معرفی زیر ساخت معاملات الگوریتمی

شرکت داتکس

پیش از هر چیز از اینکه زیر ساخت داتکس را انتخاب کرده اید از شما سپاسگزاریم.

این سند توسط شرکت تحلیلگر امید در راستای آشنایی شرکت کنندگان در مسابقه با نحوه اتصال به زیرساخت شرکت داتکس با همکاری شرکت تحلیلگر امید تهیه شده است.

توضیحات کلیدی

در این قسمت مجموعه ای از تعاریف کلی جهت آشنایی شرکت کنندگان در مسابقه با اصطلاحات رایج در در سرویس های بورسی فراهم شده است.

- **منظور از پیام چیست؟**

هر تغییری که در اطلاعات کل عمومی بازار به اصطلاح پیام های (RLC) و یا اطلاعات خصوصی بازار به اصلاح پیام های (SLE) اتفاق بیفتد تحت عنوان یک رویداد (Event) با فرمت JSON برای شما ارسال خواهد شد، مثلا با هربار تغییر در اطلاعات حقیقی/ حقوقی سهم و یا تغییر در هر یک از سفارشات یک پیام با برای شما ارسال خواهد شد.

تهران، میدان آرژانتین، خیابان بخارست،
خیابان دهم، پلاک ۲۱، طبقه‌ی سوم
(۰۲۱)۸۸۱۷۵۸۶۹-۸۸۱۷۳۲۶۸-۸۸۱۷۲۵۶۴

- **چرا پیام ها برای ما اهمیت دارند؟**

الگوریتم به تحرکات ریز در بازار و یا تغییرات ریز در سفارشات وابستگی شدیدی دارد، به عنوان مثال اگر الگوریتم شما متوجه اجرا شدن سفارشی که کمتر از یک ثانیه پیش فرستاده است، نشود و یا به اشتباه سفارشی را لغو شده در نظر بگیرد در صورتی که سفارش هنوز با وضعیت باز روی تابلو معاملات وجود دارد ممکن است بارها اقدام به ارسال سفارشات جدیدی کند. از این رو باید در زیر ساخت ارتباطی این حساسیت را در نظر گرفت و از تکنولوژی هایی با قابلیت ادامه دادن پیام از جایی که در دریافت آنها اختلال شناسایی شده "Message Resume" و یا از تکنولوژی هایی که از تایید پردازش پیام "Message Acknowledge" پشتیبانی می‌کنند استفاده کرد.

- منظور از قابلیت Message Resume در ارتباط بلادرنگ چیست؟

روشی که در آن این امکان وجود دارد که با استفاده از شماره آخرین پیام دریافتی، از سرور درخواست کرد که ادامه پیامها را برای شما بفرستد، به این ترتیب اطمینان حاصل می کنیم که هیچ پیامی از دست نمی رود.

- چگونگی می توان از روش های تایید پردازش پیام استفاده کرد؟

آشنایی با Message Broker ها: در سیستم های حساس نظیر سامانه های مالی و بانکی که پیامها از اهمیت بالایی برخوردار هستند برای تضمین تحویل و پردازش پیام از نرم افزارهای واسط جهت تبادل پیام استفاده می‌کنند، به این نرم افزارهای واسط "Message Broker"، پیام رسان و یا کارگزار پیام می‌گویند. در این روش فرستنده پیام را به Message Broker تحویل می‌دهد و پیام رسان با توجه به تنظیمات از پیش تعریف شده پیامها را در صف های پردازشی (Queue) مخصوص هر مصرف کننده قرار می‌دهد و این ضمانت را می‌دهند که تا زمانی که مصرف کننده (Consumer) این پیام را دریافت نکرده این پیام را حفظ کند. مصرف کننده پس از پردازش پیام از Message Broker می‌خواهد که این پیام را حذف کند.

** معمولا این فرایند بدون دخالت برنامه نویس و به صورت خودکار توسط پروتکل ها انجام می شود.

فرآیند استفاده از Toolkit معاملات الگوریتمی

فرآیند استفاده از Toolkit معاملات الگوریتمی شامل مراحل ذیل می‌باشد:

تهران، میدان آرژانتین، خیابان بخارست،
خیابان دهم، پلاک ۲۱، طبقه‌ی سوم
(۰۲۱)۸۸۱۷۵۸۶۹-۸۸۱۷۳۲۶۸-۸۸۱۷۲۵۶۴

۱. اتصال به زیر ساخت داتکس

۲. دریافت اطلاعات بازار

۳. مدیریت سفارشات و معاملات

۱. اتصال به زیر ساخت تحلیلگر امید

لازم به ذکر است که با استفاده از زیرساخت معاملات الگوریتمی داتکس منطق و کد مرجع برنامه می‌تواند روی سرور شخصی، رایانه شخصی و یا در سرورهای داتکس جرا شود.

- برای ارسال سفارش لازم است ابتدا در یکی از کارگزاری های طرف قرارداد داتکس ثبت نام کنید و با در دست داشتن نام کاربری و رمز حساب کاربری خود، به متصل نمودن این حساب به زیر ساخت داتکس اقدام نمایید.
- پس از فعال سازی سرویس "نام کاربری، رمز عبور، ip، port، نام صف داده های عمومی و نام صف داده های خصوصی" در اختیار شما قرار می‌گیرد، شما می‌توانید با استفاده از نمونه کدهای ارائه شده در پیوست به سرویس متصل شوید.

** فارق از اینکه شما قصد ارسال سفارش برای چه کارگزاری و یا حسابی را داشته باشید فرمت پیامهای ورودی و خروجی کاملا یکسان است.

۲. دریافت اطلاعات بازار

شرکت داتکس امکان دسترسی به دادهای Real Time و Historical را فراهم می‌کند.

۲.۱ دریافت دیتای Real-time از بازار

دیتا Real Time شامل ریز معاملات، اطلاعات تابلو سهام، عرضه و تقاضا و اطلاعات حقیقی حقوقی می‌باشد که با فرمت JSON از طریق یک صف در RabbitMQ برای شما ارسال خواهد شد، کافیست به عنوان یک Consumer به Queue ای با نام تعریف شده متصل شوید و شروع به خواندن داده های عمومی و لحظه‌ای بازار کنید و با پردازش کردن این پیام ها تصمیم به ارسال سفارش بگیرید. نمونه پیامهای داده های Real Time در ادامه تهیه شده است.

تهران، میدان آرژانتین، خیابان بخارست،
خیابان دهم، پلاک ۲۱، طبقه‌ی سوم
(۰۲۱)۸۸۱۷۵۸۶۹-۸۸۱۷۳۲۶۸-۸۸۱۷۲۵۶۴

Trade Event:


```
{
"quantity" 536
"price" 2777
"instrumentId" "IRO1MSMI0001"
"createdAt" "2019-01-02T10:36:59.000+0330"
"sellerId" "451"
"buyerId" "112"
"high" 2870
"low" 2776
"number" 950
}
```


Stock Watch Event:


```
{
"isin" "IRO3IKAZ0001"
"last" 6490
"closing" 6490
"first" 6490
"high" 6490
"low" 6490
"min" 5872
"max" 6490
"tradeValue" 8273711600
"tradeVolume" 1274840
"tradesCount" 13175
"referencePrice" 6181
"state" "A"
"event" "TRADE"
"upRangeCount" 0
"downRangeCount" 0
```

```
"lastTrade" "2019-01-02T10:38:58.000+0330"
}
```


Bid Ask Event:


```
{
"isin" "IRO1KSAD0001"
"items" [
{
"bidNumber" 1
"bidPrice" 2124
"bidQuantity" 50
"askNumber" 5
"askPrice" 2125
"askQuantity" 19586
}
{
"bidNumber" 1
"bidPrice" 2120
"bidQuantity" 956
"askNumber" 1
"askPrice" 2200
"askQuantity" 8917
}
{
"bidNumber" 1
"bidPrice" 2110
"bidQuantity" 7109
"askNumber" 1
"askPrice" 2276
"askQuantity" 5000
}
{
"bidNumber" 1
"bidPrice" 2104
"bidQuantity" 4074
"askNumber" 1
"askPrice" 2290
"askQuantity" 1710
}
{
"bidNumber" 1
"bidPrice" 2101
"bidQuantity" 2500
```

```
"askNumber" 1
"askPrice" 2798
"askQuantity" 1145
}
]
"dateTime" "2019-01-02T10:40:20.000+0330"
}
```


Clients Info Event:


```
{
"individualBuyCount" 74
"individualSellCount" 26
"individualBuyVolume" 413475
"individualSellVolume" 413475
"naturalBuyVolume" 0
"naturalSellVolume" 0
"naturalBuyCount" 0
"naturalSellCount" 0
"isin" "IRO7AZMP0001"
"dateTime" "2019-01-02T10:41:18.111+0330"
"buyerDensityValue" 28113784
"sellerDensityValue":80018864
}
```


۲.۲ دریافت اطلاعات Historical از بازار

جهت دریافت اطلاعات Historical تغییرات صف می‌توانید به آدرس های زیر مراجعه کنید:

http://history.omidanalyzer.com/BestLimits

مثال :

http://history.omidanalyzer.com/BestLimits/IRO1SIPA0001/2019/01/20190102.json

۳. مدیریت سفارشات

- برای ارسال سفارش کافیست یک پیام از جنس Order با فرمت JSON ارسال کنید.
- به محض ارسال اولین سفارش پیامهای خصوصی آن از جنس OrderNotice برای شما ارسال خواهد شد که شما را از وضعیت و تغییرات سفارشتان آگاه می‌سازد.همچنین با هر بار اجرای سفارش، پیامی مبنی بر تغییر تعداد اجرا

تهران، میدان آرژانتین، خیابان بخارست،
خیابان دهم، پلاک ۲۱، طبقه‌ی سوم
(۰۲۱)۸۸۱۷۵۸۶۹-۸۸۱۷۳۲۶۸-۸۸۱۷۲۵۶۴

شـده سفارش و تعداد باقیمانده سفارش دریافت خواهید کرد. چنانچه وضعیت سفارش به هر دلیلی تغییر کند (مثلا سفارش کامل اجرا شود و یا سفارش از طریق سامانه آنلاین لغو شود) فیلد وضعیت سفارش تغییر خواهد کرد.

- برای لغو سفارش کافیست یک پیام از جنس Cancelation با فرمت JSON ارسال کنید.

** توصیه می شود علاوه بر پردازش پیامهای دریافتی، این پیامها را در یک فایل به عنوان لاگ ذخیره کنید.

Order:

```
{
"iceberg" 0
"validityDate" null
"price" 5130
"side" "BUY"
"validity" "DAY"
"tag" "1281"
"userId" "User_id"
"quantity" 20000
"clientId" "Client_id"
"broker" "PASARGAD"
"isin" "IRO1KVRZ0001"
"senderOrderId" "sender_local_id"
}
```

Cancelation:

```
 {
"orderId" "763335"
"clientId" "Client_id"
"broker" " PASARGAD "
}
```

Order Notice:

```
{
"message" "OmsOrderChangeNotice"
"description" "Null Message"
"code" "TRADE AMOUNT ERROR"
"executed" 0
```

```
"state" "ERROR"
"level" "OMS"
"orderId" 72970
"validity" "FILL_AND_KILL"
"iceberg" 0
"price" 2560
"side" "BUY"
"cancellation" null
"tag" "357"
"userId" "User-id"
"broker" "PASARGAD"
"quantity" 449
"clientId" "ALGO-ID"
"createdAt" "2018-10-15T09:00:09.028+0000"
"isin" "IRO1BMEL0001"
"omsId" "OMS ORDER ID"
"logicBoxId" null
"senderOrderId" "YOUR_LOCAL_ORDER_ID"
}
```

نمونه کد ها

نمونه کد اتصال از طریق پایتون:

```python
#!/usr/bin/env python
import pika
connection = pika.BlockingConnection(pika.ConnectionParameters(host='ip_address'))
channel = connection.channel()
channel.queue_declare(queue='rlc-queue-name')
def callback(ch, method, properties, body):
    print(" [x] Received %r" % body)
channel.basic_consume(callback,
                      queue='rlc-queue-name',
                      no_ack=True)
print(' [*] Waiting for messages. To exit press CTRL+C')
channel.start_consuming()
```

توضیحات تکمیلی:

http://www.rabbitmq.com/tutorials/tutorial-one-python.html

نمونه کد اتصال از طریق جاوا

```java
public class Recv {
  private final static String QUEUE_NAME = "rlc_queue_name";
  public static void main(String[] argv) throws Exception {
    ConnectionFactory factory = new ConnectionFactory();
    factory.setHost("ip_address");
    Connection connection = factory.newConnection();
    Channel channel = connection.createChannel();
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);
    System.out.println(" [*] Waiting for messages. To exit press CTRL+C");
   DeliverCallback deliverCallback = (consumerTag, delivery) -> {
    String message = new String(delivery.getBody(), "UTF-8");
    System.out.println(" [x] Received '" + message + "'");
   };
   channel.basicConsume(QUEUE_NAME, true, deliverCallback, consumerTag -> { });
  }
}
```

توضیحات تکمیلی

http://www.rabbitmq.com/tutorials/tutorial-one-java.html

نمونه کد اتصال از طریق C#:

```
using RabbitMQ.Client;
```

```
using RabbitMQ.Client.Events;
using System;
using System.Text;

class Receive
{
    public static void Main()
    {
        var factory = new ConnectionFactory() { HostName = "localhost" };
        using(var connection = factory.CreateConnection())
        using(var channel = connection.CreateModel())
        {
            channel.QueueDeclare(queue: "hello",
                                 durable: false,
                                 exclusive: false,
                                 autoDelete: false,
                                 arguments: null);

            var consumer = new EventingBasicConsumer(channel);
            consumer.Received += (model, ea) =>
            {
                var body = ea.Body;
                var message = Encoding.UTF8.GetString(body);
                Console.WriteLine(" [x] Received {0}", message);
            };
            channel.BasicConsume(queue: "hello",
                                 autoAck: true,
                                 consumer: consumer);

            Console.WriteLine(" Press [enter] to exit.");
            Console.ReadLine();
        }
    }
}
```

توضیحات تکمیلی

http://www.rabbitmq.com/tutorials/tutorial-one-dotnet.html

می توانید برای دسترسی به اطلاعات بیشتر بر روی لینک های زیر کلیک کنید:

- Ruby
- PHP
- JavaScript
- Go
- Elixir
- Objective-C
- Swift
- Spring AMQP

شرکت داتکس برای تمام شما شرکت کنندگان عزیز آرزوی موفقیت دارد.

تهران، میدان آرژانتین، خیابان بخارست،
خیابـان دهـم، پـلاک ۲۱، طبقه‌ی سوم
(۰۲۱)۸۸۱۷۵۸۶۹-۸۸۱۷۳۲۶۸-۸۸۱۷۲۵۶۴