# Spring Boot使用Redis管理Session

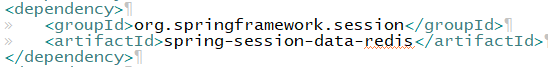
在开发后端服务的时候，我个人是比较倾向构建无状态接口的，如果需要验证可以使用OAuth2.0协议，给客户端Token，我们存入Redis等内存数据库。接口端通过AOP进行校验。

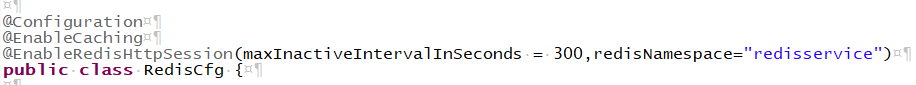
不过实际工作中还是有很多需要Session信息的项目，比如管理系统等等。

这就又牵扯出一个问题，如果我们的后端服务需要负载均衡——不管是用Nginx、还是Spring Cloud的Ribbon进行负载转发，多个服务端的Session如何统一管理？

本例采用Redis。

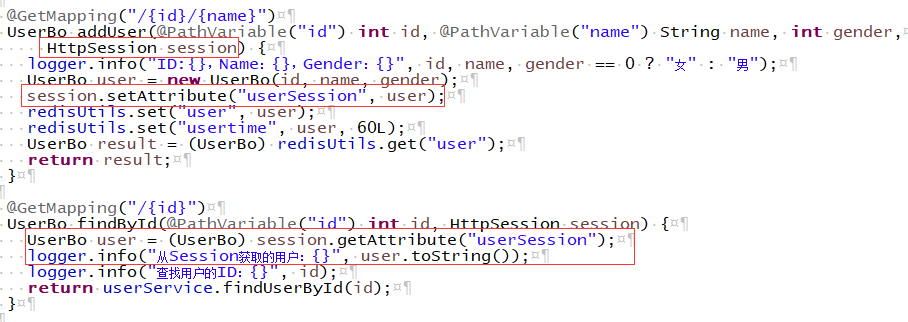
## 配置





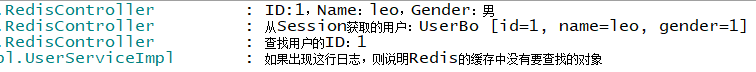
在原来的基础上增加**@EnableRedisHttpSession**注解，其中maxInactiveIntervalInSeconds是Session失效时间。在实际工作中，Redis肯定不会只管理一个服务的Session，每个服务的Session最好有所区别，redisNamespace就是用来设置命名空间的。

## Controller



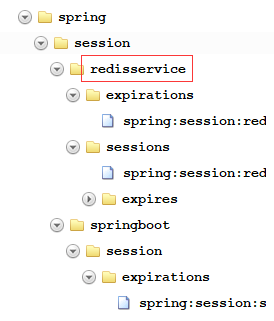
第一个方法设置Session信息，第二个方法获取Session信息。

## 结果



当执行第一个方法时，已经在Session里设置了用户信息，写入Redis了。

第二个方法从Session中取出了用户信息。



看看Redis里的数据，红框就代表这下面是我们当前服务的Session。