Artigo
Danusa Calixto · Ago. 10, 2022 7min de leitura

Open Exchange

Financial Information eXchange: o protocolo FIX finalmente chegou na IRIS

Depois deste GitHub, vamos ver como o protocolo FIX pode ser facilmente implementado usando a IRIS e o Python.

Se você não tem muito tempo, foque na seção Envie uma Cotação antes do Pedido no final, que explica em alguns minutos como enviar uma Solicitação de Cotação seguido por uma Solicitação de Pedido, além de mostrar o resultado do servidor em menos de cinco cliques.

IRIS-fix-protocol

A implementação do Protocolo FIX usando um contêiner Python da IRIS para o iniciador e um contêiner Python regular para o ativador.

O Protocolo Financial Information eXchange (FIX®) revolucionou o ambiente de negociações, provando ser fundamental na facilitação de várias tendências de negociações eletrônicas que surgiram na última década.

O FIX se tornou uma linguagem dos mercados globais financeiros muito usada por firmas de compra e venda, plataformas de negociação e até reguladores para comunicar informações de negócios.

Esta demonstração tem como objetivo simular um cliente FIX, permitindo ao usuário criar várias sessões conectadas a um servidor (ativador do FIX) e enviar solicitações de compra ou venda.

Requisitos e informações

- O QuickFix será instalado automaticamente na construção
- Se você usa VSCode (deve ter visto algumas janelas pop-up no canto direito) e pressionar open in container, todas as extensões necessárias serão instaladas. Esse passo não é obrigatório.

Veja a documentação geral neste site.

Veja a documentação das tags neste site.

Veja os tipos de mensagens neste site.

Demonstração

Começando a demonstração

Para iniciar a demonstração, você precisa usar Docker-compose na pasta iris-fix-protocol:

```
docker-compose up
```

Encerrando a demonstração

```
docker-compose down
```

Abrindo a demonstração

Se você **NÃO** estiver dentro do contêiner da IRIS, conecte-se à demonstração usando:

```
http://localhost:52795/csp/irisapp/EnsPortal.ProductionConfig.zen?PRODUCTION=INFORMAT
ION.QuickFixProduction
```

Username é SuperUser e Password é SYS

Se você estiver no contêiner:

```
http://127.0.0.1:52773/csp/irisapp/EnsPortal.ProductionConfig.zen?PRODUCTION=INFORMAT
ION.QuickFixProduction
```

Usando a demonstração do cliente: Iniciador

Configurações e sessões

Só uma sessão pode ser aberta pela operação Python.Fix---.

Temos uma **Sessão de Pedidos**, Python.FixOrder, que pode enviar qualquer mensagem/solicitação ao servidor.

Temos uma **Sessão de Cotações**, Python.FixQuote, que faz a inscrição automática na solicitação de dados do mercado e pode enviar ao servidor qualquer solicitação de cotação.

Se você quiser modificar os parâmetros dessa sessão, clique em Python.FixOrder ou Python.FixQuote e acesse settings na guia à direita, na parte Python e na parte %settings. Aqui, você pode inserir ou modificar quaisquer parâmetros (não se esqueça de pressionar apply depois de terminar).
Veja a configuração padrão para Fix Order:

```
BeginString=FIX.4.3
SenderCompID=CLIENTORDER
TargetCompID=SERVER
HeartBtInt=30
SocketConnectPort=3000
```

```
SocketConnectHost=acceptor
DataDictionary=/irisdev/app/src/fix/spec/FIX43.xml
FileStorePath=/irisdev/app/src/fix/Sessions/
ConnectionType=initiator
FileLogPath=./Logs/
StartTime=00:00:00
EndTime=00:00:00
ReconnectInterval=10
LogoutTimeout=5
LogonTimeout=30
ResetOnLogon=Y
ResetOnLogout=Y
ResetOnDisconnect=Y
SendRedundantResendRequests=Y
SocketNodelay=N
ValidateUserDefinedFields=N
ValidateFieldsOutOfOrder=N
```

Veja a configuração padrão para Fix Quote:

```
BeginString=FIX.4.3
SenderCompID=CLIENTQUOTE
TargetCompID=SERVER
HeartBtInt=30
SocketConnectPort=3000
SocketConnectHost=acceptor
DataDictionary=/irisdev/app/src/fix/spec/FIX43.xml
FileStorePath=/irisdev/app/src/fix/Sessions/
ConnectionType=initiator
FileLogPath=./Logs/
StartTime=00:00:00
EndTime=00:00:00
ReconnectInterval=10
LogoutTimeout=5
LogonTimeout=30
ResetOnLogon=Y
ResetOnLogout=Y
ResetOnDisconnect=Y
SendRedundantResendRequests=Y
SocketNodelay=N
ValidateUserDefinedFields=N
ValidateFieldsOutOfOrder=N
```

Se você quiser modificar os parâmetros da Inscrição no Mercado, clique em Python.FixQuote e acesse settings na guia à direita, na parte Python Adapter e na parte %settings. Aqui, você pode inserir ou modificar quaisquer parâmetros (não se esqueça de pressionar apply depois de terminar).
Veja a configuração padrão para o adaptador Fix Order:

```
subscribe=True
MarketDepth=0
MDUpdateType=0
SecurityType=FOR
md_types=0;1
symbols=EUR/USD;USD/CZK
products=4;4
```

Os símbolos e os produtos funcionam juntos e representam uma lista de símbolos + produtos do FIX, que deve ser usada assim: symbols=EUR/USD;EUR/CZK;USD/CZK products=4;4;4

Agora, ao iniciar/reiniciar, a nova configuração será aplicada e as novas sessões serão criadas.

Para criar várias sessões ativas ao mesmo tempo, você pode adicionar uma nova operação com o + perto da coluna Operation. Em Operation Class, selecione Python.FixOrderOperation ou Python.FixQuoteOperation e, em Operation Name, Python.FixOrder2, por exemplo. Agora, você precisa inserir a configuração desejada para a sessão.
Pode ser a mesma configuração padrão anterior ou qualquer outra configuração válida, mas qualquer nova sessão precisa ser adicionada também ao servidor. **Observe** que ter a mesma configuração de ID da sessão duas vezes pode causar problemas relacionados à conexão com o servidor. Veja mais informações neste site (na guia "Getting started" / "Configuration").

Enviar uma cotação

Primeiro, você precisa iniciar a demonstração, usando o botão verde Start, ou use Stop e Start novamente para aplicar as mudanças nas configurações.

Em seguida, ao clicar na operação Python.FixQuote escolhida e selecionar na guia à direita action, você pode aplicar test à demonstração.

Na janela test, selecione:

Nova cotação

Tipo de solicitação: Grongier.PEX.Message

Em classname, insira:

```
msg.Request
```

Para json, veja um exemplo de solicitação de cotação:

```
{
"header_field":
    {
        "35":"R"
    },
"group_field":
    {
        "146":
        {
            "40":"1;2",
            "55":"EUR/USD;USD/CZK"
        }
    }
}
```

Agora, você pode clicar em Visual Trace para ver nos detalhes o que aconteceu e visualizar os registros do

iniciador.

Enviar um pedido

Primeiro, você precisa iniciar a demonstração, usando o botão verde Start, ou use Stop e Start novamente para aplicar as mudanças nas configurações.

Em seguida, ao clicar na operação Python.FixOrder escolhida e selecionar na guia à direita action, você pode aplicar test à demonstração.

Na janela test, selecione:

Aqui, você pode enviar qualquer mensagem de quickfix para o servidor. Veja um exemplo de uma solicitação de ordem de compra, mas se você seguir o mesmo padrão em quase qualquer tipo de mensagem.

Nova ordem de compra Tipo de solicitação: Grongier.PEX.Message

Em classname, insira:

```
msg.Request
```

Para json, veja um exemplo de uma ordem de compra simples:

```
{
"header_field":
    {
        "35":"D"
    },
"message_field":
    {
        "55": "EUR/USD",
        "40": "1",
        "44": "100",
        "38": "10000",
        "54": "1",
        "21": "1"
    }
}
```

Agora, você pode clicar em Visual Trace para ver nos detalhes o que aconteceu e visualizar os registros do iniciador.

Envie uma Cotação antes da Ordem

Isso, que é lógica de aplicação, depende do funcionamento das especificações e de como o servidor processa as mensagens. No nosso exemplo, podemos usar Fix.BusinessProcess desta forma:

Tipo de solicitação: Grongier.PEX.Message

Em classname, insira:

```
msg.Request
```

Para json, veja um exemplo de uma ordem de compra simples:

```
{
"header_field":
    {
        "35":"D"
    },
"message_field":
    {
        "55": "EUR/USD",
        "40": "1",
        "44": "100",
        "38": "10000",
        "54": "1",
        "21": "1"
    }
}
```

É igual a uma solicitação de Ordem, mas no processo enviaremos antes de maneira automática uma solicitação de Cotação, com as informações corretas necessárias, para depois enviar a solicitação de Ordem.

Ao clicar em Visual Trace, você pode ver todas as mensagens.

Usando a demonstração do servidor: ativador

Registros

O ativador está isolado em outro contêiner. Para acessar os registros, você precisa acessar a pasta de origem iris-fix-protocol e a pasta acceptor/Logs. Aqui, você ver os registros para cada sessão.

Nova sessão do ativador

Se você conhece o protocolo FIX, você sabe que criar uma sessão no lado do cliente sem adicioná-la ao lado do servidor causará um erro, o que faz sentido, já que segurança e confiança são palavras-chave para o protocolo FIX.

Para adicionar uma nova sessão ao ativador, você precisa acessar acceptor/server.cfg e incluir as sessões que você quer adicionar no final do arquivo.

Agora, você pode inserir no terminal:

```
docker-compose up -d --build acceptor
```

Isso fechará e reiniciará o ativador, então talvez cause problemas relacionados às solicitações enviadas e aos dados armazenados.

#Embedded Python #Ferramentas #Interoperabilidade #Python #InterSystems IRIS #VSCode
Confira o aplicativo relacionado no InterSystems Open Exchange

URL de origem:https://pt.community.intersystems.com/post/financial-information-exchange-o-protocolo-fix-finalmente-chegou-na-iris