更新头像路径方法，再刷新网站，使得新的头像能被显示出来）里设置

分享（服务端直传）：我们在将分享的图片存入本地时，用的是消息队列来处理生成需要分享的图片，上传至云服务器同样用消息队列，而且仍然需要上节课的wk工具在本地生成一个需要分享的图片，之后将这个图片再去传给云服务器，需要注意的点是，在用wk生成图片时，我们是调用底层系统令启线程来执行的wk命令生成分享图片，现在需要在wk生成分享图片后，将这个图片传给云服务器，在服务器直传时，不能像客户端上传一样，通过前端给云服务器上传而是需要在后端手动处理，那现在就有一个问题，不能知道什么时候这个线程执行完毕生成分享图片，所以需要启动一个定时任务来将这个分享的图片传给云服务器，定时查看这个线程是否执行完毕，如果执行完毕则将生成的图片传给云服务器，这里的定时任务没有用quartz而是用的spring的定时任务线程池，这是因为消息队列消费者在运行的时候已经确保了同一事件只能有一个服务器来访问，拥有有抢占机制，不会造成多台服务器去消费一个事件，执行同一件事情，而造成问题，而且你如果用quartz的话，在配置好之后，过了对应的延迟时间，就会立即执行，那你现在如果还没有需要将分享图片上传值云服务器的需求，也是问题，在启动上传至云服务器的线程时，如果上传云服务器失败不能一直让这个线程执行着，所以在上传云服务器三次未成功（可能是网络原因，也可能是云服务器的原因），或者超过了30s（几乎和生成分享图片的线程一起执行，但分享图片的线程执行完毕（生成了图片）用不到3s，所以时间是足够的，剩余的时间是为了防止上传至云服务器失败，让其多上传几次，而且上传也需要时间，这些总的加起来30s够用）就让这个线程停止运行，停止运行用了线程的方法

判断线程是否停止（略）：上传服务器三次未成功（每失败一次则对应的参数加一），超过30s（在线程启动时通过参数将现在时间记录，每运行一次线程则判断一次，当前时间与该参数的差值）

将图片上传至云服务器：处理响应结果就是或许到响应信息





获取分享图片看上一个总结