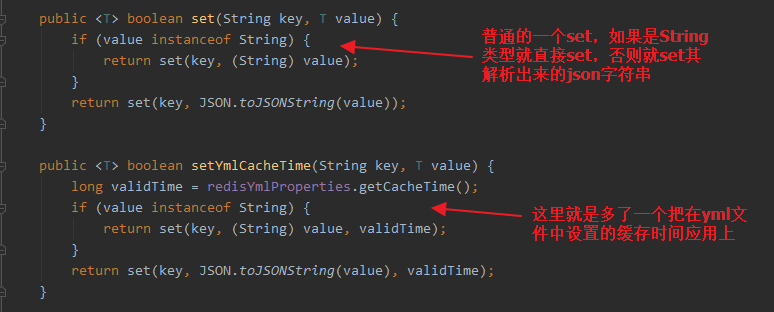
# Redis封装类的小解

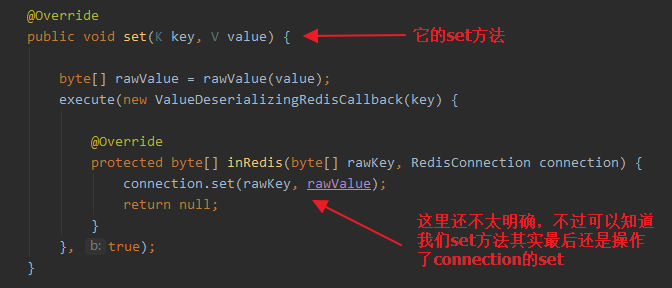
项目依赖于springboot，工具类分别提供了静态方法和注入bean的方法；工具类中已经配置了redis的缓存配置。同时封装扩展了一些缓存注解

## RedisStrUtil

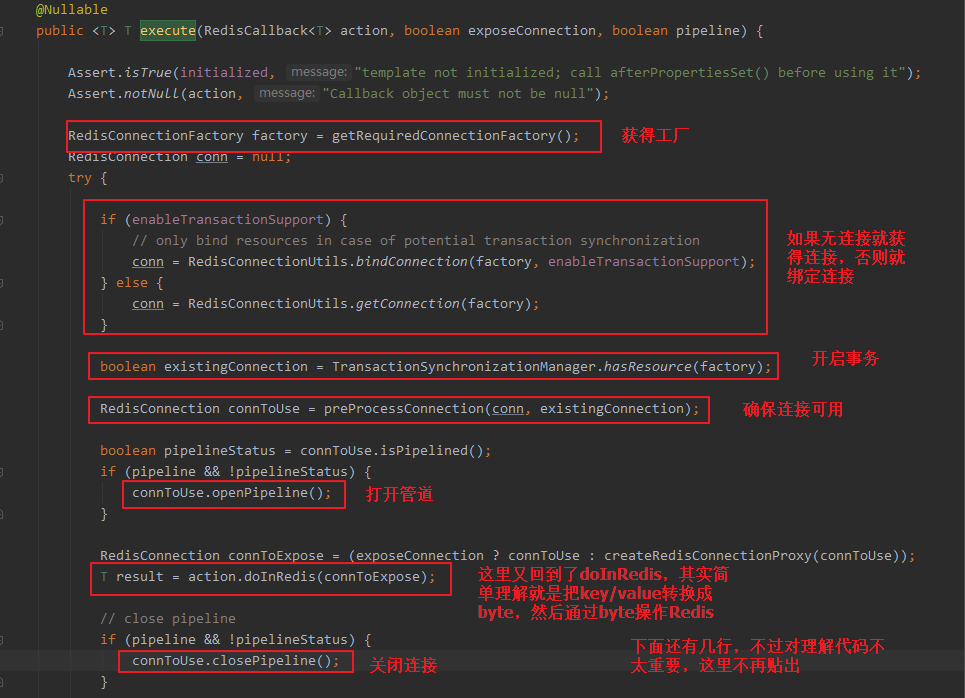




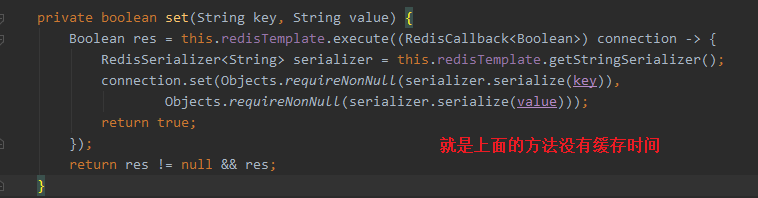
**我们具体看看spring的RedisTemplate的实现，也许可以更加理解上面的代码，这是RedisTemplate的opsForValue方法返回的一个实现类**

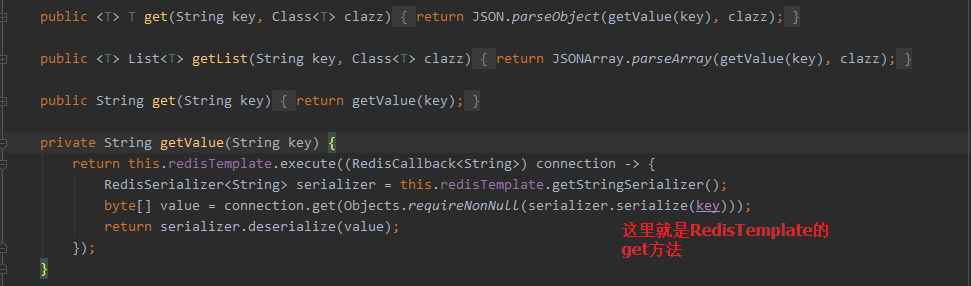
****

**我们最后看看Execute方法**

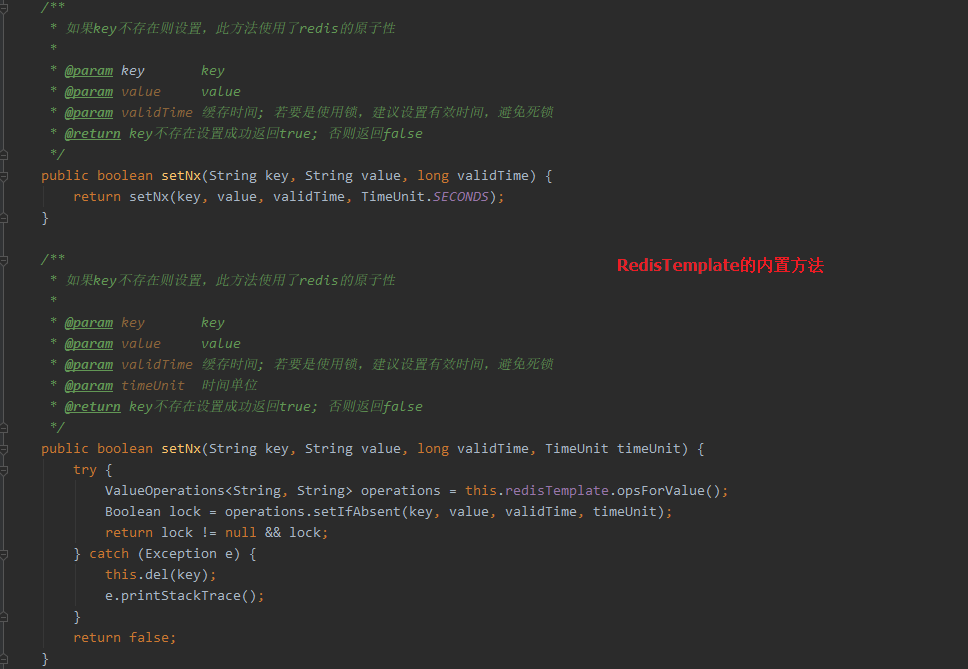
****

**所以总结一下，个人认为这里就是模仿Spring的RedisTemplate进行缓存数据的增添，它只是把key/value转换成byte，然后根据connection来进行赋值的步骤糅合在Execute方法里面做了（其实就是省略了inRedis步骤）**

****

**再多说一句，Redis的Set/Get…等方法其实到最后都是通过Connection的操作流来实现，所以这里都调用了execute方法。**

****

****

****

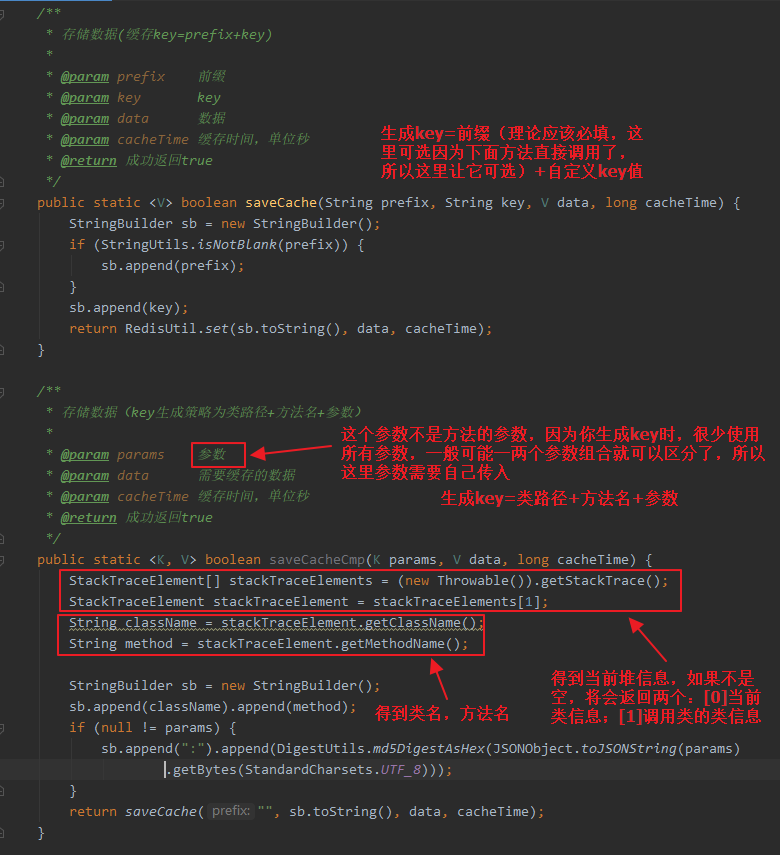
****

## RedisUtil

这里的方法全部都是调用的RedisStrUtil里面的方法，这里只是把方法做了二次封装，并且将方法变成了静态方法，方便调用，加了一些Log的异常日志，方便在服务器中查看异常信息

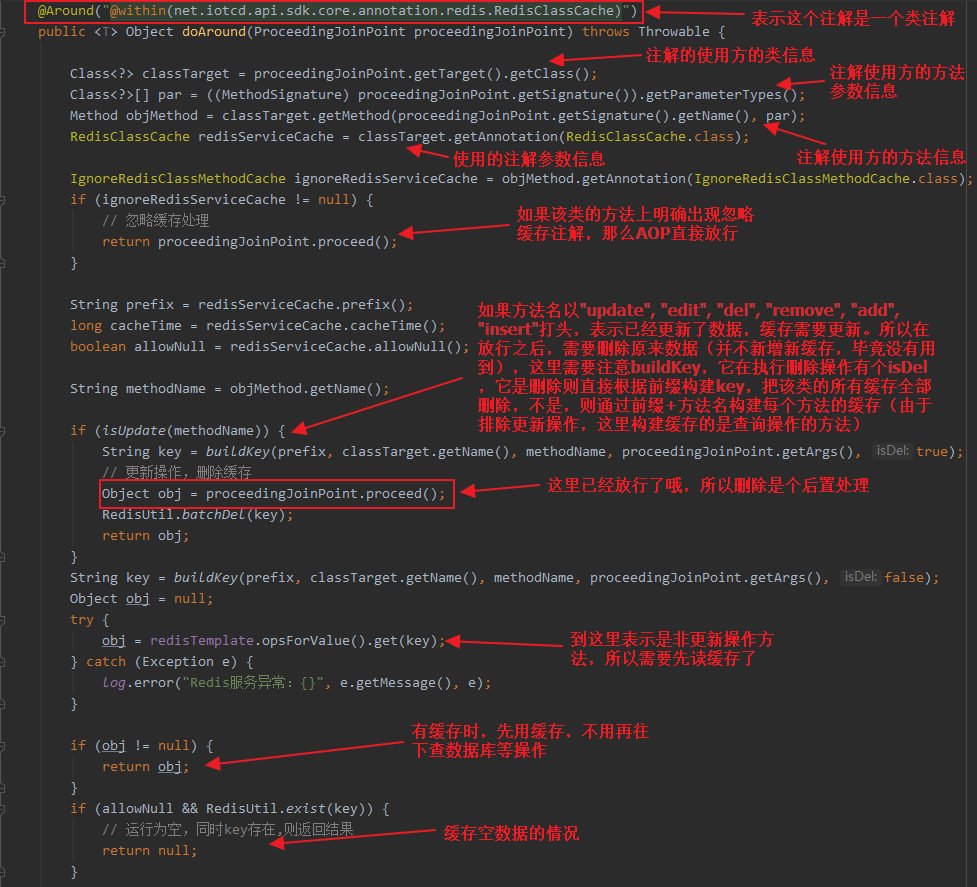
## RedisAutoKeyUtil

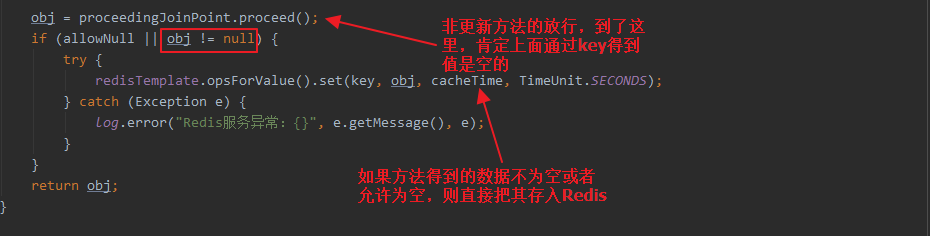
这个类封装的是根据一些固定的策略，自动的生成reids的key的缓存工具，由于这个类里面就是各种key的生成策略以及调用之前的Util方法，这里只做一个方法来展示下，想了解该类的其它方法直接去代码中看该类即可

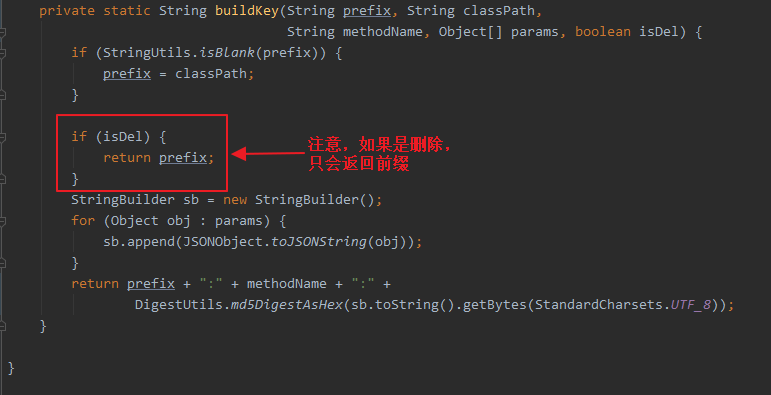


## RedisClassCache

这是一个类注解，一般用于Service层的使用，通过注解自动对Redis的更新，如果删除，将删除类的整个缓存

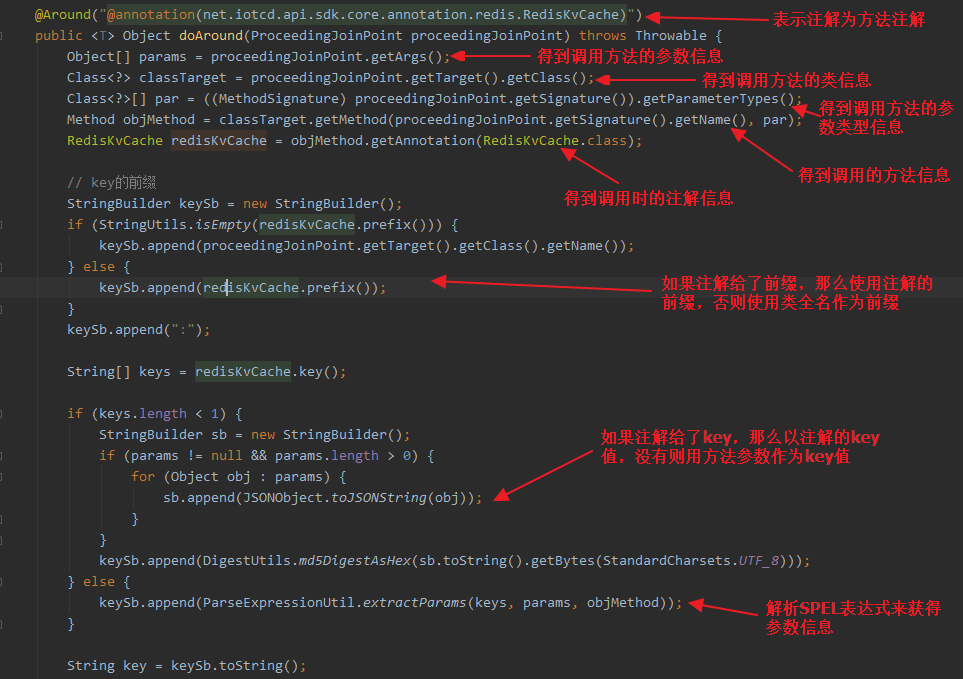


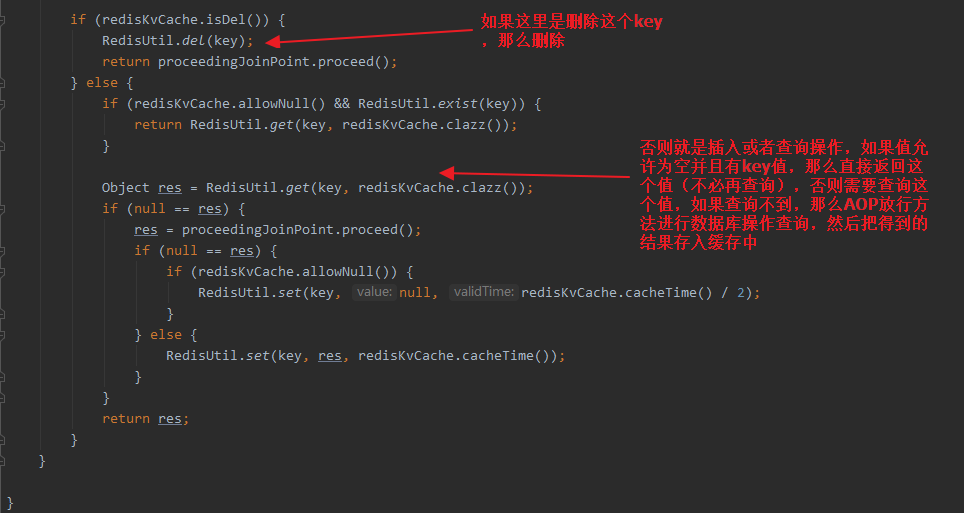




## RedisKvCache

这是一个方法注解，表示给该方法添加缓存





## RedisListCache

这里并不是用Reids的List进行缓存。实际场景是这样：一个查询方法如果返回的是集合（List）那么把它们的值解析成一个Json字符串进行缓存，而调用缓存信息时，也是将Json字符串进行解析（对于Redis而言，它也只是一个<String,String>的简单缓存





## RedisPageCacheSet与RedisPageCacheDel

这个注解是进行分页的缓存的设置和进行分页缓存的删除。这里存的Redis集合，total和数据分开存储（因为这里是缓存分页，所以可能只需要部分数据，这里存成Redis的集合好进行数据取出）。这里先查了1000条数据（默认），如果超过了1000，那么会再去查下一个1000，然后分别在不同的key/value中。例如：如果1001条数据，假设前端（也就是注解分页参数）刚好999，1000，1001是一页，那么它会取两次，一次从1-1000数据取到999，1000第二次从1001-2000（假如数据够，不够则直接到结尾）取1001。（注解里面的分页信息是对Redis的分页，表示Redis每次从数据库取多少存入，前端的业务分页信息是从接口里面参数得到的）

