Popular Methods

- sync ( )
- set ( )
- multi ( )
- get ( )
- syncAndReturnAll ( )
- del ( )
- expire ( )
- exec ( )
- sadd ( )
- incr ( )

View more ...

Related Classes

- org.junit.Before
- java.util.concurrent.TimeUnit
- java.io.UnsupportedEncodingException
- java.util.Calendar
- java.sql.SQLException
- java.net.UnknownHostException
- java.util.concurrent.Callable
- java.util.concurrent.atomic.AtomicLong
- com.fasterxml.jackson.core.JsonProcessingException
- com.google.common.base.Throwables
- com.alibaba.fastjson.JSON
- redis.clients.jedis.Jedis
- com.alibaba.fastjson.JSONObject
- redis.clients.jedis.JedisPool
- com.google.common.primitives.Longs
- com.alibaba.fastjson.JSONArray
- com.alibaba.fastjson.JSONException
- redis.clients.jedis.ShardedJedis
- redis.clients.jedis.exceptions.JedisException
- redis.clients.jedis.exceptions.JedisDataException
- redis.clients.jedis.Tuple
- redis.clients.jedis.Response
- javax.xml.bind.ValidationException
- org.quartz.Calendar
- net.md_5.bungee.api.connection.PendingConnection

Java Code Examples for redis.clients.jedis.Pipeline

The following are top voted examples for showing how to use redis.clients.jedis.Pipeline. These examples are extracted from open source projects. You can vote up the examples you like and your votes will be used in our system to generate

more good examples.

+ Save this class

Example 1

Project: *JRediClients* File: *PipelinedGetSetBenchmark.java* View source code 7 votes

```
public static void main(String[] args) throws UnknownHostException, IOException {
  Jedis jedis = new Jedis(hnp.getHost(), hnp.getPort());
  jedis.connect();
  jedis.auth("foobared");
  jedis.flushAll();

  long begin = Calendar.getInstance().getTimeInMillis();

  Pipeline p = jedis.pipelined();
  for (int n = 0; n <= TOTAL_OPERATIONS; n++) {
    String key = "foo" + n;
    p.set(key, "bar" + n);
    p.get(key);
  }
  p.sync();

  long elapsed = Calendar.getInstance().getTimeInMillis() - begin;

  jedis.disconnect();

  System.out.println(((1000 * 2 * TOTAL_OPERATIONS) / elapsed) + " ops");
}
```

Example 2

Project: *cachecloud* File: *PipeliningTest.java* View source code 7 votes

```
@Test
public void testEvalshaKeyAndArg() {
  String key = "test";
  String arg = "3";
  String script = "redis.call('INCRBY', KEYS[1], ARGV[1]) redis.call('INCRBY', KEYS[1], ARGV[1])";
  String sha1 = jedis.scriptLoad(script);

  assertTrue(jedis.scriptExists(sha1));

  Pipeline p = jedis.pipelined();
  p.set(key, "0");
  Response<Object> result0 = p.evalsha(sha1, Arrays.asList(key), Arrays.asList(arg));
  p.incr(key);
  Response<Object> result1 = p.evalsha(sha1, Arrays.asList(key), Arrays.asList(arg));
  Response<String> result2 = p.get(key);
  p.sync();

  assertNull(result0.get());
  assertNull(result1.get());
  assertEquals("13", result2.get());
}
```

Example 3

Project: *RedisDirectory* File: *JedisPoolStream.java* View source code 6 votes

```
/**
 * Use transactions to delete index file
 *
 * @param fileLengthKey
 * @param fileDataKey
 * @param field
 * @param blockSize
 */
@Override
public void deleteFile(String fileLengthKey, String fileDataKey, String field, long blockSize) {
    Jedis jedis = null;
    try {
        jedis = jedisPool.getResource();
        Pipeline pipelined = jedis.pipelined();
```

```
        //delete file length
        pipelined.hdel(fileLengthKey.getBytes(), field.getBytes());
        //delete file content
        for (int i = 0; i < blockSize; i++) {
            byte[] blockName = getBlockName(field, i);
            pipelined.hdel(fileDataKey.getBytes(), blockName);
        }
        pipelined.sync();
    } finally {
        jedis.close();
    }
}
```

Example 4

Project: *RedisDirectory* File: *JedisPoolStream.java* *View source code* 6 votes

```
@Override
public void rename(String fileLengthKey, String fileDataKey, String oldField, String newField, List<byte[]> values, long
        fileLength) {
    long blockSize = 0;
    Jedis jedis = null;
    try {
        jedis = jedisPool.getResource();
        Pipeline pipelined = jedis.pipelined();
        //add new file length
        pipelined.hset(fileLengthKey.getBytes(), newField.getBytes(), Longs.toByteArray(fileLength));
        //add new file content
        blockSize = getBlockSize(fileLength);
        for (int i = 0; i < blockSize; i++) {
            pipelined.hset(fileDataKey.getBytes(), getBlockName(newField, i), compressFilter(values.get(i)));
        }
        values.clear();
        pipelined.sync();
    } finally {
        jedis.close();
        deleteFile(fileLengthKey, fileDataKey, oldField, blockSize);
    }
}
```

Example 5

Project: *RedisDirectory* File: *JedisPoolStream.java* *View source code* 6 votes

```
@Override
public void saveFile(String fileLengthKey, String fileDataKey, String fileName, List<byte[]> values, long fileLength) {
    Jedis jedis = null;
    try {
        jedis = jedisPool.getResource();
        Pipeline pipelined = jedis.pipelined();
        pipelined.hset(fileLengthKey.getBytes(), fileName.getBytes(), Longs.toByteArray(fileLength));
        Long blockSize = getBlockSize(fileLength);
        for (int i = 0; i < blockSize; i++) {
            pipelined.hset(fileDataKey.getBytes(), getBlockName(fileName, i), compressFilter(values.get(i)));
            if (i % Constants.SYNC_COUNT == 0) {
                pipelined.sync();
                pipelined = jedis.pipelined();
            }
        }
        values.clear();
        pipelined.sync();
    } finally {
        jedis.close();
    }
}
```

Example 6

Project: *RedisDirectory* File: *JedisStream.java* *View source code* 6 votes

```
@Override
public void rename(String fileLengthKey, String fileDataKey, String oldField, String newField, List<byte[]> values, long
        fileLength) {
    Jedis jedis = openJedis();
    Pipeline pipelined = jedis.pipelined();
```

```
    //add new file length
    pipelined.hset(fileLengthKey.getBytes(), newField.getBytes(), Longs.toByteArray(fileLength));
    //add new file content
    Long blockSize = getBlockSize(fileLength);
    for (int i = 0; i < blockSize; i++) {
        pipelined.hset(fileDataKey.getBytes(), getBlockName(newField, i), compressFilter(values.get(i)));
    }
    pipelined.sync();
    jedis.close();
    values.clear();
    deleteFile(fileLengthKey, fileDataKey, oldField, blockSize);
}
```

Example 7

Project: *RedisDirectory* File: *JedisStream.java* *View source code* 6 votes

```
@Override
public void saveFile(String fileLengthKey, String fileDataKey, String fileName, List<byte[]> values, long fileLength) {
    Jedis jedis = openJedis();
    Pipeline pipelined = jedis.pipelined();
    pipelined.hset(fileLengthKey.getBytes(), fileName.getBytes(), Longs.toByteArray(fileLength));
    Long blockSize = getBlockSize(fileLength);
    for (int i = 0; i < blockSize; i++) {
        pipelined.hset(fileDataKey.getBytes(), getBlockName(fileName, i), compressFilter(values.get(i)));
        if (i % Constants.SYNC_COUNT == 0) {
            pipelined.sync();
            pipelined = jedis.pipelined();
        }
    }
    pipelined.sync();
    jedis.close();
    values.clear();
}
```

Example 8

Project: *JInsight* File: *JedisPipelineInstrumentationTest.java* *View source code* 6 votes

```
@Test
public void testSync() throws Exception {
  int rnd = ThreadLocalRandom.current().nextInt(0, presetElements.size());
  String key = presetElementKeys.get(rnd);
  String value = String.valueOf(presetElements.get(key));

  Pipeline pipeline = jedis.pipelined();

  Snapshot snapshot = commandTracker.snapshot();
  Snapshot discardSnapshot = discardTracker.snapshot();
  Snapshot txsnapshot = execTracker.snapshot();
  Snapshot pipelineSnapshot = pipelineTracker.snapshot();
  pipelineSnapshot.increment();
  Response<Long> added = pipeline.sadd(key, value);
  pipeline.sync();
  assertEquals(1, (long) added.get());
  pipelineSnapshot.validate();
  txsnapshot.validate();
  snapshot.validate();
  discardSnapshot.validate();
}
```

Example 9

Project: *JInsight* File: *JedisPipelineInstrumentationTest.java* *View source code* 6 votes

```
@Test
public void testCloseSyncs() throws Exception {
  int rnd = ThreadLocalRandom.current().nextInt(0, presetElements.size());
  String key = presetElementKeys.get(rnd);
  String value = String.valueOf(presetElements.get(key));

  Pipeline pipeline = jedis.pipelined();

  Snapshot snapshot = commandTracker.snapshot();
  Snapshot discardSnapshot = discardTracker.snapshot();
```

```
  Snapshot txsnapshot = execTracker.snapshot();
  Snapshot pipelineSnapshot = pipelineTracker.snapshot();
  pipelineSnapshot.increment();
  Response<Long> added = pipeline.sadd(key, value);
  pipeline.close();
  assertEquals(1, (long) added.get());
  pipelineSnapshot.validate();
  txsnapshot.validate();
  snapshot.validate();
  discardSnapshot.validate();
}
```

Example 10

Project: *gedis* File: *ServerMapCache.java* *View source code* 6 votes

```
@Override
public boolean containsKeys(Collection<K> keys) {
    if (keys == null || keys.size() == 0) {
        return false;
    }
    // 使用 Redis 提供的管道进行批处理，提高效率
    Pipeline pipeline = jedisProxy.pipelined();
    Set<Response<Boolean>> responses = new HashSet<Response<Boolean>>(keys.size());
    for (K key : keys) {
        if (localMapCache != null && localMapCache.containsKey(key)) {
            continue;
        }
        responses.add(pipeline.hexists(SerializeUtil.serialize(getName()), SerializeUtil.serialize(key)));
    }
    if (responses.size() < 1) {
        return true;
    }
    pipeline.sync();
    for (Response<Boolean> response : responses) {
        if (!response.get()) {
            return false;
        }
    }
    return true;
}
```

Example 11

Project: *gedis* File: *ServerMapCache.java* *View source code* 6 votes

```
@Override
public Map<K, V> gets(Collection<K> keys) {
    Map<K, V> result = new HashMap<>(keys.size());
    Pipeline pipeline = jedisProxy.pipelined();
    byte[] keyBytes = SerializeUtil.serialize(getName());
    Map<K, Response<byte[]>> responseMap = new HashMap<>(keys.size());
    for (K key : keys) {
        if (localMapCache != null && localMapCache.containsKey(key)) {
            result.put(key, localMapCache.get(key));
            continue;
        }
        responseMap.put(key, pipeline.hget(keyBytes, SerializeUtil.serialize(key)));
    }
    if (responseMap.size() < 1) {
        return result;
    }
    pipeline.sync();
    for (Map.Entry<K, Response<byte[]>> entry : responseMap.entrySet()) {
        result.put(entry.getKey(), (V) SerializeUtil.deserialize(entry.getValue().get()));
    }
    return result;
}
```

Example 12

Project: *dooo* File: *StatisticsCacheServiceAbstract.java* *View source code* 6 votes

```
private List<Map<String, String>> getAll(final List<String> keys) {
    try {
```

```
        return localCacheOfRedisData.get(keys.hashCode() + "", new Callable<List<Map<String, String>>>() {
            @Override
            public List<Map<String, String>> call() throws Exception {
                return redisConfig.execute(new RedisCallback<List<Map<String, String>>>() {
                    @Override
                    public List<Map<String, String>> call(Jedis jedis) {
                        Pipeline pipeline = jedis.pipelined();
                        for (String key : keys) {
                            pipeline.hgetAll(key);
                        }

                        List result = pipeline.syncAndReturnAll();
                        return result;
                    }
                });
            }
        });
    } catch (Exception e) {
        LOGGER.error(keys.toString(), e);
    }
    return new ArrayList<>();
}
```

Example 13

Project: *analytics-with-rfx* File: *RealtimeTrackingUtil.java* *View source code* 6 votes

```
public static boolean updateKafkaLogEvent(int unixtime, final String event){
        Date date = new Date(unixtime*1000L);
        final String dateStr = DateTimeUtil.formatDate(date ,DATE_FORMAT_PATTERN);
        final String dateHourStr = DateTimeUtil.formatDate(date ,YYYY_MM_DD_HH);

        boolean commited = new RedisCommand<Boolean>(jedisPool) {
                @Override
                protected Boolean build() throws JedisException {
                        String keyD = MONITOR_PREFIX+dateStr;
                        String keyH = MONITOR_PREFIX+dateHourStr;

                        Pipeline p = jedis.pipelined();
                        p.hincrBy(keyD, "e:"+event , 1L);
                        p.expire(keyD, AFTER_4_DAYS);
                        p.hincrBy(keyH, "e:"+event , 1L);
                        p.expire(keyH, AFTER_2_DAYS);
                        p.sync();
                        return true;
                }
        }.execute();

        return commited;
}
```

Example 14

Project: *analytics-with-rfx* File: *RealtimeTrackingUtil.java* *View source code* 6 votes

```
public static boolean updateMonitorEvent(int unixtime, final String event){
        Date date = new Date(unixtime*1000L);
        final String dateStr = DateTimeUtil.formatDate(date ,DATE_FORMAT_PATTERN);
        final String dateHourStr = DateTimeUtil.formatDate(date ,YYYY_MM_DD_HH);

        boolean commited = new RedisCommand<Boolean>(jedisPool) {
                @Override
                protected Boolean build() throws JedisException {
                        String keyD = MONITOR_PREFIX+dateStr;
                        String keyH = MONITOR_PREFIX+dateHourStr;
                        Pipeline p = jedis.pipelined();
                        p.hincrBy(keyD, event , 1L);
                        p.expire(keyD, AFTER_2_DAYS);
                        p.hincrBy(keyH, event , 1L);
                        p.expire(keyH, AFTER_1_DAY);
                        p.sync();
                        return true;
                }
        }.execute();
        return commited;
}
```

Example 15

Project: *analytics-with-rfx* File: *UserRedisUtil.java* *View source code* 6 votes

```
public static boolean addUser(String keyPrefix, int unixTime, String metric, final String uuid) {
    Date date = new Date(unixTime*1000L);
    final String dateStr = DateTimeUtil.formatDate(date, DateTimeUtil.DATE_FORMAT_PATTERN);

    return new RedisCommand<Boolean>(redisAdDataStats) {
        @Override
        protected Boolean build() throws JedisException {
            Pipeline p = jedis.pipelined();
            String keyTotal = keyPrefix+"t";
            String keyDaily = keyPrefix+dateStr+":t";
            String keyHourly = keyPrefix+dateStr+":"+metric;

            p.pfadd(keyTotal, uuid);
            p.pfadd(keyDaily, uuid);
            p.pfadd(keyHourly, uuid);
            p.expire(keyDaily, AFTER_7_DAYS);
            p.expire(keyHourly, AFTER_3_DAYS);

            p.sync();
            return true;
        }
    }.execute();
}
```

Example 16

Project: *RedisDirectory* File: *JedisPoolStream.java* *View source code* 6 votes

```
/**
 * Use transactions to delete index file
 *
 * @param fileLengthKey
 * @param fileDataKey
 * @param field
 * @param blockSize
 */
@Override
public void deleteFile(String fileLengthKey, String fileDataKey, String field, long blockSize) {
    Jedis jedis = null;
    try {
        jedis = jedisPool.getResource();
        Pipeline pipelined = jedis.pipelined();
        //delete file length
        pipelined.hdel(fileLengthKey.getBytes(), field.getBytes());
        //delete file content
        for (int i = 0; i < blockSize; i++) {
            byte[] blockName = getBlockName(field, i);
            pipelined.hdel(fileDataKey.getBytes(), blockName);
        }
        pipelined.sync();
    } finally {
        jedis.close();
    }
}
```

Example 17

Project: *RedisDirectory* File: *JedisPoolStream.java* *View source code* 6 votes

```
@Override
public void rename(String fileLengthKey, String fileDataKey, String oldField, String newField, List<byte[]> values, long
        fileLength) {
    long blockSize = 0;
    Jedis jedis = null;
    try {
        jedis = jedisPool.getResource();
        Pipeline pipelined = jedis.pipelined();
        //add new file length
        pipelined.hset(fileLengthKey.getBytes(), newField.getBytes(), Longs.toByteArray(fileLength));
        //add new file content
        blockSize = getBlockSize(fileLength);
        for (int i = 0; i < blockSize; i++) {
```

```
            pipelined.hset(fileDataKey.getBytes(), getBlockName(newField, i), compressFilter(values.get(i)));
        }
        values.clear();
        pipelined.sync();
    } finally {
        jedis.close();
        deleteFile(fileLengthKey, fileDataKey, oldField, blockSize);
    }
}
```

Example 18

Project: *RedisDirectory* File: *JedisPoolStream.java* *View source code* 6 votes

```
@Override
public void saveFile(String fileLengthKey, String fileDataKey, String fileName, List<byte[]> values, long fileLength) {
    Jedis jedis = null;
    try {
        jedis = jedisPool.getResource();
        Pipeline pipelined = jedis.pipelined();
        pipelined.hset(fileLengthKey.getBytes(), fileName.getBytes(), Longs.toByteArray(fileLength));
        Long blockSize = getBlockSize(fileLength);
        for (int i = 0; i < blockSize; i++) {
            pipelined.hset(fileDataKey.getBytes(), getBlockName(fileName, i), compressFilter(values.get(i)));
            if (i % Constants.SYNC_COUNT == 0) {
                pipelined.sync();
                pipelined = jedis.pipelined();
            }
        }
        values.clear();
        pipelined.sync();
    } finally {
        jedis.close();
    }
}
```

Example 19

Project: *RedisDirectory* File: *JedisStream.java* *View source code* 6 votes

```
@Override
public void rename(String fileLengthKey, String fileDataKey, String oldField, String newField, List<byte[]> values, long
        fileLength) {
    Jedis jedis = openJedis();
    Pipeline pipelined = jedis.pipelined();
    //add new file length
    pipelined.hset(fileLengthKey.getBytes(), newField.getBytes(), Longs.toByteArray(fileLength));
    //add new file content
    Long blockSize = getBlockSize(fileLength);
    for (int i = 0; i < blockSize; i++) {
        pipelined.hset(fileDataKey.getBytes(), getBlockName(newField, i), compressFilter(values.get(i)));
    }
    pipelined.sync();
    jedis.close();
    values.clear();
    deleteFile(fileLengthKey, fileDataKey, oldField, blockSize);
}
```

Example 20

Project: *RedisDirectory* File: *JedisStream.java* *View source code* 6 votes

```
@Override
public void saveFile(String fileLengthKey, String fileDataKey, String fileName, List<byte[]> values, long fileLength) {
    Jedis jedis = openJedis();
    Pipeline pipelined = jedis.pipelined();
    pipelined.hset(fileLengthKey.getBytes(), fileName.getBytes(), Longs.toByteArray(fileLength));
    Long blockSize = getBlockSize(fileLength);
    for (int i = 0; i < blockSize; i++) {
        pipelined.hset(fileDataKey.getBytes(), getBlockName(fileName, i), compressFilter(values.get(i)));
        if (i % Constants.SYNC_COUNT == 0) {
            pipelined.sync();
            pipelined = jedis.pipelined();
        }
    }
```

```
    pipelined.sync();
    jedis.close();
    values.clear();
}
```

Example 21

Project: *Qihua* File: *RedisTransactionTest.java* View source code 6 votes

```
public void pipelineWithTransaction() {
  Jedis jedis = pool.getResource();
  try {
    Pipeline p = jedis.pipelined();
    p.multi();
    for (int i = 0; i < rowCount; i++) {
      String key = RandomStringUtils.randomAlphabetic(8);
      p.set(key, RandomStringUtils.randomNumeric(5));
      p.expire(key, 5 * 60);
    }
    p.exec();
    p.sync();
  } catch (Exception e) {
    pool.returnResource(jedis);
  }
}
```

Example 22

Project: *ReSearch* File: *FullTextFacetedIndex.java* View source code 6 votes

```
public List<Entry> execute() {
    Jedis conn = pool.getResource();
    Pipeline pipe = conn.pipelined();

    String tmpKey = root.execute(pipe);


    pipe.zrevrangeWithScores(tmpKey, query.sort.offset, query.sort.offset + query.sort.limit);

    List<Object> res = pipe.syncAndReturnAll();
    conn.close();


    Set<Tuple> ids = (Set<Tuple>) res.get(res.size() - 1);
    List<Entry> entries = new ArrayList<>(ids.size());
    for (Tuple t : ids) {
        entries.add(new Entry(t.getElement(), t.getScore()));
    }
    return entries;
}
```

Example 23

Project: *ReSearch* File: *FullTextFacetedIndex.java* View source code 6 votes

```
@Override
String execute(Pipeline pipe) {

    if (tokens.size() == 1) {
        return tokenKey(tokens.get(0).text);
    }

    String[] keys = new String[tokens.size()+1];
    keys[0] = makeTmpKey(raw);
    for (int i = 0; i < tokens.size(); i++) {
        keys[i+1] = tokenKey(tokens.get(i).text);
    }

    intersectTokens.execute(pipe, 1, keys);
    pipe.expire(keys[0], DEFAULT_EXPIRATION);
    return keys[0];
}
```

Example 24

Project: *ReSearch* File: *FullTextFacetedIndex.java* *View source code* 6 votes

```
@Override
String execute(Pipeline pipe) {

    // Since the circle query doesn't have pre-defined precision, we need to calculate all the
    // relevant geohashes in our precision manually
    Set<String> hashKeys = getSearchHashes();


    // now let's union them
    String[] keysArr = hashKeys.toArray(new String[hashKeys.size()]);
    double[] scoresArr = new double[hashKeys.size()];//all weights are zero
    String tmpKey = makeTmpKey(keysArr);

    pipe.zunionstore(tmpKey, new ZParams().aggregate(ZParams.Aggregate.MAX)
            .weightsByDouble(scoresArr), keysArr);
    return tmpKey;
}
```

Example 25

Project: *ReSearch* File: *JSONStore.java* *View source code* 6 votes

```
@Override
public void store(Document... docs) {

    Jedis conn = pool.getResource();
    Pipeline pipe = conn.pipelined();
    try {
        for (Document doc : docs) {

            String encoded = gson.toJson(doc);
            pipe.set(key(doc.getId()), encoded);

        }
        pipe.sync();
    } finally {
        conn.close();
    }
}
```

Example 26

Project: *fiat* File: *RedisPermissionsRepository.java* *View source code* 6 votes

```
private Table<String, ResourceType, Response<Map<String, String>>> getAllFromRedis(Set<String> userIds) {
  if (userIds.size() == 0) {
    return HashBasedTable.create();
  }
  try (Jedis jedis = jedisSource.getJedis()) {
    Table<String, ResourceType, Response<Map<String, String>>> responseTable =
        ArrayTable.create(userIds, new ArrayIterator<>(ResourceType.values()));

    Pipeline p = jedis.pipelined();
    for (String userId : userIds) {
      for (ResourceType r : ResourceType.values()) {
        responseTable.put(userId, r, p.hgetAll(userKey(userId, r)));
      }
    }
    p.sync();
    return responseTable;
  } catch (Exception e) {
    log.error("Storage exception reading all entries.", e);
  }
  return null;
}
```

Example 27

Project: *fiat* File: *RedisPermissionsRepository.java* *View source code* 6 votes

```
@Override
public void remove(@NonNull String id) {
  try (Jedis jedis = jedisSource.getJedis()) {
    Map<String, String> userRolesById = jedis.hgetAll(userKey(id, ResourceType.ROLE));

    Pipeline p = jedis.pipelined();

    p.srem(allUsersKey(), id);
    for (String roleName : userRolesById.keySet()) {
      p.srem(roleKey(roleName), id);
    }

    for (ResourceType r : ResourceType.values()) {
      p.del(userKey(id, r));
    }
    p.sync();
  } catch (Exception e) {
    log.error("Storage exception reading " + id + " entry.", e);
  }
}
```

Example 28

Project: *cachecloud* File: *PipelinedGetSetBenchmark.java* *View source code* 6 votes

```
public static void main(String[] args) throws UnknownHostException, IOException {
  Jedis jedis = new Jedis(hnp.getHost(), hnp.getPort());
  jedis.connect();
  jedis.auth("foobared");
  jedis.flushAll();

  long begin = Calendar.getInstance().getTimeInMillis();

  Pipeline p = jedis.pipelined();
  for (int n = 0; n <= TOTAL_OPERATIONS; n++) {
    String key = "foo" + n;
    p.set(key, "bar" + n);
    p.get(key);
  }
  p.sync();

  long elapsed = Calendar.getInstance().getTimeInMillis() - begin;

  jedis.disconnect();

  System.out.println(((1000 * 2 * TOTAL_OPERATIONS) / elapsed) + " ops");
}
```

Example 29

Project: *cachecloud* File: *PipeliningTest.java* *View source code* 6 votes

```
@Test
public void multiWithMassiveRequests() {
  Pipeline p = jedis.pipelined();
  p.multi();

  List<Response<?>> responseList = new ArrayList<Response<?>>();
  for (int i = 0; i < 100000; i++) {
    // any operation should be ok, but shouldn't forget about timeout
    responseList.add(p.setbit("test", 1, true));
  }

  Response<List<Object>> exec = p.exec();
  p.sync();

  // we don't need to check return value
  // if below codes run without throwing Exception, we're ok
  exec.get();

  for (Response<?> resp : responseList) {
    resp.get();
  }
}
```

Example 30

Project: *cachecloud* File: *PipeliningTest.java* View source code 6 votes

```
@Test
public void multiWithSync() {
  jedis.set("foo", "314");
  jedis.set("bar", "foo");
  jedis.set("hello", "world");
  Pipeline p = jedis.pipelined();
  Response<String> r1 = p.get("bar");
  p.multi();
  Response<String> r2 = p.get("foo");
  p.exec();
  Response<String> r3 = p.get("hello");
  p.sync();

  // before multi
  assertEquals("foo", r1.get());
  // It should be readable whether exec's response was built or not
  assertEquals("314", r2.get());
  // after multi
  assertEquals("world", r3.get());
}
```

Example 31

Project: *cachecloud* File: *PipeliningTest.java* View source code 6 votes

```
@Test
public void testEvalKeyAndArg() {
  String key = "test";
  String arg = "3";
  String script = "redis.call('INCRBY', KEYS[1], ARGV[1]) redis.call('INCRBY', KEYS[1], ARGV[1])";

  Pipeline p = jedis.pipelined();
  p.set(key, "0");
  Response<Object> result0 = p.eval(script, Arrays.asList(key), Arrays.asList(arg));
  p.incr(key);
  Response<Object> result1 = p.eval(script, Arrays.asList(key), Arrays.asList(arg));
  Response<String> result2 = p.get(key);
  p.sync();

  assertNull(result0.get());
  assertNull(result1.get());
  assertEquals("13", result2.get());
}
```

Example 32

Project: *cachecloud* File: *PipeliningTest.java* View source code 6 votes

```
@Test
public void testEvalKeyAndArgWithBinary() {
  // binary
  byte[] bKey = SafeEncoder.encode("test");
  byte[] bArg = SafeEncoder.encode("3");
  byte[] bScript = SafeEncoder
      .encode("redis.call('INCRBY', KEYS[1], ARGV[1]) redis.call('INCRBY', KEYS[1], ARGV[1])");

  Pipeline bP = jedis.pipelined();
  bP.set(bKey, SafeEncoder.encode("0"));
  Response<Object> bResult0 = bP.eval(bScript, Arrays.asList(bKey), Arrays.asList(bArg));
  bP.incr(bKey);
  Response<Object> bResult1 = bP.eval(bScript, Arrays.asList(bKey), Arrays.asList(bArg));
  Response<byte[]> bResult2 = bP.get(bKey);
  bP.sync();

  assertNull(bResult0.get());
  assertNull(bResult1.get());
  assertArrayEquals(SafeEncoder.encode("13"), bResult2.get());
}
```

Example 33

Project: *cachecloud* File: *PipeliningTest.java* View source code 6 votes

```
@Test
public void testEvalshaKeyAndArgWithBinary() {
  byte[] bKey = SafeEncoder.encode("test");
  byte[] bArg = SafeEncoder.encode("3");
  String script = "redis.call('INCRBY', KEYS[1], ARGV[1]) redis.call('INCRBY', KEYS[1], ARGV[1])";
  byte[] bScript = SafeEncoder.encode(script);
  byte[] bSha1 = jedis.scriptLoad(bScript);

  assertTrue(jedis.scriptExists(bSha1) == 1);

  Pipeline p = jedis.pipelined();
  p.set(bKey, SafeEncoder.encode("0"));
  Response<Object> result0 = p.evalsha(bSha1, Arrays.asList(bKey), Arrays.asList(bArg));
  p.incr(bKey);
  Response<Object> result1 = p.evalsha(bSha1, Arrays.asList(bKey), Arrays.asList(bArg));
  Response<byte[]> result2 = p.get(bKey);
  p.sync();

  assertNull(result0.get());
  assertNull(result1.get());
  assertArrayEquals(SafeEncoder.encode("13"), result2.get());
}
```

Example 34

Project: *cachecloud* File: *PipeliningTest.java* *View source code* 6 votes

```
@Test
public void testSyncWithNoCommandQueued() {
  // we need to test with fresh instance of Jedis
  Jedis jedis2 = new Jedis(hnp.getHost(), hnp.getPort(), 500);

  Pipeline pipeline = jedis2.pipelined();
  pipeline.sync();

  jedis2.close();

  jedis2 = new Jedis(hnp.getHost(), hnp.getPort(), 500);

  pipeline = jedis2.pipelined();
  List<Object> resp = pipeline.syncAndReturnAll();
  assertTrue(resp.isEmpty());

  jedis2.close();
}
```

Example 35

Project: *cachecloud* File: *PipeliningTest.java* *View source code* 6 votes

```
@Test
public void testCloseable() throws IOException {
  // we need to test with fresh instance of Jedis
  Jedis jedis2 = new Jedis(hnp.getHost(), hnp.getPort(), 500);
  jedis2.auth("foobared");

  Pipeline pipeline = jedis2.pipelined();
  Response<String> retFuture1 = pipeline.set("a", "1");
  Response<String> retFuture2 = pipeline.set("b", "2");

  pipeline.close();

  // it shouldn't meet any exception
  retFuture1.get();
  retFuture2.get();
}
```

Example 36

Project: *kafka-consumer-redis* File: *ConsumerThread.java* *View source code* 6 votes

```
@Override
public void run() {
    Thread.currentThread().setName("ConsumerThread-"+topicConfig.getTopic());
```

```
    try {
        while (running) {

            ConsumerRecords<String, String> records = consumer.poll(1000);
            logger.info(String.format("poll count:"+records.count()));
            Pipeline jedisPipe = jedis.pipelined();
            for (ConsumerRecord<String, String> record : records) {
                String key = record.key();
                String value = record.value();
                if(key!=null && value!=null) {
                    jedisPipe.set(key,value);
                }
            }
            jedisPipe.sync();
        }
    }finally {
        if (consumer!=null) {
            consumer.close();
        }
    }
}
```

Example 37

Project: *java-platform* File: *RedisExample.java* *View source code* 6 votes

```
public void testPipelined() {// 0.076秒
        Jedis jedis = new Jedis("120.25.241.144", 6379);
        jedis.auth("b840fc02d52404542994");

        long start = System.currentTimeMillis();
        Pipeline pipeline = jedis.pipelined();
        for (int i = 0; i < 1000; i++) {
                pipeline.set("n" + i, "n" + i);
                System.out.println(i);
        }
        pipeline.syncAndReturnAll();
        long end = System.currentTimeMillis();
        System.out.println("共花费：" + (end - start) / 1000.0 + "秒");

        jedis.disconnect();
        try {
                Closeables.close(jedis, true);
        } catch (IOException e) {
                e.printStackTrace();
        }
}
```

Example 38

Project: *java-platform* File: *RedisExample.java* *View source code* 6 votes

```
public void testCombPipelineTrans() {// 0.099秒
        Jedis jedis = new Jedis("120.25.241.144", 6379);
        jedis.auth("b840fc02d52404542994");

        long start = System.currentTimeMillis();
        Pipeline pipeline = jedis.pipelined();
        pipeline.multi();
        for (int i = 0; i < 1000; i++) {
                pipeline.set("n" + i, "n" + i);
                System.out.println(i);
        }
        pipeline.exec();
        pipeline.syncAndReturnAll();
        long end = System.currentTimeMillis();
        System.out.println("共花费：" + (end - start) / 1000.0 + "秒");

        jedis.disconnect();
        try {
                Closeables.close(jedis, true);
        } catch (IOException e) {
                e.printStackTrace();
        }
}
```

Example 39

Project: *g2* File: *JedisTemplate.java* *View source code* 6 votes

```
/**
 * Execute with a call back action with result in pipeline.
 */
public List<Object> execute(PipelineAction pipelineAction) throws JedisException {
        Jedis jedis = null;
        boolean broken = false;
        try {
                jedis = jedisPool.getResource();
                Pipeline pipeline = jedis.pipelined();
                pipelineAction.action(pipeline);
                return pipeline.syncAndReturnAll();
        } catch (JedisException e) {
                broken = handleJedisException(e);
                throw e;
        } finally {
                closeResource(jedis, broken);
        }
}
```

Example 40

Project: *g2* File: *JedisTemplate.java* *View source code* 6 votes

```
/**
 * Execute with a call back action without result in pipeline.
 */
public void execute(PipelineActionNoResult pipelineAction) throws JedisException {
        Jedis jedis = null;
        boolean broken = false;
        try {
                jedis = jedisPool.getResource();
                Pipeline pipeline = jedis.pipelined();
                pipelineAction.action(pipeline);
                pipeline.sync();
        } catch (JedisException e) {
                broken = handleJedisException(e);
                throw e;
        } finally {
                closeResource(jedis, broken);
        }
}
```

Terms of Use Privacy Support & Contact

ProgramCreek