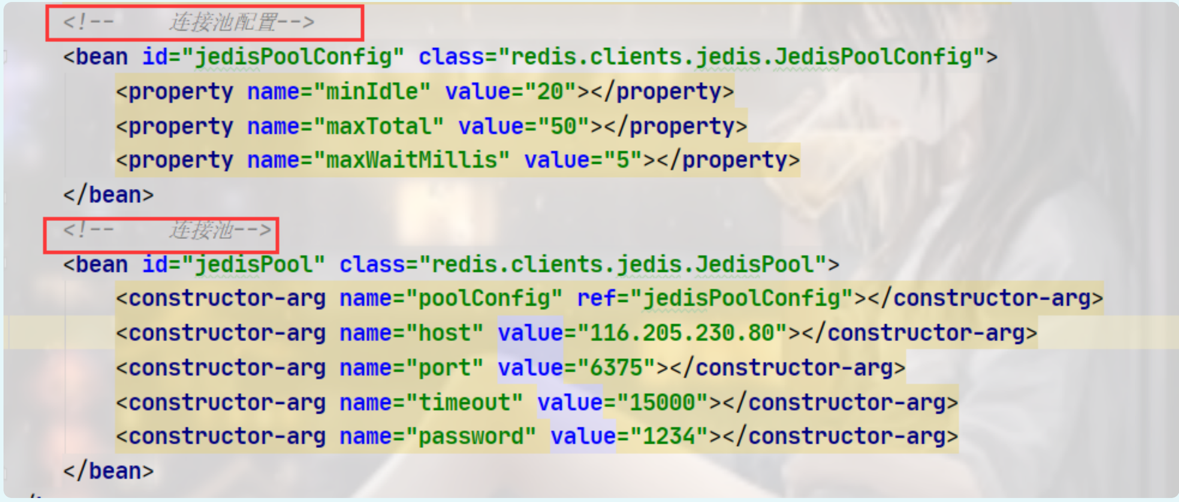
redis-offcn

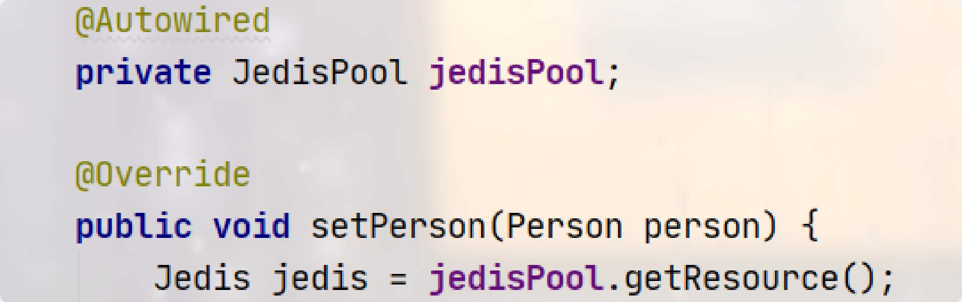
1. redis容易忘记的命令

> - 存活周期：
>
> expire
>
> setex key value
>
> ttl key
>
> - string数据结构:
>
> del key
>
> setnx
>
> mset mget
>
> - hash数据结构
>
> hset hget hmget hvals hkeys hgetall
>
> - list数据结构
>
> lpush lrange
>
> - set数据结构
>
> sadd spop随机出来几个key smembers 交并差
>
> - zset数据结构
>
> 查手册吧

2.spring整合redis

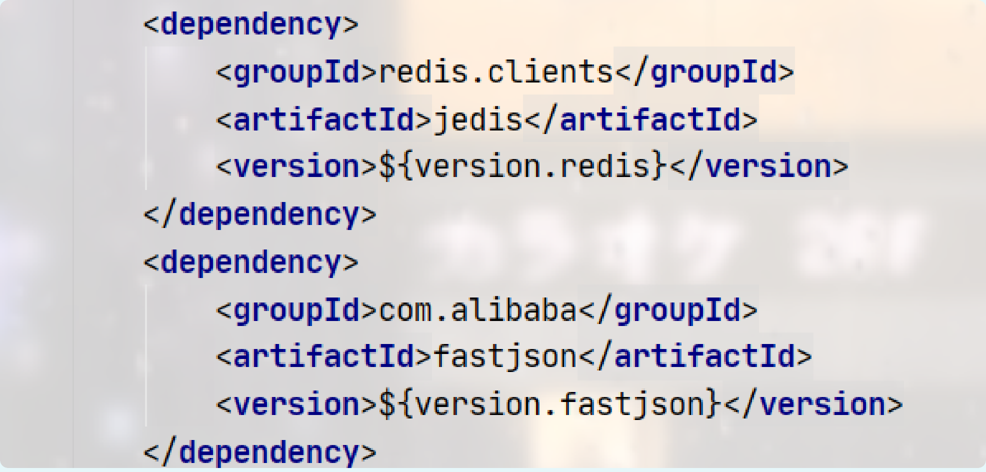


```
<!--      连接池配置-->
<bean id="jedisPoolConfig" class="redis.clients.jedis.JedisPoolConfig">
    <property name="minIdle" value="20"></property>
    <property name="maxTotal" value="50"></property>
    <property name="maxWaitMillis" value="5"></property>
</bean>
<!--      连接池-->
<bean id="jedisPool" class="redis.clients.jedis.JedisPool">
    <constructor-arg name="poolConfig" ref="jedisPoolConfig"></constructor-arg>
    <constructor-arg name="host" value="116.205.230.80"></constructor-arg>
    <constructor-arg name="port" value="6375"></constructor-arg>
    <constructor-arg name="timeout" value="15000"></constructor-arg>
    <constructor-arg name="password" value="1234"></constructor-arg>
</bean>
```



```
@Autowired
private JedisPool jedisPool;

@Override
public void setPerson(Person person) {
    Jedis jedis = jedisPool.getResource();
```



```
<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>${version.redis}</version>
</dependency>
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>${version.fastjson}</version>
</dependency>
```

3. springboot整合redis

redis配置类

```java
@Configuration
@EnableCaching
public class RedisConfig extends CachingConfigurerSupport {
    @Bean
    public RedisTemplate<String, Object> redisTemplate(RedisConnectionFactory factory) {
        RedisTemplate<String, Object> template = new RedisTemplate<>();
        RedisSerializer<String> redisSerializer = new StringRedisSerializer();
        Jackson2JsonRedisSerializer jackson2JsonRedisSerializer = new Jackson2JsonRedisSerializer(Object.class);
        ObjectMapper om = new ObjectMapper();
        om.setVisibility(PropertyAccessor.ALL, JsonAutoDetect.Visibility.ANY);
        om.enableDefaultTyping(ObjectMapper.DefaultTyping.NON_FINAL);
        jackson2JsonRedisSerializer.setObjectMapper(om);
        template.setConnectionFactory(factory);
        //key序列化方式
        template.setKeySerializer(redisSerializer);
        //value序列化
        template.setValueSerializer(jackson2JsonRedisSerializer);
        //value hashmap序列化
        template.setHashValueSerializer(jackson2JsonRedisSerializer);
        return template;
    }

    @Bean
    public CacheManager cacheManager(RedisConnectionFactory factory) {
        RedisSerializer<String> redisSerializer = new StringRedisSerializer();
        Jackson2JsonRedisSerializer jackson2JsonRedisSerializer = new Jackson2JsonRedisSerializer(Object.class);
        //解决查询缓存转换异常的问题
        ObjectMapper om = new ObjectMapper();
        om.setVisibility(PropertyAccessor.ALL, JsonAutoDetect.Visibility.ANY);
        om.enableDefaultTyping(ObjectMapper.DefaultTyping.NON_FINAL);
        jackson2JsonRedisSerializer.setObjectMapper(om);
        // 配置序列化（解决乱码的问题）,过期时间600秒
        RedisCacheConfiguration config = RedisCacheConfiguration.defaultCacheConfig()
                .entryTtl(Duration.ofSeconds(600))
                .serializeKeysWith(RedisSerializationContext.SerializationPair.fromSerializer(redisSerializer))
                .serializeValuesWith(RedisSerializationContext.SerializationPair.fromSerializer(jackson2JsonRedisSerializer))
                .disableCachingNullValues();
        RedisCacheManager cacheManager = RedisCacheManager.builder(factory)
                .cacheDefaults(config)
                .build();
        return cacheManager;
    }

}
```

application.yml Java | 复制代码

```
#Redis 服务器地址
spring:
  redis:
    port: 6375
    password: 1234
    host: 116.205.230.80
    timeout: 15000
    database: 0
    lettuce:
      pool:
        max-active: 20
        max-idle: 5
        max-wait: -1
        min-idle: 0
```

pom.xml XML | 复制代码

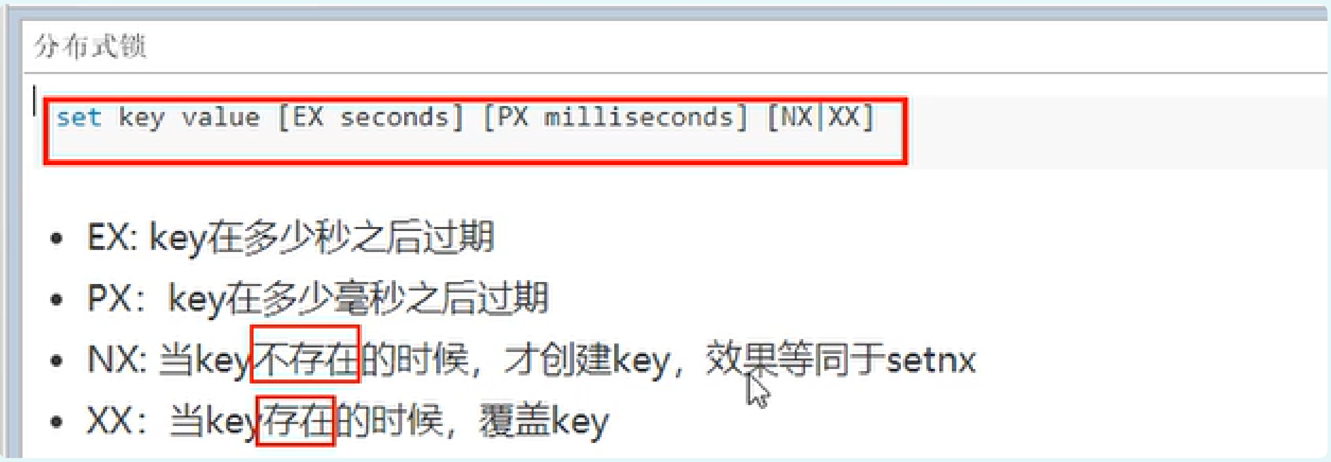
```
<!-- redis -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
<!-- spring2.X集成redis所需common-pool2-->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-pool2</artifactId>
        </dependency>
```

4. redis的命令手册关键

- 查看redis版本命令

  redis-server -v

  进入客户端连接：info
- 网站：http://www.redis.cn/commands.html
- bitmap HyperLogLog GEO Stream
- redis命令不区分大小写， 而key是区分大小写的
- help @类型名词(String list)



分布式锁

```
set key value [EX seconds] [PX milliseconds] [NX|XX]
```

- EX: key在多少秒之后过期
- PX：key在多少毫秒之后过期
- NX: 当key不存在的时候，才创建key，效果等同于setnx
- XX：当key存在的时候，覆盖key