《软件架构与中间件》实验指导书

——实验四：数据缓存架构的设计与实现

- 实验 Redis 使用

1.1 基本开发环境准备

- 安装 IntelliJ IDEA，并注册学生账户以获得免费使用权。
- 安装 redis
  - option1 直接安装 redis
  - option2 使用 docker 安装 redis

option1 直接安装 redis

redis for Windows Current Version: 3.0.503 (June 28, 2016) 下载地址

1 解压 redis64-latest.zip 到任意文件夹，如 C:\redis

2 启动 redis-server：

```
$ cd C:\redis
$ redis-server.exe redis.windows.conf
```

3 使用 redis-cli：

```
$ cd c:\redis
$ redis-cli.exe
127.0.0.1:6379> set foo bar
OK
127.0.0.1:6379> get foo
"bar"
```

redis for Linux/Mac latest 下载地址

1 下载后解压、编译：

```
$ tar xzf redis-5.0.4.tar.gz
$ cd redis-5.0.4
$ make
```

此时可执行文件在`redis-5.0.4/src` 目录下

2 启动 redis-server：

```
$ src/redis-server
```

3 使用 redis-cli：

```
$ src/redis-cli
redis> set foo bar
OK
redis> get foo
"bar"
```

option2 使用 docker 安装 redis

1 安装 docker

docker for Windows 下载地址 - 要求 64 位 windows 10 专业版或企业版

docker for Mac 下载地址 - 要求 Mac OS Sierra 10.12 以上

docker Linux 安装命令：

```
$ curl -fsSL get.docker.com -o get-docker.sh
$ sudo sh get-docker.sh --mirror Aliyun
```

2 安装 redis

```
$ docker pull redis
$ docker run --name my-redis -p 6379:6379 redis:latest
```

此时 redis-server 已启动

3 使用 redis-cli：

```
$ docker exec -it my-redis bash
$ redis-cli
127.0.0.1:6379> set foo bar
OK
127.0.0.1:6379> get foo
"bar"
```

1.2 redis-cli 的使用

- 数据类型：string

```
# 赋值 set [key] [value]
127.0.0.1:6379> set name zhangsan
OK

# 取值 get [key]
127.0.0.1:6379> get name
"zhangsan"

# 删除 del [key]
127.0.0.1:6379> del name
(integer) 1
127.0.0.1:6379> get name
(nil)
```

- 数据类型：hash

hash 可以存储多个键值对之间的映射

```
# 赋值
# hset [key] [field] [value]
# hmset [key] [field1] [value1] [field2] [value2] ...
# 取值
# hget [key] [field]
# hmget [key] [field1] [field2] ...
127.0.0.1:6379> hset myhash name zhangsan
(integer) 1
127.0.0.1:6379> hget myhash name
```

```
"zhangsan"
127.0.0.1:6379> hmset myhash name zhangsan age 18 class 2
OK
127.0.0.1:6379> hmget myhash name age class
1) "zhangsan"
2) "18"
3) "2"

# 删除 hdel [key] [field]
127.0.0.1:6379> hdel myhash name
(integer) 1
127.0.0.1:6379> hget myhash name
(nil)
```

- 数据类型：list

list 的顺序是按照插入的顺序，可以在头部跟尾部插入数据

```
# 两端添加
# lpush [key] [value1] [value2] ...
# rpush [key] [value1] [value2] ...
# 查看
# lrange [key] [start_index] [stop_index]
127.0.0.1:6379> lpush mylist a b c
(integer) 3
127.0.0.1:6379> lpush mylist 1 2 3
(integer) 6
127.0.0.1:6379> lrange mylist 0 -1
1) "3"
2) "2"
3) "1"
4) "c"
5) "b"
6) "a"
127.0.0.1:6379> rpush mylist d e f
(integer) 9
127.0.0.1:6379> lrange mylist 0 -1
1) "3"
2) "2"
3) "1"
4) "c"
5) "b"
6) "a"
7) "d"
```

```
8) "e"
9) "f"

# 指定位置添加 lset [key] [index] [value]
127.0.0.1:6379> lset mylist 3 x
OK
127.0.0.1:6379> lrange mylist 0 -1
1) "3"
2) "2"
3) "1"
4) "x"
5) "b"
6) "a"
7) "d"
8) "e"
9) "f"

# 两端弹出
# lpop [key]
# rpop [key]
127.0.0.1:6379> lpop mylist
"3"
127.0.0.1:6379> rpop mylist
"f"
```

- 数据类型：set

set 中不允许出现重复的元素，没有顺序

```
# 添加 sadd [key] [member1] [member2] ...
# 删除 srem [key] [member1] [member2] ...
# 查看 smembers [key]
127.0.0.1:6379> sadd myset a b c 1 2 3
(integer) 6
127.0.0.1:6379> sadd myset a
(integer) 0
127.0.0.1:6379> srem myset a b c
(integer) 3
127.0.0.1:6379> smembers myset
1) "1"
2) "3"
3) "2"
```

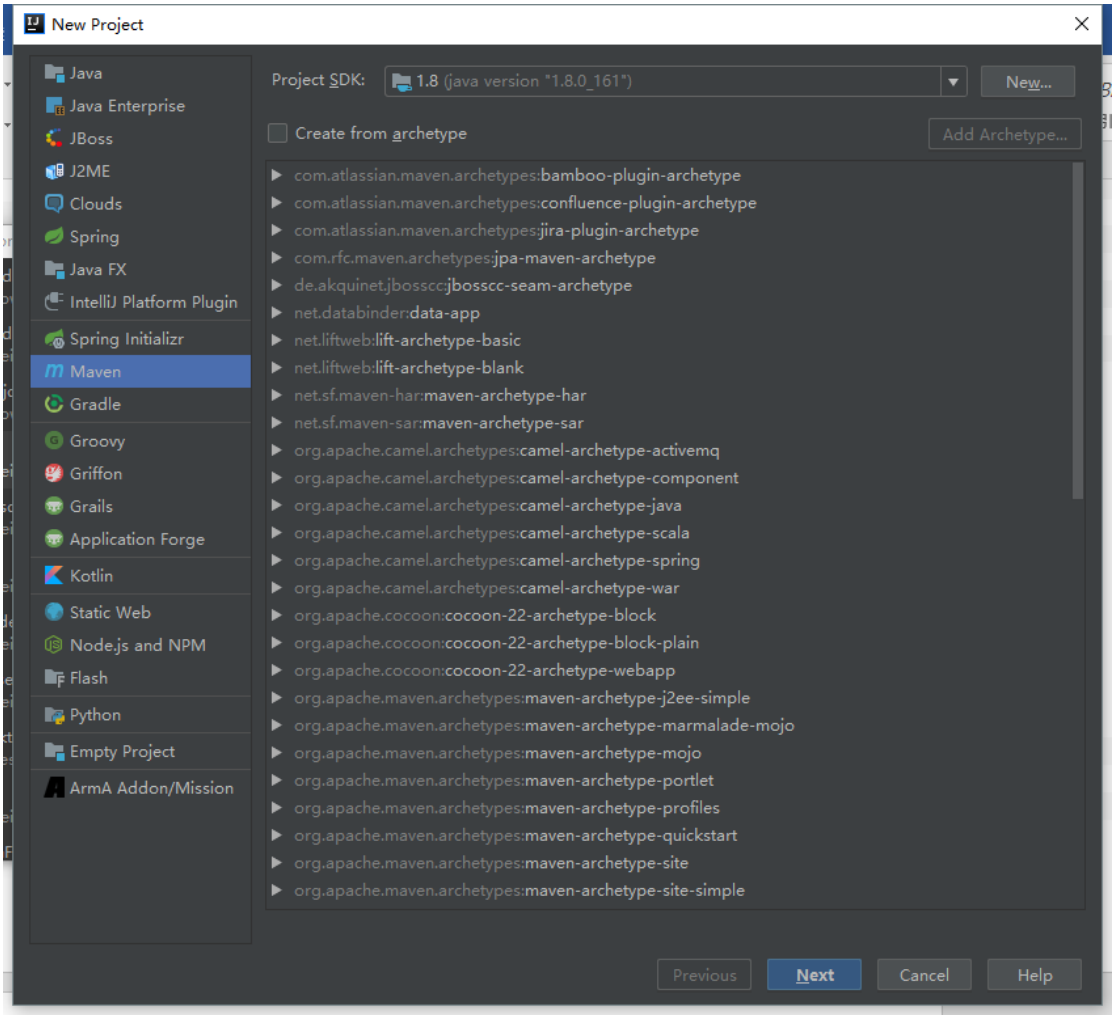
- 数据类型：sorted set

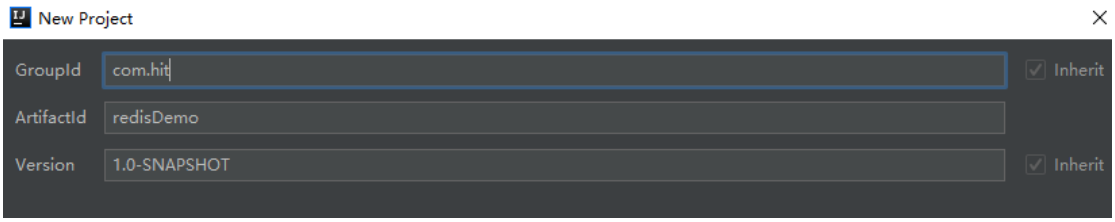
sorted set 有顺序，从小到大排序

```
# 添加 zadd [key] [score1] [member1] [score2] [member2] ...
127.0.0.1:6379> zadd mysset 10 zhangsan 30 lisi 20 wangwu
(integer) 3
# 查看 zrange [key] [start_index] [end_index] <withscores>
127.0.0.1:6379> zrange mysset 0 -1
1) "zhangsan"
2) "wangwu"
3) "lisi"
127.0.0.1:6379> zrange mysset 0 -1 withscores
1) "zhangsan"
2) "10"
3) "wangwu"
4) "20"
5) "lisi"
6) "30"
# 查看 score: zscore [key] [member]
127.0.0.1:6379> zscore mysset zhangsan
"10"
# 删除 zrem [key] [member1] [member2] ...
127.0.0.1:6379> zrem mysset zhangsan
(integer) 1
```

1.3 redis java client: Jedis 的使用

- 使用 idea 创建 maven 项目





- 修改 pom.xml 添加 jedis 依赖，修改后的文件内容如下：

```
1.  <?xml version="1.0" encoding="UTF-8"?>
2.  <project xmlns="http://maven.apache.org/POM/4.0.0"
3.           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4.           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.
    apache.org/xsd/maven-4.0.0.xsd">
5.      <modelVersion>4.0.0</modelVersion>
6.
7.      <groupId>com.hit</groupId>
8.      <artifactId>redisDemo</artifactId>
```

```
    <version>1.0-SNAPSHOT</version>

    <dependencies>
        <dependency>
            <groupId>redis.clients</groupId>
            <artifactId>jedis</artifactId>
            <version>3.0.1</version>
        </dependency>
    </dependencies>

</project>
```

- 新建文件/src/main/java/Demo.java，文件内容如下：

```
import redis.clients.jedis.Jedis;

public class Demo {
    public static void main(String[] args) {
        Jedis jedis = new Jedis("localhost", 6379, 100000);
        jedis.set("foo", "bar");
        System.out.println(jedis.get("foo"));
    }
}
```

运行 Demo.main()：

输出：**bar**