

東方通 | **TongTech**[®]
400-650-7088



TongLinkQ9.0

C 客户端编程手册

版本: V9.0.0



目 录

1. 概述.....	1
1.1 目的.....	1
1.2 预期读者.....	1
2. 应用编写中用到的宏.....	1
3. 应用编写中用到的数据结构.....	1
3.1 数据结构.....	2
3.1.1 TLQConfig 结构概述.....	2
3.1.2 域详细说明.....	2
3.1.3 TLQMessage 结构概述.....	3
3.1.4 域详细说明.....	3
3.1.5 TLQBatchMessage 结构概述.....	3
3.1.6 域详细说明.....	3
3.1.7 TLQMultiMessage 结构概述.....	3
3.1.8 域详细说明.....	3
4. 函数定义.....	4
4.1 函数声明.....	4
4.2 基本应用函数.....	4
4.2.1 客户端配置相关函数.....	4
4.2.2 发布订阅同步模式客户端生产者相关函数.....	7
4.2.3 发布订阅异步模式客户端生产者相关函数.....	12
4.2.4 发布订阅同步模式客户端消费者相关函数.....	17
4.2.5 发布订阅异步模式客户端消费者拉取模式相关函数.....	25
4.2.6 发布订阅异步模式客户端消费者推送模式相关函数.....	33
4.2.7 点对点同步模式客户端生产者相关函数.....	40
4.2.8 点对点异步模式客户端生产者相关函数.....	44
4.2.9 点对点同步模式客户端消费者相关函数.....	49
4.2.10 点对点异步模式客户端消费者相关函数.....	55
4.2.11 BUFF 消息相关函数.....	61
4.2.12 批量消息相关函数.....	64
4.2.13 异步模式客户端生产者发送/消费者拉取消息回调相关函数.....	65
4.2.14 异步模式客户端生产者发送消息成功或失败获取结果相关函数.....	66
附录 1 错误编码信息.....	67

1. 概述

1.1 目的

本文档用于描述 TongLINK/Q 客户端 C 接口有关的应用编写中用到的相关宏、数据结构及函数接口定义等。通过阅读此手册能快速查找到应用编写中用到的函数及用法等。

1.2 预期读者

本接口设计说明书的预期读者是消息中间件 TLQ 版本产品用户以及对消息中间件感兴趣的读者。

2. 应用编写中用到的宏

宏名称	代表值	含义
TLQ_DOMAIN_LENGTH	64	域名的最大长度
TLQ_GROUP_LENGTH	64	组名的最大长度
TLQ_TOPIC_LENGTH	256	主题的最大长度
TLQ_FILE_PATH_LENGTH	256	文件路径加文件名的最大长度

3. 应用编写中用到的数据结构

应用编写中用到的概念和数据结构有如下：

同步模式客户端生产者：客户端发送同步消息机制，即当客户端发送消息成功后必须等到对应消息的确认消息才能继续发下一条消息；

异步模式客户端生产者：客户端发送异步消息机制，即当客户端发送消息不用等到消息发送完就可以继续发送下一条消息；

同步模式客户端消费者：客户端拉取同步消息机制，即当客户端发送拉取消息的请求之后需要等待返回的消息数据结果；

异步模式客户端消费者：客户端拉取异步消息机制，即当客户端注册异步回调函数后在消费者注销之前都可以拉取消息数据；

BUFF 消息：指内存中一段用户数据，包括消息头和消息内容，消息内容最大长度 4M；

批量消息：一种包含多个 BUFF 消息的数据结构；

管理节点：详细设计请参看系统使用手册。

数据结构名称	含义
TLQConfig	配置信息
TLQMessage	BUFF 消息
TLQBatchMessage	批量消息
TLQMultiMessage	多个 BUFF 消息
TLQLightProducer	同步模式客户端生产者实例，发布订阅模式
TLQProducer	异步模式客户端生产者实例，发布订阅模式
TLQLightPullConsumer	同步模式客户端消费者实例，发布订阅模式
TLQPullConsumer	异步模式客户端消费者实例，发布订阅模式
TLQLightPairProducer	同步模式客户端生产者实例，点对点模式
TLQLightPairConsumer	同步模式客户端消费者实例，点对点模式
TLQPairProducer	异步模式客户端生产者实例，点对点模式
TLQPairConsumer	异步模式客户端消费者实例，点对点模式
TLQResult	异步客户端生产者回调函数成功返回结果
TLQException	异步客户端生产者回调函数失败返回结果

3.1 数据结构

3.1.1 TLQConfig 结构概述

域名	类型	含义	初始值
pData	void*	配置信息的数据结构指针	NULL

3.1.2 域详细说明

3.1.2.1 pData

在初始化配置信息的时候动态创建的一个数据结构体指针。

3.1.3 TLQMessage 结构概述

域名	类型	含义	初始值
pData	void*	BUFF 消息的数据结构指针	NULL

3.1.4 域详细说明

3.1.4.1 pData

在初始化 BUFF 消息的时候动态创建的一个数据结构体指针。

3.1.5 TLQBatchMessage 结构概述

域名	类型	含义	初始值
pData	void*	批量消息的数据结构指针	NULL

3.1.6 域详细说明

3.1.6.1 pData

在初始化批量消息的时候动态创建的一个数据结构体指针。

3.1.7 TLQMultiMessage 结构概述

域名	类型	含义	初始值
status	TLQMultiStatus	枚举类型： TLQ_MULTI_INIT = 0 TLQ_MULTI_FOUND TLQ_MULTI_NO_NEW_MSG TLQ_MULTI_TIMEOUT	0
minOffset	char	最小的消息偏移量	0
maxOffset	char	最大的消息偏移量	0
messageArray	TLQMessage*	一个或多个 BUFF 消息的数组	NULL
messageCount	int	BUFF 消息的条数	0

3.1.8 域详细说明

3.1.8.1 status

客户端消费者拉取多条 BUFF 消息的状态。

3.1.8.2 minOffset

客户端消费者拉取消息的最小偏移量。

3.1.8.3 maxOffset

客户端消费者拉取消息的最大偏移量。

3.1.8.4 messageArray

客户端消费者存储拉取的一条或多条 BUFF 消息。

3.1.8.5 messageCount

客户端消费者拉取到的 BUFF 消息的条数。

备注：其他未列出的数据结构如：TLQLightProducer、TLQProducer、TLQLightPullConsumer、TLQPullConsumer、TLQLightPairProducer、TLQPairProducer、TLQLightPairConsumer、TLQPairConsumer、TLQResult、TLQException 的数据成员都是 void* pData,初始化值都为 NULL，使用方法见第 4 章的函数使用方法。

4. 函数定义

本部分将详细介绍有关基本应用接口函数，主要从各函数功能、原型、位置、说明等几方面进行具体介绍。没有做特别说明的接口函数既适用于 UNIX 环境又适用于 Windows 环境。

4.1 函数声明

在 C 编程环境下，编写 C 的应用程序时，必须包含头文件 TLQ.h。

4.2 基本应用函数

4.2.1 客户端配置相关函数

4.2.1.1 TLQConfigInit

【功能】客户端配置信息初始化。

【原型】int TLQConfigInit(TLQConfig* outConfig);

【参数说明】

输入：

outConfig 配置信息；

【返回值】

0 成功;
-1 失败。

【使用说明】

用于初始化配置信息，在创建同步/异步生产者或消费者实例之前必须先调用此函数。

4.2.1.2 TLQConfigSetManager

【功能】 客户端设置管理节点地址。

【原型】 `int TLQConfigSetManager(TLQConfig config, const char* addr);`

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
addr 管理节点的地址, 格式: tcp://IP:Port

【返回值】

0 成功;
-1 失败。

【使用说明】

用于指定客户端应用所要连接的管理节点的地址。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 其次设置客户端连接管理节点地址 TLQConfigSetManager。

4.2.1.3 TLQConfigSetClientID

【功能】 设置客户端 ID 唯一标志。

【原型】 `int TLQConfigSetClientID(TLQConfig config, const char* clientid);`

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
clientid 客户端唯一 ID 标志。

【返回值】

0 成功;
-1 失败。

【使用说明】

用于指定客户端应用唯一的 ID 标志。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID。

4.2.1.4 TLQConfigSetLogLevel

【功能】 设置客户端打印日志级别。

【原型】 `int TLQConfigSetLogLevel(TLQConfig config, int level);`

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
level 客户端打印日志级别: 0:DEBUG 1:INFO 2:WARN 3:ERROR 4:FATAL
默认日志级别是 0。

【返回值】

0 成功;
-1 失败。

【使用说明】

用于指定客户端打印日志的级别, 日志级别数字越小, 打印越详细, 日志体量也越大。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel。

备注: TLQConfigSetManager、TLQConfigSetClientID、TLQConfigSetLogLevel 三个函数接口是没有顺序关系的, 在 TLQConfigInit 之后都可以设置。

4.2.1.5 TLQConfigSetFile

【功能】通过设置配置文件来设置配置相关信息。

【原型】int TLQConfigSetFile(TLQConfig config, const char* filepath);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
filepath 配置文件的路径。

【返回值】

0 成功;
-1 失败。

【使用说明】

通过设置配置文件来设置配置信息, 配置文件中可以配置 manager, clientID, logLevel, netFluxMaxSize 等, 如果通过配置文件设置, 4.2.1.2-4.2.1.4 这三个接口函数就可以不用调用。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 其次通过配置文件设置相关配置信息 TLQConfigSetFile。

4.2.1.6 TLQConfigDestroy

【功能】删除客户端配置信息。

【原型】int TLQConfigDestroy(TLQConfig* config);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体地址。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

删除客户端创建的配置信息数据。

1. 调用先后顺序为:

1) 首先初始化配置信息 TLQConfigInit;

2) 其次删除生产者或消费者实例数据之前调用 TLQConfigDestroy。

4.2.2 发布订阅同步模式客户端生产者相关函数

4.2.2.1 TLQLightProducerInit

[[功能]] 同步模式客户端生产者实例初始化函数。

[[原型]] int TLQLightProducerInit(TLQConfig config, TLQLightProducer* outProducer);

[[参数说明]]

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;

outProducer 初始化创建的客户端生产者实例。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

用于创建一个客户端生产者实例。

1. 调用先后顺序为:

1) 首先初始化配置信息 TLQConfigInit;

2) 设置客户端连接管理节点地址 TLQConfigSetManager;

3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;

4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;

5) 创建客户端生产者实例 TLQLightProducerInit。

4.2.2.2 TLQLightProducerSetDomain

[[功能]] 设置同步模式客户端生产者实例的属性值: 域名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

[[原型]] int TLQLightProducerSetDomain

(TLQLightProducer producer, const char* domain);

[[参数说明]]

输入:

producer 客户端生产者实例;

domain 域名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端生产者实例的域名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个生产者类型的实例如 TLQLightProducerInit;
- 2) 其次设置这个实例的域名属性 TLQLightProducerSetDomain。

4.2.2.3 TLQLightProducerSetGroup

【功能】设置同步模式客户端生产者实例的属性值: 组名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightProducerSetGroup(TLQLightProducer producer, const char* group);

【参数说明】

输入:

producer	客户端生产者实例;
group	组名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端生产者实例的组名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个生产者类型的实例如 TLQLightProducerInit;
- 2) 其次设置这个实例的组名属性 TLQLightProducerSetGroup。

4.2.2.4 TLQLightProducerSetTopic

【功能】设置同步模式客户端生产者实例的属性值: 主题名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightProducerSetTopic(TLQLightProducer producer, const char* topic);

【参数说明】

输入:

producer	客户端生产者实例;
topic	主题名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端生产者实例的主题名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个生产者类型的实例如 TLQLightProducerInit;
- 2) 其次设置这个实例的主题名属性 TLQLightProducerSetTopic。

4.2.2.5 TLQLightProducerSetProxy

【功能】设置同步模式客户端生产者实例的属性值：发布订阅模式，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightProducerSetProxy(TLQLightProducer producer, int isProxy);

【参数说明】

输入：

producer 客户端生产者实例；
isProxy 发布订阅模式，默认是 0 值：本地发布订阅 1：本地和全局发布订阅 2：全局发布订阅。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端生产者实例的发布订阅模式属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQLightProducerInit;
- 2) 其次设置这个实例的发布订阅模式属性 TLQLightProducerSetProxy。

4.2.2.6 TLQLightProducerStart

【功能】同步模式客户端生产者实例启动函数。

【原型】int TLQLightProducerStart(TLQLightProducer producer);

【参数说明】

输入：

producer 经初始化创建的客户端生产者实例。

【返回值】

0 成功；

-1 失败。

【使用说明】

用于启动一个客户端生产者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightProducerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightProducerSetTopic;
- 7) 启动一个实例 TLQLightProducerStart。

4.2.2.7 TLQLightPutMessage

【功能】同步模式客户端生产者发送一条 BUFF 消息。

【原型】 int TLQLightPutMessage(TLQLightProducer producer, TLQMessage message);

【参数说明】

输入:

producer 经过 TLQLightProducerInit 初始化的客户端生产者实例结构体;
message 需要发送的 BUFF 消息数据, 详见 [BUFF 消息相关函数](#)。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者发送 BUFF 消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightProducerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQLightProducerSetTopic;
- 7) 启动实例 TLQLightProducerStart;
- 8) 该客户端生产者发送一条 BUFF 消息 TLQLightPutMessage。

4.2.2.8 TLQLightPutBatchMessage

【功能】 同步模式客户端生产者发送一条批量消息。

【原型】 int TLQLightPutBatchMessage(TLQLightProducer producer, TLQBatchMessage batchMessage);

【参数说明】

输入:

producer 经过 TLQLightProducerInit 初始化的客户端生产者实例结构体;
batchMessage 需要发送的批量消息数据, 详见[批量消息相关函数](#)。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者发送批量消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightProducerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQLightProducerSetTopic;
- 7) 启动实例 TLQLightProducerStart;

8) 该客户端生产者发送一条批量消息 TLQLightPutBatchMessage。

4.2.2.9 TLQLightPutFile

【功能】同步模式客户端生产者发送一条 FILE 消息。

【原型】int TLQLightPutFile(TLQLightProducer producer, const char* filePathName);

【参数说明】

输入：

producer 经过 TLQLightProducerInit 初始化的客户端生产者实例结构体；
filePathName 需要发送的文件消息的全路径，包括路径和文件名。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端生产者发送文件消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightProducerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightProducerSetTopic;
- 7) 启动实例 TLQLightProducerStart;
- 8) 该客户端生产者发送一条文件消息 TLQLightPutFile。

4.2.2.10 TLQLightProducerDestroy

【功能】删除同步模式客户端生产者实例。

【原型】int TLQLightProducerDestroy(TLQLightProducer* producer);

【参数说明】

输入：

producer 经过 TLQLightProducerInit 初始化的客户端生产者实例结构体地址。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端生产者实例发送完消息退出应用之前需要删除该实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightProducerInit;

- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightProducerSetTopic;
- 7) 启动实例 TLQLightProducerStart;
- 8) 该客户端生产者发送一条 BUFF 消息 TLQLightPutMessage;
- 9) 删除该客户端生产者实例 TLQLightProducerDestroy。

4.2.3 发布订阅异步模式客户端生产者相关函数

4.2.3.1 TLQProducerInit

【功能】异步模式客户端生产者实例初始化函数。

【原型】int TLQProducerInit(TLQConfig config, TLQProducer* outProducer);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
outProducer 初始化创建的客户端生产者实例。

【返回值】

0 成功;
-1 失败。

【使用说明】

用于创建一个客户端生产者实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明;
- 2) 创建客户端生产者实例 TLQProducerInit。

4.2.3.2 TLQProducerSetDomain

【功能】设置异步模式客户端生产者实例的属性值：域名，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQProducerSetDomain(TLQProducer producer, const char* domain);

【参数说明】

输入:

producer 客户端生产者实例;
domain 域名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端生产者实例的域名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个生产者类型的实例如 TLQProducerInit;
- 2) 其次设置这个实例的域名属性 TLQProducerSetDomain。

4.2.3.3 TLQProducerSetGroup

【功能】设置异步模式客户端生产者实例的属性值：组名，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQProducerSetGroup(TLQProducer producer, const char* group);

【参数说明】

输入：

producer	客户端生产者实例；
group	组名。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端生产者实例的组名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQProducerInit;
- 2) 其次设置这个实例的组名属性 TLQProducerSetGroup。

4.2.3.4 TLQProducerSetTopic

【功能】设置异步模式客户端生产者实例的属性值：主题名，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQProducerSetTopic(TLQProducer producer, const char* topic);

【参数说明】

输入：

producer	客户端生产者实例；
topic	主题名。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端生产者实例的主题名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQProducerInit;
- 2) 其次设置这个实例的主题名属性 TLQProducerSetTopic。

4.2.3.5 TLQProducerSetProxy

【功能】设置异步模式客户端生产者实例的属性值：发布订阅模式，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQProducerSetProxy(TLQProducer producer, int isProxy);

【参数说明】

输入：

producer	客户端生产者实例；
----------	-----------

isProxy 发布订阅模式，默认是 0 值：本地发布订阅 1：本地和全局发布订阅 2:全局发布订阅。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

设置客户端生产者实例的发布订阅模式属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQProducerInit;
- 2) 其次设置这个实例的发布订阅模式属性 TLQProducerSetProxy。

4.2.3.6 TLQProducerStart

[[功能]] 异步模式客户端生产者实例启动函数。

[[原型]] int TLQProducerStart(TLQProducer producer);

[[参数说明]]

输入：

producer 经初始化创建的客户端生产者实例。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

用于启动一个客户端生产者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQProducerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQProducerSetTopic;
- 7) 启动一个实例 TLQProducerStart。

4.2.3.7 TLQPutMessageAsync

[[功能]] 异步模式客户端生产者发送一条 BUFF 消息。

[[原型]] int TLQPutMessageAsync(TLQProducer producer, TLQMessage* message, TLQCallBackSuccess successCallback, TLQCallBackException exceptionCallback, void* args);

[[参数说明]]

输入：

producer 经过 TLQProducerInit 初始化的客户端生产者实例结构体;
message 需要发送的 BUFF 消息数据地址，详见 [BUFF 消息相关函数](#);
successCallback 消息发送成功的回调函数接口，详见 [TLQCallBackSuccess](#)

的详细说明;

exceptionCallback 消息发送失败的回调函数接口, 详见 [TLQCallBackException](#) 的详细说明;

args 需要传入回调函数接口的参数。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

客户端生产者发送 BUFF 消息。

1. 调用先后顺序为:

1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明

2) 创建客户端生产者实例 TLQProducerInit;

3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQProducerSetTopic;

4) 启动一个实例 TLQProducerStart;

5) 客户端生产者发送一条 BUFF 消息 TLQPutMessageAsync。

4.2.3.8 TLQPutBatchMessageAsync

[[功能]] 异步模式客户端生产者发送一条批量消息。

[[原型]] int TLQPutBatchMessageAsync(TLQProducer producer, TLQBatchMessage batchMessage, TLQCallBackSuccess successCallback, TLQCallBackException exceptionCallback, void* args);

[[参数说明]]

输入:

producer 经过 TLQProducerInit 初始化的客户端生产者实例结构体;

batchMessage 需要发送的批量消息数据, 详见[批量消息相关函数](#);

successCallback 消息发送成功的回调函数接口, 详见 [TLQCallBackSuccess](#) 的详细说明;

exceptionCallback 消息发送失败的回调函数接口, 详见 [TLQCallBackException](#) 的详细说明;

args 需要传入回调函数接口的参数。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

客户端生产者发送批量消息。

1. 调用先后顺序为:

1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明

2) 创建客户端生产者实例 TLQProducerInit;

3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQProducerSetTopic;

4) 启动一个实例 TLQProducerStart;

5) 客户端生产者发送一条批量消息 TLQPutBatchMessageAsync。

4.2.3.9 TLQPutFileAsync

【功能】异步模式客户端生产者发送一条文件消息。

【原型】int TLQPutFileAsync(TLQProducer producer, const char* filePathName, TLQCallBackSuccess successCallback, TLQCallBackException exceptionCallback, void* args);

【参数说明】

输入:

producer 经过 TLQProducerInit 初始化的客户端生产者实例结构体;
filePathName 需要发送的文件消息的全路径, 包括路径和文件名。
successCallback 消息发送成功的回调函数接口, 详见 [TLQCallBackSuccess](#)

的详细说明;

exceptionCallback 消息发送失败的回调函数接口, 详见 [TLQCallBackException](#) 的详细说明;

args 需要传入回调函数接口的参数。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者发送文件消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明
- 2) 创建客户端生产者实例 TLQProducerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名

TLQProducerSetTopic;

4) 启动一个实例 TLQProducerStart;

5) 客户端生产者发送一条文件消息 TLQPutFileAsync。

4.2.3.10 TLQProducerDestroy

【功能】删除异步模式客户端生产者实例。

【原型】int TLQProducerDestroy(TLQProducer* producer);

【参数说明】

输入:

producer 经过 TLQProducerInit 初始化的客户端生产者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者实例发送完消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端生产者实例 TLQProducerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名

TLQProducerSetTopic;

- 4) 启动一个实例 TLQProducerStart;
- 5) 客户端生产者发送一条 BUFF 消息 TLQPutMessageAsync;
- 6) 删除客户端生产者实例 TLQProducerDestroy。

4.2.4 发布订阅同步模式客户端消费者相关函数

4.2.4.1 TLQLightPullConsumerInit

【功能】同步模式客户端消费者实例初始化函数。

【原型】int TLQLightPullConsumerInit(TLQConfig config, TLQLightPullConsumer* outConsumer);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
outConsumer 初始化创建的客户端消费者实例。

【返回值】

0 成功;
-1 失败。

【使用说明】

用于创建一个客户端消费者实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit。

4.2.4.2 TLQLightPullConsumerSetDomain

【功能】设置同步模式客户端消费者实例的属性值: 域名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightPullConsumerSetDomain
(TLQLightPullConsumer consumer, const char* domain);

【参数说明】

输入:

consumer 客户端消费者实例;
domain 域名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例的域名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQLightPullConsumerInit;
- 2) 其次设置这个实例的域名属性 TLQLightPullConsumerSetDomain。

4.2.4.3 TLQLightPullConsumerSetTopic

【功能】设置同步模式客户端消费者实例的属性值: 主题名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightPullConsumerSetTopic
(TLQLightPullConsumer consumer, const char* topic);

【参数说明】

输入:

consumer	客户端消费者实例;
topic	主题名。

【返回值】

0	成功;
-1	失败。

【使用说明】

设置客户端消费者实例的主题名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQLightPullConsumerInit;
- 2) 其次设置这个实例的主题名属性 TLQLightPullConsumerSetTopic。

4.2.4.4 TLQLightPullConsumerSetGroup

【功能】设置同步模式客户端消费者实例的属性值: 组名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightPullConsumerSetGroup
(TLQLightPullConsumer consumer, const char* group);

【参数说明】

输入:

consumer	客户端消费者实例;
group	组名。

【返回值】

0	成功;
-1	失败。

【使用说明】

设置客户端消费者实例的组名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQLightPullConsumerInit;
- 2) 其次设置这个实例的组名属性 TLQLightPullConsumerSetGroup。

4.2.4.5 TLQLightPullConsumerSetProxy

【功能】设置同步模式客户端消费者实例的属性值: 发布订阅模式, 需要在该客户端实

公司地址: 北京市海淀区中关村南大街2号数码大厦A座22层 100080

电话: (010) 82652228

网址: www.tongtech.com

例初始化函数调用之后才能使用此接口设置实例属性。

[[原型]] int TLQLightPullConsumerSetProxy
(TLQLightPullConsumer consumer, int isProxy);

[[参数说明]]

输入:

consumer 客户端消费者实例;

isProxy 发布订阅模式, 默认是 0 值: 本地发布订阅 1: 本地和全局发布订阅 2: 全局发布订阅。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

设置客户端消费者实例的发布订阅模式属性值。

1. 调用先后顺序为:

1) 首先初始化一个消费者类型的实例如 TLQLightPullConsumerInit;

2) 其次设置这个实例的发布订阅模式属性 TLQLightPullConsumerSetProxy。

4.2.4.6 TLQLightPullConsumerSetConsumeFlag

[[功能]] 设置同步模式客户端消费者实例的属性值: 订阅类型, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

[[原型]] int TLQLightPullConsumerSetConsumeFlag
(TLQLightPullConsumer consumer, int consumeFlag);

[[参数说明]]

输入:

consumer 客户端消费者实例;

consumeFlag 订阅类型, 默认是 0 值: 持久订阅类型; 1: 非持久订阅类型。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

设置客户端消费者实例在发布订阅模式下消费者的订阅类型。

1. 调用先后顺序为:

1) 首先初始化一个消费者类型的实例如 TLQLightPullConsumerInit;

2) 其次设置这个实例的订阅类型属性 TLQLightPullConsumerSetConsumeFlag。

4.2.4.7 TLQLightPullConsumerSetAckMode

[[功能]] 设置同步模式客户端消费者实例的属性值: 主题使用历史记录模式下回复 ACK 消息的模式: 自动回复或手动回复, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

[[原型]] int TLQLightPullConsumerSetAckMode
(TLQLightPullConsumer consumer, int manualAck);

[[参数说明]]

输入:

consumer 客户端消费者实例;

manualAck 自动回复或手动回复, 默认是 0 值: 自动回复 ACK; 1: 手动回复 ACK。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

设置客户端消费者实例在主题使用历史记录模式的时候拉取到消息后回复 ACK 的模式, 自动或手动回复 ACK。

1. 调用先后顺序为:

1) 首先初始化一个消费者类型的实例如 TLQLightPullConsumerInit;

2) 其次设置这个实例 ACK 模式属性 TLQLightPullConsumerSetAckMode。

4.2.4.8 TLQLightPullConsumerStart

[[功能]] 同步模式客户端消费者实例启动函数。

[[原型]] int TLQLightPullConsumerStart(TLQLightPullConsumer consumer);

[[参数说明]]

输入:

consumer 初始化创建的客户端消费者实例。

[[返回值]]

0 成功;

-1 失败。

[[使用说明]]

用于启动一个客户端消费者实例。

1. 调用先后顺序为:

1) 首先初始化配置信息 TLQConfigInit;

2) 设置客户端连接管理节点地址 TLQConfigSetManager;

3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;

4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;

5) 创建客户端消费者实例 TLQLightPullConsumerInit;

6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQLightPullConsumerSetTopic;

7) 启动一个实例 TLQLightPullConsumerStart。

4.2.4.9 TLQLightPullMessage

[[功能]] 同步模式客户端消费者按偏移量 0 拉取一条 BUFF 消息。

[[原型]] int TLQLightPullMessage(TLQLightPullConsumer consumer, TLQMessage* message);

[[参数说明]]

输入:

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体;

message 拉取到的 BUFF 消息数据。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者拉取一条 BUFF 消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQLightPullConsumerSetTopic;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者拉取一条 BUFF 消息 TLQLightPullMessage。

4.2.4.10 TLQLightPullMultiMessage

【功能】同步模式客户端消费者按偏移量 0 拉取一条或多条 BUFF 消息。

【原型】int TLQLightPullMultiMessage(TLQLightPullConsumer consumer, int pullNum, TLQMultiMessage* multiMessage);

【参数说明】

输入:

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体;
pullNum 期望拉取消息的条数;
multiMessage 拉取到的多条 BUFF 消息数据, 详见 3.1.7 TLQMultiMessage

结构概述。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者拉取多条 BUFF 消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQLightPullConsumerSetTopic;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者拉取多条 BUFF 消息 TLQLightPullMultiMessage。

4.2.4.11 TLQLightPullMultiRollbackMessage

【功能】同步模式客户端消费者按偏移量拉取多条回溯消息。

【原型】int TLQLightPullMultiRollbackMessage(TLQLightPullConsumer consumer, long long offset, int pullMaxNum, TLQMultiMessage* multiMessage);

【参数说明】

输入：

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体；
offset 消息条数的偏移量，offset 决定拉取消息的模式有三种：-1/-2/>=0，默认值是 0 值；-2 拉取最新消息，当 offset 为-2 时 pullNum 只能是 1；-1 是使用服务端历史记录拉取；>=0 是使用指定 offset 拉取指定偏移量的消息；
pullMaxNum 期望拉取消息的最多条数；
multiMessage 拉取到的多条回溯消息数据。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者按偏移量拉取多条回溯消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightConsumerSetTopic;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者按偏移量拉取多条回溯消息 TLQLightPullMultiRollbackMessage。

4.2.4.12 TLQLightPullFile

【功能】同步模式客户端消费者拉取一条 File 消息。

【原型】int TLQLightPullFile(TLQLightPullConsumer consumer, const char* filePath);

【参数说明】

输入：

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体；
filePath 需要存储拉取到的文件的路径。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者拉取一条文件消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightPullConsumerSetTopic;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者拉取一条文件消息 TLQLightPullFile。

4.2.4.13 TLQLightPullFileAck

【功能】同步模式客户端消费者拉取一条 File 消息完成后回复确认 ACK。

【原型】int TLQLightPullFileAck(TLQLightPullConsumer consumer);

【参数说明】

输入:

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者拉取一条文件消息完成后回复 ACK 确认。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightPullConsumerSetTopic;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者拉取一条文件消息 TLQLightPullFile;
- 9) 拉取文件消息成功完成后回复 ACK 确认 TLQLightPullFileAck。

4.2.4.14 TLQLightPullConsumerDestroy

【功能】删除同步模式客户端消费者实例。

【原型】int TLQLightPullConsumerDestroy(TLQLightPullConsumer* consumer);

【参数说明】

输入:

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端消费者实例拉取消息退出应用之前需要删除该实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightPullConsumerSetTopic;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者拉取一条 BUFF 消息 TLQLightPullMessage;
- 9) 删除客户端消费者实例 TLQLightPullConsumerDestroy。

4.2.4.15 TLQLightPullConsumerAckMessage

【功能】同步模式客户端消费者在使用历史记录拉取消息完成后回复确认 ACK。

【原型】int TLQLightPullConsumerAckMessage(TLQLightPullConsumer consumer);

【参数说明】

输入：

consumer 经过 TLQLightPullConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

- 0 成功；
-1 失败。

【使用说明】

客户端消费者在使用历史记录拉取消息完成后回复 ACK 确认。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQLightPullConsumerSetTopic; 这里需要设置 offset 为-1 模式;
- 7) 启动一个实例 TLQLightPullConsumerStart;
- 8) 客户端消费者拉取 BUFF 消息 TLQLightPullMultiRollbackMessage;
- 9) 拉取消息成功完成后回复 ACK 确认 TLQLightPullConsumerAckMessage。

4.2.5 发布订阅异步模式客户端消费者拉取模式相关函数

4.2.5.1 TLQPullConsumerInit

【功能】异步模式客户端消费者实例初始化函数。

【原型】int TLQPullConsumerInit(TLQConfig config, TLQPullConsumer* outConsumer);

【参数说明】

输入：

config 经过 TLQConfigInit 初始化的配置信息结构体；

outConsumer 初始化创建的客户端消费者实例。

【返回值】

0 成功；

-1 失败。

【使用说明】

用于创建一个客户端消费者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPullConsumerInit。

4.2.5.2 TLQPullConsumerSetDomain

【功能】设置异步模式客户端消费者实例的属性值：域名，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPullConsumerSetDomain

(TLQPullConsumer consumer, const char* domain);

【参数说明】

输入：

consumer 客户端消费者实例；

domain 域名。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端消费者实例的域名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 TLQPullConsumerInit；
- 2) 其次设置这个实例的域名属性 TLQPullConsumerSetDomain。

4.2.5.3 TLQPullConsumerSetTopic

【功能】设置异步模式客户端消费者实例的属性值：主题名，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPullConsumerSetTopic

(TLQPullConsumer consumer, const char* topic);

【参数说明】

输入:

consumer 客户端消费者实例;
topic 主题名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例的主题名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPullConsumerInit;
- 2) 其次设置这个实例的主题名属性 TLQPullConsumerSetTopic。

4.2.5.4 TLQPullConsumerSetGroup

【功能】设置异步模式客户端消费者实例的属性值: 组名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPullConsumerSetGroup

(TLQPullConsumer consumer, const char* group);

【参数说明】

输入:

consumer 客户端消费者实例;
group 组名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例的组名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPullConsumerInit;
- 2) 其次设置这个实例的组名属性 TLQPullConsumerSetGroup。

4.2.5.5 TLQPullConsumerSetProxy

【功能】设置异步模式客户端消费者实例的属性值: 发布订阅模式, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPullConsumerSetProxy

(TLQPullConsumer consumer, int isProxy);

【参数说明】

输入:

consumer 客户端消费者实例;
isProxy 发布订阅模式, 默认是 0 值: 本地发布订阅 1: 本地和全局发布订阅 2: 全局发布订阅。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例的发布订阅模式属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPullConsumerInit;
- 2) 其次设置这个实例的发布订阅模式属性 TLQPullConsumerSetProxy。

4.2.5.6 TLQPullConsumerSetConsumeFlag

【功能】设置异步模式客户端消费者实例的属性值: 订阅类型, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPullConsumerSetConsumeFlag

(TLQPullConsumer consumer, int consumeFlag);

【参数说明】

输入:

consumer 客户端消费者实例;

consumeFlag 订阅类型, 默认是 0 值: 持久订阅类型; 1: 非持久订阅类型。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例在发布订阅模式下消费者的订阅类型。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPullConsumerInit;
- 2) 其次设置这个实例的订阅类型属性 TLQPullConsumerSetConsumeFlag。

4.2.5.7 TLQPullConsumerSetAckMode

【功能】设置异步模式客户端消费者实例的属性值: 主题使用历史记录模式下回复 ACK 消息的模式: 自动回复或手动回复, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPullConsumerSetAckMode

(TLQPullConsumer consumer, int manualAck);

【参数说明】

输入:

consumer 客户端消费者实例;

manualAck 自动回复或手动回复, 默认是 0 值: 自动回复 ACK; 1: 手动回复 ACK。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例在主题使用历史记录模式的时候拉取到消息后回复 ACK 的

模式，自动或手动回复 ACK。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 `TLQPullConsumerInit`;
- 2) 其次设置这个实例 ACK 模式属性 `TLQPullConsumerSetAckMode`。

4.2.5.8 TLQPullConsumerSetOffsetPullNum

【功能】设置异步模式客户端消费者实例的属性值：只有在发布订阅模式且使用回调函数拉取消息时对消费者实例属性设置 `offset` 和 `pullNum` 来控制拉取消息的偏移量和拉取消息条数，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】`int TLQPullConsumerSetOffsetPullNum`

`(TLQPullConsumer consumer, int offset, int pullNum);`

【参数说明】

输入：

`consumer` 客户端消费者实例；

`offset` 偏移量，决定拉取消息的模式有三种：-1/-2/>=0，默认值是 0 值；-2 拉取最新消息，当 `offset` 为-2 时 `pullNum` 只能是 1；-1 是使用服务端历史记录拉取；>=0 是使用指定 `offset` 拉取指定偏移量的消息；

`pullNum` 拉取消息条数，默认值为 1。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端消费者实例在发布订阅模式且使用回调函数拉取消息时对消费者实例属性设置 `offset` 和 `pullNum` 来控制拉取消息的偏移量和拉取消息条数。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 `TLQPullConsumerInit`;
- 2) 其次设置拉取消息偏移和条数属性 `TLQPullConsumerSetOffsetPullNum`。

4.2.5.9 TLQPullConsumerStart

【功能】异步模式客户端消费者实例启动函数。

【原型】`int TLQPullConsumerStart(TLQPullConsumer consumer);`

【参数说明】

输入：

`consumer` 初始化创建的客户端消费者实例。

【返回值】

0 成功；

-1 失败。

【使用说明】

用于启动一个客户端消费者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 `TLQConfigInit`;
- 2) 设置客户端连接管理节点地址 `TLQConfigSetManager`;

- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQPullConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPullConsumerSetTopic;
- 7) 启动一个实例 TLQPullConsumerStart。

4.2.5.10 TLQPullRegisterCallBack

【功能】异步模式客户端消费者注册拉取 BUFF 消息的回调函数接口。

【原型】int TLQPullRegisterCallBack(TLQPullConsumer consumer, TLQCallback callback, void* args);

【参数说明】

输入:

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体;
callback 拉取消息的回调函数接口, 详见 [TLQCallback](#) 的详细说明;
args 需要传入回调函数接口的参数。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者拉取 BUFF 消息需要注册这个回调函数。

1. 调用先后顺序为:
 - 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
 - 2) 创建客户端消费者实例 TLQPullConsumerInit;
 - 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQPullConsumerSetTopic;
 - 4) 启动一个实例 TLQPullConsumerStart;
 - 5) 客户端消费者注册拉取 BUFF 消息的回调函数 TLQPullRegisterCallBack。

4.2.5.11 TLQPullUnRegisterCallBack

【功能】异步模式客户端消费者注销拉取消息的回调函数接口。

【原型】int TLQPullUnRegisterCallBack(TLQPullConsumer consumer);

【参数说明】

输入:

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者停止拉取消息需要注销这个回调函数。

1. 调用先后顺序为:
 - 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;

- 2) 创建客户端消费者实例 TLQPullConsumerInit;
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPullConsumerSetTopic;
- 4) 启动一个实例 TLQPullConsumerStart;
- 5) 客户端消费者注册拉取消息的回调函数 TLQPullRegisterCallBack;
- 6) 拉取一定时间后，如 sleep(1)，停止拉取消息：TLQPullUnRegisterCallBack。

4.2.5.12 TLQPullMultiRollbackMessage

【功能】异步模式客户端消费者按偏移量拉取多条回溯消息。

【原型】int TLQPullMultiRollbackMessage(TLQPullConsumer consumer, long long offset, int pullMaxNum, TLQMultiMessage* multiMessage);

【参数说明】

输入：

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体；

offset 消息条数的偏移量，可以从 0 开始的整数，即传入 0 代表从第一条开始拉取；决定拉取消息的模式有三种：-1/-2/>=0，默认值是 0 值；-2 拉取最新消息，当 offset 为 -2 时 pullNum 只能是 1；-1 是使用服务端历史记录拉取；>=0 是使用指定 offset 拉取指定偏移量的消息；

pullMaxNum 期望拉取消息的最多条数；

multiMessage 拉取到的多条回溯消息数据。

【返回值】

0 成功；

-1 失败。

【使用说明】

客户端消费者按偏移量拉取多条回溯消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPullConsumerInit;
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPullConsumerSetTopic;
- 4) 启动一个实例 TLQPullConsumerStart;
- 5) 客户端消费者按偏移量拉取多条回溯消息 TLQPullMultiRollbackMessage。

4.2.5.13 TLQPullMessage

【功能】异步模式客户端消费者按偏移量 0 拉取一条 BUFF 消息。

【原型】int TLQPullMessage(TLQPullConsumer consumer, TLQMessage* message);

【参数说明】

输入：

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体；

message 拉取到的一条消息。

【返回值】

0 成功；

-1 失败。

【使用说明】

客户端消费者拉取一条 BUFF 消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPullConsumerInit；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPullConsumerSetTopic；
- 4) 启动一个实例 TLQPullConsumerStart；
- 5) 客户端消费者拉取一条消息 TLQPullMessage。

4.2.5.14 TLQPullMultiMessage

【功能】异步模式客户端消费者按偏移量 0 拉取一条或多条 BUFF 消息。

【原型】int TLQPullMultiMessage(TLQPullConsumer consumer, int pullMaxNum, TLQMultiMessage* multiMessage);

【参数说明】

输入：

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体；
pullMaxNum 期望拉取消息的最大条数；
multiMessage 拉取到的消息。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者拉取一条或多条 BUFF 消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPullConsumerInit；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPullConsumerSetTopic；
- 4) 启动一个实例 TLQPullConsumerStart；
- 5) 客户端消费者拉取一条或多条消息 TLQPullMultiMessage。

4.2.5.15 TLQPullFileRegisterCallBack

【功能】异步模式客户端消费者注册拉取文件消息的回调函数接口。

【原型】int TLQPullFileRegisterCallBack(TLQPullConsumer consumer, const char* filePath, TLQCallback callback, void* args);

【参数说明】

输入：

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体；
filePath 需要存储拉取到的文件的路径；
callback 拉取消息的回调函数接口，详见 [TLQCallback](#) 的详细说明；

args 需要传入回调函数接口的参数。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者拉取文件消息需要注册这个回调函数。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 `TLQPullConsumerInit`；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 `TLQPullConsumerSetTopic`；
- 4) 启动一个实例 `TLQPullConsumerStart`；
- 5) 客户端消费者注册拉取文件消息的回调函数 `TLQPullFileRegisterCallBack`。

4.2.5.16 TLQPullFileAck

【功能】异步模式客户端消费者拉取文件消息成功完成后回复确认 ACK。

【原型】`int TLQPullFileAck(TLQPullConsumer consumer);`

【参数说明】

输入：

consumer 经过 `TLQPullConsumerInit` 初始化的客户端消费者实例结构体。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者拉取文件消息成功完成后回复确认 ACK。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 `TLQPullConsumerInit`；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 `TLQPullConsumerSetTopic`；
- 4) 启动一个实例 `TLQPullConsumerStart`；
- 5) 客户端消费者注册拉取文件消息的回调函数 `TLQPullFileRegisterCallBack`；
- 6) 拉取文件成功后回复 ACK 确认 `TLQPullFileAck`。

4.2.5.17 TLQPullConsumerAckMessage

【功能】异步模式客户端消费者使用历史记录拉取 BUFF 消息成功完成后回复确认 ACK。

【原型】`int TLQPullConsumerAckMessage(TLQPullConsumer consumer);`

【参数说明】

输入：

consumer 经过 `TLQPullConsumerInit` 初始化的客户端消费者实例结构体。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者使用历史记录拉取 BUFF 消息成功完成后回复确认 ACK。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPullConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQPullConsumerSetTopic; 这里需要设置 offset 为-1 模式;
- 4) 启动一个实例 TLQPullConsumerStart;
- 5) 客户端消费者注册拉取 BUFF 消息的回调函数 TLQPullRegisterCallBack;
- 6) 拉取消息成功后回复 ACK 确认 TLQPullConsumerAckMessage。

4.2.5.18 TLQPullConsumerDestroy

【功能】删除异步模式客户端消费者实例。

【原型】int TLQPullConsumerDestroy(TLQPullConsumer* consumer);

【参数说明】

输入:

consumer 经过 TLQPullConsumerInit 初始化的客户端消费者实例结构体地址。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者实例拉取消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPullConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQPullConsumerSetTopic;
- 4) 启动一个实例 TLQPullConsumerStart;
- 5) 客户端消费者拉取消息, 如回溯消息 TLQPullMultiRollbackMessage;
- 6) 删除客户端消费者实例 TLQPullConsumerDestroy。

4.2.6 发布订阅异步模式客户端消费者推送模式相关函数

4.2.6.1 TLQPushConsumerInit

【功能】异步模式客户端消费者实例初始化函数。

【原型】int TLQPushConsumerInit(TLQConfig config, TLQPushConsumer* outConsumer);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;

outConsumer 初始化创建的客户端消费者实例。

【返回值】

0 成功;
-1 失败。

【使用说明】

用于创建一个客户端消费者实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPushConsumerInit。

4.2.6.2 TLQPushConsumerSetDomain

【功能】设置异步模式客户端消费者实例的属性值: 域名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetDomain

(TLQPushConsumer consumer, const char* domain);

【参数说明】

输入:

consumer 客户端消费者实例;
domain 域名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例的域名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit;
- 2) 其次设置这个实例的域名属性 TLQPushConsumerSetDomain。

4.2.6.3 TLQPushConsumerSetTopic

【功能】设置异步模式客户端消费者实例的属性值: 主题名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetTopic

(TLQPushConsumer consumer, const char* topic);

【参数说明】

输入:

consumer 客户端消费者实例;
topic 主题名。

【返回值】

0 成功;
-1 失败。

【使用说明】

设置客户端消费者实例的主题名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit;
- 2) 其次设置这个实例的主题名属性 TLQPushConsumerSetTopic。

4.2.6.4 TLQPushConsumerSetGroup

【功能】设置异步模式客户端消费者实例的属性值: 组名, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetGroup

(TLQPushConsumer consumer, const char* group);

【参数说明】

输入:

consumer	客户端消费者实例;
group	组名。

【返回值】

0	成功;
-1	失败。

【使用说明】

设置客户端消费者实例的组名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit;
- 2) 其次设置这个实例的组名属性 TLQPushConsumerSetGroup。

4.2.6.5 TLQPushConsumerSetProxy

【功能】设置异步模式客户端消费者实例的属性值: 发布订阅模式, 需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetProxy

(TLQPushConsumer consumer, int isProxy);

【参数说明】

输入:

consumer	客户端消费者实例;
isProxy	发布订阅模式, 默认是 0 值: 本地发布订阅 1: 本地和全局发布订阅 2: 全局发布订阅。

【返回值】

0	成功;
-1	失败。

【使用说明】

设置客户端消费者实例的发布订阅模式属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit;
- 2) 其次设置这个实例的发布订阅模式属性 TLQPushConsumerSetProxy。

4.2.6.6 TLQPushConsumerSetConsumeFlag

【功能】设置异步模式客户端消费者实例的属性值：订阅类型，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetConsumeFlag
(TLQPushConsumer consumer, int consumeFlag);

【参数说明】

输入：

consumer 客户端消费者实例；

consumeFlag 订阅类型，默认是 0 值：持久订阅类型；1：非持久订阅类型。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端消费者实例在发布订阅模式下消费者的订阅类型。

1. 调用先后顺序为：

1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit；

2) 其次设置这个实例的订阅类型属性 TLQPushConsumerSetConsumeFlag。

4.2.6.7 TLQPushConsumerSetAckMode

【功能】设置异步模式客户端消费者实例的属性值：主题使用历史记录模式下回复 ACK 消息的模式：自动回复或手动回复，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetAckMode
(TLQPushConsumer consumer, int manualAck);

【参数说明】

输入：

consumer 客户端消费者实例；

manualAck 自动回复或手动回复，默认是 0 值：自动回复 ACK；1：手动回复 ACK。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端消费者实例在主题使用历史记录模式的时候拉取到消息后回复 ACK 的模式，自动或手动回复 ACK。

1. 调用先后顺序为：

1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit；

2) 其次设置这个实例 ACK 模式属性 TLQPushConsumerSetAckMode。

4.2.6.8 TLQPushConsumerSetOffsetPullNum

【功能】设置异步模式客户端消费者实例的属性值：只有在发布订阅模式且使用回调函

数拉取消息时对消费者实例属性设置 offset 和 pullNum 来控制拉取消息的偏移量和拉取消息条数，需要在该客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPushConsumerSetOffsetPullNum

(TLQPushConsumer consumer, int offset, int pullNum);

【参数说明】

输入：

consumer 客户端消费者实例；

offset 偏移量，决定拉取消息的模式有三种：-1/-2/>=0，默认值是 0 值；-2 拉取最新消息，当 offset 为-2 时 pullNum 只能是 1；-1 是使用服务端历史记录拉取；>=0 是使用指定 offset 拉取指定偏移量的消息；

pullNum 拉取消息条数，默认值为 1。

【返回值】

0 成功；

-1 失败。

【使用说明】

设置客户端消费者实例在发布订阅模式且使用回调函数拉取消息时对消费者实例属性设置 offset 和 pullNum 来控制拉取消息的偏移量和拉取消息条数。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 TLQPushConsumerInit；
- 2) 其次设置拉取消息偏移和条数属性 TLQPushConsumerSetOffsetPullNum。

4.2.6.9 TLQPushConsumerStart

【功能】异步模式客户端消费者实例启动函数。

【原型】int TLQPushConsumerStart(TLQPushConsumer consumer);

【参数说明】

输入：

consumer 初始化创建的客户端消费者实例。

【返回值】

0 成功；

-1 失败。

【使用说明】

用于启动一个客户端消费者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit；
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager；
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID；
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel；
- 5) 创建客户端消费者实例 TLQPushConsumerInit；

6) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPushConsumerSetTopic；

7) 启动一个实例 TLQPushConsumerStart。

4.2.6.10 TLQPushRegisterCallBack

【功能】异步模式客户端消费者注册获取 BUFF 消息的回调函数接口。

【原型】int TLQPushRegisterCallBack(TLQPushConsumer consumer, TLQCallback callback, void* args);

【参数说明】

输入：

consumer 经过 TLQPushConsumerInit 初始化的客户端消费者实例结构体；
callback 获取消息的回调函数接口，详见 [TLQCallback](#) 的详细说明；
args 需要传入回调函数接口的参数。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者获取 BUFF 消息需要注册这个回调函数。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPushConsumerInit；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPushConsumerSetTopic；
- 4) 启动一个实例 TLQPushConsumerStart；
- 5) 客户端消费者注册获取 BUFF 消息的回调函数 TLQPushRegisterCallBack。

4.2.6.11 TLQPushUnRegisterCallBack

【功能】异步模式客户端消费者注销获取消息的回调函数接口。

【原型】int TLQPushUnRegisterCallBack(TLQPushConsumer consumer);

【参数说明】

输入：

consumer 经过 TLQPushConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功；
-1 失败。

【使用说明】

客户端消费者停止获取消息需要注销这个回调函数。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPushConsumerInit；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置主题名 TLQPushConsumerSetTopic；
- 4) 启动一个实例 TLQPushConsumerStart；
- 5) 客户端消费者注册获取消息的回调函数 TLQPushRegisterCallBack；
- 6) 一定时间后，如 sleep(1)，停止获取消息：TLQPushUnRegisterCallBack。

4.2.6.12 TLQPushConsumerAckMessage

【功能】异步模式客户端消费者使用历史记录获取 BUFF 消息成功完成后回复确认 ACK。

【原型】int TLQPushConsumerAckMessage(TLQPushConsumer consumer);

【参数说明】

输入:

consumer 经过 TLQPushConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端消费者使用历史记录获取 BUFF 消息成功完成后回复确认 ACK。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPushConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQPushConsumerSetTopic; 这里需要设置 offset 为-1 模式;
- 4) 启动一个实例 TLQPushConsumerStart;
- 5) 客户端消费者注册获取 BUFF 消息的回调函数 TLQPushRegisterCallBack;
- 6) 获取消息成功后回复 ACK 确认 TLQPushConsumerAckMessage。

4.2.6.13 TLQPushConsumerDestroy

【功能】删除异步模式客户端消费者推送模式的实例。

【原型】int TLQPushConsumerDestroy(TLQPushConsumer* consumer);

【参数说明】

输入:

consumer 经过 TLQPushConsumerInit 初始化的客户端消费者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端消费者实例获取消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPushConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQPushConsumerSetTopic;
- 4) 启动一个实例 TLQPushConsumerStart;
- 5) 客户端消费者注册获取 BUFF 消息的回调函数 TLQPushRegisterCallBack;
- 6) 一定时间后, 如 sleep(1), 注销回调函数: TLQPushUnRegisterCallBack;
- 7) 删除客户端消费者实例 TLQPushConsumerDestroy。

4.2.7 点对点同步模式客户端生产者相关函数

4.2.7.1 TLQLightPairProducerInit

【功能】客户端生产者实例初始化函数。

【原型】`int TLQLightPairProducerInit`
(`TLQConfig config`, `TLQLightPairProducer* outPairProducer`);

【参数说明】

输入：

`config` 经过 `TLQConfigInit` 初始化的配置信息结构体；
`outPairProducer` 初始化创建的客户端生产者实例。

【返回值】

0 成功；
-1 失败。

【使用说明】

用于创建一个客户端生产者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 `TLQConfigInit`;
- 2) 设置客户端连接管理节点地址 `TLQConfigSetManager`;
- 3) 设置客户端唯一 ID 标志 `TLQConfigSetClientID`;
- 4) 选择设置客户端打印日志级别 `TLQConfigSetLogLevel`;
- 5) 创建客户端生产者实例 `TLQLightPairProducerInit`。

4.2.7.2 TLQLightPairProducerSetDomain

【功能】设置客户端生产者实例的属性值：域名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】`int TLQLightPairProducerSetDomain`
(`TLQLightPairProducer pairProducer`, `const char* domain`);

【参数说明】

输入：

`pairProducer` 客户端生产者实例；
`domain` 域名。

【返回值】

0 成功；
-1 失败。

【使用说明】

设置客户端生产者实例的域名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 `TLQLightPairProducerInit`;
- 2) 其次设置这个实例的域名属性 `TLQLightPairProducerSetDomain`。

4.2.7.3 TLQLightPairProducerSetGroup

【功能】设置客户端生产者实例的属性值：组名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightPairProducerSetGroup
(TLQLightPairProducer pairProducer, const char* group);

【参数说明】

输入：

pairProducer	客户端生产者实例；
group	组名。

【返回值】

0	成功；
-1	失败。

【使用说明】

设置客户端生产者实例的组名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQLightPairProducerInit;
- 2) 其次设置这个实例的组名属性 TLQLightPairProducerSetGroup。

4.2.7.4 TLQLightPairProducerSetTopic

【功能】设置客户端生产者实例的属性值：队列名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQLightPairProducerSetTopic
(TLQLightPairProducer pairProducer, const char* topic);

【参数说明】

输入：

pairProducer	客户端生产者实例；
topic	队列名。

【返回值】

0	成功；
-1	失败。

【使用说明】

设置客户端生产者实例的队列名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQLightPairProducerInit;
- 2) 其次设置这个实例的队列名属性 TLQLightPairProducerSetTopic。

4.2.7.5 TLQLightPairProducerStart

【功能】客户端生产者实例启动函数。

【原型】int TLQLightPairProducerStart(TLQLightPairProducer pairProducer);

【参数说明】

输入：

pairProducer 经初始化创建的客户端生产者实例。

【返回值】

0 成功;
-1 失败。

【使用说明】

用于启动一个客户端生产者实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightPairProducerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQLightPairProducerSetTopic;
- 7) 启动一个实例 TLQLightPairProducerStart。

4.2.7.6 TLQLightPairPutMessage

【功能】客户端生产者发送一条 BUFF 消息。

【原型】int TLQLightPairPutMessage(TLQLightPairProducer pairProducer, TLQMessage message);

【参数说明】

输入:

pairProducer 经过 TLQLightPairProducerInit 初始化的客户端生产者实例结构体;
message 需要发送的 BUFF 消息数据, 详见 [BUFF 消息相关函数](#)。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端生产者发送 BUFF 消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightPairProducerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQLightPairProducerSetTopic;
- 7) 启动实例 TLQLightPairProducerStart;
- 8) 客户端生产者发送一条 BUFF 消息 TLQLightPairPutMessage。

4.2.7.7 TLQLightPairPutBatchMessage

【功能】客户端生产者发送一条批量消息。

【原型】 int TLQLightPairPutBatchMessage(TLQLightPairProducer pairProducer, TLQBatchMessage batchMessage);

【参数说明】

输入:

pairProducer 经过 TLQLightPairProducerInit 初始化的客户端生产者实例结构体;
batchMessage 需要发送的批量消息数据, 详见[批量消息相关函数](#)。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端生产者发送批量消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightPairProducerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQLightPairProducerSetTopic;
- 7) 启动实例 TLQLightPairProducerStart;
- 8) 客户端生产者发送一条批量消息 TLQLightPairPutBatchMessage。

4.2.7.8 TLQLightPairPutFile

【功能】 客户端生产者发送一条 FILE 消息。

【原型】 int TLQLightPairPutFile

(TLQLightPairProducer pairProducer, const char* filePathName);

【参数说明】

输入:

pairProducer 经过 TLQLightPairProducerInit 初始化的客户端生产者实例结构体;

filePathName 需要发送的文件消息的全路径, 包括路径和文件名。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端生产者发送文件消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQLightPairProducerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名

TLQLightPairProducerSetTopic;

7) 启动实例 TLQLightPairProducerStart;

8) 客户端生产者发送一条文件消息 TLQLightPairPutFile。

4.2.7.9 TLQLightPairProducerDestroy

【功能】删除客户端生产者实例。

【原型】int TLQLightPairProducerDestroy(TLQLightPairProducer* pairProducer);

【参数说明】

输入:

pairProducer 经过 TLQLightPairProducerInit 初始化的客户端生产者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者实例发送完消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

1) 首先初始化配置信息 TLQConfigInit;

2) 设置客户端连接管理节点地址 TLQConfigSetManager;

3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;

4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;

5) 创建客户端生产者实例 TLQLightPairProducerInit;

6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名

TLQLightPairProducerSetTopic;

7) 启动实例 TLQLightPairProducerStart;

8) 客户端生产者发送一条 BUFF 消息 TLQLightPairPutMessage;

9) 删除客户端生产者实例 TLQLightPairProducerDestroy。

4.2.8 点对点异步模式客户端生产者相关函数

4.2.8.1 TLQPairProducerInit

【功能】客户端生产者实例初始化函数。

【原型】int TLQPairProducerInit(TLQConfig config, TLQPairProducer* outPairProducer);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;

outPairProducer 初始化创建的客户端生产者实例。

【返回值】

0 成功;

-1 失败。

【使用说明】

用于创建一个客户端生产者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQPairProducerInit。

4.2.8.2 TLQPairProducerSetDomain

【功能】设置客户端生产者实例的属性值：域名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPairProducerSetDomain

(TLQPairProducer pairProducer, const char* domain);

【参数说明】

输入：

pairProducer 客户端生产者实例；
domain 域名。

【返回值】

0 成功；
-1 失败。

【使用说明】

设置客户端生产者实例的域名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQPairProducerInit;
- 2) 其次设置这个实例的域名属性 TLQPairProducerSetDomain。

4.2.8.3 TLQPairProducerSetGroup

【功能】设置客户端生产者实例的属性值：组名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPairProducerSetGroup

(TLQPairProducer pairProducer, const char* group);

【参数说明】

输入：

pairProducer 客户端生产者实例；
group 组名。

【返回值】

0 成功；
-1 失败。

【使用说明】

设置客户端生产者实例的组名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQPairProducerInit;

2) 其次设置这个实例的组名属性 TLQPairProducerSetGroup。

4.2.8.4 TLQPairProducerSetTopic

【功能】设置客户端生产者实例的属性值：队列名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPairProducerSetTopic
(TLQPairProducer pairProducer, const char* topic);

【参数说明】

输入：

pairProducer 客户端生产者实例；
topic 队列名。

【返回值】

0 成功；
-1 失败。

【使用说明】

设置客户端生产者实例的队列名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个生产者类型的实例如 TLQPairProducerInit;
- 2) 其次设置这个实例的队列名属性 TLQPairProducerSetTopic。

4.2.8.5 TLQPairProducerStart

【功能】客户端生产者实例启动函数。

【原型】int TLQPairProducerStart(TLQPairProducer pairProducer);

【参数说明】

输入：

pairProducer 经初始化创建的客户端生产者实例。

【返回值】

0 成功；
-1 失败。

【使用说明】

用于启动一个客户端生产者实例。

1. 调用先后顺序为：

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端生产者实例 TLQPairProducerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置队列名 TLQPairProducerSetTopic;
- 7) 启动一个实例 TLQPairProducerStart。

4.2.8.6 TLQPairPutMsgAsync

【功能】客户端生产者发送一条 BUFF 消息。

【原型】int TLQPairPutMsgAsync(TLQPairProducer pairProducer, TLQMessage* message, TLQCallBackSuccess successCallback, TLQCallBackException exceptionCallback, void* args);

【参数说明】

输入：

pairProducer 经过 TLQPairProducerInit 初始化的客户端生产者实例结构体；

message 需要发送的 BUFF 消息数据地址，详见 [BUFF 消息相关函数](#)；

successCallback 消息发送成功的回调函数接口，详见 [TLQCallBackSuccess](#) 的详细说明；

exceptionCallback 消息发送失败的回调函数接口，详见 [TLQCallBackException](#) 的详细说明；

args 需要传入回调函数接口的参数。

【返回值】

0 成功；

-1 失败。

【使用说明】

客户端生产者发送 BUFF 消息。

1. 调用先后顺序为：

1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明

2) 创建客户端生产者实例 TLQPairProducerInit；

3) 可以设置实例的一个或多个属性值，如调用接口设置队列名 TLQPairProducerSetTopic；

4) 启动一个实例 TLQPairProducerStart；

5) 客户端生产者发送一条 BUFF 消息 TLQPairPutMsgAsync。

4.2.8.7 TLQPairPutBatchMsgAsync

【功能】客户端生产者发送一条批量消息。

【原型】int TLQPairPutBatchMsgAsync(TLQPairProducer pairProducer, TLQBatchMessage batchMessage, TLQCallBackSuccess successCallback, TLQCallBackException exceptionCallback, void* args);

【参数说明】

输入：

pairProducer 经过 TLQPairProducerInit 初始化的客户端生产者实例结构体；

batchMessage 需要发送的批量消息数据，详见 [批量消息相关函数](#)；

successCallback 消息发送成功的回调函数接口，详见 [TLQCallBackSuccess](#) 的详细说明；

exceptionCallback 消息发送失败的回调函数接口，详见 [TLQCallBackException](#) 的详细说明；

args 需要传入回调函数接口的参数。

【返回值】

- 0 成功;
- 1 失败。

【使用说明】

客户端生产者发送批量消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明
- 2) 创建客户端生产者实例 `TLQPairProducerInit`;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置队列名

`TLQPairProducerSetTopic`;

- 4) 启动一个实例 `TLQPairProducerStart`;
- 5) 客户端生产者发送一条批量消息 `TLQPairPutBatchMsgAsync`。

4.2.8.8 TLQPairPutFileAsync

【功能】客户端生产者发送一条文件消息。

【原型】`int TLQPairPutFileAsync(TLQPairProducer pairProducer, const char* filePathName, TLQCallBackSuccess successCallback, TLQCallBackException exceptionCallback, void* args);`

【参数说明】

输入:

`pairProducer` 经过 `TLQPairProducerInit` 初始化的客户端生产者实例结构体;

`filePathName` 需要发送的文件消息的全路径, 包括路径和文件名。

`successCallback` 消息发送成功的回调函数接口, 详见 [TLQCallBackSuccess](#) 的详细说明;

`exceptionCallback` 消息发送失败的回调函数接口, 详见 [TLQCallBackException](#) 的详细说明;

`args` 需要传入回调函数接口的参数。

【返回值】

- 0 成功;
- 1 失败。

【使用说明】

客户端生产者发送文件消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明
- 2) 创建客户端生产者实例 `TLQPairProducerInit`;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置主题名

`TLQPairProducerSetTopic`;

- 4) 启动一个实例 `TLQPairProducerStart`;
- 5) 客户端生产者发送一条文件消息 `TLQPairPutFileAsync`。

4.2.8.9 TLQPairProducerDestroy

【功能】删除客户端生产者实例。

【原型】`int TLQPairProducerDestroy(TLQPairProducer* pairProducer);`

【参数说明】

输入:

pairProducer 经过 TLQPairProducerInit 初始化的客户端生产者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者实例发送完消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;

2) 创建客户端生产者实例 TLQPairProducerInit;

3) 可以设置实例的一个或多个属性值, 如调用接口设置队列名

TLQPairProducerSetTopic;

4) 启动一个实例 TLQPairProducerStart;

5) 客户端生产者发送一条 BUFF 消息 TLQPairPutMsgAsync;

6) 删除客户端生产者实例 TLQPairProducerDestroy。

4.2.9 点对点同步模式客户端消费者相关函数

4.2.9.1 TLQLightPairConsumerInit

【功能】客户端消费者实例初始化函数。

【原型】int TLQLightPairConsumerInit(TLQConfig config, TLQLightPairConsumer* outPairConsumer);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;

outPairConsumer 初始化创建的客户端消费者实例。

【返回值】

0 成功;

-1 失败。

【使用说明】

用于创建一个客户端消费者实例。

1. 调用先后顺序为:

1) 首先初始化配置信息 TLQConfigInit;

2) 设置客户端连接管理节点地址 TLQConfigSetManager;

3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;

4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;

5) 创建客户端消费者实例 TLQLightPairConsumerInit。

4.2.9.2 TLQLightPairConsumerSetDomain

【功能】设置客户端消费者实例的属性值: 域名, 需要在客户端实例初始化函数调用之

后才能使用此接口设置实例属性。

【原型】 `int TLQLightPairConsumerSetDomain`

`(TLQLightPairConsumer pairConsumer, const char* domain);`

【参数说明】

输入：

<code>pairConsumer</code>	客户端消费者实例；
<code>domain</code>	域名。

【返回值】

0	成功；
-1	失败。

【使用说明】

设置客户端消费者实例的域名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 `TLQLightPairConsumerInit`;
- 2) 其次设置这个实例的域名属性 `TLQLightPairConsumerSetDomain`。

4.2.9.3 TLQLightPairConsumerSetTopic

【功能】设置客户端消费者实例的属性值：队列名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】 `int TLQLightPairConsumerSetTopic`

`(TLQLightPairConsumer pairConsumer, const char* topic);`

【参数说明】

输入：

<code>pairConsumer</code>	客户端消费者实例；
<code>topic</code>	队列名。

【返回值】

0	成功；
-1	失败。

【使用说明】

设置客户端消费者实例的队列名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 `TLQLightPairConsumerInit`;
- 2) 其次设置这个实例的队列名属性 `TLQLightPairConsumerSetTopic`。

4.2.9.4 TLQLightPairConsumerSetGroup

【功能】设置客户端消费者实例的属性值：组名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】 `int TLQLightPairConsumerSetGroup`

`(TLQLightPairConsumer pairConsumer, const char* group);`

【参数说明】

输入：

<code>pairConsumer</code>	客户端消费者实例；
---------------------------	-----------

group 组名。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

设置客户端消费者实例的组名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQLightPairConsumerInit;
- 2) 其次设置这个实例的组名属性 TLQLightPairConsumerSetGroup。

4.2.9.5 TLQLightPairConsumerSetAckMode

[[功能]] 设置客户端消费者实例的属性值: 回复 ACK 模式, 需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

[[原型]] int TLQLightPairConsumerSetAckMode

(TLQLightPairConsumer pairConsumer, int manualAck);

[[参数说明]]

输入:

pairConsumer 客户端消费者实例;

manualAck 回复 ACK 的模式设置, 默认是 0 值: 自动回复 ACK 模式;

1 值: 手动回复 ACK 模式。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

设置客户端消费者实例回复 ACK 模式的属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQLightPairConsumerInit;
- 2) 其次设置这个实例的 ACK 回复模式属性 TLQLightPairConsumerSetAckMode。

4.2.9.6 TLQLightPairConsumerStart

[[功能]] 客户端消费者实例启动函数。

[[原型]] int TLQLightPairConsumerStart(TLQLightPairConsumer pairConsumer);

[[参数说明]]

输入:

pairConsumer 初始化创建的客户端消费者实例。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

用于启动一个客户端消费者实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;

- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPairConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置队列名 TLQLightPairConsumerSetTopic;
- 7) 启动一个实例 TLQLightPairConsumerStart;

4.2.9.7 TLQLightPairGetMessage

【功能】客户端消费者获取一条 BUFF 消息。

【原型】int TLQLightPairGetMessage

(TLQLightPairConsumer pairConsumer, TLQMessage* message);

【参数说明】

输入:

pairConsumer 经过 TLQLightPairConsumerInit 初始化的客户端消费者实例结构体;

message 获取到的 BUFF 消息数据。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端消费者获取一条 BUFF 消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPairConsumerInit;
- 6) 可以设置实例的一个或多个属性值，如调用接口设置队列名 TLQLightPairConsumerSetTopic;
- 7) 启动一个实例 TLQLightPairConsumerStart;
- 8) 客户端消费者获取一条 BUFF 消息 TLQLightPairGetMessage。

4.2.9.8 TLQLightPairGetFile

【功能】客户端消费者拉取一条 File 消息。

【原型】int TLQLightPairGetFile

(TLQLightPairConsumer pairConsumer, const char* filePath);

【参数说明】

输入:

pairConsumer 经过 TLQLightPairConsumerInit 初始化的客户端消费者实例结构体;

filePath 需要存储拉取到的文件的路径。

【返回值】

0 成功;
-1 失败。

【使用说明】

客户端消费者拉取一条文件消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPairConsumerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置主题名 TLQLightPairConsumerSetTopic;
- 7) 启动一个实例 TLQLightPairConsumerStart;
- 8) 客户端消费者拉取一条文件消息 TLQLightPairGetFile。

4.2.9.9 TLQLightPairAckMessage

【功能】点对点模式客户端消费者对获取到的消息手动回复确认 ACK。

【原型】int TLQLightPairAckMessage(TLQLightPairConsumer pairConsumer);

【参数说明】

输入:

pairConsumer 经过 TLQLightPairConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功;
-1 失败。

【使用说明】

点对点模式下客户端消费者获取一条消息后需要对其回复 ACK 确认消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPairConsumerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQLightPairConsumerSetTopic;
- 7) 启动一个实例 TLQLightPairConsumerStart;
- 8) 客户端消费者获取一条 BUFF 消息 TLQLightPairGetMessage;
- 9) 客户端消费者对收到的消息回复 ACK 确认消息 TLQLightPairAckMessage。

4.2.9.10 TLQLightPairAckFile

【功能】点对点模式客户端消费者对获取到的 FILE 消息手动回复确认 ACK。

【原型】 int TLQLightPairAckFile(TLQLightPairConsumer pairConsumer);

【参数说明】

输入:

pairConsumer 经过 TLQLightPairConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功;

-1 失败。

【使用说明】

点对点模式下客户端消费者获取一条 FILE 消息后需要对其回复 ACK 确认消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPairConsumerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQLightPairConsumerSetTopic;
- 7) 启动一个实例 TLQLightPairConsumerStart;
- 8) 客户端消费者获取一条 FILE 消息 TLQLightPairGetFile;
- 9) 客户端消费者收到 FILE 消息后回复 ACK 确认消息 TLQLightPairAckFile。

4.2.9.11 TLQLightPairConsumerDestroy

【功能】删除客户端消费者实例。

【原型】 int TLQLightPairConsumerDestroy(TLQLightPairConsumer* pairConsumer);

【参数说明】

输入:

pairConsumer 经过 TLQLightPairConsumerInit 初始化的客户端消费者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端消费者实例获取消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQLightPairConsumerInit;
- 6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQLightPairConsumerSetTopic;
- 7) 启动一个实例 TLQLightPairConsumerStart;

- 8) 客户端消费者获取一条 BUFF 消息 TLQLightPairGetMessage;
- 9) 客户端消费者对收到的消息做 ACK 确认回复 TLQLightPairAckMessage;
- 10) 删除客户端消费者实例 TLQLightPairConsumerDestroy。

4.2.10 点对点异步模式客户端消费者相关函数

4.2.10.1 TLQPairConsumerInit

【功能】客户端消费者实例初始化函数。

【原型】int TLQPairConsumerInit

(TLQConfig config, TLQPairConsumer* outPairConsumer);

【参数说明】

输入:

config 经过 TLQConfigInit 初始化的配置信息结构体;
outPairConsumer 初始化创建的客户端消费者实例。

【返回值】

0 成功;

-1 失败。

【使用说明】

用于创建一个客户端消费者实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息 TLQConfigInit;
- 2) 设置客户端连接管理节点地址 TLQConfigSetManager;
- 3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;
- 4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;
- 5) 创建客户端消费者实例 TLQPairConsumerInit。

4.2.10.2 TLQPairConsumerSetDomain

【功能】设置客户端消费者实例的属性值: 域名, 需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPairConsumerSetDomain

(TLQPairConsumer pairConsumer, const char* domain);

【参数说明】

输入:

pairConsumer 客户端消费者实例;
domain 域名。

【返回值】

0 成功;

-1 失败。

【使用说明】

设置客户端消费者实例的域名属性值。

1. 调用先后顺序为:

- 1) 首先初始化一个消费者类型的实例如 TLQPairConsumerInit;
- 2) 其次设置这个实例的域名属性 TLQPairConsumerSetDomain。

4.2.10.3 TLQPairConsumerSetTopic

【功能】设置客户端消费者实例的属性值：队列名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPairConsumerSetTopic

(TLQPairConsumer pairConsumer, const char* topic);

【参数说明】

输入：

pairConsumer	客户端消费者实例；
topic	队列名。

【返回值】

0	成功；
-1	失败。

【使用说明】

设置客户端消费者实例的队列名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 TLQPairConsumerInit;
- 2) 其次设置这个实例的队列名属性 TLQPairConsumerSetTopic。

4.2.10.4 TLQPairConsumerSetGroup

【功能】设置客户端消费者实例的属性值：组名，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】int TLQPairConsumerSetGroup

(TLQPairConsumer pairConsumer, const char* group);

【参数说明】

输入：

pairConsumer	客户端消费者实例；
group	组名。

【返回值】

0	成功；
-1	失败。

【使用说明】

设置客户端消费者实例的组名属性值。

1. 调用先后顺序为：

- 1) 首先初始化一个消费者类型的实例如 TLQPairConsumerInit;
- 2) 其次设置这个实例的组名属性 TLQPairConsumerSetGroup。

4.2.10.5 TLQPairConsumerSetAckMode

【功能】设置客户端消费者实例的属性值：回复 ACK 模式，需要在客户端实例初始化函数调用之后才能使用此接口设置实例属性。

【原型】 int TLQPairConsumerSetAckMode

(TLQPairConsumer pairConsumer, int manualAck);

【参数说明】

输入:

pairConsumer 客户端消费者实例;

manualAck 回复 ACK 的模式设置, 默认是 0 值: 自动回复 ACK 模式; 1

值: 手动回复 ACK 模式。

【返回值】

0 成功;

-1 失败。

【使用说明】

设置客户端消费者实例回复 ACK 模式的属性值。

1. 调用先后顺序为:

1) 首先初始化一个消费者类型的实例如 TLQPairConsumerInit;

2) 其次设置这个实例的 ACK 回复模式属性 TLQPairConsumerSetAckMode。

4.2.10.6 TLQPairConsumerStart

【功能】 客户端消费者实例启动函数。

【原型】 int TLQPairConsumerStart(TLQPairConsumer pairConsumer);

【参数说明】

输入:

pairConsumer 初始化创建的客户端消费者实例。

【返回值】

0 成功;

-1 失败。

【使用说明】

用于启动一个客户端消费者实例。

1. 调用先后顺序为:

1) 首先初始化配置信息 TLQConfigInit;

2) 设置客户端连接管理节点地址 TLQConfigSetManager;

3) 设置客户端唯一 ID 标志 TLQConfigSetClientID;

4) 选择设置客户端打印日志级别 TLQConfigSetLogLevel;

5) 创建客户端消费者实例 TLQPairConsumerInit;

6) 可以设置实例的一个或多个属性值, 如调用接口设置队列名

TLQPairConsumerSetTopic;

7) 启动一个实例 TLQPairConsumerStart;

4.2.10.7 TLQPairRegisterCallBack

【功能】 客户端消费者注册获取 BUFF 消息的回调函数接口。

【原型】 int TLQPairRegisterCallBack

(TLQPairConsumer pairConsumer, TLQCallback callback, void* args);

【参数说明】

输入:

pairConsumer 经过 TLQPairConsumerInit 初始化的客户端消费者实例结构体;
callback 获取消息的回调函数接口, 详见 [TLQCallback](#) 的详细说明。
args 需要传入回调函数接口的参数。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

客户端消费者获取 BUFF 消息需要注册这个回调函数。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPairConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQPairConsumerSetTopic;
- 4) 启动一个实例 TLQPairConsumerStart;
- 5) 客户端消费者注册获取 BUFF 消息的回调函数 TLQPairRegisterCallBack。

4.2.10.8 TLQPairUnRegisterCallBack

[[功能]] 客户端消费者注销获取消息的回调函数接口。

[[原型]] int TLQPairUnRegisterCallBack(TLQPairConsumer pairConsumer);

[[参数说明]]

输入:

pairConsumer 经过 TLQPairConsumerInit 初始化的客户端消费者实例结构体。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

客户端消费者停止获取消息需要注销这个回调函数。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPairConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQPairConsumerSetTopic;
- 4) 启动一个实例 TLQPairConsumerStart;
- 5) 客户端消费者注册获取消息的回调函数 TLQPairRegisterCallBack;
- 6) 获取一定时间后, 如 sleep(1), 停止获取消息: TLQPairUnRegisterCallBack。

4.2.10.9 TLQPairFileRegisterCallBack

[[功能]] 客户端消费者注册拉取文件消息的回调函数接口。

[[原型]] int TLQPairFileRegisterCallBack(TLQPairConsumer pairConsumer, const char* filePath, TLQCallback callback, void* args);

[[参数说明]]

输入：
pairConsumer 经过 TLQPairConsumerInit 初始化的客户端消费者实例结构体；

filePath 需要存储拉取到的文件的路径；
callback 拉取消息的回调函数接口，详见 [TLQCallback](#) 的详细说明。
args 需要传入回调函数接口的参数。

[[返回值]]

0 成功；
-1 失败。

[[使用说明]]

客户端消费者拉取文件消息需要注册这个回调函数。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPairConsumerInit；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置队列名 TLQPairConsumerSetTopic；
- 4) 启动一个实例 TLQPairConsumerStart；
- 5) 客户端消费者注册拉取文件消息的回调函数 TLQPairFileRegisterCallBack。

4.2.10.10 TLQPairAckMessage

[[功能]] 点对点模式客户端消费者对获取到的消息后手动回复确认 ACK。

[[原型]] int TLQPairAckMessage(TLQPairConsumer pairConsumer);

[[参数说明]]

输入：

pairConsumer 经过 TLQPairConsumerInit 初始化的客户端消费者实例结构体。

[[返回值]]

0 成功；
-1 失败。

[[使用说明]]

点对点模式下客户端消费者获取一条消息后需要对其回复 ACK 确认消息。

1. 调用先后顺序为：

- 1) 首先初始化配置信息和设置管理节点等配置信息，详见 4.2.1 的使用说明；
- 2) 创建客户端消费者实例 TLQPairConsumerInit；
- 3) 可以设置实例的一个或多个属性值，如调用接口设置队列名 TLQPairConsumerSetTopic；
- 4) 启动一个实例 TLQPairConsumerStart；
- 5) 客户端消费者注册获取消息的回调函数 TLQPairRegisterCallBack；
- 6) 客户端消费者对收到的消息回复 ACK 确认消息 TLQPairAckMessage；
- 7) 获取一定时间后，如 sleep(1)，停止获取消息：TLQPairUnRegisterCallBack。

4.2.10.11 TLQPairAckFile

[[功能]] 点对点模式客户端消费者对获取到的 FILE 消息手动回复确认 ACK。

【原型】 int TLQPairAckFile(TLQPairConsumer pairConsumer);

【参数说明】

输入:

pairConsumer 经过 TLQPairConsumerInit 初始化的客户端消费者实例结构体。

【返回值】

0 成功;

-1 失败。

【使用说明】

点对点模式下客户端消费者获取一条 FILE 消息后需要对其回复 ACK 确认消息。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPairConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQPairConsumerSetTopic;
- 4) 启动一个实例 TLQPairConsumerStart;
- 5) 客户端消费者注册获取文件消息的回调函数 TLQPairFileRegisterCallBack;
- 6) 客户端消费者收到文件完成后回复 ACK 确认消息 TLQPairAckFile;
- 7) 获取一定时间后, 如 sleep(1), 停止获取消息: TLQPairUnRegisterCallBack。

4.2.10.12 TLQPairConsumerDestroy

【功能】 删除客户端消费者实例。

【原型】 int TLQPairConsumerDestroy(TLQPairConsumer* pairConsumer);

【参数说明】

输入:

pairConsumer 经过 TLQPairConsumerInit 初始化的客户端消费者实例结构体地址。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端消费者实例获取消息退出应用之前需要删除该实例。

1. 调用先后顺序为:

- 1) 首先初始化配置信息和设置管理节点等配置信息, 详见 4.2.1 的使用说明;
- 2) 创建客户端消费者实例 TLQPairConsumerInit;
- 3) 可以设置实例的一个或多个属性值, 如调用接口设置队列名 TLQPairConsumerSetTopic;
- 4) 启动一个实例 TLQPairConsumerStart;
- 5) 客户端消费者注册获取消息的回调函数 TLQPairRegisterCallBack;
- 6) 客户端消费者对收到的消息回复 ACK 确认消息 TLQPairAckMessage;
- 7) 获取一定时间后, 如 sleep(1), 停止获取消息: TLQPairUnRegisterCallBack;
- 8) 删除客户端消费者实例 TLQPairConsumerDestroy。

4.2.11 BUFF 消息相关函数

4.2.11.1 TLQMessageInit

〔功能〕 BUFF 消息结构体的初始化。

〔原型〕 `int TLQMessageInit(TLQMessage* msg);`

〔参数说明〕

输入:

msg 需要发送或拉取的 BUFF 消息结构体。

〔返回值〕

0 成功;

-1 失败。

〔使用说明〕

初始化一条 BUFF 消息，在生产者发送 BUFF 消息之前需要调用此函数。

4.2.11.2 TLQMessageSetOption

〔功能〕 设置 BUFF 消息可选配置属性，如消息持久性，主题类型，优先级及有效期。

〔原型〕 `int TLQMessageSetOption(TLQMessage msg, int option, void* inVal);`

〔参数说明〕

输入:

msg 需要设置的 BUFF 消息结构体;

option 可选配置属性标识，默认值都为 0，目前支持四种:

TLQ_MESSAGE_PERSISTENCE 消息持久性，0 非持久，1 持久

TLQ_MESSAGE_TOPIC_TYPE 状态型主题 1 或事件型主题 0

TLQ_MESSAGE_PRIORITY 消息优先级，0-9，9 是最高优先级

TLQ_MESSAGE_EXPIRY 消息有效期，int32 的一个整数，单位 ms，

目前该功能不支持，只支持设置数值

inVal 可选配置属性的值。

〔返回值〕

0 成功;

-1 失败。

〔使用说明〕

初始化一条 BUFF 消息之后可以设置消息的可选属性值。

1. 调用先后顺序为:

1) 首先初始化一条 BUFF 消息 TLQMessageInit;

2) 选择配置修改 BUFF 消息的属性值 TLQMessageSetOption。

4.2.11.3 TLQMessageGetOption

〔功能〕 获取 BUFF 消息可选配置属性，如消息持久性等。

〔原型〕 `int TLQMessageGetOption(TLQMessage msg, int option, void* outVal);`

〔参数说明〕

输入:

msg 需要发送或拉取的 BUFF 消息结构体；
option 可选配置属性标识，目前只支持四种，详见 TLQMessageSetOption；
输出：
outVal 可选配置属性的值。

【返回值】

0 成功；
-1 失败。

【使用说明】

初始化一条 BUFF 消息之后可以获取消息的可选属性值。

1. 调用先后顺序为：

- 1) 首先初始化一条 BUFF 消息 TLQMessageInit；
- 2) 获取 BUFF 消息的可选属性值 TLQMessageGetOption。

4.2.11.4 TLQMessageSetData

【功能】设置 BUFF 消息的数据。

【原型】int TLQMessageSetData(TLQMessage msg, void* inData, int inDataSize);

【参数说明】

输入：

msg 需要发送的 BUFF 消息结构体；
inData 需要发送的数据体；
inDataSize 数据体的长度。

【返回值】

0 成功；
-1 失败。

【使用说明】

初始化一条 BUFF 消息之后设置消息实际需要发送的数据体。

1. 调用先后顺序为：

- 1) 首先初始化一条 BUFF 消息 TLQMessageInit；
- 2) 设置 BUFF 消息的数据体 TLQMessageSetData。

4.2.11.5 TLQMessageGetData

【功能】获取 BUFF 消息的数据。

【原型】int TLQMessageGetData(TLQMessage msg, void** outData, int* outLen);

【参数说明】

输入：

msg 需要发送或拉取的 BUFF 消息结构体；

输出：

outData BUFF 消息的数据体；
outLen BUFF 消息的数据体的长度。

【返回值】

0 成功；
-1 失败。

【使用说明】

初始化一条 BUFF 消息之后获取消息实际的数据体。

1. 调用先后顺序为：

- 1) 首先初始化一条 BUFF 消息 TLQMessageInit;
- 2) 设置 BUFF 消息的数据体 TLQMessageSetData;
- 3) 获取 BUFF 消息的数据体 TLQMessageGetData。

4.2.11.6 TLQMessageAttrSetString

【功能】设置 BUFF 消息的自定义属性的名字和值，属性类型为字符串类型。

【原型】

```
int TLQMessageAttrSetString(TLQMessage msg, const char* attrName, const char* val);
```

【参数说明】

输入：

msg	需要设置的 BUFF 消息结构体；
attrName	自定义属性的名字；
val	属性的值。

【返回值】

0	成功；
-1	失败。

【使用说明】

初始化一条 BUFF 消息之后设置消息的字符串类型的自定义属性。

1. 调用先后顺序为：

- 1) 首先初始化一条 BUFF 消息 TLQMessageInit;
- 2) 设置 BUFF 消息的字符串类型的自定义属性 TLQMessageAttrSetString。

4.2.11.7 TLQMessageAttrGetString

【功能】获取 BUFF 消息的字符串类型的自定义属性值。

【原型】

```
char* TLQMessageAttrGetString(TLQMessage msg, const char* attrName);
```

【参数说明】

输入：

msg	需要获取自定义属性的 BUFF 消息结构体；
attrName	自定义属性的名字。

【返回值】

非 NULL	返回获取到的字符串类型的属性值；
NULL	获取失败。

【使用说明】

初始化一条 BUFF 消息并设置消息的字符串类型的自定义属性后获取该属性的值。

1. 调用先后顺序为：

- 1) 首先初始化一条 BUFF 消息 TLQMessageInit;
- 2) 设置 BUFF 消息的字符串类型的自定义属性 TLQMessageAttrSetString;
- 3) 获取 BUFF 消息的字符串类型的自定义属性值 TLQMessageAttrGetString。

4.2.11.8 TLQMessageDestroy

【功能】删除 BUFF 消息结构体。

【原型】int TLQMessageDestroy(TLQMessage* msg);

【参数说明】

输入:

msg 经过 TLQMessageInit 产生的 BUFF 消息结构体。

【返回值】

0 成功;

-1 失败。

【使用说明】

删除一条 BUFF 消息，一般是在发送 BUFF 消息不成功的时候需要删除此消息。

4.2.12 批量消息相关函数

4.2.12.1 TLQBatchMessageInit

【功能】批量消息结构体的初始化，批量消息只用于生产者发送。

【原型】int TLQBatchMessageInit(TLQBatchMessage* batchMessage);

【参数说明】

输入:

batchMessage 需要发送的批量消息结构体。

【返回值】

0 成功;

-1 失败。

【使用说明】

初始化一条批量消息，在生产者发送批量消息之前需要调用此函数。

4.2.12.2 TLQBatchMessageAdd

【功能】给一条批量消息中添加多个 BUFF 消息。

【原型】int TLQBatchMessageAdd(TLQBatchMessage batchMessage, TLQMessage* message, ...);

【参数说明】

输入:

batchMessage 需要发送的批量消息结构体;

message 需要添加到批量消息中的 BUFF 消息。

【返回值】

0 成功;

-1 失败。

【使用说明】

给一条批量消息中添加一个或多个 BUFF 消息。

1. 调用先后顺序为:

1) 首先初始化一条批量消息 TLQBatchMessageInit;

2) 添加一个或多个 BUFF 消息到批量消息中 TLQBatchMessageAdd。

4.2.12.3 TLQBatchMessageDestroy

【功能】删除一条批量消息。

【原型】int TLQBatchMessageDestroy(TLQBatchMessage* batchMessage);

【参数说明】

输入:

batchMessage 经 TLQBatchMessageInit 的批量消息结构体。

【返回值】

0 成功;

-1 失败。

【使用说明】

删除一条批量消息。

1. 调用先后顺序为:

1) 首先初始化一条批量消息 TLQBatchMessageInit;

2) 添加一个或多个 BUFF 消息到批量消息中 TLQBatchMessageAdd;

3) 发送批量消息, 例如同步模式客户端生产者发送批量消息 TLQLightPutBatchMessage;

4) 删除一条批量消息 TLQBatchMessageDestroy。

4.2.13 异步模式客户端生产者发送/消费者拉取消息回调相关函数

4.2.13.1 TLQCallBackSuccess

【功能】客户端生产者发送异步消息成功时的回调函数接口。

【原型】int (*TLQCallBackSuccess)(TLQResult result, void* args);

【参数说明】

输入:

result 客户端消息发送成功后工作节点执行该回调函数传入的结果数据;

args 客户端消息发送成功后需要传入回调函数的参数。

【返回值】

0 成功;

-1 失败。

【使用说明】

客户端生产者发送异步消息时需要传入该回调函数接口, 详见 [TLQPutMessageAsync](#) 函数说明。

4.2.13.2 TLQCallBackException

【功能】客户端生产者发送异步消息失败时的回调函数接口。

【原型】int (*TLQCallBackException)(TLQException exp, void* args);

【参数说明】

输入:

exp 客户端消息发送失败后工作节点执行该回调函数传入的结果数据;
args 客户端消息发送失败后需要传入回调函数的参数。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

客户端生产者发送异步消息时需要传入该回调函数接口，详见 [TLQPutMessageAsync](#) 函数说明。

4.2.13.3 TLQCallback

[[功能]] 客户端消费者拉取异步消息时的回调函数接口。

[[原型]] `int (*TLQCallback)(TLQMessage message, void* args);`

[[参数说明]]

输入:

message 客户端消费者拉取到的 BUFF 消息;
args 客户端消费者注册拉取消息的回调函数时传入的参数。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

客户端消费者拉取异步消息注册回调函数时需要使用该接口，详见 [TLQPullRegisterCallBack](#) 函数说明。

4.2.14 异步模式客户端生产者发送消息成功或失败获取结果相关函数

4.2.14.1 TLQResultGetMessageID

[[功能]] 客户端生产者实例发送异步消息成功回调后获取消息的 ID。

[[原型]] `int TLQResultGetMessageID(TLQResult result, char** msgid, int* len);`

[[参数说明]]

输入:

result 客户端异步消息发送成功后工作节点执行回调函数传入的结果数据，回调接口详见 [TLQCallBackSuccess](#) 函数说明;

msgid 发送成功的消息 ID;

len 消息 ID 的长度。

[[返回值]]

0 成功;
-1 失败。

[[使用说明]]

客户端生产者期望发送异步消息成功回调时获取消息的 ID 可以调用该接口。

4.2.14.2 TLQExceptionGetErrMsg

〔功能〕 客户端生产者实例发送异步消息失败回调后获取失败原因。

〔原型〕 `int TLQExceptionGetErrMsg(TLQException except, char** errmsg, int* len);`

〔参数说明〕

输入：

`except` 客户端异步消息发送失败后工作节点执行回调函数传入的结果数据，回调接口详见 [TLQCallbackException](#) 函数说明；

`errmsg` 失败原因；

`len` 失败原因的长度。

〔返回值〕

0 成功；

-1 失败。

〔使用说明〕

客户端生产者期望发送异步消息失败回调时获取失败原因可以调用该接口。

附录 1 错误编码信息

列出目前会产生错误编码信息：

错误信息	错误数值	错误原因
MESSAGE TYPE INVALID.	-490	不支持的协议消息类型
API ARGS INVALID.	-446	函数传入参数有误
API MSGTYPE INVALID.	-439	函数传入参数中的消息类型有误
API ARGS CHECK INVALID.	-431	检查参数为 NULL
CONFIG TYPE INVALID.	-420	配置类型错误
DIDN'T SET TOPIC NAME OR TOPIC NAME IS INVALID.	-410	没有设置 topic
REGISTER CONSUMER MESSAGE SERIALIZE FAILED.	-409	消费者注册协议序列化失败
UNREGISTER CONSUMER MESSAGE SERIALIZE FAILED.	-408	消费者注销协议序列化失败
REGISTER PRODUCER MESSAGE SERIALIZE FAILED.	-407	生产者注册协议序列化失败
UNREGISTER PRODUCER MESSAGE SERIALIZE FAILED.	-406	生产者注销协议序列化失败

		化失败
--	--	-----

