Article
Lucas Enard · Jul 27, 2022 7m read

Open Exchange

Financial Information eXchange : The FIX protocol finally on IRIS

Following this GitHub we will see how the FIX protocol can be implemented easily using IRIS and Python.

If you don't have much time focus on the Send a Quote before the Order part near the end, as it will, in a matter of minute, tell you how to send a Quote Request followed by an Order Request and show you the result from the server, and that in no more than five clicks.

IRIS-fix-protocol

Implementation of the fix protocol using an IRIS python container for the initiator and a regular python container for the acceptor.

The Financial Information eXchange (FIX®) Protocol has revolutionized the trading environment, proving fundamental in facilitating many of the electronic trading trends that have emerged over the past decade.

FIX has become the language of the global financial markets used extensively by buy and sell-side firms, trading platforms and even regulators to communicate trade information.

This demo has for objective to simulate a FIX client, allowing the user to create multiple sessions connected to a server (fix acceptor) and send buy or sell requests.

Requirements and information

- QuickFix will be installed automatically when building
- If you use VSCode, you should have seen some pop up in the right corner, if you press open in container all the needed extensions will be installed but this step is not necessary.

See this website for the general documentation.

See this website for the tags documentation.

See this website for the Message Types.

The demo

Starting the demo

To start the demo you have to use docker-compose in the iris-fix-protocol folder:

```
docker-compose up
```

Closing the demo

```
docker-compose down
```

Opening the demo

If you are **NOT** inside the IRIS container connect to the demo using :

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

The Username is SuperUser and the Password is SYS

Else, if you are inside the container :

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

Using the client demo : Initiator

Settings and Sessions

Note that only one session can be opened by Python.Fix--- operation.

We have an **Order Session**, Python.FixOrder that can send to the server any message/request.

We have a **Quote Session**, Python.FixQuote that automatically subscribe to market data request and can send to the server any quote request.

If you want to modify the parameters of these session, click on the Python.FixOrder or Python.FixQuote then go to settings in the right tab, then in the Python part, then in the %settings part.
Here, you can enter or modify any parameters (don't forget to press apply once your are done).

Here's the default configuration for the 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
```

Here's the default configuration for the 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
```

If you want to modify the parameters of the Market Subscription, click on the Python.FixQuote then go to settings in the right tab, then in the Python Adapter part, then in the %settings part.
Here, you can enter or modify any parameters (don't forget to press apply once your are done).

Here's the default configuration for the Fix Order adapter:

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

Note that symbols and product works together and represent a list of fix symbol + product, it should be used like this :
symbols=EUR/USD;EUR/CZK;USD/CZK
products=4;4;4

Now, on start/restart, the new configuration will apply and the new sessions will be created.

To create multiple sessions and them being active at the same time, you can add a new operation with the + near the Operation column.
In Operation Class select Python.FixOrderOperation or Python.FixQuoteOperation and in the Operation Name Python.FixOrder2 for example, now you need to enter the configuration you want for your session.

It can be the same as the default configuration seen earlier or any other valid configuration but don't forget that any new session needs to be added to the your server too.
Note that having two time the same session ID configuration can lead to issues concerning the connection to the server.
See this website (in the 'Getting started' / 'Configuration' tab) for more information.

Send a quote

You must first start the demo, using the green Start button or Stop and Start it again to apply your config changes.

Then, by clicking on the operation Python.FixQuote of your choice, and selecting in the right tab action, you can test the demo.

In this test window, select :

New Quote

Type of request : Grongier.PEX.Message

For the classname you must enter :

```
msg.Request
```

And for the json, here is an example of quote request :

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

Now you can click on Visual Trace to see in details what happened and see the logs of the initiator.

Send an order

You must first start the demo, using the green Start button or Stop and Start it again to apply your config changes.

Then, by clicking on the operation Python.FixOrder of your choice, and selecting in the right tab action, you can test the demo.

In this test window, select :

Here, you can send any quickfix message to the server.
Here is an example for an Order Buy request, but if you follow the same pattern almost any type of message.

New Order Buy
Type of request : Grongier.PEX.Message

For the classname you must enter :

```
msg.Request
```

And for the json, here is an example of a simple buy order :

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

Now you can click on Visual Trace to see in details what happened and see the logs of the initiator.

Send a Quote before the Order

This, is application logic, it depends on how your spec are working and how your server process messages.
In our example, we can use the Fix.BusinessProcess with this :

Type of request : Grongier.PEX.Message

For the classname you must enter :

```
msg.Request
```

And for the json, here is an example of a simple buy order :

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

It's the same a an Order request, but in the process we will automatically send a Quote request before, with the right information needed, then send the Order request.

By clicking on the Visual Trace you can see the whole messages

Using the server demo : Acceptor

Logs

The acceptor is isolated in another container, to access it's logs you must go to the source folder iris-fix-protocol then in the acceptor/Logs folder.
Here you can see the logs for each session.

New acceptor session

If you are familiar with the FIX protocol, you now that creating a session on the client side without adding it to the server side will lead to an error, which is logical as security and reliability are key words for the FIX protocol.

To add a new session to the acceptor you must go to acceptor/server.cfg and add at the end of the file the sessions you wish to add.

Now you can enter in the terminal :

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

Note that this will close the acceptor and restart it so it may lead to issues regarding sent requests and stored data

#Embedded Python #Interoperability #Python #Tools #InterSystems IRIS #VSCode
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/financial-information-exchange-fix-protocol-finally-iris