trc-market-data-client-sdk:1.2.0-SNAPSHOT Sample Application

This is a sample application of using trc-market-data-client-sdk:1.2.0-SNAPSHOT

Dependencies

1. java 11 sdk
2. trc-market-data-client-sdk:1.2.0-SNAPSHOT

trc-market-data-client-sdk

Changelog

Version 1.2.0

- **Added**:
  - `FxInstrumentSnapshot` was introduced in place of `InstrumentSnapshot` to better represent the type of data this structure it's carrying.
  - `MarketDataCallback.onInstrumentSnapshot(MarketDataSubscriber marketDataSubscriber, FxInstrumentSnapshot instrumentSnapshot)` to replace the other callback by the same name.
  - `MarketDataCallback.onSubscriptionFailure(MarketDataSubscriber, SubscriptionDetails, SubscriptionError, String)` to replace the other callback by the same name. The parameter `SubscriptionDetails` was added to be able to identify the subscription that failed.

- **Deprecated**:

  > The following two callbacks will be removed in the next release. However, for backward compatibility, they will still be invoked during this release.

  - `MarketDataCallback.onSubscriptionFailure(MarketDataSubscriber, SubscriptionError, String)` method. Please override the other callback by the same name (mentioned above) instead.
  - `MarketDataCallback.onInstrumentSnapshot(MarketDataSubscriber, InstrumentSnapshot)` method. Please override the other callback by the same name (mentioned above) instead.
  - `InstrumentSnapshot` class is deprecated and is replaced by `FxInstrumentSnapshot`.

Subscribing to fx market data

1. Create market data listener

```
public class MarketDataListener implements MarketDataCallback {
     @Override
     public void onInstrumentSnapshot(MarketDataSubscriber
marketDataSubscriber, FxInstrumentSnapshot fxInstrumentSnapshot) {
         System.out.println("On Data :" + fxInstrumentSnapshot);
     }

     @Override
     public void onSubscriptionFailure(MarketDataSubscriber
marketDataSubscriber,
                                       SubscriptionDetails
subscriptionDetails,
                                       SubscriptionError subscriptionError,
                                       String s) {
         System.out.printf("onSubscriptionFailure. InstrumentId:%s.
ErrorCode:%s. Message:%s",
                 subscriptionDetails.getSubscribedInstrument(),
                 subscriptionError,
                 s);
     }
}
```

2. Implement connection listener

```
public class ConnectionListener implements ConnectionCallback {
    @Override
    public void onConnect(MarketDataSubscriber marketDataSubscriber) {

    }

    @Override
    public void onDisconnect(MarketDataSubscriber marketDataSubscriber,
ConnectionError connectionError,
                             String message) {

    }

    @Override
    public void onConnectionFailure(MarketDataSubscriber
marketDataSubscriber, ConnectionError connectionError,
                                    String message) {
    }
}
```

3. Subscribe for currency pairs in "onConnect" callback

```
public class ConnectionListener implements ConnectionCallback {
  @Override
  public void onConnect(MarketDataSubscriber marketDataSubscriber) {
     System.out.println("OnConnect");
     MarketDataListener marketDataListener = new MarketDataListener();
     marketDataSubscriber.subscribeFx("CHFJPY", marketDataListener);
  }
}
```

4. Create subscriber

```
MarketDataSubscriber subscriber =
MarketDataSubscriberFactory.createSubscriber();
```

5. Update trustStorePath path with client.trustStore path

```
String trustStorePath = "app/src/main/resources/client.truststore";
```

7. Connect

```
SslConfig sslConfig = new SslConfig(null, null, trustStorePath,
"gxw9dck*czu5XQW8azp");
subscriber.connect("13.250.15.157", 56100, "test-user", "test-pw",
sslConfig, connectionListener);
```

> Since this is a test client username and password will be ignored. Details about these parameters will be shared in the next release

8. Listen for data

```
On Data :InstrumentSnapshot{timestamp=2023-06-15T07:44:04.095054Z,
instrumentId='CHFJPY', bidPricePoint=PricePoint{price=153.601,
quantity=1000000}, askPricePoint=PricePoint{price=153.611,
quantity=1000000}}
On Data :InstrumentSnapshot{timestamp=2023-06-15T07:44:05.022826Z,
instrumentId='CHFJPY', bidPricePoint=PricePoint{price=153.689,
quantity=1000000}, askPricePoint=PricePoint{price=153.709,
quantity=1000000}}
On Data :InstrumentSnapshot{timestamp=2023-06-15T07:44:06.022660Z,
instrumentId='CHFJPY', bidPricePoint=PricePoint{price=153.613,
quantity=1000000}, askPricePoint=PricePoint{price=153.621,
quantity=1000000}}
On Data :InstrumentSnapshot{timestamp=2023-06-15T07:44:07.022487Z,
instrumentId='CHFJPY', bidPricePoint=PricePoint{price=153.572,
quantity=1000000}, askPricePoint=PricePoint{price=153.591,
quantity=1000000}}
```

Supported currency pairs

- GBPJPY
- CHFJPY
- AUDJPY
- CADJPY
- NZDJPY
- SGDJPY
- EURJPY
- CNHJPY
- HKDJPY
- USDJPY