# Redis

### 学习目标

* 了解redis的常用5种数据类型及其操作
* 了解redis的持久化方案
* 掌握使用jedis操作redis
* 掌握使用redis充当缓存案例

### 课程内容

* redis数据结构
* jedis操作redis
* redis案例

### 第一堂课

本节知识点

1.1 今日内容

1.2 redis概述

1.3 redis下载和安装

视频时间

25分17秒

本节目标

了解redis

学会下载和安装redis

#### 1.1 今日内容

##### 1.1.1知识概述

本节课主要梳理今天的学习内容，知道今天主要讲解的内容即可。

##### 1.1.2视频详情



##### 1.1.3总结与补充

今日内容：

1. redis数据结构

2. jedis操作redis

3. redis案例

##### 1.1.4课堂提问与练习

无

##### 1.1.5习题答案

无

##### 1.1.6练习答案

无

##### 1.1.7视频缺陷

无

##### 1.1.8视频扩展

无

#### 1.2 redis概述

##### 1.2.1知识概述

Redis是一款高性能的非关系型的nosql的数据库。目前市场使用率非常高，了解什么是非关系型以及它与关系型数据库的异同。

##### 1.2.2视频详情



##### 1.2.3总结与补充

无

##### 1.2.4课堂提问与练习

无

##### 1.2.5习题答案

无

##### 1.2.6练习答案

无

##### 1.2.7视频缺陷

无

##### 1.2.8视频扩展

无

#### 1.3 redis下载&安装

##### 1.3.1知识概述

学会下载和安装redis

##### 1.3.2视频详情



##### 1.3.3总结与补充

安装redis

启动redis

##### 1.3.4课堂提问与练习

无

##### 1.3.5习题答案

无

##### 1.3.6 练习答案

无

##### 1.3.7 视频缺陷

无

##### 1.3.8 视频扩展

无

### 第二堂课

本节知识点：

2.1 redis常用数据结构介绍

2.2 redis命令操作string&hash

2.3 redis命令操作list

2.4 redis命令操作set&sortedset

2.5 redis命令操作通用命令

视频时间：

23分27秒

本节目标 ：

掌握redis的五种基本数据类型

掌握redis五种数据类型的基本命令操作

#### 2.1 redis常用数据结构介绍

##### 2.1.1知识概述

redis的数据结构：

\*redis存储的是：key,value格式的数据，其中key都是字符串，value有5种不同的数据结构

\* value的数据结构：

1) 字符串类型 string

2) 哈希类型 hash ： map格式

3) 列表类型 list ： linkedlist格式。支持重复元素

4) 集合类型 set ： 不允许重复元素

5) 有序集合类型 sortedset：不允许重复元素，且元素有顺序

##### 2.1.2视频详情



##### 2.1.3总结与补充

redis的数据结构：

\*redis存储的是：key,value格式的数据，其中key都是字符串，value有5种不同的数据结构

\* value的数据结构：

1) 字符串类型 string

2) 哈希类型 hash ： map格式

3) 列表类型 list ： linkedlist格式。支持重复元素

4) 集合类型 set ： 不允许重复元素

5) 有序集合类型 sortedset：不允许重复元素，且元素有顺序

##### 2.1.4课堂提问与练习

无

##### 2.1.5习题答案

无

##### 2.1.6 练习答案

无

##### 2.1.7 视频缺陷

无

##### 2.1.8 视频扩展

无

#### 2.2 redis命令操作string&hash

##### 2.2.1知识概述

字符串类型 string

1. 存储： set key value

127.0.0.1:6379> set username zhangsan

OK

2. 获取： get key

127.0.0.1:6379> get username

"zhangsan"

3. 删除： del key

127.0.0.1:6379> del age

(integer) 12.2.2

哈希类型 hash

1. 存储： hset key field value

127.0.0.1:6379> hset myhash username lisi

(integer) 1

127.0.0.1:6379> hset myhash password 123

(integer) 1

2. 获取：

\* hget key field: 获取指定的field对应的值

127.0.0.1:6379> hget myhash username

"lisi"

\* hgetall key：获取所有的field和value

127.0.0.1:6379> hgetall myhash

1) "username"

2) "lisi"

3) "password"

4) "123"

3. 删除： hdel key field

127.0.0.1:6379> hdel myhash username

(integer) 1

##### 2.2.2视频详情



##### 2.2.3总结与补充

字符串类型 string

1. 存储： set key value

127.0.0.1:6379> set username zhangsan

OK

2. 获取： get key

127.0.0.1:6379> get username

"zhangsan"

3. 删除： del key

127.0.0.1:6379> del age

(integer) 12.2.2

哈希类型 hash

1. 存储： hset key field value

127.0.0.1:6379> hset myhash username lisi

(integer) 1

127.0.0.1:6379> hset myhash password 123

(integer) 1

2. 获取：

\* hget key field: 获取指定的field对应的值

127.0.0.1:6379> hget myhash username

"lisi"

\* hgetall key：获取所有的field和value

127.0.0.1:6379> hgetall myhash

1) "username"

2) "lisi"

3) "password"

4) "123"

3. 删除： hdel key field

127.0.0.1:6379> hdel myhash username

(integer) 1

##### 2.2.4课堂提问与练习

无

##### 2.2.5习题答案

无

##### 2.2.6 练习答案

无

##### 2.2.7 视频缺陷

无

##### 2.2.8 视频扩展

无

#### 2.3 redis命令操作list

##### 2.3.1知识概述

列表类型 list:可以添加一个元素到列表的头部（左边）或者尾部（右边）

1. 添加：

1. lpush key value: 将元素加入列表左表

2. rpush key value：将元素加入列表右边

127.0.0.1:6379> lpush myList a

(integer) 1

127.0.0.1:6379> lpush myList b

(integer) 2

127.0.0.1:6379> rpush myList c

(integer) 3

2. 获取：

\* lrange key start end ：范围获取

127.0.0.1:6379> lrange myList 0 -1

1) "b"

2) "a"

3) "c"

3. 删除：

\* lpop key： 删除列表最左边的元素，并将元素返回

\* rpop key： 删除列表最右边的元素，并将元素返回

##### 2.3.2视频详情



##### 2.3.3总结与补充

列表类型 list:可以添加一个元素到列表的头部（左边）或者尾部（右边）

1. 添加：

1. lpush key value: 将元素加入列表左表

2. rpush key value：将元素加入列表右边

127.0.0.1:6379> lpush myList a

(integer) 1

127.0.0.1:6379> lpush myList b

(integer) 2

127.0.0.1:6379> rpush myList c

(integer) 3

2. 获取：

\* lrange key start end ：范围获取

127.0.0.1:6379> lrange myList 0 -1

1) "b"

2) "a"

3) "c"

3. 删除：

\* lpop key： 删除列表最左边的元素，并将元素返回

\* rpop key： 删除列表最右边的元素，并将元素返回

##### 2.3.4课堂提问与练习

无

##### 2.3.5习题答案

无

##### 2.3.6 练习答案

无

##### 2.3.7 视频缺陷

无

##### 2.3.8 视频扩展

无

#### 2.4 redis命令操作set&sortedset

##### 2.4.1知识概述

集合类型 set ： 不允许重复元素

1. 存储：sadd key value

127.0.0.1:6379> sadd myset a

(integer) 1

127.0.0.1:6379> sadd myset a

(integer) 0

2. 获取：smembers key:获取set集合中所有元素

127.0.0.1:6379> smembers myset

1) "a"

3. 删除：srem key value:删除set集合中的某个元素

127.0.0.1:6379> srem myset a

(integer) 1

6. 有序集合类型 sortedset：不允许重复元素，且元素有顺序.每个元素都会关联一个double类型的分数。redis

正是通过分数来为集合中的成员进行从小到大的排序。

1. 存储：zadd key score value

127.0.0.1:6379> zadd mysort 60 zhangsan

(integer) 1

127.0.0.1:6379> zadd mysort 50 lisi

(integer) 1

127.0.0.1:6379> zadd mysort 80 wangwu

(integer) 1

2. 获取：zrange key start end [withscores]

127.0.0.1:6379> zrange mysort 0 -1

1) "lisi"

2) "zhangsan"

3) "wangwu"

127.0.0.1:6379> zrange mysort 0 -1 withscores

1) "zhangsan"

2) "60"

3) "wangwu"

4) "80"

5) "lisi"

6) "500"

3. 删除：zrem key value

127.0.0.1:6379> zrem mysort lisi

(integer) 1

##### 2.4.2视频详情



##### 2.4.3总结与补充

集合类型 set ： 不允许重复元素

1. 存储：sadd key value

127.0.0.1:6379> sadd myset a

(integer) 1

127.0.0.1:6379> sadd myset a

(integer) 0

2. 获取：smembers key:获取set集合中所有元素

127.0.0.1:6379> smembers myset

1) "a"

3. 删除：srem key value:删除set集合中的某个元素

127.0.0.1:6379> srem myset a

(integer) 1

6. 有序集合类型 sortedset：不允许重复元素，且元素有顺序.每个元素都会关联一个double类型的分数。redis

正是通过分数来为集合中的成员进行从小到大的排序。

1. 存储：zadd key score value

127.0.0.1:6379> zadd mysort 60 zhangsan

(integer) 1

127.0.0.1:6379> zadd mysort 50 lisi

(integer) 1

127.0.0.1:6379> zadd mysort 80 wangwu

(integer) 1

2. 获取：zrange key start end [withscores]

127.0.0.1:6379> zrange mysort 0 -1

1) "lisi"

2) "zhangsan"

3) "wangwu"

127.0.0.1:6379> zrange mysort 0 -1 withscores

1) "zhangsan"

2) "60"

3) "wangwu"

4) "80"

5) "lisi"

6) "500"

3. 删除：zrem key value

127.0.0.1:6379> zrem mysort lisi

(integer) 1

##### 2.4.4课堂提问与练习

无

##### 2.4.5习题答案

无

##### 2.4.6 练习答案

无

##### 2.4.7 视频缺陷

无

##### 2.4.8 视频扩展

无

#### 2.5 redis命令操作-通用命令

##### 2.5.1知识概述

通用命令

1. keys \* : 查询所有的键

2. type key ： 获取键对应的value的类型

3. del key：删除指定的key value

##### 2.5.2视频详情



##### 2.5.3总结与补充

通用命令

1. keys \* : 查询所有的键

2. type key ： 获取键对应的value的类型

3. del key：删除指定的key value

##### 2.5.4课堂提问与练习

无

##### 2.5.5习题答案

无

##### 2.5.6 练习答案

无

##### 2.5.7 视频缺陷

无

##### 2.5.8 视频扩展

无

### 第三堂课

本节知识点：

3.1 redis\_持久化\_RDB

3.2 redis\_持久化\_AOF

视频时间：

10分43秒

本节目标 ：

掌握redis的两种持久化方案

#### 3.1 redis持久化-RDB

##### 3.1.1知识概述

1. redis是一个内存数据库，当redis服务器重启，获取电脑重启，数据会丢失，我们可以将redis内存中的数据

持久化保存到硬盘的文件中。

2. redis持久化机制：

1. RDB：默认方式，不需要进行配置，默认就使用这种机制

\* 在一定的间隔时间中，检测key的变化情况，然后持久化数据

1. 编辑redis.windwos.conf文件

# after 900 sec (15 min) if at least 1 key changed

save 900 1

# after 300 sec (5 min) if at least 10 keys changed

save 300 10

# after 60 sec if at least 10000 keys changed

save 60 10000

2. 重新启动redis服务器，并指定配置文件名称

D:\JavaWeb2018\day23\_redis\资料\redis\windows-64\redis-2.8.9>redis-server.exe

redis.windows.conf

##### 3.1.2视频详情



##### 3.1.3总结与补充

1. redis是一个内存数据库，当redis服务器重启，获取电脑重启，数据会丢失，我们可以将redis内存中的数据

持久化保存到硬盘的文件中。

2. redis持久化机制：

1. RDB：默认方式，不需要进行配置，默认就使用这种机制

\* 在一定的间隔时间中，检测key的变化情况，然后持久化数据

1. 编辑redis.windwos.conf文件

# after 900 sec (15 min) if at least 1 key changed

save 900 1

# after 300 sec (5 min) if at least 10 keys changed

save 300 10

# after 60 sec if at least 10000 keys changed

save 60 10000

2. 重新启动redis服务器，并指定配置文件名称

D:\JavaWeb2018\day23\_redis\资料\redis\windows-64\redis-2.8.9>redis-server.exe

redis.windows.conf

##### 3.1.4课堂提问与练习

无

##### 3.1.5习题答案

无

##### 3.1.6练习答案

无

##### 3.1.7视频缺陷

无

##### 3.1.8视频扩展

无

#### 3.2 redis持久化-AOF

##### 3.2.1知识概述

AOF：日志记录的方式，可以记录每一条命令的操作。可以每一次命令操作后，持久化数据

1. 编辑redis.windwos.conf文件

appendonly no（关闭aof） --> appendonly yes （开启aof）

# appendfsync always ： 每一次操作都进行持久化

appendfsync everysec ： 每隔一秒进行一次持久化

# appendfsync no ： 不进行持久化

##### 3.2.2视频详情



##### 3.2.3总结与补充

AOF：日志记录的方式，可以记录每一条命令的操作。可以每一次命令操作后，持久化数据

1. 编辑redis.windwos.conf文件

appendonly no（关闭aof） --> appendonly yes （开启aof）

# appendfsync always ： 每一次操作都进行持久化

appendfsync everysec ： 每隔一秒进行一次持久化

# appendfsync no ： 不进行持久化

##### 3.2.4课堂提问与练习

无

##### 3.2.5习题答案

无

##### 练习答案

无

##### 3.2.7视频缺陷

无

##### 3.2.8视频扩展

无

### 第四堂课

本节知识点：

4.1 Jedis\_快速入门

4.2 Jedis\_操作string

4.3 Jedis\_操作hash

4.4 Jedis\_操作list

4.5 Jedis\_操作set&sortedset

视频时间

23分47秒

本节目标 ：

掌握使用jedis操作reids

#### 4.1 Jedis\_快速入门

##### 4.1.1知识概述

\* Jedis: 一款java操作redis数据库的工具.

\* 使用步骤：

1. 下载jedis的jar包

2. 使用

//1. 获取连接

Jedis jedis = new Jedis("localhost",6379);

//2. 操作

jedis.set("username","zhangsan");

//3. 关闭连接

jedis.close();

##### 4.1.2视频详情



##### 4.1.3总结与补充

\* Jedis: 一款java操作redis数据库的工具.

\* 使用步骤：

1. 下载jedis的jar包

2. 使用

//1. 获取连接

Jedis jedis = new Jedis("localhost",6379);

//2. 操作

jedis.set("username","zhangsan");

//3. 关闭连接

jedis.close();

##### 4.1.4课堂提问与练习

无

##### 4.1.5习题答案

无

##### 练习答案

无

##### 视频缺陷

无

##### 视频扩展

无

#### 4.2 Jedis\_操作string

##### 4.2.1知识概述

Jedis操作string

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

//存储

jedis.set("username","zhangsan");

//获取

String username = jedis.get("username");

System.out.println(username);

//可以使用setex()方法存储可以指定过期时间的 key value

jedis.setex("activecode",20,"hehe");//将activecode：hehe键值对存入redis，并且20秒后自

动删除该键值对

//3. 关闭连接

jedis.close();

##### 4.2.2视频详情



##### 4.2.3总结与补充

Jedis操作string

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

//存储

jedis.set("username","zhangsan");

//获取

String username = jedis.get("username");

System.out.println(username);

//可以使用setex()方法存储可以指定过期时间的 key value

jedis.setex("activecode",20,"hehe");//将activecode：hehe键值对存入redis，并且20秒后自

动删除该键值对

//3. 关闭连接

jedis.close();

##### 4.2.4课堂提问与练习

无

##### 4.2.5习题答案

无

##### 练习答案

无

##### 4.2.7视频缺陷

无

##### 4.2.8视频扩展

无

#### 4.3 Jedis\_操作hash

##### 4.3.1知识概述

哈希类型 hash ： map格式

hset

hget

hgetAll

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// 存储hash

jedis.hset("user","name","lisi");

jedis.hset("user","age","23");

jedis.hset("user","gender","female");

// 获取hash

String name = jedis.hget("user", "name");

System.out.println(name);

// 获取hash的所有map中的数据

Map<String, String> user = jedis.hgetAll("user");

// keyset

Set<String> keySet = user.keySet();

for (String key : keySet) {

//获取value

String value = user.get(key);

System.out.println(key + ":" + value);

}

//3. 关闭连接

jedis.close();

##### 4.3.2视频详情



##### 4.3.3总结与补充

哈希类型 hash ： map格式

hset

hget

hgetAll

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// 存储hash

jedis.hset("user","name","lisi");

jedis.hset("user","age","23");

jedis.hset("user","gender","female");

// 获取hash

String name = jedis.hget("user", "name");

System.out.println(name);

// 获取hash的所有map中的数据

Map<String, String> user = jedis.hgetAll("user");

// keyset

Set<String> keySet = user.keySet();

for (String key : keySet) {

//获取value

String value = user.get(key);

System.out.println(key + ":" + value);

}

//3. 关闭连接

jedis.close();

##### 4.3.4课堂提问与练习

无

##### 4.3.5习题答案

无

##### 4.3.6练习答案

无

##### 4.3.7视频缺陷

无

##### 4.3.8视频扩展

无

#### 4.4 Jedis\_操作list

##### 4.4.1知识概述

列表类型 list ： linkedlist格式。支持重复元素

lpush / rpush

lpop / rpop

lrange start end : 范围获取

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// list 存储

jedis.lpush("mylist","a","b","c");//从左边存

jedis.rpush("mylist","a","b","c");//从右边存

// list 范围获取

List<String> mylist = jedis.lrange("mylist", 0, -1);

System.out.println(mylist);

// list 弹出

String element1 = jedis.lpop("mylist");//c

System.out.println(element1);

String element2 = jedis.rpop("mylist");//c

System.out.println(element2);

// list 范围获取

List<String> mylist2 = jedis.lrange("mylist", 0, -1);

System.out.println(mylist2);

//3. 关闭连接

jedis.close();

##### 4.4.2视频详情



##### 4.4.3总结与补充

列表类型 list ： linkedlist格式。支持重复元素

lpush / rpush

lpop / rpop

lrange start end : 范围获取

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// list 存储

jedis.lpush("mylist","a","b","c");//从左边存

jedis.rpush("mylist","a","b","c");//从右边存

// list 范围获取

List<String> mylist = jedis.lrange("mylist", 0, -1);

System.out.println(mylist);

// list 弹出

String element1 = jedis.lpop("mylist");//c

System.out.println(element1);

String element2 = jedis.rpop("mylist");//c

System.out.println(element2);

// list 范围获取

List<String> mylist2 = jedis.lrange("mylist", 0, -1);

System.out.println(mylist2);

//3. 关闭连接

jedis.close();

##### 4.4.4课堂提问与练习

无

##### 4.4.5习题答案

无

##### 4.4.6练习答案

无

##### 4.4.7视频缺陷

无

##### 4.4.8视频扩展

无

#### 4.5 Jedis操作set&sortedset

##### 4.5.1知识概述

4）集合类型 set ： 不允许重复元素

sadd

smembers:获取所有元素

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// set 存储

jedis.sadd("myset","java","php","c++");

// set 获取

Set<String> myset = jedis.smembers("myset");

System.out.println(myset);

//3. 关闭连接

jedis.close();

5) 有序集合类型 sortedset：不允许重复元素，且元素有顺序

zadd

zrange

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// sortedset 存储

jedis.zadd("mysortedset",3,"亚瑟");

jedis.zadd("mysortedset",30,"后裔");

jedis.zadd("mysortedset",55,"孙悟空");

// sortedset 获取

Set<String> mysortedset = jedis.zrange("mysortedset", 0, -1);

System.out.println(mysortedset);

//3. 关闭连接

jedis.close();

##### 4.5.2视频详情



##### 4.5.3总结与补充

4）集合类型 set ： 不允许重复元素

sadd

smembers:获取所有元素

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// set 存储

jedis.sadd("myset","java","php","c++");

// set 获取

Set<String> myset = jedis.smembers("myset");

System.out.println(myset);

//3. 关闭连接

jedis.close();

5) 有序集合类型 sortedset：不允许重复元素，且元素有顺序

zadd

zrange

//1. 获取连接

Jedis jedis = new Jedis();//如果使用空参构造，默认值 "localhost",6379端口

//2. 操作

// sortedset 存储

jedis.zadd("mysortedset",3,"亚瑟");

jedis.zadd("mysortedset",30,"后裔");

jedis.zadd("mysortedset",55,"孙悟空");

// sortedset 获取

Set<String> mysortedset = jedis.zrange("mysortedset", 0, -1);

System.out.println(mysortedset);

//3. 关闭连接

jedis.close();

##### 4.5.4课堂提问与练习

无

##### 4.5.5习题答案

无

##### 4.5.6练习答案

无

##### 4.5.7视频缺陷

无

##### 4.5.8视频扩展

无

### 第五堂课

本节知识点：

5.1 Jedis\_连接池

5.2 Jedis\_连接池工具类

视频时间

14分53秒

本节目标 ：

熟练掌握redis连接池和jedis连接池工具类

#### 5.1 Jedis\_连接池

##### 5.1.1知识概述

.使用jedis连接池能优化redis的使用性能

##### 5.1.2视频详情



##### 5.1.3总结与补充

无

##### 5.1.4课堂提问与练习

无

##### 5.1.5习题答案

无

##### 5.1.6练习答案

无

##### 5.1.7视频缺陷

无

##### 5.1.8视频扩展

无

#### 5.2 Jedis\_连接池工具类

##### 5.2.1知识概述

##### 5.2.2视频详情



##### 5.2.3总结与补充

无

##### 5.2.4课堂提问与练习

无

##### 5.2.5习题答案

无

##### 5.2.6练习答案

无

##### 5.2.7视频缺陷

无

##### 5.2.8视频扩展

无

### 第六堂课

本节知识点：

6.1 案例\_分析&环境搭建

6.2 案例\_实现\_查询数据库

6.3 案例\_实现\_redis缓存优化

视频时间

33分40秒

本节目标 ：

熟练掌握使用reids的缓存优化案例

#### 6.1 案例\_分析&环境搭建

##### 6.1.1知识概述

案例需求：

1. 提供index.html页面，页面中有一个省份 下拉列表

2. 当 页面加载完成后 发送ajax请求，加载所有省份

##### 6.1.2视频详情



##### 6.1.3总结与补充

案例需求：

1. 提供index.html页面，页面中有一个省份 下拉列表

2. 当 页面加载完成后 发送ajax请求，加载所有省份

##### 6.1.4课堂提问与练习

无

##### 6.1.5习题答案

无

##### 6.1.6练习答案

无

##### 6.1.7视频缺陷

无

##### 6.1.8视频扩展

无

#### 6.2 案例\_实现\_查询数据库

##### 6.2.1知识概述

##### 6.2.2视频详情



##### 6.2.3总结与补充

无

##### 6.2.4课堂提问与练习

无

##### 6.2..5习题答案

无

##### 6.2.6练习答案

无

##### 6.2.7视频缺陷

无

##### 6.2.8视频扩展

无

#### 6.3 案例\_实现\_redis缓存优化

##### 6.3.1知识概述

注意：使用redis缓存一些不经常发生变化的数据。

\* 数据库的数据一旦发生改变，则需要更新缓存。

\* 数据库的表执行 增删改的相关操作，需要将redis缓存数据情况，再次存入

\* 在service对应的增删改方法中，将redis数据删除。

##### 6.3.2视频详情



##### 6.3.3总结与补充

无

##### 6.3.4课堂提问与练习

无

##### 6.3..5习题答案

无

##### 6.3.6练习答案

无

##### 6.3.7视频缺陷

无

##### 6.3.8 视频扩展

无