commons.data-integrate模块SDK抽取方案

# SDK开放点

本次抽取SDK主要开放的是访问和操作redis的能力，具体开放点如下：

* 新增了操作redis的API

不再将redis连接池直接开放给上层组件，取而代之是新增一套redis的操作的API

取如果有未覆盖到的API，请及时反馈给蒋维佳。

# 业务组件开发

## 去掉如下maven依赖

<dependency>

<groupId>com.huawei.jaguar</groupId>

<artifactId>commons.data-integrate</artifactId>

</dependency>

## 添加新的maven依赖

<dependency>

<groupId>com.huawei.jaguar</groupId>

<artifactId>commons.sdk. data-integrate </artifactId>

</dependency>

<dependency>

<groupId>com.huawei.jaguar</groupId>

<artifactId>commons.sdk. data-integrate.impl</artifactId>

</dependency>

## 代码改造

各组件的封装的redis操作类，统一切换到新的commons.sdk. data-integrate.impl包里面的redis访问接口。

# 缓存API说明

## 初始化操作

### void reload(String connectUrls);

功能说明：使用新的url初始化缓存配置,可用于动态刷新缓存对象类

参数说明：connectUrls格式 IP：PORT**|**IP：PORT**|**IP：PORT

## 单value操作

### <T> T get(String key);

功能说明：获取key对应的缓存对象

参数说明：缓存的key值

### <T> T getSet(String key, T value);

功能说明：将给定 key的值设为 value。并返回 key的旧值(old value)。

参数说明：

Key：缓存的key值

Value：key值对应的value对象

### <T> String set(String key, T value);

功能说明：对象放置到缓存当中

参数说明：

Key：缓存的key值

Value：key值对应的value对象

### <T> String set(String key, int seconds, T value);

功能说明：将对象放置到缓存当中，并指定缓存自动失效时间

参数说明：

Key：缓存的key值

Seconds：失效时间，单位为秒

Value：key值对应的value对象

### <T> long setnx(String key, T value);

功能说明：若给定的 key不存在则set，存在则 setnx不做任何动作

参数说明：

Key：缓存的key值

Seconds：失效时间，单位为秒

Value：key值对应的value对象

### <T> String setex(String key, T value, int expireSeconds);

功能说明：

该函数用于将对象放置到缓存当中，并绑定一个ttl，是一个原子化的操作

如果 key已经存在， SETEX 命令将覆写旧值

如果 key不存在， SETEX 则设置key对应的值

参数说明：

Key：缓存的key值

Value：key值对应的value对象

expireSeconds：失效时间，单位为秒

### long delete(String key);

功能说明：

删除key对应的缓存

参数说明：

Key：缓存的key值

### boolean exists(String key);

功能说明：

判断key对应的缓存存在不存在

参数说明：

Key：缓存的key值

## Hash操作

### <T> T hget(String key, String field);

功能说明：该函数用于获取Hash表类型的缓存值

参数说明：

Key：缓存的key值

Field：缓存的field域值

### <T> Map<String, T> hgetAll(String key);

功能说明：获取哈希表 key中.所有的域和值

参数说明：

Key：缓存的key值

### <T> long hset(String key, String field, T value);

功能说明：

如果 key不存在，一个新的哈希表被创建并进行 HSET操作。

如果域 field已经存在于哈希表中，旧值将被覆盖。

参数说明：

Key：缓存的key值

Field：缓存的field域值

Value：key值对应的value对象

### <T> List<T> hmget(String key, String... fields);

功能说明：

获取返回哈希表 key中,一个或多个给定域的值

参数说明：

Key：缓存的key值

Field：缓存的field域值

### <T> String hmset(String key, Map<String, T> hash);

功能说明：

同时将多个 field-value (域-值)对设置到哈希表 key中

参数说明：

Key：缓存的key值

hash： field-value的对应的Map

### boolean hexists(String key, String field);

功能说明：

查看哈希表 key中，给定域 field是否存在

参数说明：

Key：缓存的key值

Field：缓存的field域值

### long hdel(String key, String... fields);

功能说明：

删除哈希表 key 中的一个或多个指定域，不存在的域将被忽略

参数说明：

Key：缓存的key值

Field：缓存的field域值

## Set操作

### <T> long sadd(String key, T... members);

功能说明：

该函数用于将一个或多个 member对象加入到集合 key当中，已经存在于集合的 member元素将被忽略

参数说明：

Key：缓存的key值

Members：存入缓存中，集合类型的members对象

### <T> Set<T> smembers(String key);

功能说明：

返回集合 key 中的所有成员。不存在的 key被视为空集合

参数说明：

Key：缓存的key值

### <T> boolean sismember(String key, T member);

功能说明：

判断 member元素是否集合 key的成员

参数说明：

Key：缓存的key值

Members：集合类型的members对象

### <T> long srem(String key, T... members);

功能说明：

该函数用于移除集合 key中的一个或多个 member元素，不存在的 member元素会被忽略

参数说明：

Key：缓存的key值

Members：集合类型的members对象

## 自增和自减操作

### Long incr(String key);

功能说明：

缓存中的值加一，要保证缓存中存放的是一个整数

参数说明：

Key：缓存的key值

### Long decr(String key);

功能说明：

缓存中的值减一，要保证缓存中存放的是一个整数

参数说明：

Key：缓存的key值

### Long incrBy(String key, long increment);

功能说明：

缓存中的值加上增量increment，要保证缓存中存放的是一个整数

参数说明：

Key：缓存的key值

Increment：增量

### Long decrBy(String key, long decrement);

功能说明：

缓存中的值减去减量，要保证缓存中存放的是一个整数

参数说明：

Key：缓存的key值

Decrement：减量

## 批量操作(pipeline)

### <T> List<T> batchGet(List<String> keyList);

功能说明：

批量获取缓存数据,对应redis的pipeline操作

参数说明：

keyList：缓存键名列表

### <T> void batchSet(List<CacheItem<T>> cache);

功能说明：

批量保存缓存数据，对应redis的pipeline操作

参数说明：

cache：缓存数据列表

### <T> void batchDelete(List<String> keyList);

功能说明：

批量获取缓存数据,对应redis的pipeline操作

参数说明：

keyList：缓存键名列表

## 获取服务器信息

### List<ServerInfo> getServerInfo();

功能说明：

获取服务器信息，

参数说明：

对于redis来说，

如果是sentinel模式，返回的是当前所有master节点的信息

如果是普通shard模式，返回的是所有redis节点信息

对应于之前老的 List<HostAndPort> getAllRedisInfo();方法

# 数据库API说明

## 功能说明

### 见API文档

API接口抽取自ibatis的com.ibatis.sqlmap.engine.impl.SqlMapClientImpl对象，在SqlMapClientImpl的基础上增加了超时日志的功能

# Spring配置指导

## Redis缓存配置示范

<bean id=*"commons.jedisPoolConfig"* class=*"redis.clients.jedis.JedisPoolConfig"*>

<property name=*"minIdle"* value=*"${commons.redis.pool.minIdle}"* />

<property name=*"maxIdle"* value=*"${commons.redis.pool.maxIdle}"* />

<property name=*"maxTotal"* value=*"${commons.redis.pool.maxTotal}"* />

<property name=*"testOnBorrow"* value=*"${commons.redis.pool.testOnBorrow}"* />

<property name=*"maxWaitMillis"* value=*"${commons.redis.pool.maxWait}"* />

<property name=*"singleKeyTimeOut"* value=*"${commons.redis.pool.singleKeyTimeOut}"* />

<property name=*"mutiKeyTimeOut"* value=*"${commons.redis.pool.mutiKeyTimeOut}"* />

</bean>

<bean id=*"commons.cacheManager"*

class=*"com.huawei.jaguar.commons.sdk.data.integrate.impl.cache.RedisCacheManagerImpl"*

init-method=*"init"*>

<property name=*"jedisPoolConfig"* ref=*"commons.jedisPoolConfig"* />

<property name=*"connectUrls"* value=*"${redis.connect}"* />

<!--

<property name="enable" value="${commons.redis.enable}" />

<property name="throwRedisException" value="${commons.redis.throwRedisException}" />

-->

</bean>

