# 技术关键点

#### 转发与登录校验

本项目使用Spring Cloud Gateway 作为微服务转发和登录校验的网关。网关通过在 Spring Cloud Nacos 发现服务，我们为其配置了路由转发以及负载均衡的配置文件。

在实际测试时，我们发现云服务器在 nacos 注册的 IP 地址为服务器内网地址，会造成 403 服务不可达错误，于是我们在不同微服务的配置文件下注明了每个服务器的外网地址，帮助 gateway 成功完成转发。

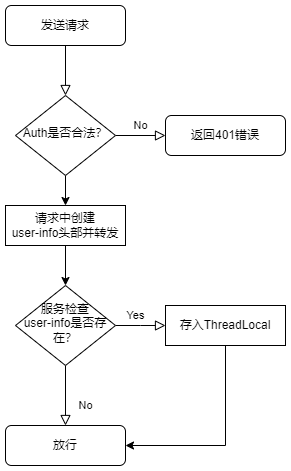
有趣的是，我们在分别服务器上远程部署了每个微服务后，在我们的电脑本地又开启了微服务，我们发现，nacos 上注册了我们本地的服务，但是地址为 192.168.x.x，此时我们部分地遇到了服务请求时间过长问题，这变相地说明我们的负载均衡配置成功了。

在用户 Token 方面，我们使用了 JWT 作为用户 Token 生成器，在 UserService 微服务登录时生成，有效期 10 分钟。

对于登录校验，我们为 gateway 配置了 globalFilter，首先获取请求头中的 Authorization，将其用 JWT 解析，如果解析无效或者校验过期，则不允许放行，返回401错误；反之，如果验证了数据有效，则将其转换为 userId，新建 user-info的请求头部，然后进行转发。

微服务接收到请求后，进入UserInfo拦截器解析user-info，并将其存入 ThreadLocal 中，方便服务内使用。

流程图如下所示：



#### 将图片上传至阿里云oss并返回oss链接

本项目有两个功能用到此技术，分别是用户上传头像及作者上传小说封面。  
本功能主要过程为：  
 1.前端获取需要上传的图片

2.前端将图片传给后端

3.后端将图片上传至阿里云oss容器内，并返回图片在阿里云oss容器内的对应链接

4.后端将链接传给前端

5.前端读取链接即可展示对应图片

流程图如下所示



#### 自动化构建部署

我们在服务打包和上传的过程中不堪其扰，繁琐的打包上传工作几乎每次commit都要进行，尤其是我们在把微服务配置好之后。

因此，我们在github上做了自动化构建部署工作，通过获取每台服务器的ip地址、用户名、ssh端口号和ssh密钥。

通过使用 scp 文件上传以及 ssh-action ssh执行命令，我们实现了每次 main 分支 commit 时，微服务打包为jar包，自动部署到远端服务器，并终止之前已经运行的程序，重启服务。

流程图如下：

