Eureka核心功能服务注册(register)：Eureka Client会通过发送REST请求的方式向Eureka Server注册自己的服务，提供自身的元数 据，比如ip地址、端口、运行状况指标的url、主页地址等信息。Eureka Server接收到注册请求后，就会把这些元数 据信息存储在一个双层的Map中。

服务续约(renew)：在服务注册后，Eureka Client会维护一个心跳来持续通知Eureka Server，说明服务一直处于可 用状态，防止被剔除。Eureka Client在默认的情况下会每隔30秒(eureka.instance.leaseRenewalIntervalInSeconds)发送一次心跳来进行服务续约。

服务同步(replicate)：Eureka Server之间会互相进行注册，构建Eureka Server集群，不同Eureka Server之间会进 行服务同步，用来保证服务信息的一致性。

获取服务(get registry)：服务消费者（Eureka Client）在启动的时候，会发送一个REST请求给Eureka Server，获 取上面注册的服务清单，并且缓存在Eureka Client本地，默认缓存30秒 (eureka.client.registryFetchIntervalSeconds)。同时，为了性能考虑，Eureka Server也会维护一份只读的服务清 单缓存，该缓存每隔30秒更新一次

服务调用(call)：服务消费者在获取到服务清单后，就可以根据清单中的服务列表信息，查找到其他服务的地址，从而进行 远程调用。Eureka有Region和Zone的概念，一个Region可以包含多个Zone，在进行服务调用时，优先访问处于同 一个Zone中的服务提供者。

服务下线(cancel)：当Eureka Client需要关闭或重启时，就不希望在这个时间段内再有请求进来，所以，就需要提前 先发送REST请求给Eureka Server，告诉Eureka Server自己要下线了，Eureka Server在收到请求后，就会把该服务 状态置为下线（DOWN），并把该下线事件传播出去。

服务剔除(evict)：有时候，服务实例可能会因为网络故障等原因导致不能提供服务，而此时该实例也没有发送请求给 Eureka Server来进行服务下线，所以，还需要有服务剔除的机制。Eureka Server在启动的时候会创建一个定时任 务，每隔一段时间（默认60秒），从当前服务清单中把超时没有续约（默认90秒， eureka.instance.leaseExpirationDurationInSeconds）的服务剔除。

自我保护：既然Eureka Server会定时剔除超时没有续约的服务，那就有可能出现一种场景，网络一段时间内发生了 异常，所有的服务都没能够进行续约，Eureka Server就把所有的服务都剔除了，这样显然不太合理。所以，就有了 自我保护机制，当短时间内，统计续约失败的比例，如果达到一定阈值，则会触发自我保护的机制，在该机制下，Eureka Server不会剔除任何的微服务，等到正常后，再退出自我保护机制。自我保护开关(eureka.server.enable- self-preservation: false)