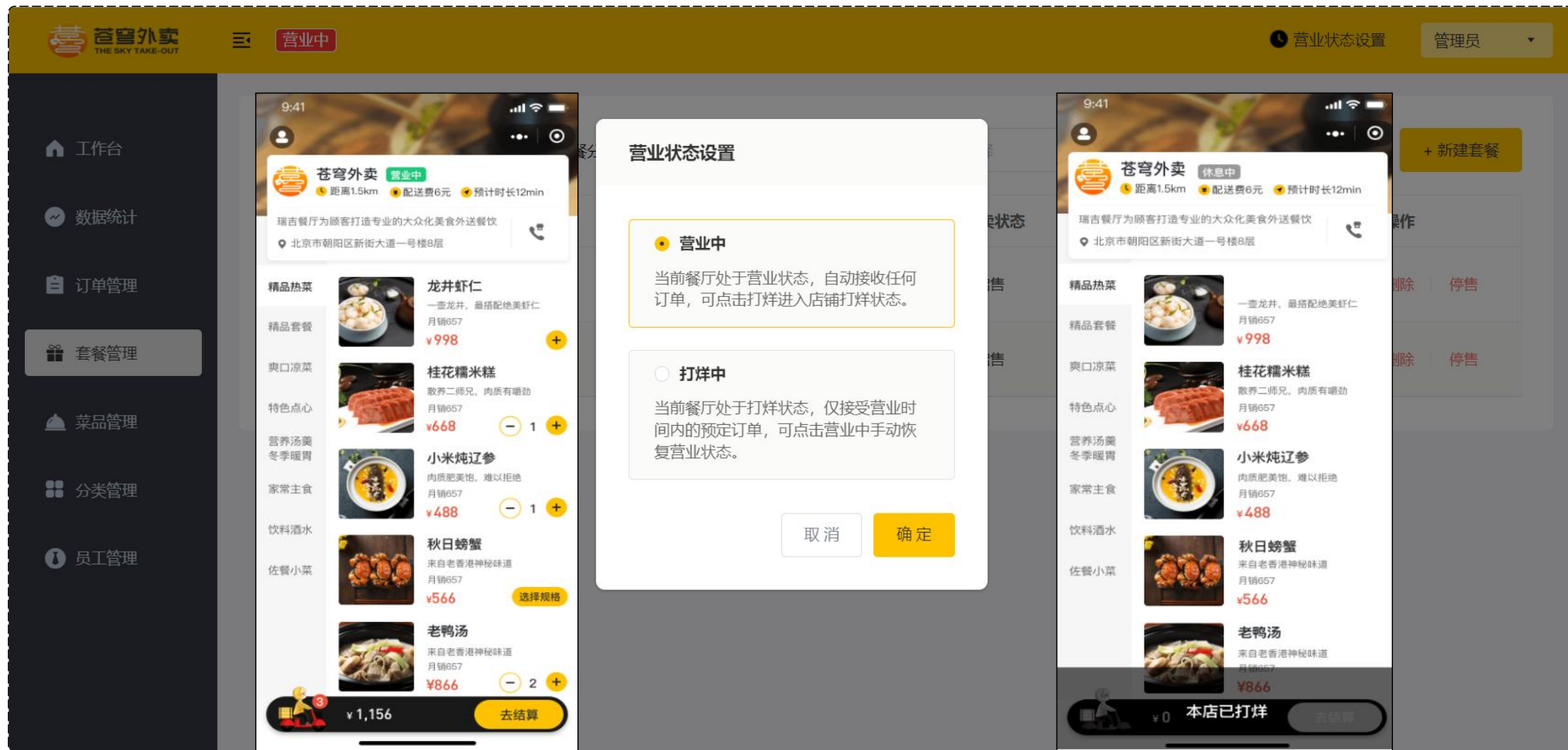


店铺营业状态设置



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌





目录

Contents

- ◆ Redis入门
- ◆ Redis数据类型
- ◆ Redis常用命令
- ◆ 在Java中操作Redis
- ◆ 店铺营业状态设置



Redis入门

- Redis简介
- Redis下载与安装
- Redis服务启动与停止

Redis简介

Redis是一个基于**内存**的 key-value 结构数据库。

- 基于内存存储，读写性能高
- 适合存储热点数据（热点商品、资讯、新闻）
- 企业应用广泛

key	value
id	101
name	小智
city	北京

官网: <https://redis.io>

中文网: <https://www.redis.net.cn/>





Redis入门

- Redis简介
- Redis下载与安装
- Redis服务启动与停止

Redis下载与安装

Redis安装包分为 Windows 版和 Linux 版：

- Windows版下载地址：<https://github.com/microsoftarchive/redis/releases>
- Linux版下载地址：<https://download.redis.io/releases/>














redis-4.0.0.tar.gz



Redis-x64-3.2.100.zip

Redis下载与安装

Redis的Windows版属于绿色软件，直接解压即可使用，解压后目录结构如下：

-  EventLog.dll
-  Redis on Windows Release Notes.docx
-  Redis on Windows.docx
-  redis.windows.conf **Redis配置文件**
-  redis.windows-service.conf
-  redis-benchmark.exe
-  redis-benchmark.pdb
-  redis-check-aof.exe
-  redis-check-aof.pdb
-  redis-cli.exe **Redis客户端**
-  redis-cli.pdb
-  redis-server.exe **Redis服务端**
-  redis-server.pdb
-  Windows Service Documentation.docx



Redis入门

- Redis简介
- Redis下载与安装
- Redis服务启动与停止

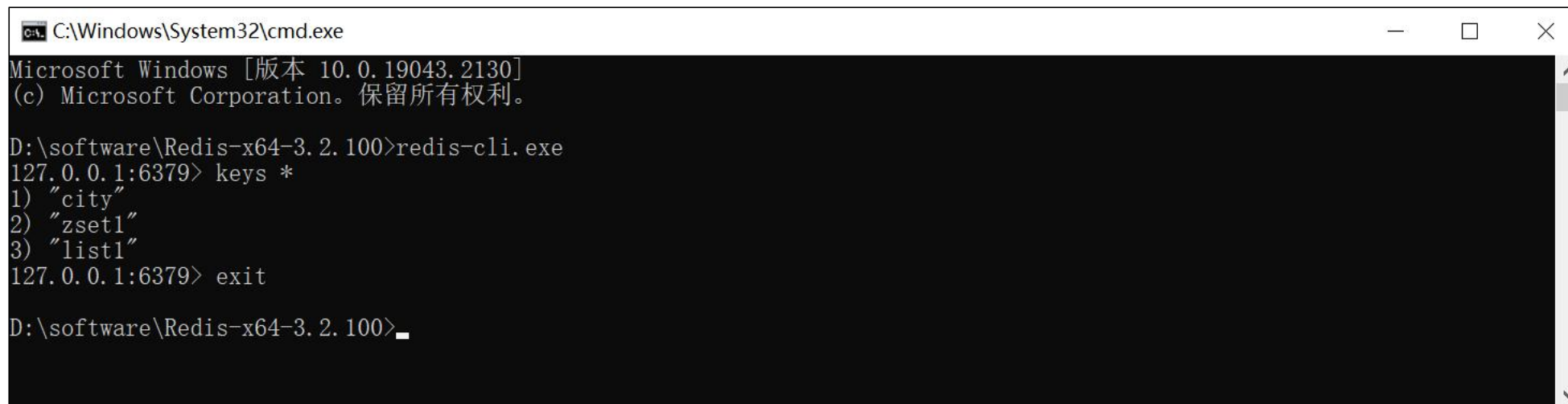


服务启动命令: `redis-server.exe redis.windows.conf`



Redis服务启动与停止

客户端连接命令：**redis-cli.exe**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19043.2130]
(c) Microsoft Corporation. 保留所有权利。

D:\software\Redis-x64-3.2.100>redis-cli.exe
127.0.0.1:6379> keys *
1) "city"
2) "zset1"
3) "list1"
127.0.0.1:6379> exit

D:\software\Redis-x64-3.2.100>_
```

通过redis-cli.exe命令默认连接的是本地的redis服务，并且使用默认6379端口。也可以通过指定如下参数连接：

- h ip地址
- p 端口号
- a 密码（如果需要）

Redis服务启动与停止

设置Redis服务密码，修改redis.windows.conf

```
requirepass 123456
```

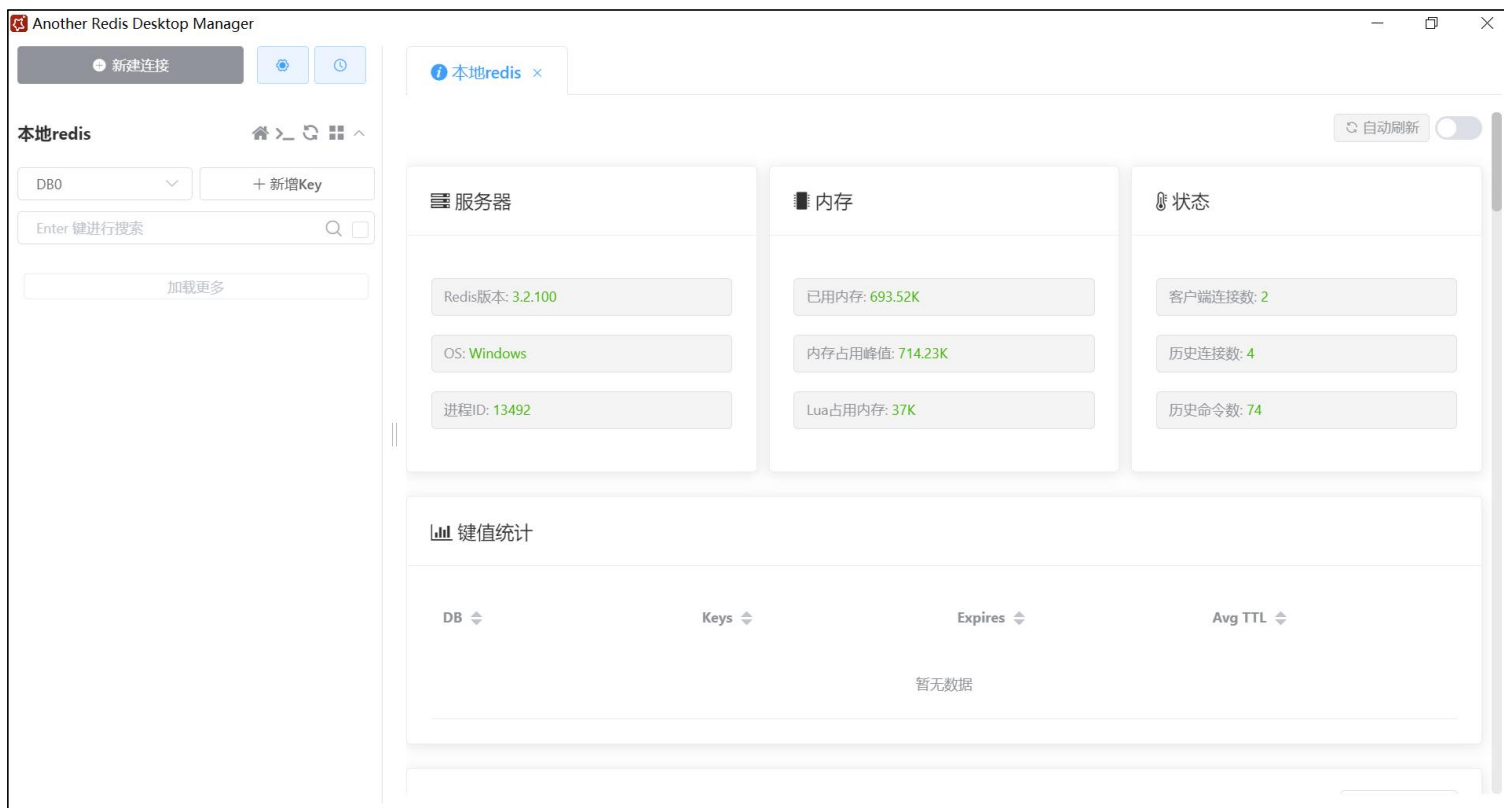
注意：

- 修改密码后需要重启Redis服务才能生效
- Redis配置文件中 # 表示注释

Redis服务启动与停止

Redis客户端图形工具：

 Another-Redis-Desktop-Manager.1.5.5.exe





目录

Contents

- ◆ Redis入门
- ◆ Redis数据类型
- ◆ Redis常用命令
- ◆ 在Java中操作Redis
- ◆ 店铺营业状态设置



Redis数据类型

- 5种常用数据类型介绍
- 各种数据类型的特点

5种常用数据类型介绍

Redis存储的是key-value结构的数据，其中key是字符串类型，value有5种常用的数据类型：

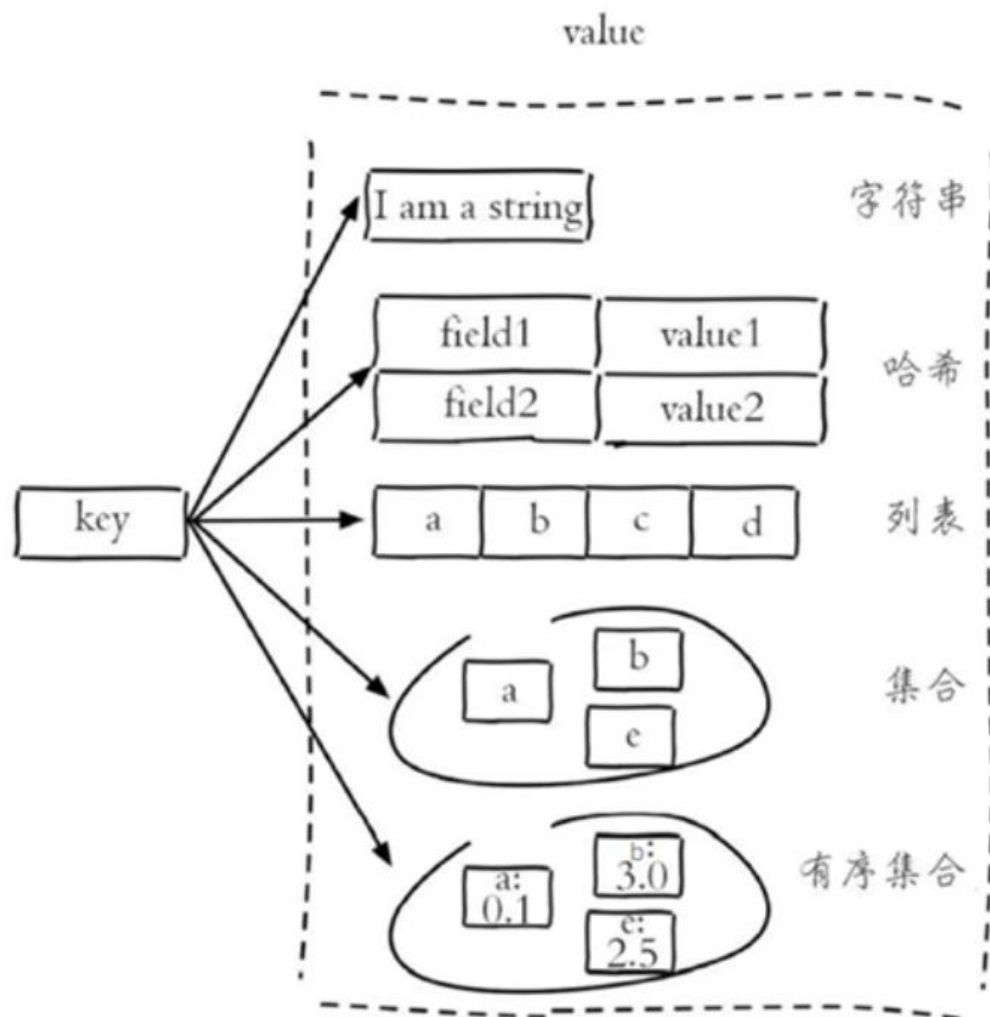
- 字符串 string
- 哈希 hash
- 列表 list
- 集合 set
- 有序集合 sorted set / zset



Redis数据类型

- 5种常用数据类型介绍
- 各种数据类型的特点

各种数据类型的特点



- 字符串(string): 普通字符串, Redis中最简单的数据类型
- 哈希(hash): 也叫散列, 类似于Java中的HashMap结构
- 列表(list): 按照插入顺序排序, 可以有重复元素, 类似于Java中的LinkedList
- 集合(set): 无序集合, 没有重复元素, 类似于Java中的HashSet
- 有序集合(sorted set / zset): 集合中每个元素关联一个分数(score), 根据分数升序排序, 没有重复元素



目录

Contents

- ◆ Redis入门
- ◆ Redis数据类型
- ◆ Redis常用命令
- ◆ 在Java中操作Redis
- ◆ 店铺营业状态设置



Redis常用命令

- 字符串操作命令
- 哈希操作命令
- 列表操作命令
- 集合操作命令
- 有序集合操作命令
- 通用命令

字符串操作命令

Redis 字符串类型常用命令：

- SET key value 设置指定key的值
- GET key 获取指定key的值
- SETEX key seconds value 设置指定key的值，并将 key 的过期时间设为 seconds 秒
- SETNX key value 只有在 key 不存在时设置 key 的值



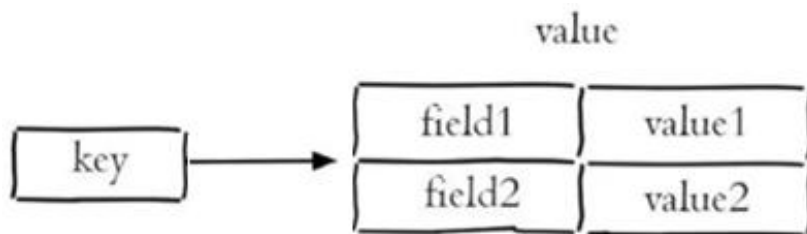
Redis常用命令

- 字符串操作命令
- 哈希操作命令
- 列表操作命令
- 集合操作命令
- 有序集合操作命令
- 通用命令

哈希操作命令

Redis hash 是一个string类型的 field 和 value 的映射表，hash特别适合用于存储对象，常用命令：

- HSET key field value 将哈希表 key 中的字段 field 的值设为 value
- HGET key field 获取存储在哈希表中指定字段的值
- HDEL key field 删除存储在哈希表中的指定字段
- HKEYS key 获取哈希表中所有字段
- HVALS key 获取哈希表中所有值





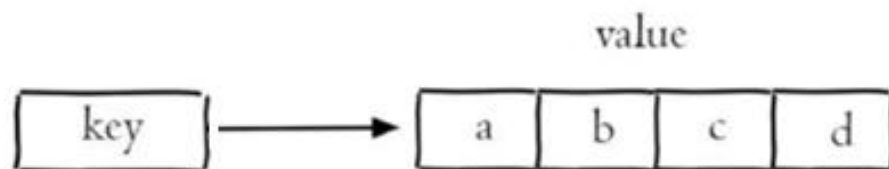
Redis常用命令

- 字符串操作命令
- 哈希操作命令
- 列表操作命令
- 集合操作命令
- 有序集合操作命令
- 通用命令

列表操作命令

Redis 列表是简单的字符串列表，按照插入顺序排序，常用命令：

- LPUSH key value1 [value2] 将一个或多个值插入到列表头部(左边)
- LRANGE key start stop 获取列表指定范围内的元素
- RPOP key 移除并获取列表最后一个元素(右边)
- LLEN key 获取列表长度





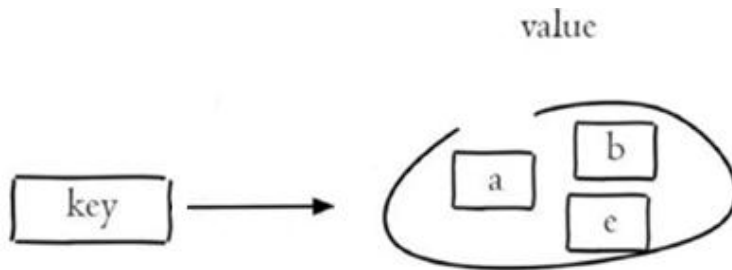
Redis常用命令

- 字符串操作命令
- 哈希操作命令
- 列表操作命令
- 集合操作命令
- 有序集合操作命令
- 通用命令

集合操作命令

Redis set 是string类型的无序集合。集合成员是唯一的，集合中不能出现重复的数据，常用命令：

- SADD key member1 [member2] 向集合添加一个或多个成员
- SMEMBERS key 返回集合中的所有成员
- SCARD key 获取集合的成员数
- SINTER key1 [key2] 返回给定所有集合的交集
- SUNION key1 [key2] 返回所有给定集合的并集
- SREM key member1 [member2] 删除集合中一个或多个成员





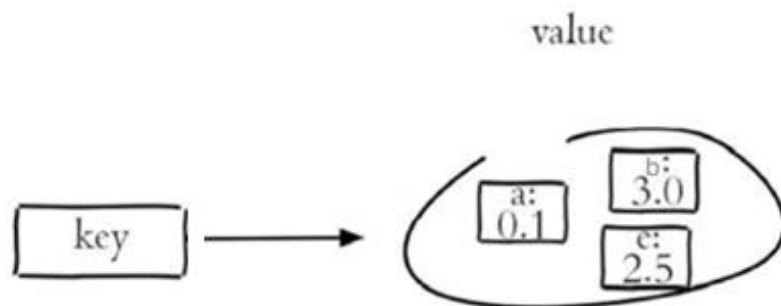
Redis常用命令

- 字符串操作命令
- 哈希操作命令
- 列表操作命令
- 集合操作命令
- 有序集合操作命令
- 通用命令

有序集合操作命令

Redis有序集合是string类型元素的集合，且不允许有重复成员。每个元素都会关联一个double类型的分数。常用命令：

- `ZADD key score1 member1 [score2 member2]` 向有序集合添加一个或多个成员
- `ZRANGE key start stop [WITHSCORES]` 通过索引区间返回有序集合中指定区间内的成员
- `ZINCRBY key increment member` 有序集合中对指定成员的分数加上增量 increment
- `ZREM key member [member ...]` 移除有序集合中的一个或多个成员





Redis常用命令

- 字符串操作命令
- 哈希操作命令
- 列表操作命令
- 集合操作命令
- 有序集合操作命令
- 通用命令

通用命令

Redis的通用命令是不分数据类型的，都可以使用的命令：

- KEYS pattern 查找所有符合给定模式(pattern)的 key
- EXISTS key 检查给定 key 是否存在
- TYPE key 返回 key 所储存的值的类型
- DEL key 该命令用于在 key 存在是删除 key



目录

Contents

- ◆ Redis入门
- ◆ Redis数据类型
- ◆ Redis常用命令
- ◆ 在Java中操作Redis
- ◆ 店铺营业状态设置



在Java中操作Redis

- Redis的Java客户端
- Spring Data Redis使用方式

Redis的Java客户端

Redis 的 Java 客户端很多，常用的几种：

- Jedis
- Lettuce
- Spring Data Redis

Spring Data Redis 是 Spring 的一部分，对 Redis 底层开发包进行了高度封装。

在 Spring 项目中，可以使用Spring Data Redis来简化操作。



在Java中操作Redis

- Redis的Java客户端
- Spring Data Redis使用方式

Spring Data Redis使用方式

操作步骤：

- ① 导入Spring Data Redis 的maven坐标
- ② 配置Redis数据源
- ③ 编写配置类，创建RedisTemplate对象
- ④ 通过RedisTemplate对象操作Redis

```
@Configuration
@Slf4j
public class RedisConfiguration {

    @Bean
    public RedisTemplate redisTemplate(RedisConnectionFactory redisConnectionFactory){
        log.info("开始创建redis模板类...");
        RedisTemplate redisTemplate = new RedisTemplate();
        // 设置Key的序列化器，默认为JdkSerializationRedisSerializer
        redisTemplate.setKeySerializer(new StringRedisSerializer());
        redisTemplate.setConnectionFactory(redisConnectionFactory);
        return redisTemplate;
    }
}
```

Spring Data Redis使用方式

RedisTemplate 针对大量api进行了归类封装,将同一数据类型的操作封装为对应的Operation接口，具体分类如下：

- ValueOperations： string数据操作
- SetOperations： set类型数据操作
- ZSetOperations： zset类型数据操作
- HashOperations： hash类型的数据操作
- ListOperations： list类型的数据操作



目录

Contents

- ◆ Redis入门
- ◆ Redis数据类型
- ◆ Redis常用命令
- ◆ 在Java中操作Redis
- ◆ 店铺营业状态设置

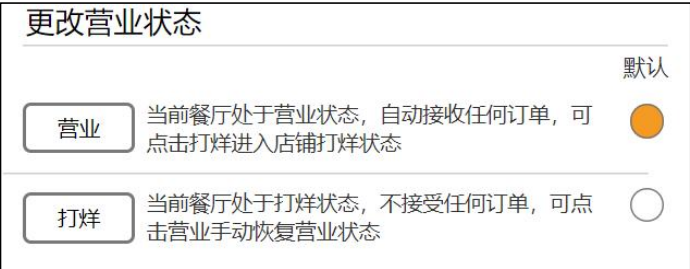


店铺营业状态设置

- 需求分析和设计
- 代码开发
- 功能测试

需求分析和设计

产品原型:



状态	状态说明
营业	客户可在小程序下单点餐
打烊	客户无法下单点餐

需求分析和设计

接口设计:

- 设置营业状态
- 管理端查询营业状态
- 用户端查询营业状态

本项目约定:

- **管理端** 发出的请求，统一使用
- **用户端** 发出的请求，统一使用

基本信息

Path: /user/shop/status

Method: GET

接口描述:

请求参数

返回数据

名称	类型	是否必须	默认值	备注
code	integer	必须		
data	integer	必须		店铺状态: 1为营业, 0为打烊
msg	string	非必须		

返回数据

名称	类型	是否必须	默认值	备注	其他信息
code	integer	必须			format: int32
data	string	非必须			
msg	string	非必须			

基本信息

Path: /admin/shop/status

Method: GET

接口描述:

请求参数

返回数据

名称	类型	是否必须	默认值	备注
code	integer	必须		
data	integer	必须		店铺营业状态: 1为营业, 0为打烊
msg	string	非必须		

需求分析和设计

营业状态数据存储方式：基于Redis的字符串来进行存储

key	value
SHOP_STATUS	1

约定：1表示营业 0表示打烊



店铺营业状态设置

- 需求分析和设计
- 代码开发
- 功能测试

代码开发

```
@RestController("adminShopController")
@RequestMapping("/admin/shop")
@Api(tags = "店铺操作相关接口")
@Slf4j
public class ShopController {

    public static final String KEY = "SHOP_STATUS";

    @Autowired
    private RedisTemplate redisTemplate;

    /**
     * 设置店铺营业状态
     *
     * @param status
     * @return
     */
    @PutMapping("/{status}")
    @ApiOperation("设置店铺营业状态")
    public Result<String> setStatus(@PathVariable Integer status) {
        log.info("设置营业状态为: {}", status == 1 ? "营业中" : "打烊中");
        redisTemplate.opsForValue().set(KEY, status);
        return Result.success();
    }
}
```

代码开发

```
/**
 * 查询店铺营业状态
 *
 * @return
 */
@GetMapping("/status")
@ApiOperation("查询店铺营业状态")
public Result<Integer> getStatus() {
    Integer status = (Integer) redisTemplate.opsForValue().get(KEY);
    log.info("查询店铺营业状态为: {}", status == 1 ? "营业中" : "打烊中");
    return Result.success(status);
}
```

代码开发

```
@RestController("userShopController")
@RequestMapping("/user/shop")
@Api(tags = "店铺操作相关接口")
@Slf4j
public class ShopController {

    public static final String KEY = "SHOP_STATUS";

    @Autowired
    private RedisTemplate redisTemplate;

    /**
     * 查询店铺营业状态
     *
     * @return
     */
    @GetMapping("/status")
    @ApiOperation("查询店铺营业状态")
    public Result<Integer> getStatus() {
        Integer status = (Integer) redisTemplate.opsForValue().get(KEY);
        log.info("查询店铺营业状态为: {}", status == 1 ? "营业中" : "打烊中");
        return Result.success(status);
    }
}
```



店铺营业状态设置

- 需求分析和设计
- 代码开发
- 功能测试

功能测试

可以通过如下方式进行测试：

- swagger接口文档测试
- 前后端联调测试



传智教育旗下高端IT教育品牌