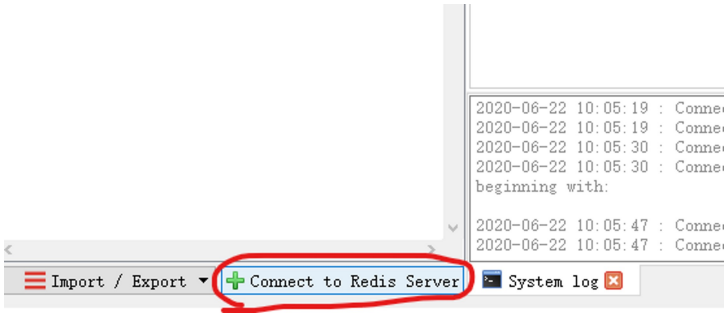
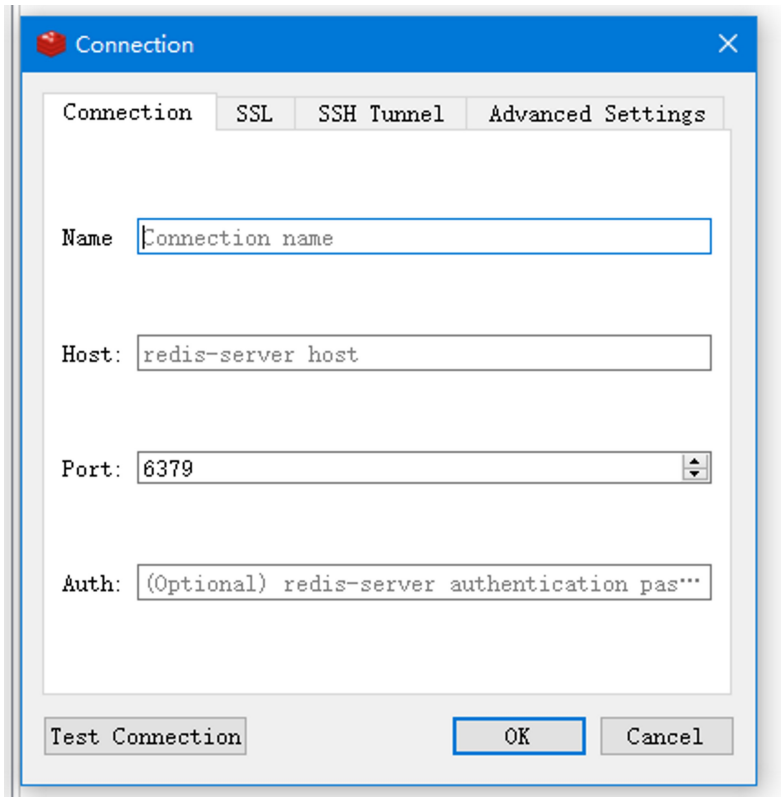
14、整合redis

2020年6月20日 10:58

redis是一个开源（BSD许可）的，内存中的数据结构存储系统，可以用作数据库、缓存和消息中间件。

- 常用命令：中文官网、菜鸟教程
- 1、docker安装redis
  - 1、拉取镜像
    docker pull redis
  - 2、启动redis
    docker run -d -p 6379:6379 --name myredis 235592615444
- 2、使用可视化工具连接至redis
  - 1、如果是阿里云服务器先去安全组开放6379端口
  - 2、打开RedisDesktop
    - 

    - 

- 3、引入redis的starter

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

- 4、配置redis

```
spring.redis.host=39.101.170.233
```

- 5、初步使用

- 1）操作k-v都是字符串使用StringRedisTemplate

  String、List、Set、Hash[散列]、ZSet[有序集合]

  stringRedisTemplate.opsForValue()[String]

  stringRedisTemplate.opsForList()[list]

  stringRedisTemplate.opsForHash()[Hash]

  stringRedisTemplate.opsForSet()[Set]

  stringRedisTemplate.opsForZSet()[ZSet]

```
@Autowired
StringRedisTemplate stringRedisTemplate;
@Test
  public void myTest01(){
      //给redis中保存数据
      stringRedisTemplate.opsForValue().append("msg","hello");
      String msg = stringRedisTemplate.opsForValue().get("msg");
      System.out.println(msg);
      stringRedisTemplate.opsForList().leftPush("mylist","1");
      stringRedisTemplate.opsForList().leftPush("mylist","1");
  }
```

- 2）k-v都是对象RedisTemplate

```
@Autowired
RedisTemplate redisTemplate;//k-v都是对象
@Test
public void myTest02(){
      Employee byId = employeeMapper.getById(1);
      redisTemplate.opsForValue().set("emp-01",byId);
     }
```

- 3）默认保存对象，使用jdk序列化机制，序列化后的数据保存到redis中
  - 1、将数据以json方式保存
    - （1）自己将对象转为json（不推荐）
    - （2）redisTemplate默认的序列化规则；改变默认的序列化规则；

      编写MyRedisConfig

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.redis.connection.RedisConnectionFactory;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.data.redis.serializer.Jackson2JsonRedisSerializer;
@Configuration
public class MyRedisConfig {
      @Bean
      public RedisTemplate<Object, Employee>
empRedisTemplate(RedisConnectionFactory redisConnectionFactory){
            RedisTemplate<Object,Employee> template = new RedisTemplate<>();
            template.setConnectionFactory(redisConnectionFactory);
            Jackson2JsonRedisSerializer<Employee> ser = new
Jackson2JsonRedisSerializer<Employee>(Employee.class);
            template.setDefaultSerializer(ser);
```

```
            return template;
        }
    }
    测试
    @Test
    public void myTest02(){
        Employee byId = employeeMapper.getById(1);
        employeeRedisTemplate.opsForValue().set("emp-01",byId);
    }
```

- 6、测试缓存

  原理CacheManager==Cache缓存组件来实际给缓存中存取数据

  - 1）引入redis的starter，容器中保存的是RedisCacheManager
  - 2）RedisCacheManager帮我们创建RedisCache来作为缓存组件；RedisCache通过操作redis缓存
  - 3）默认保存数据k-v都是obj；利用序列化保存；如何保存json
    - 1、引入redis的starter，CacheManager变为RedisCacheManager；
    - 2、默认创建的RedisCacheManager操作redis的时候使用的是RedisTemplate<Object,Object>
    - 3、RedisTemplate<Object,Object>是默认使用jdk的序列化机制
  - 4）自定义CacheManager

```
@Bean
public CacheManager cacheManager(RedisConnectionFactory factory) {
    RedisCacheConfiguration cacheConfiguration =
                    RedisCacheConfiguration.defaultCacheConfig()
                            .entryTtl(Duration.ofDays(1))
                            .disableCachingNullValues()
                            .serializeValuesWith(RedisSerializationContext.SerializationPair
                            .fromSerializer(new GenericJackson2JsonRedisSerializer()));
    return RedisCacheManager.builder(factory).cacheDefaults(cacheConfiguration).build();
}
```