

## Market Data Line Redis 客户端参考

## 目录

1. 概述.....	3
2. 支持的 Redis 命令 .....	3
3. 支持的消息及定义.....	4
4. 客户端调用流程.....	6
5. 测试客户端 python .....	6
6. 测试客户端 java .....	6
7. 测试客户端 csharp .....	7
8. 连通性测试工具 redis-cli .....	7
9. 参考资料.....	8

## 1. 概述

通联数据行情服务采用二进制 TCP 协议，这个协议为全市场行情推送和订阅做了优化，并且能够适应各种网络条件。为了支持更多的客户端语言以及更多的订阅功能，行情服务还支持简化版的 Redis 协议。

下表是行情服务 TCP 协议和 REDIS 协议的对比，可以看到 2 者相互补充，TCP 协议有利于传输，Redis 协议有利于最终用户使用。

功能	TCP 协议	Redis 协议
在低速网络上进行数据压缩传输	支持	不支持
在高速网络上进行低延迟转发	二进制消息头,序列化和反序列化时间为 0	基于字符串的消息头
消息编码	动态的支持多种消息编码	只允许一种固定的消息编码
支持客户端语言	C++	第三方 redis 客户端支持十几种语言
按消息 ID 订阅	支持	支持
按股票 ID 订阅	不支持	支持
查询股票列表	不支持	支持
查询股票快照	不支持	支持

## 2. 支持的 Redis 命令

作为简化版的 Redis 协议，行情服务支持以下几种 Redis 命令，未提及的 Redis 命令和功能都未实现。

`auth user_token`

说明: 登陆, `user_token` 由系统管理员提供

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/auth>

`subscribe keypace.* [keypace.* ...]`

说明: 订阅某一类消息

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/subscribe>

示例: 订阅所有上交所 Level1 指数 `subscribe mdl.3.3.*`

`unsubscribe keypace.* [keypace.* ...]`

说明: 取消订阅某一类消息

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/unsubscribe>

示例: 取消订阅所有上交所 Level1 指数 `unsubscribe mdl.3.3.*`

`subscribe keypace.ticksymbol [keypace.ticksymbol ...]`

说明: 订阅指定 `ticksymbol` 的消息

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/subscribe>

示例: 订阅上证指数, 平安银行 `subscribe mdl.3.3.000001 mdl.5.2.000001`

`unsubscribe keypace.ticksymbol [keypace.ticksymbol ...]`

说明: 取消订阅指定 `ticksymbol` 的消息

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/unsubscribe>

示例: 取消订阅上证指数, 平安银行 `unsubscribe mdl.3.3.000001 mdl.5.2.000001`

`keys keypace.*`

说明: 查询某一类消息的所有 `ticksymbol`

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/keys>

示例: 查询深交所股票列表 `keys mdl.5.2.*`

`get keypace.ticksymbol`

说明: 取得指定 `ticksymbol` 的最后消息快照

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/get>

示例: 取得上证指数快照 `get mdl.3.3.000001`

`mget keypace.ticksymbol [keypace.ticksymbol ...]`

说明: 取得多个 `ticksymbol` 的消息快照

返回值: 请参考第三方客户端文档和官方文档 <http://redis.io/commands/mget>

示例: 取得上证指数, 平安银行的消息快照 `mget mdl.3.3.000001 mdl.5.2.000001`

`ping`

请参考第三方客户端文档和官方文档 <http://redis.io/commands/ping>

`info`

部分支持, 请参考第三方客户端文档和官方文档 <http://redis.io/commands/info>

注意事项:

- `Subscribe/unsubscribe keypace.*` 和 `subscribe/unsubscribe keypace.ticksymbol` 是 2 组独立的订阅命令, `unsubscribe keypace.ticksymbol` 不会影响 `subscribe keypace.*` 得到的数据, `unsubscribe keypace.*` 不会影响 `subscribe keypace.ticksymbol` 得到的数据, 如果同时 `subscribe keypace.*` 和 `subscribe keypace.ticksymbol`, 会得到 2 次推送。
- 为了服务器发挥最大性能, 同一服务器不会同时支持查询命令 (例如 `get/mget`) 和订阅命令 (例如 `subscribe/unsubscribe`), 某些测试服务器除外。

### 3. 支持的消息及定义

每个 Redis 消息都有一个 key, 我们把 `mdl.` 开头的 key 作为行情消息专有的 `keyspace`。并用 `mdl.serviceID.messageID` 的 key 格式来区分每一种消息类型。

`protobuf` 是 google 的开源消息封装格式, 具有适中的体积和性能, 并且被多种语言支持, 因此我们使用 `protobuf` 来作为消息编码格式。

下表是目前支持的所有消息类型的 `ServiceID`, `MessageID` 定义和 `protobuf` 消息定义。

ServiceID		MessageID		消息定义文件	消息定义名称	TickSymbol
2	系统消息	4	心跳	无需解析处理		
3	上交所 Level1	3	指数	mdl_sh11_msg.proto	Indexes	IndexID
		4	股票		Equity	SecurityID
		5	债券		Bonds	SecurityID

		6	基金		Funds	SecurityID
4	上交所 Level2	3	逐笔成交	mdl_shl2_msg.proto	SHL2Transaction	SecurityID
		4	市场行情		SHL2MarketData	SecurityID
		5	虚拟竞价集合		SHL2VirtualAuctionPrice	SecurityID
		6	指数行情		SHL2Index	SecurityID
		7	市场总览		SHL2MarketOverview	SecurityID
		9	期权行情		OPTLevel1	SecurityID
5	深交所 Level1	1	指数	mdl_szl1_msg.proto	SZL1Index	IndexID
		2	股票		SZL1Stock	SecurityID
6	深交所 Level2	12	证券实时状态	mdl_szl2_msg.proto	SecurityStatus	
		19	公告		Bulletin	
		28	集中竞价交易快照行情		Snapshot300111_v2	SecurityID
		29	指数行情快照		Snapshot309011_v2	SecurityID
		30	成交量统计指标行情快照		Snapshot309111_v2	SecurityID
		31	盘后定价交易业务行情快照		Snapshot300611_v2	SecurityID
		33	集中竞价业务逐笔委托行情		Order300192_v2	SecurityID
		34	协议交易业务逐笔委托行情		Order300592_v2	SecurityID
		35	转融通证券出借业务逐笔委托行情		Order300792_v2	SecurityID
		36	集中竞价业务逐笔成交行情		Transaction300191_v2	SecurityID
		37	协议交易业务逐笔成交行情		Transaction300591_v2	SecurityID
		38	转融通证券出借业务逐笔成交行情		Transaction300791_v2	SecurityID
7	中金所期货	1	期货	mdl_cffex_msg.proto	CTPFuture	InstruID
8	郑交所期货	1	期货	mdl_czce_msg.proto	CTPFuture	InstruID
9	上期所期货	1	期货	mdl_shfe_msg.proto	CTPFuture	InstruID
10	大商所期货	1	期货	mdl_dce_msg.proto	CTPFuture	InstruID
11	港股	2	股票	mdl_hkex_msg.proto	OMDMarketData	TickerSymbol
		3	恒生指数		HangSengIndex	IndexType_SecNo
13	衍生数据	1	上交所股票分钟线	mdl_bar_msg.proto	XSHGStockMinuteBar	SecurityID
		2	深交所股票分钟线		XSHEStockMinuteBar	SecurityID
		3	港交所股票分钟线		HKEXStockMinuteBar	SecurityID
		4	大商所期货分钟线		DCEFuturesMinuteBar	SecurityID
		5	上期所期货分钟线		SHFEFutureMinuteBar	SecurityID
		6	郑商所期货分钟线		CZCEFutureMinuteBar	SecurityID
		7	中金所期货分钟线		CFFEXFutureMinuteBar	SecurityID

		8	上交所期权分钟线		XSHGOptionMinuteBar	SecurityID
		9	上交所指数分钟线		XSHGIndexMinuteBar	SecurityID
		10	深交所指数分钟线		XSHEIndexMinuteBar	SecurityID
		11	上交所股票资金流向		XSHGCapitalFlow	SecurityID
		12	深交所股票资金流向		XSHECapitalFlow	SecurityID
		13	行业资金流向		IndustryCapitalFlow	SecurityID

## 4. 客户端调用流程

- 连接到服务器
- 发送 `auth` 命令请求登陆，用户名任意，但不能为空
- 发送 `subscribe` 命令订阅行情，如果需要服务端发送心跳，请订阅 `mdl.2.4.*`
- 解析服务端推送的 `protobuf` 消息，如果遇到消息不能正常解析的情况，客户端重新连接服务器，记录日志，并报告系统管理员。
- 客户端接收到的 `protobuf` 消息可能包含空值，客户端应该能处理这些空值字段。
- 服务端每隔 10 秒发送心跳消息，心跳消息无需解析，只是帮助客户端判断服务器是否存活。
- 不需要使用时，断开连接

## 5. 测试客户端 python

开发包中包含了用 python 订阅行情的示例代码 `mdl_test.py`

要运行 `mdl_test.py` 需要以下步骤:

- 安装 python
- 安装 python-redis
- 安装 protobuf for python
- 编译消息文件，运行 `redis_protocol\msg_compiler\compile_python_msg.bat`
- 更改代码中的 `host` 为代理服务器地址
- 运行示例代码，在交易时间，代码会输出行情数据

## 6. 测试客户端 java

开发包中包含了用 java 订阅行情的示例代码 `MDLTester.java`

要运行 `MDLTester.java` 需要以下步骤:

- 安装 java 的 redis 客户端
- 安装 java 的 protobuf jar 包和编译器
- 编译消息文件，运行 `redis_protocol\msg_compiler\compile_java_msg.bat`。
- 更改代码中的 `host` 为代理服务器地址
- 运行示例代码，在交易时间，代码会输出行情数据

## 7. 测试客户端 csharp

开发包中包含了用 csharp 订阅行情的示例代码 Form1.cs

要运行 Form1.cs 需要以下步骤:

- 安装 csharp 的 redis 客户端, 推荐 Sider
- 安装 csharp 的 protobuf 解析包, protobuf-net
- 编译消息文件, 运行 redis\_protocol\msg\_compiler\compile\_csharp\_msg.bat。
- 更改代码中的 host 为代理服务器地址
- 运行示例代码, 在交易时间, 代码会输出行情数据

## 8. 连通性测试工具 redis-cli

redis-cli 是 redis 官方提供的测试程序, 可以用来验证 redis 服务是否正常工作

redis-cli 命令行参数为 redis-cli -h hostname -p 9379, 启动 redis-cli 后, 可以输入 redis 命令和行情服务交互。此处 feeder01-dev.datayes.com 需更换为行情代理服务器的地址。

```
c:\redis-cli>redis-cli -h feeder01-dev.datayes.com -p 9379
feeder01-dev.datayes.com:9379>
```

示例 1: 用 info 查询 redis 状态

```
feeder01-dev.datayes.com:9379> info
# Server
redis_version:2.6.0
redis_mode:standalone
os_name:linux
arch_bits:64
compiler_version:937
process_id:31329
uptime_in_seconds:28635
uptime_in_days:0
# Memory
used_memory:973078528
used_memory_human:928.00M
used_memory_rss:98947072
used_memory_peak:1216364544
used_memory_peak_human:1.13G
# CPU
used_cpu_sys:4655.85
used_cpu_user:4866.77
# Clients
connected_clients:1
client_longest_output_list:0B
# Stats
total_connections_received:113
total_commands_processed:44346
pubsub_channels:76320
# Keyspace
db0:keys=76320,hashmap=0,map=0,expires=0,avg_ttl=0
```

示例 2: 用 keys 查询上交所指数列表

```
feeder01-dev.datayes.com:9379> keys mdl.3.3.*
1> "0000001"
2> "0000002"
3> "0000003"
4> "0000004"
5> "0000005"
6> "0000006"
7> "0000007"
8> "0000008"
9> "0000009"
10> "0000010"
11> "0000011"
12> "0000012"
13> "0000013"
14> "0000015"
15> "0000016"
16> "0000017"
```

示例 3: 用 get 查询上证指数

注意, 服务器返回的是 protobuf 格式, 非可读文本。

```
feeder01-dev.datayes.com:9379> get mdl.3.3.0000001
"\n\x060000001\x12\x0c\xe4\xb8\x8a\xe8\xaf\x81\xe6\x8c\x87\xe6\x95\
2\x05\b\x82\xd0\xe2\x0f:\x05\b\xe3\xba\xc3\x0fB\x05\b\x9e\x9e\xc7\
feeder01-dev.datayes.com:9379>
```

示例 4: 订阅上交所所有行情数据

```
feeder01-dev.datayes.com:9379> subscribe mdl.3.3.* mdl.3.4.* mdl.3.5.* mdl.3.6.*
Reading messages... (press Ctrl-C to quit)
1> "subscribe"
2> "mdl.3.3.*"
3> <integer> 1
1> "subscribe"
2> "mdl.3.4.*"
3> <integer> 2
1> "subscribe"
2> "mdl.3.5.*"
3> <integer> 3
1> "subscribe"
2> "mdl.3.6.*"
3> <integer> 4
```

## 9. 参考资料

protobuf 解码: <https://github.com/google/protobuf>

Reis 客户端: <http://redis.io/clients>