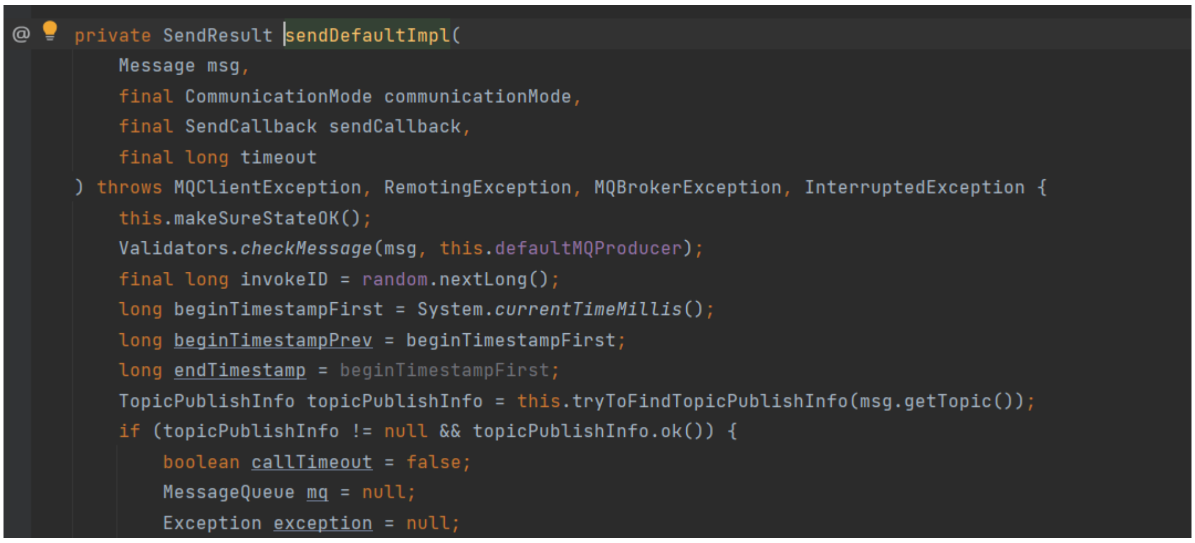
问题一：Demo没有创建Topic，却可以使用

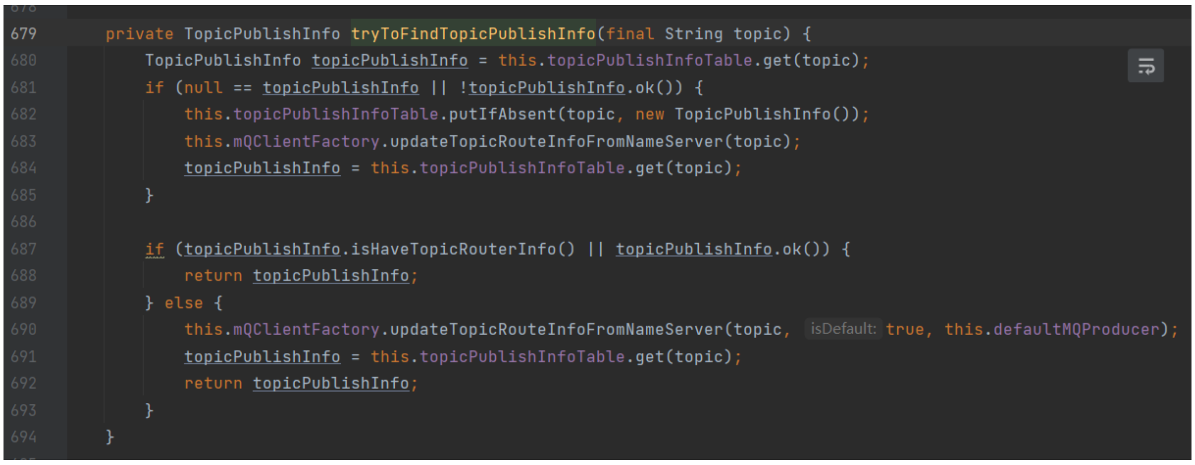
1. 首先查看.sendDefaultImpl可知



```java
private SendResult sendDefaultImpl(
        Message msg,
        final CommunicationMode communicationMode,
        final SendCallback sendCallback,
        final long timeout
) throws MQClientException, RemotingException, MQBrokerException, InterruptedException {
        this.makeSureStateOK();
        Validators.checkMessage(msg, this.defaultMQProducer);
        final long invokeID = random.nextLong();
        long beginTimestampFirst = System.currentTimeMillis();
        long beginTimestampPrev = beginTimestampFirst;
        long endTimestamp = beginTimestampFirst;
        TopicPublishInfo topicPublishInfo = this.tryToFindTopicPublishInfo(msg.getTopic());
        if (topicPublishInfo != null && topicPublishInfo.ok()) {
            boolean callTimeout = false;
            MessageQueue mq = null;
            Exception exception = null;
```

找到正确的TopicPublishInfo

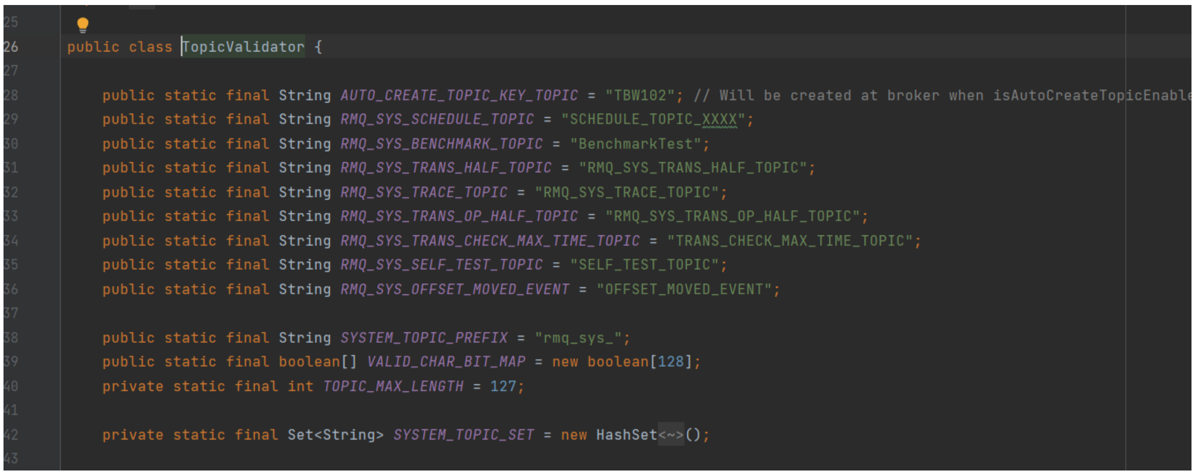
2. 进入.tryToFindTopicPublishInfo



```java
    private TopicPublishInfo tryToFindTopicPublishInfo(final String topic) {
        TopicPublishInfo topicPublishInfo = this.topicPublishInfoTable.get(topic);
        if (null == topicPublishInfo || !topicPublishInfo.ok()) {
            this.topicPublishInfoTable.putIfAbsent(topic, new TopicPublishInfo());
            this.mQClientFactory.updateTopicRouteInfoFromNameServer(topic);
            topicPublishInfo = this.topicPublishInfoTable.get(topic);
        }

        if (topicPublishInfo.isHaveTopicRouterInfo() || topicPublishInfo.ok()) {
            return topicPublishInfo;
        } else {
            this.mQClientFactory.updateTopicRouteInfoFromNameServer(topic, isDefault: true, this.defaultMQProducer);
            topicPublishInfo = this.topicPublishInfoTable.get(topic);
            return topicPublishInfo;
        }
    }
```

可以看到在broker创建的时候如果设置了isAutoCreateTopicEnable，会自动缓存TBW102 这个topic。



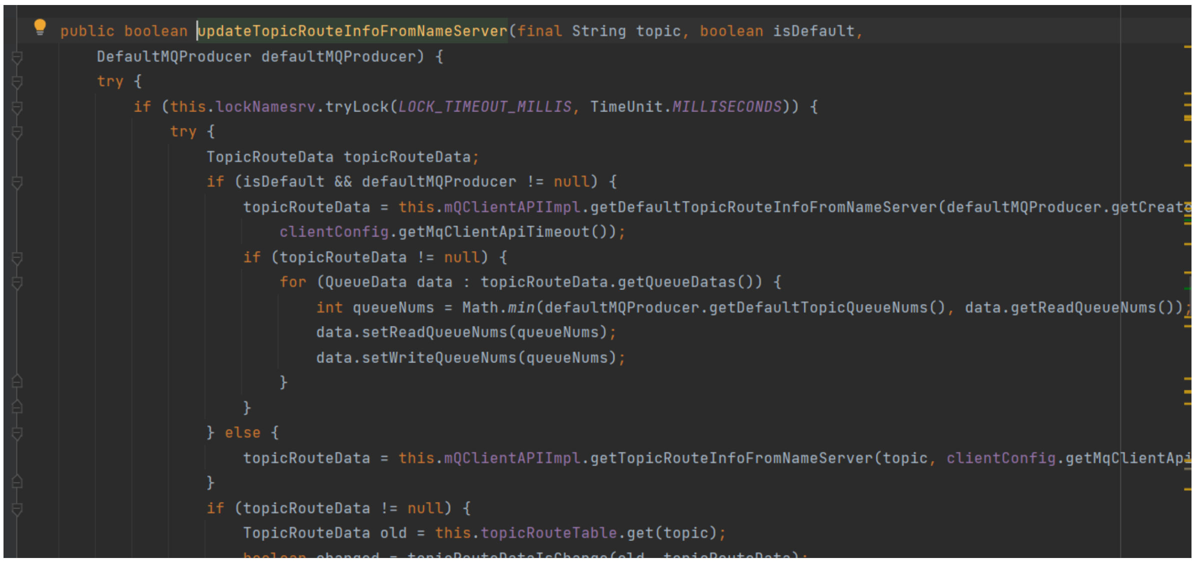
```java
public class TopicValidator {

    public static final String AUTO_CREATE_TOPIC_KEY_TOPIC = "TBW102"; // Will be created at broker when isAutoCreateTopicEnable
    public static final String RMQ_SYS_SCHEDULE_TOPIC = "SCHEDULE_TOPIC_XXXX";
    public static final String RMQ_SYS_BENCHMARK_TOPIC = "BenchmarkTest";
    public static final String RMQ_SYS_TRANS_HALF_TOPIC = "RMQ_SYS_TRANS_HALF_TOPIC";
    public static final String RMQ_SYS_TRACE_TOPIC = "RMQ_SYS_TRACE_TOPIC";
    public static final String RMQ_SYS_TRANS_OP_HALF_TOPIC = "RMQ_SYS_TRANS_OP_HALF_TOPIC";
    public static final String RMQ_SYS_TRANS_CHECK_MAX_TIME_TOPIC = "TRANS_CHECK_MAX_TIME_TOPIC";
    public static final String RMQ_SYS_SELF_TEST_TOPIC = "SELF_TEST_TOPIC";
    public static final String RMQ_SYS_OFFSET_MOVED_EVENT = "OFFSET_MOVED_EVENT";

    public static final String SYSTEM_TOPIC_PREFIX = "rmq_sys_";
    public static final boolean[] VALID_CHAR_BIT_MAP = new boolean[128];
    private static final int TOPIC_MAX_LENGTH = 127;

    private static final Set<String> SYSTEM_TOPIC_SET = new HashSet<~>();
```

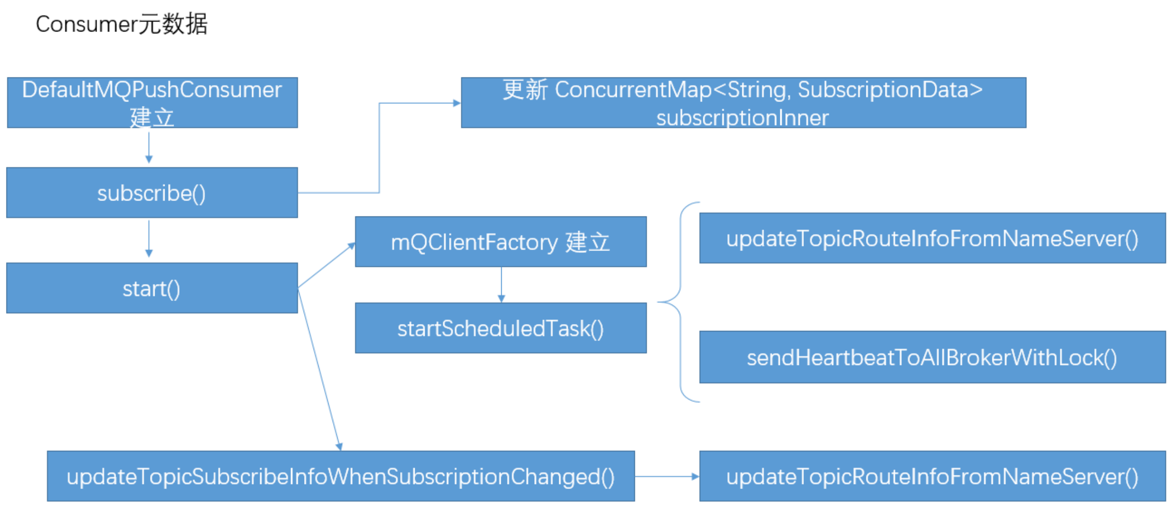
3. 进入.updateTopicRouteInfoFromNameServer



```java
public boolean updateTopicRouteInfoFromNameServer(final String topic, boolean isDefault,
    DefaultMQProducer defaultMQProducer) {
    try {
        if (this.lockNamesrv.tryLock(LOCK_TIMEOUT_MILLIS, TimeUnit.MILLISECONDS)) {
            try {
                TopicRouteData topicRouteData;
                if (isDefault && defaultMQProducer != null) {
                    topicRouteData = this.mQClientAPIImpl.getDefaultTopicRouteInfoFromNameServer(defaultMQProducer.getCreate
                        clientConfig.getMqClientApiTimeout());
                    if (topicRouteData != null) {
                        for (QueueData data : topicRouteData.getQueueDatas()) {
                            int queueNums = Math.min(defaultMQProducer.getDefaultTopicQueueNums(), data.getReadQueueNums());
                            data.setReadQueueNums(queueNums);
                            data.setWriteQueueNums(queueNums);
                        }
                    }
                } else {
                    topicRouteData = this.mQClientAPIImpl.getTopicRouteInfoFromNameServer(topic, clientConfig.getMqClientApi
                }
                if (topicRouteData != null) {
                    TopicRouteData old = this.topicRouteTable.get(topic);
                    boolean changed = topicRouteDataIsChange(old, topicRouteData);
```

可以发现在broker启动时，会将TBW102Topic的路由信息注册到nameserver，在使用TopicTest这 个Topic去获取的其实是TBW102的路由信息，并根据该路由信息构建一个TopicPublishInfo来使用。所以TopicTest的信息会发到TBW102所在的broker中。

问题二：元数据的生命周期


Consumer元数据
DefaultMQPushConsumer 建立
subscribe()
start()
更新 ConcurrentMap<String, SubscriptionData> subscriptionInner
mQClientFactory 建立
startScheduledTask()
updateTopicRouteInfoFromNameServer()
sendHeartbeatToAllBrokerWithLock()
updateTopicSubscribeInfoWhenSubscriptionChanged()
updateTopicRouteInfoFromNameServer()

Consumer订阅的Topic信息存储在`ConcurrentMap<String , SubscriptionData> subscriptionInner`结构中。
在Consumer开始运行的时候，建立了两个定时任务来维护Broker的相关信息：
`updateTopicRouteInfoFromNameServer()`从NameServer拉取Topic对应的Broker地址。
`sendHeartbeatToAllBrokerWithLock()`向所有的Broker发送心跳消息，更新自己订阅的Topic信息和其他设置。