1. 添加依赖

<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>  
 <version>2.6.0</version>  
</dependency>

1. 改yml

spring.redis.host=192.168.44.165  
spring.redis.port=6379  
spring.redis.database= 0  
spring.redis.timeout=1800000  
  
spring.redis.lettuce.pool.max-active=20  
spring.redis.lettuce.pool.max-wait=-1  
#最大阻塞等待时间(负数表示没限制)  
spring.redis.lettuce.pool.max-idle=5  
spring.redis.lettuce.pool.min-idle=0

1. 写配置

@Configuration  
@EnableCaching  
public class RedisConfig {  
 */\*\*  
 \* 自定义key规则  
 \** ***@return*** *\*/* @Bean  
 public KeyGenerator keyGenerator() {  
 return new KeyGenerator() {  
 @Override  
 public Object generate(Object target, Method method, Object... params) {  
 StringBuilder sb = new StringBuilder();  
 sb.append(target.getClass().getName());  
 sb.append(method.getName());  
 for (Object obj : params) {  
 sb.append(obj.toString());  
 }  
 return sb.toString();  
 }  
 };  
 }  
  
 */\*\*  
 \* 设置RedisTemplate规则  
 \** ***@param*** *redisConnectionFactory  
 \** ***@return*** *\*/* @Bean  
 public RedisTemplate<Object, Object> redisTemplate(RedisConnectionFactory redisConnectionFactory) {  
 RedisTemplate<Object, Object> redisTemplate = new RedisTemplate<>();  
 redisTemplate.setConnectionFactory(redisConnectionFactory);  
 Jackson2JsonRedisSerializer jackson2JsonRedisSerializer = new Jackson2JsonRedisSerializer(Object.class);  
  
//解决查询缓存转换异常的问题  
 ObjectMapper om = new ObjectMapper();  
// 指定要序列化的域，field,get和set,以及修饰符范围，ANY是都有包括private和public  
 om.setVisibility(PropertyAccessor.*ALL*, JsonAutoDetect.Visibility.*ANY*);  
// 指定序列化输入的类型，类必须是非final修饰的，final修饰的类，比如String,Integer等会跑出异常  
 om.enableDefaultTyping(ObjectMapper.DefaultTyping.*NON\_FINAL*);  
 jackson2JsonRedisSerializer.setObjectMapper(om);  
  
//序列号key value  
 redisTemplate.setKeySerializer(new StringRedisSerializer());  
 redisTemplate.setValueSerializer(jackson2JsonRedisSerializer);  
 redisTemplate.setHashKeySerializer(new StringRedisSerializer());  
 redisTemplate.setHashValueSerializer(jackson2JsonRedisSerializer);  
  
 redisTemplate.afterPropertiesSet();  
 return redisTemplate;  
 }  
  
 */\*\*  
 \* 设置CacheManager缓存规则  
 \** ***@param*** *factory  
 \** ***@return*** *\*/* @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;  
 }  
}

1. 启动redis开始使用

往redis里面存值加注解：@Cacheable(value = "dict", keyGenerator = "keyGenerator")

删除redis里面的缓存加注解：@CacheEvict(value = "dict", allEntries = true)

1. 还可以使用RedisTemplate来操作redis

RedisTemplate.opsForValue().get(“key”)

redisTemplate.opsForValue().set(“key”, “value”)

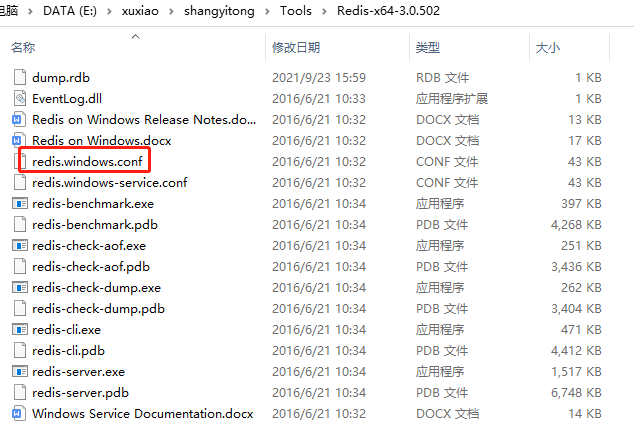
redisTemplate.opsForValue().set(“key”, “value”, 5, TimeUnit.SECONDS)

Window上安装redis

1. 下载redis（window版本）

<https://github.com/microsoftarchive/redis/tags>

1. 进入到redis的配置文件



1. 启动redis命令

redis-server redis.windows.conf