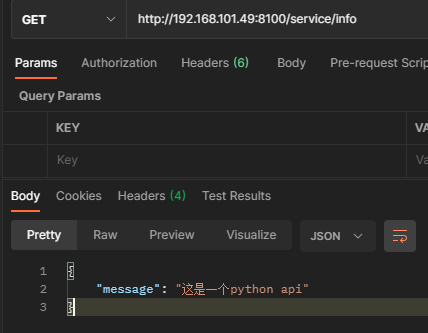
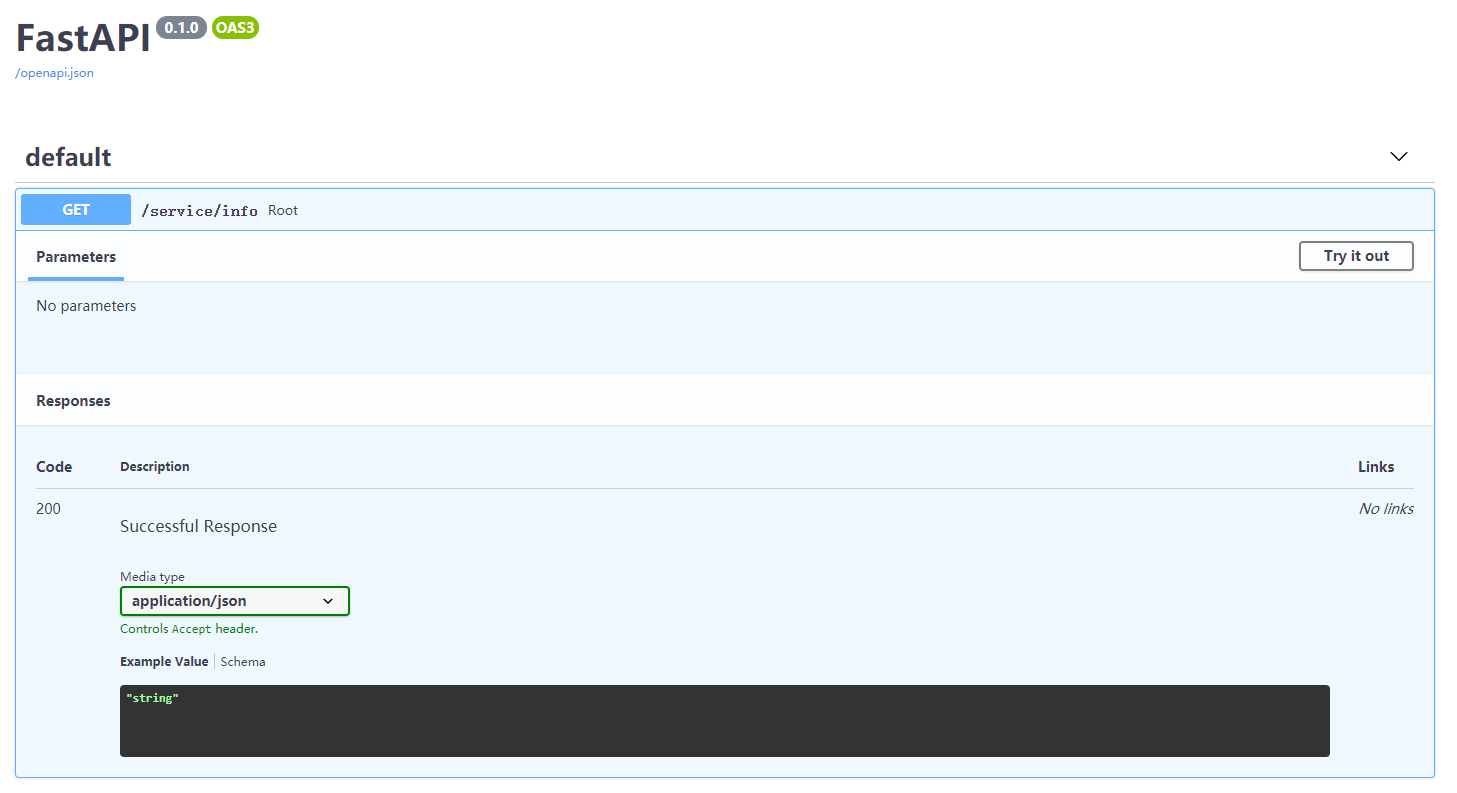
## 通过mesher实现其他微服务调用python的rest接口

## 1. python rest接口开发

使用fastapi开发一个rest接口，效果如下：



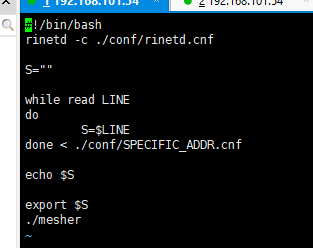


fastapi接口会自动生成openapi.json，需要根据这个json生成服务契约（schema）。

## 2. mesher配置

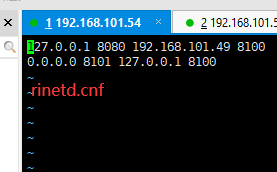
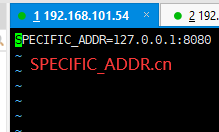
这里我将mesher放在了另外一台物理机上的docker容器中，其中物理机的ip地址为192.168.101.54，docker容器的ip地址为172.17.0.2。

docker容器的基础镜像是centos，在其中又加装了rinetd端口转发工具。制作镜像时将mesher整个目录copy到了镜像的/mesher目录下，镜像工作目录为/mesher。同时在镜像/mesher中添加启动脚本start.sh如下：

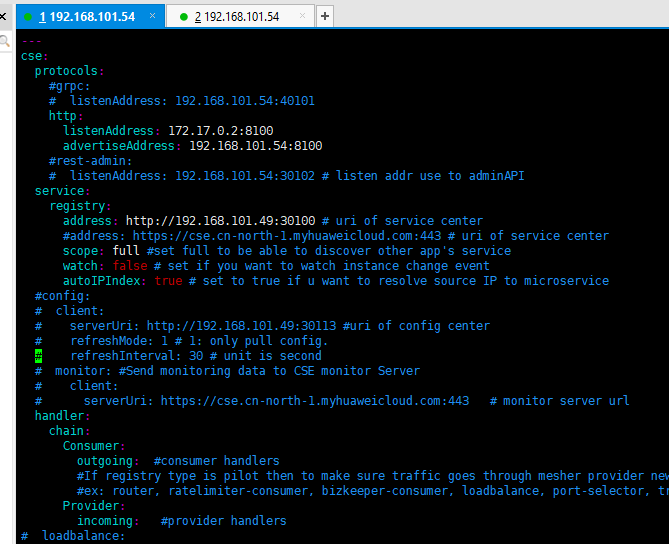


启动脚本中包含了开启容器端口映射，设置SPECIFIC\_ADDR和启动mesher三个步骤。然后启动这个镜像的docker容器。

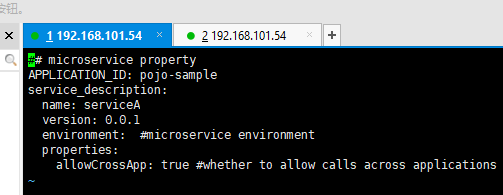
在/mesher/conf中添加rinetd.cnf和SPECIFIC\_ADDR.cnf如下：



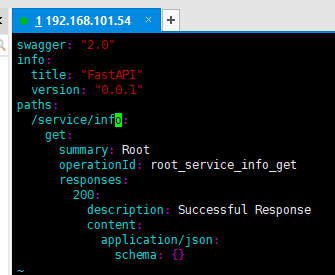
修改容器中/mesher/conf/chassis.yaml如下：



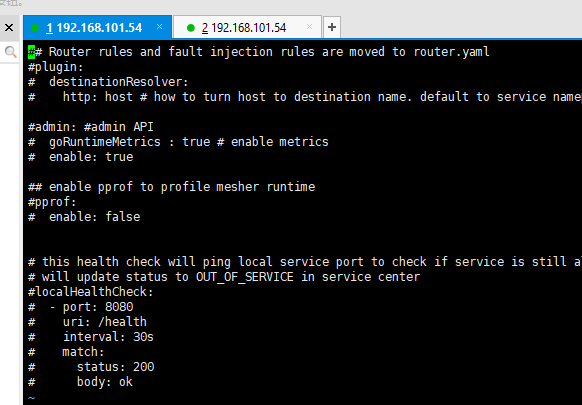
修改容器中/mesher/conf/microservice.yaml如下：



在/mesher/conf中添加serviceA/schema/serviceA.yaml（接口契约）如下：

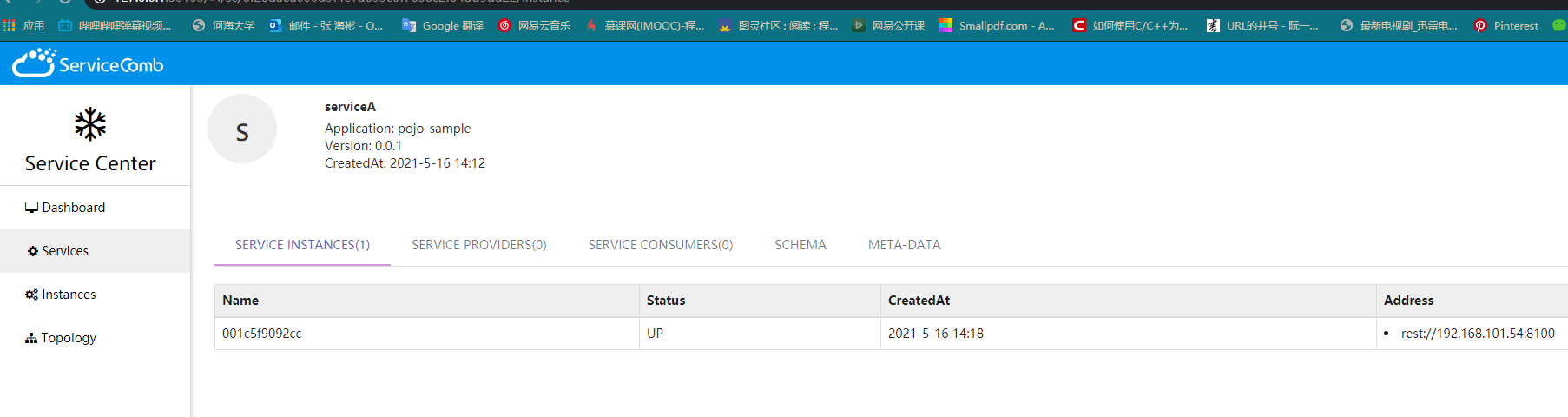


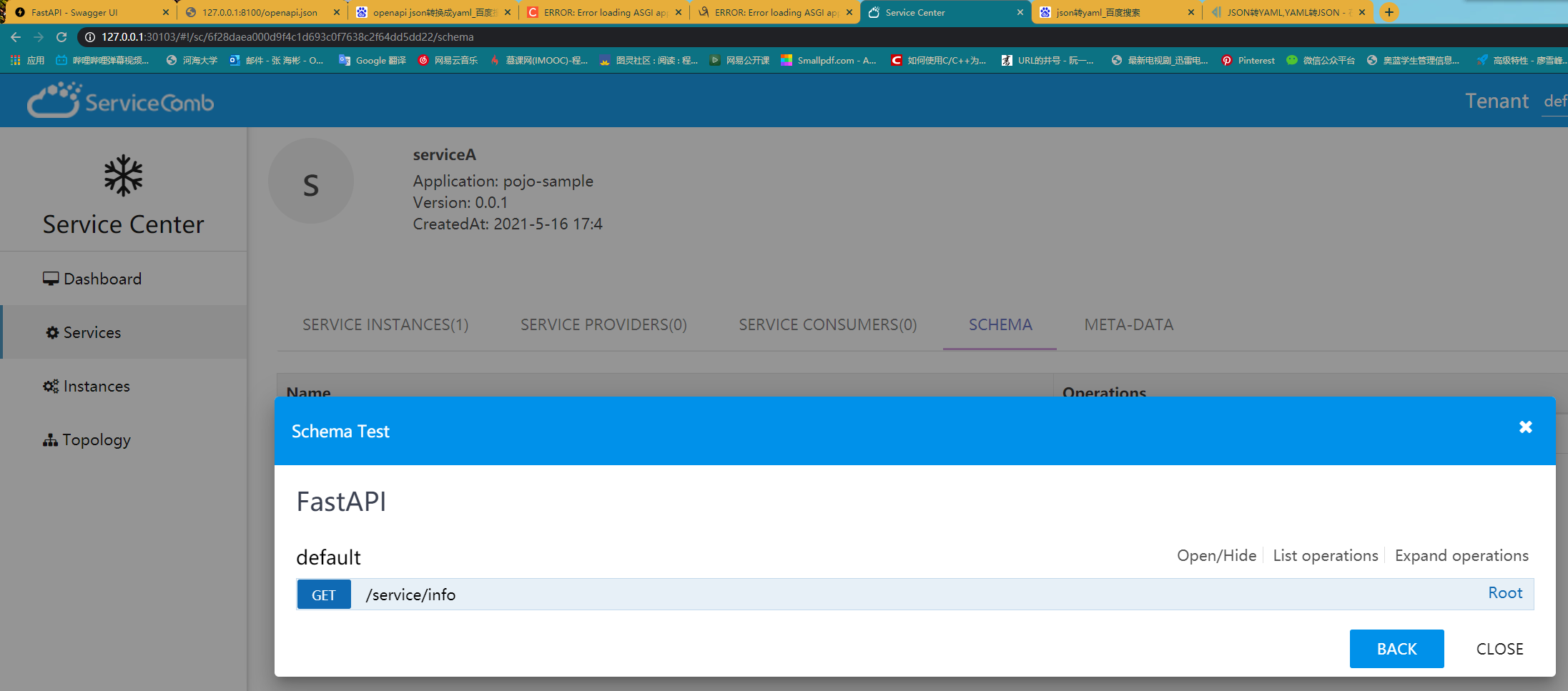
修改容器中/mesher/conf/mesher.yaml如下：



当然，这个docker容器的/mesher/conf可以通过docker run -v进行映射。

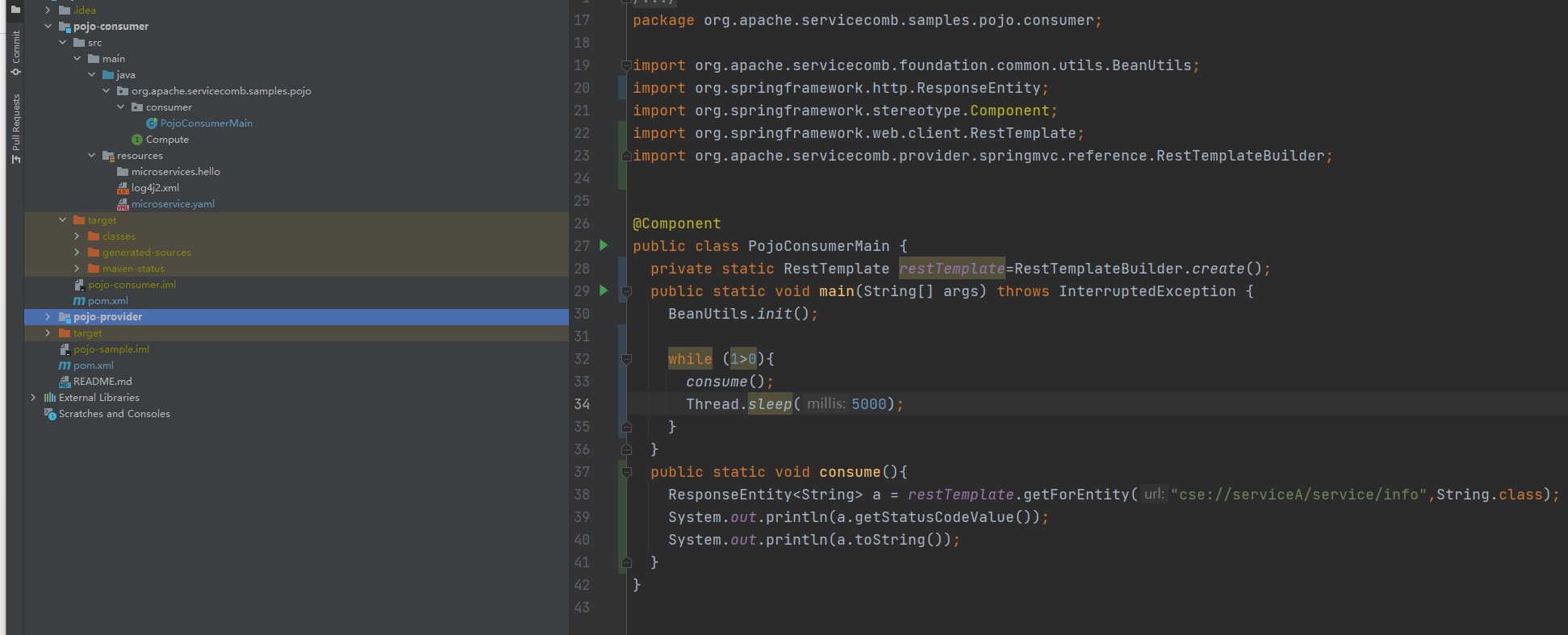
在配置好mesher/conf文件夹中对应文件之后，使用命令：docker run -it -p 8100:8100 -p 8101:8101 -v /home/mesher/serviceA\_mesher:/mesher/conf mesher\_image bash start.sh（这里我将/home/mesher/serviceA\_mesher映射到了容器的/mesher/conf目录，避免进入容器进行配置）运行。运行之后效果如下：



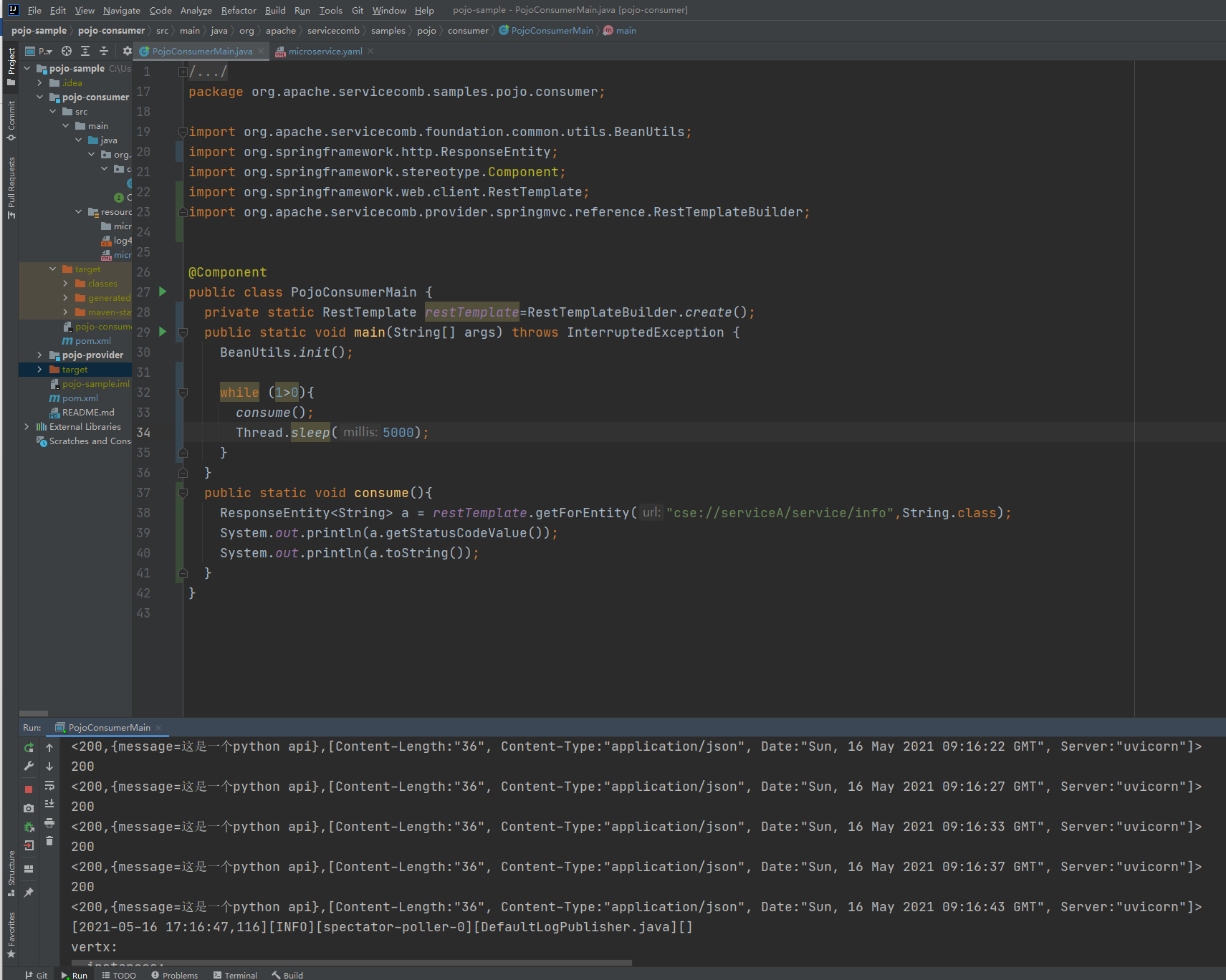


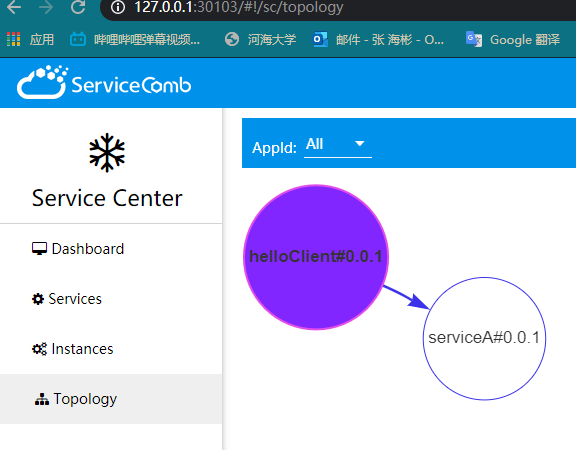
## 3. 测试

经过以上两个步骤，运行在192.168.101.49的rest接口和运行在192.168.101.54的docker容器中的mesher都启动了。在192.168.101.49中再注册一个consumer微服务到同一个注册中心进行python微服务接口的调用：



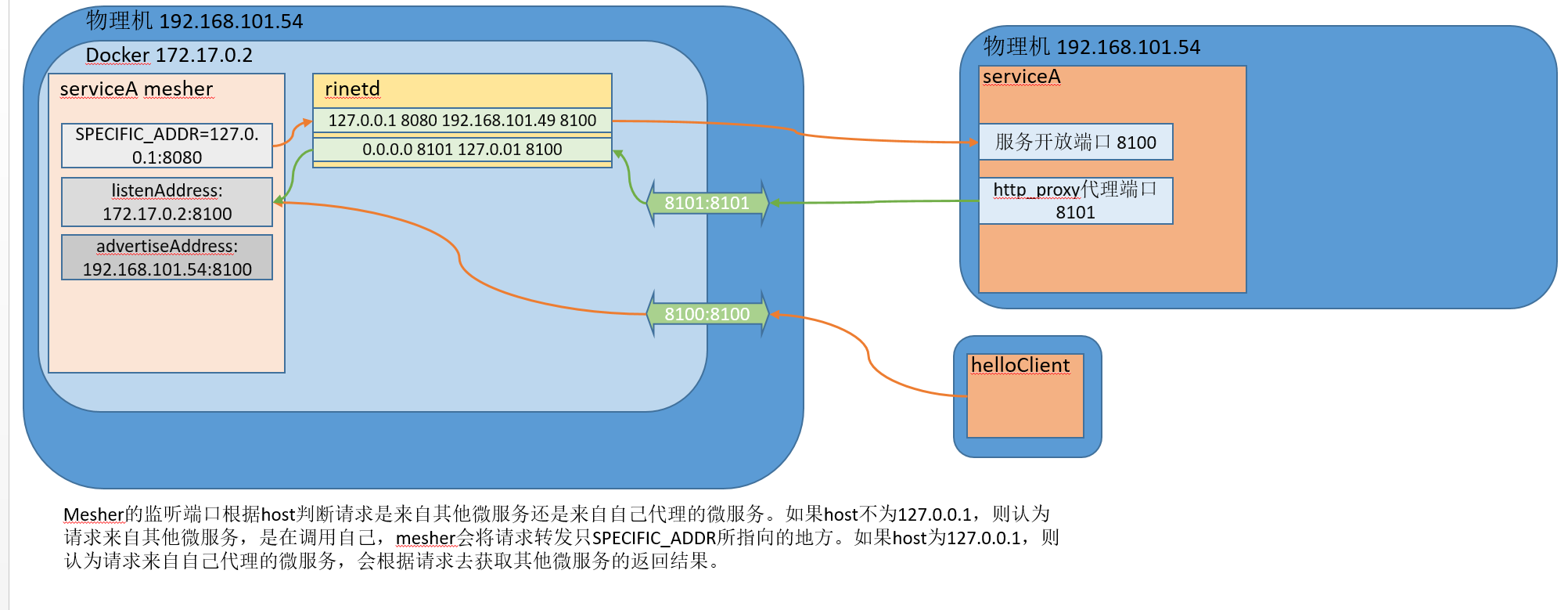
效果如下：





## 4. 总结

以上就是将python的rest接口接入注册中心使其他微服务能够调用的全流程。通过端口映射的方式可以实现将mesher和业务代码分离开。



这个python微服务通过mesher调用其他微服务时，http请求最好用python的urllib这个标准库，使用requests会出现域名中大写字母被变成小写字母的情况。