

redis Clients

Java SpringBoot New

- Introduction
- Strings
- Lists
- Sets**
- ZSets
- Hashes
- Streams
- Common Keys
- Pipelining
- Pub/Sub
- Master/Replica
- Sentinel
- Cluster
-
- Auto Config
- Manual Config
- Load Balancing (readFrom)
- RedisTemplate
- Connection Pool & Thread
- Async Spring & Lettuce
- DB select
- Spring Multi Data Source
- Lettuce Multi Data Source
- Spring Project Create
- Spring Project Eclipse
- Spring Project IntelliJ
-
- Spring Session Standalone
- Spring Session MasterRepli
- Spring Session Sentinel
- Spring Session Cluster

Java Lettuce(Spring)New

Java Lettuce(Plain)

Java Jedis

Java Redisson

C Hiredis

C# StackExchange

PHP PhpRedis

PHP Predis

Redis Admin & Monitoring Tool

Spring Data Redis Sets

[레디스 개발자 교육 신청](#)[레디스 정기점검/기술지원
Redis Technical Support](#)[레디스 엔터프라이즈 서버
Redis Enterprise Server](#)

Spring Data Redis Sets

Java Spring Framework를 사용한 레디스 셋(Sets) 명령 사용법입니다.

Sets 소스

Redis03_Set.java

```
package com.redisgate.redis;

import lombok.extern.slf4j.Slf4j;
import org.springframework.data.redis.core.*;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RestController;

import java.util.Set;

@RestController
@Slf4j
public class Redis03_Set {

    private final StringRedisTemplate stringRedisTemplate;
    private final SetOperations<String,String> setOperations;

    public Redis03_Set(StringRedisTemplate stringRedisTemplate) {
        this.stringRedisTemplate = stringRedisTemplate;
        this.setOperations = stringRedisTemplate.opsForSet();
    }
    // 여기에 각 명령(메서드) 별 소스가 들어갑니다.
}
```

각 명령(메서드) 별 표시

SADD

```
// 예제 1) SADD: 집합에 데이터를 추가
// SADD key member [member ...]
// http://localhost:8080/sadd/myset1
@GetMapping("/sadd/{key}")
public String sadd(@PathVariable("key") String key) {
    String msg = "예제 1) SADD(add) -> ";
    Long result = 0L;
    result = setOperations.add(key, "member1", "member2", "member3");
    System.out.println(msg+result);
    String[] members = {"member4", "member5", "member6"}; // 배열로 입력
    result = setOperations.add(key, members);
    System.out.println(msg+result);
    return msg+result;
}
```

SREM

```
// 예제 3) SREM: 데이터 삭제
// SREM key member [member ...]
// http://localhost:8080/srem/myset1
@GetMapping("/srem/{key}")
public String srem(@PathVariable("key") String key) {
    String msg = "예제 3) SREM(remove) -> ";
    Long result = setOperations.remove(key, "member1");
    System.out.println(msg+result);
    return msg+result;
}
```

SMEMBERS

```
// 예제 2) SMEMBERS: 데이터 조회
// SMEMBERS key
// http://localhost:8080/smembers/myset1
@GetMapping("/smembers/{key}")
public String smembers(@PathVariable("key") String key) {
    String msg = "예제 2) SMEMBERS(members) -> ";
    Set<String> result = setOperations.members(key);
    System.out.println(msg+result);
    return msg+result;
}
```

SCARD

```
// 예제 4) SCARD: 데이터 개수 조회
// SCARD key
// http://localhost:8080/scard/myset1
@GetMapping("/scard/{key}")
public String scard(@PathVariable("key") String key) {
    String msg = "예제 4) SCARD(size) -> ";
    Long result = setOperations.size(key);
    System.out.println(msg+result);
    return msg+result;
}
```

SUNION

```

// 예제 5) SUNION: 합집합을 구함,
// SINTER(교집합,intersect)/SDIFF(차집합,difference)
// SUNION key [key ...]
// > sadd myset2 member10 member11 member12    -> 테스트 데이터 입력
// http://localhost:8080/sunion/myset1:myset2
@GetMapping("/sunion/{key}")
public String sunion(@PathVariable("key") String key) {
    String msg = "예제 5) SUNION(union) -> ";
    String[] keys = key.split(":");
    Set<String> result = setOperations.union(keys[0], keys[1]);
    System.out.println(msg+result);
    return msg+result;
}

```

SUNIONSTORE

```

// 예제 6) SUNIONSTORE: 합집합을 구해서 새로운 집합(키)에 저장,
// SINTERSTORE(교집합,intersectAndStore)/SDIFFSTORE(차집합,differenceAndStore)
// UNIONSTORE destination_key source_key [source_key ...]
// http://localhost:8080/sunionstore/myset3
@GetMapping("/sunionstore/{key}")
public String sunionstore(@PathVariable("key") String key) {
    String msg = "예제 6) SUNIONSTORE(unionAndStore) -> ";
    // Long unionAndStore(K key, K otherKey, K destKey); -> dest key가 마지막이다.
    Long result = setOperations.unionAndStore("myset1","myset2", key);
    System.out.println(msg+result);
    return msg+result;
}

```

SISMEMBER

```

// 예제 7) SISMEMBER: 집합에 데이터(멤버)가 존재하는지 확인, SMISMEMBER(여러 개)
// SISMEMBER key member
// http://localhost:8080/sismember/myset1:member2    -> true
// http://localhost:8080/sismember/myset1:member9    -> false
@GetMapping("/sismember/{key}")
public String sismember(@PathVariable("key") String key) {
    String msg = "예제 7) SISMEMBER(isMember) -> ";
    String[] members = key.split(":");
    Boolean result = setOperations.isMember(members[0],members[1]);
    System.out.println(msg+result);
    return msg+result;
}

```

SPOP

```

// 예제 8) SPOP: 집합에서 무작위로 데이터(멤버)를 가져옴
// SPOP key [count]
// http://localhost:8080/spop/myset1
@GetMapping("/spop/{key}")
public String spop(@PathVariable("key") String key) {
    String msg = "예제 8) SPOP(pop) -> ";
    String result = setOperations.pop(key);
    System.out.println(msg+result);
    return msg+result;
}

```

SSCAN

```
// 예제 9) SSCAN: 일정 단위 개수 만큼씩 조회
// SSCAN key cursor [MATCH pattern] [COUNT count]
// http://localhost:8080/sscan/myset5, myset5:*, myset5:member, myset5:value
@GetMapping("/sscan/{key}")
public String sscan(@PathVariable("key") String key) {
    String msg = "예제 9) SSCAN(scan) -> ";
    // key와 pattern을 나눈다. pattern을 입력하지 않았을 경우를 고려
    String[] keyPattern = key.split(":");
    String key1 = null;
    String pattern = null;
    if (keyPattern.length == 1) {
        key1 = keyPattern[0];
        pattern = "*";
    } else {
        key1 = keyPattern[0];
        pattern = keyPattern[1];
    }
    // 테스트 데이터 입력 (키가 없을 경우에만 데이터를 입력)
    Long card = setOperations.size(key1);
    if (card == 0) {
```

<< Lists

Sets

ZSets >>



redisgate@gmail.com



02.503.2235



서울시 강남구 강남대로 342 역삼빌딩 5층 (역삼동) 우 06242

Copyright © 2014-2024 redisGate
All right reserved