

Eze EMS

Release Date: May 14, 2025

Eze EMS xAPI Python Sample Application Guide

This document contains information about getting started with Eze EMS xAPI using Python.



Confidentiality Notice: The information included in this document is confidential information of SS&C Eze and is intended only for SS&C Eze and its affiliates, Eze EMS clients, and their respective employees.

Table of Contents

Revision History	4
Introduction	5
Eze EMS xAPI Basics	5
Eze EMS xAPI Use Restrictions	5
Eze EMS xAPI Version	5
Download EMS xAPI	6
Developer Support	6
Getting Started Using EMS xAPI Python Sample Application	7
Prerequisites	7
Installation Procedure	7
Windows	7
Linux	11
Compile Protobuf File	13
Eze EMS xAPI Sample Application Workspace	15
Initial Setup	16
Logging into EMS xAPI Python Sample Application	16
Get User Accounts	17
Placing a Single Order	18

Placing a Limit Order	19
Get Net Positions	20
Get Brokendown Positions	21
Get Todays Activity	21
Subscribe Order Info	21

Legal Information

Copyright

This document is the copyrighted work of Eze Castle Software LLC ("SS&C Eze"). SS&C Eze distributes this document pursuant to a subscription agreement containing confidentiality and license provisions and is solely for the benefit of its authorized licensees. This documentation may not be copied or transmitted, in whole or in part, in any form or by any means without the express written consent of SS&C Eze.

© 1997 to 2025 Eze Castle Software LLC. All Rights Reserved.

Content

Information in this document is subject to change without notice. In the event that you are using a version of SS&C Eze products other than the most recent version, there may be discrepancies between the content of this documentation and the operation or visual presentation of your older version of the product. SS&C Eze does not warrant that this documentation is error free.

Trademarks

SS&C Eze is a trademark of SS&C Technologies, Inc. All SS&C Eze company and product names are trademarks or registered trademarks of SS&C Technologies, Inc. or SS&C Eze.

All other company or product names mentioned herein are the trademarks or registered trademarks of their respective companies.

SS&C Eze

<http://www.ezesoft.com/>

Revision History

The table below provides a snap-shot of the updates in each revision of this document. A bar is displayed on the right side of the page to help you identify updates in the current release.

Version No.	Date	Summary of Update
v2025.3.0.6, v2025.2.1.1912	NA	There are no documentation updates in these EMS xAPI releases.
v2025.2.0.1873	Apr 07, 2025	Updated the Git links.
v2025.1.0.1822, v2024.7.1.1651, v2024.7.0.1636, v2024.6.1.1634, v2024.6.0.1615, v2024.5.0.1576, v2024.4.0.1473, v2024.2.0.1355, v2024.1.0.1209, v2023.8.1.1167, v2023.8.0.1146, v2023.7.0.887, v2023.6.0.723, v2023.5.0.947	NA	There are no documentation updates in these EMS xAPI releases.
v2023.4.0.809	Jul 12, 2023	Updated the commands for gRPC and SRP installation on the Windows and Linux operating systems.
v2023.3.0.638, v2023.2.0.478, v2023.1.0.2	NA	There are no documentation updates in these EMS xAPI releases.
v2022.8.0.4	Dec 21, 2022	Added additional information for generating protobuf files on Windows and Linux .
v2022.5.0.0	Sep 09, 2022	Updated utilities.proto and order.proto file naming convention in Compile Protobuf File .
v2022.3.0.0, v2022.2.3.0, v2022.2.2.0, v2022.2.1.0, v2022.2.0.0, v2021.9.0.0, v2021.8.1.0, v2021.7.0.0	NA	There are no documentation updates in these EMS xAPI releases.
v2021.5.0.0	Jul 16, 2021	Initial release.

Introduction

The purpose of this document is to help clients get started with the EMS xAPI application on Python UI. This document provides a step-by-step process of installing Python on Windows and Linux and start using the APIs.

Eze EMS xAPI is robust and easy-to-use application that allows programmers and trading businesses to complete various trading workflows, and also access key information, including:

- Automating order routing - to smart order routers, algorithms and other trading systems.
- Routing orders to multiple brokers, dark pools, ATS, and MTFs via the Eze EMS Global Routing Network - across asset classes.
- Staging or routing single or pairs orders.
- Accessing balances, positions, executions, and other order details.
- Accessing comprehensive list and basket capabilities.

Although EMS xAPI can operate with all gRPC compatible languages, only Python language information is provided in this document as an example. Refer to this [link](#) for more information on gRPC.

Eze EMS xAPI Basics

The Eze EMS xAPI operates in conjunction with your existing Eze EMS account permissioning and entitlements. **The Eze EMS xAPI is not a standalone data feed application that is provided to you independent of the Eze EMS.** Please contact Eze Client Service if you need to request or make changes to appropriate permissions for your account.

Eze EMS xAPI Use Restrictions

As an Eze EMS xAPI user, you are prohibited from retransmitting any Eze Market Data using the Eze EMS xAPI, without the express prior written consent of Eze EMS and the exchanges or other third-party data providers (referred to as “**Sources**” in your end user agreement). Any unauthorized retransmission of Eze Market Data is a breach of your end user agreement and will cause immediate termination of your use of the Eze EMS, Eze Market Data, and the Eze EMS xAPI.

Any non-display usage of Eze Market Data, such as use of real-time data in algorithmic trading or program trading, is subject to the rules, regulations, and policies of the applicable exchanges and additional exchange fees may apply. In addition, you may have a non-display usage of Eze Market Data even if a display of real-time data occurs. Please review your Eze EMS end user agreement, and the exchanges’ and third-party data providers’ rules, regulations, and policies that apply to your use of the Eze EMS API (which apply to Eze EMS xAPI) and/or Eze Market Data. It is the sole responsibility of the Eze EMS xAPI user and each user receiving, directly or indirectly accessing or otherwise using Eze Market Data to determine whether your receipt, access or use is reportable and/or fee liable.

Eze EMS xAPI Version

This document covers all the APIs and updates to the Eze EMS xAPI that are part of 2025.3.0.6 release.

Download EMS xAPI

Contact your SS&C Eze client service representative for downloading Eze EMS xAPI.

Developer Support

- If you are an existing Eze EMS User, [log in](#) to access developer support documentation and sample code.
- You can [contact us](#) or [request a demo](#) if you want to explore more about EMS xAPI.
- You can send us an e-mail apisupport@ezesoft.com or call +1 312-442-8122.

Getting Started Using EMS xAPI Python Sample Application

Prerequisites

Eze EMS xAPI sample application can run on Python version 3 or above.

Installation Procedure

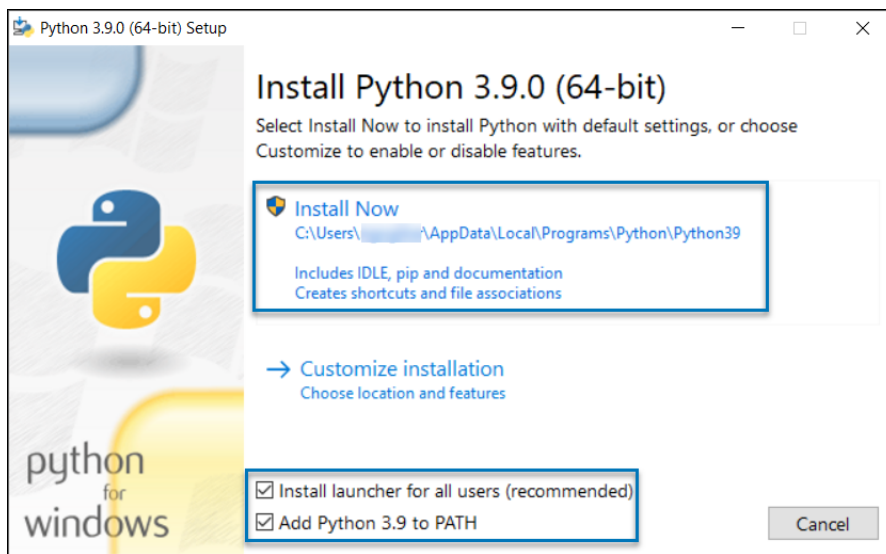
Windows

Follow the steps below to download and install Python for Windows.

1. To download the latest Python version for Windows OS visit the Python web page for [downloads](#). Download the Python version compatible with your system configuration, as shown below.



2. Run the **Python Installer** after download.
3. Ensure that the **Install Now** and **Add Python to PATH** checkboxes are enabled, as shown below. Enabling **Add Python to PATH** will place the interpreter in the execution path. Click **Install Now**.



4. To verify or know the Python version installed on the machine, run command prompt and type **Python --version** or **Python**, as shown below. Close the command prompt.

```
C:\Users\USERNAME>Python --version
```

or

```
C:\Users\USERNAME>Python
```

Follow the steps below to create EMS xAPI repository and run EMS xAPI client on Windows.

1. Run the following commands in command prompt to install **gRPC / gRPC tools** and **SRP**:

```
pip install grpcio==1.46.0
pip install grpcio-tools==1.46.0
pip install srp
pip install pycryptodomex
pip install protobuf==3.20.1
```



Note: For more information refer to this [link](#).

2. Create a folder on your local machine (e.g., C:\EMSxAPI).
3. Contact your SS&C Eze client service representative for EMS xAPI sample files (XapiClient.zip). Unzip and place all the files in the same folder you created in step 2.

4. Download **roots.pem** file:
 - a. Click open the [gRPC GitHub](#) repository.
 - b. Click **Raw**.
 - c. Right-click on the page and select **Save as...**
 - d. Save the **roots.pem** file in the same folder that was created by unzipping **xAPIClient** in step 3, as shown below. Make sure the file is saved as **roots.pem** without any additional extension and the **Save as type** must be set to **All Files (*.*)**.



5. Navigate to the **XapiClient** folder extracted in step 3, and open **xapi_client.py** file using notepad or any other text editor of your choice. Update the **host** with the actual EMS xAPI server you are trying to connect, as shown below.

```

8
9 def establishserverconnection(self):
10     ''' Function to establish connection for gRPC server '''
11     try:
12         host = 'EMSUAIXAPI.taltrade.com'
13         server_port = 9000

```



Note: Contact your SS&C Eze client service representative for more information about **host** and **port number** details.

6. Follow the steps below to run the Python UI.
 - a. Open command prompt and run the following command to navigate to the **XapiClient** folder.



Note: The command might vary based on the folder name you used in step 2.

```
>cd C:\EMSxAPI\XapiClient
```

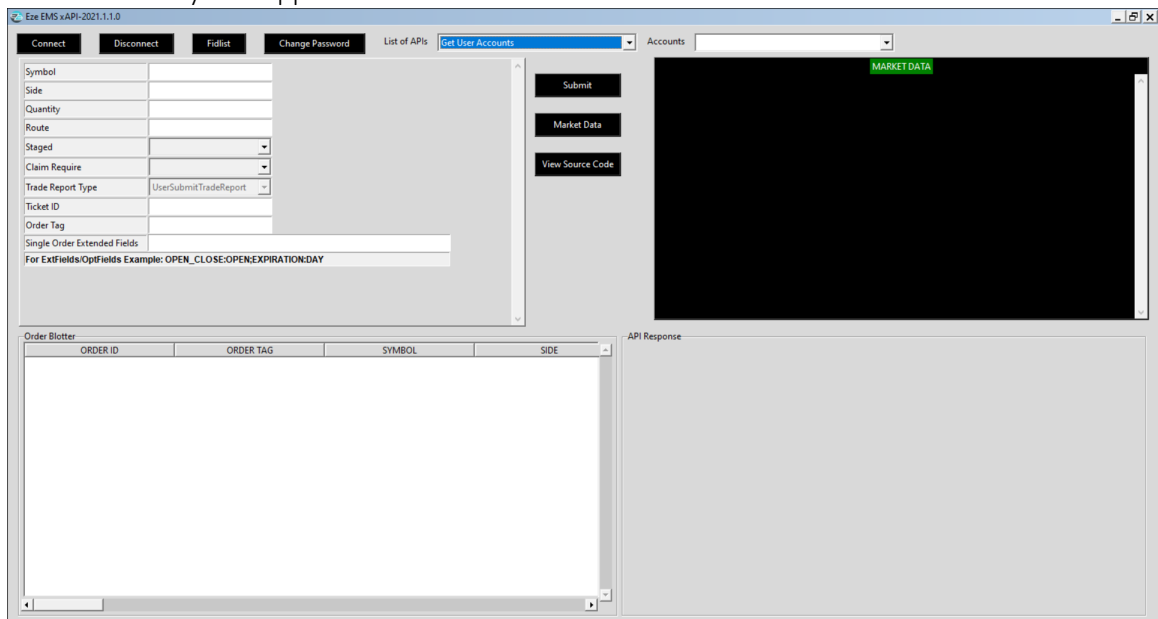
- b. Run the following command to invoke EMS xAPI Python application.

```
C:\EMSxAPI\XapiClient>python xapi_client.py
```



Note: The **roots.pem** file must be available in the same folder that you created in step 3.

7. The EMS xAPI Python application is launched.



8. Click **Connect** to login. Refer to section [Initial Setup](#) to start using the client.

Linux

Follow the steps below to install and run Python application on Linux:

Ubuntu 20.04 and other versions of Debian Linux ship with pre-installed Python 3 version. If you want to update Python version, visit this [link](#), and follow the instructions to install/upgrade Python.

1. To verify or know the Python version installed on the machine, open **Terminal** from **Applications** and type the command **python3 --version**, as shown below. Close the command prompt.

```
user@ubuntu: ~$ python3 --version
```

2. Run the following commands in Terminal to install **pip** and **tkinter** tools:

```
sudo apt install python3-pip
```

Enter your password [sudo] password for user:



Note: If the following prompt is displayed: **Do you want to continue? [Y/n]**. Type **Y**.

```
sudo apt-get install python3-tk
```



Note: If the following prompt is displayed: **Do you want to continue? [Y/n]**. Type **Y**.

3. Run the following commands in Terminal to install **gRPC/gRPC tools** and other required packages:

```
pip3 install grpcio==1.46.0
pip3 install grpcio-tools==1.46.0
pip3 install --upgrade protobuf
pip3 install srp
pip3 install pycryptodomex
pip3 install protobuf==3.20.1
```



Note: For more information refer to this [link](#).

4. Create a folder on your desktop (e.g., EMSxAPI).
5. Contact your SS&C Eze client service representative for EMS xAPI sample files (XapiClient.zip). Unzip and place all the files in the same folder you created in step 4.
6. Follow the steps below to download **roots.pem** file.
 - a. Click open the [gRPC GitHub](#) repository.
 - b. Click **Raw**.
 - c. Right-click on the page and select **Save Page As...**
 - d. Save the **roots.pem** file in the same folder that was created by unzipping **xAPIClient** in step 5, as shown below. Make sure the file is saved as **roots.pem** without any additional extension and select **All Files**.



Note: Contact your SS&C Eze client service representative for more information.

7. Navigate to **XapiClient** folder you created in step 5, and open **xapi_client.py** file using text editor or any other editor of your choice (e.g., vim, nano, etc.). Update the **host** with the actual EMS xAPI server you are trying to connect, as shown below.

```

8
9 def establishserverconnection(self):
10     ''' Function to establish connection for gRPC server '''
11     try:
12         host = 'EMSUATXAPI.taltrade.com'
13         server_port = 9000

```



Note: Contact your SS&C Eze client service representative for more information about **host** and **port number** details.

8. Follow the steps below to run the Python UI.

a. Run the following command in Terminal to navigate to the **XapiClient** folder.



Note: The command might vary based on the folder name you used in step 4.

```
cd ~/Desktop/EMSxAPI
```

b. Run the following command to invoke EMS xAPI Python application.

```
$ python3 xapi_client.py
```



Note: The **roots.pem** file must be available in the same folder that you created in step 5.

9. The EMS xAPI Python application is launched.

10. Click **Connect** to login. Refer to section [Initial Setup](#) to start using the client.

Compile Protobuf File

You can compile the protobuf in your preferred language. Follow the steps below to compile the protobuf file in Python language:

1. Create a **Protobuf** folder on your local machine (e.g., C:\EMSxAPI\Protobuf).
2. Download the **utilities.proto** file from [GitHub](#) or contact your SS&C client service representative for the file. Place this file in the Protobuf folder you created in above step.
3. Open command prompt and execute the following command.

```
python -m grpc_tools.protoc -I .\Protobuf --python_out=. --grpc_python_out=.\Protobuf\utilities.proto
```



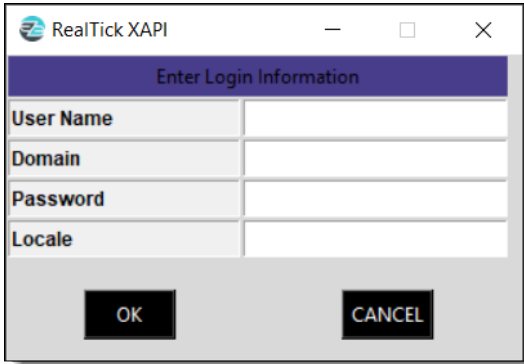


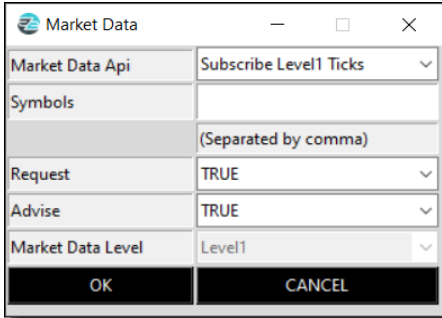
Note: Replace **.\Protobuf** with your local machine directory path. There are two (2) instances of it in the command.

This generates **utilities_pb2.py** and **utilities_pb2_grpc.py** files. The **utilities_pb2.py** file contains the request and response classes and **utilities_pb2_grpc.py** contains client and server classes.

Follow the same procedure for compiling **order.proto** and **market_data.proto** files in Python language. For more information on using Python with a gRPC API, refer to this [link](#).

Eze EMS xAPI Sample Application Workspace

The Python user interface provides you features to use APIs. The table below provides information about main features available on EMS xAPI Python Sample Application.

Feature	Description
Connect	<p>Click to login. Fill in the credentials and click OK. On successful login you will receive a user token.</p>  <p> Note: Contact your SS&C Eze client service representative for any issues related to login.</p>
Disconnect	Click to end the connection. Once disconnected the user token gets expired, you need to login again to use the client.
FidList	<p>Click to find additional information on extended and optional fields.</p> <p> Note: Contact your SS&C Eze client service representative to learn more about FidList and Extended Fields.</p>
Market Data	<p>Click to open Market Data window. This window provides you an option to subscribe/unsubscribe the Level 1 and 2 data.</p> 

Feature	Description
List of APIs	Click to view the API's in a dropdown list.
Accounts	Click to view the list of available user accounts. These accounts are used to run the API's.
Submit	Click to submit an order.
View Source Code	Click to view the source code of API. Select an API from List of APIs or Market Data window and click View Source Code .
Order Blotter	Order Blotter provides you the complete details of order fills. You can Change or Cancel an order by selecting it in this window.
API Response	This window provides the data for API's requested today. Select a request API from List of APIs and click Submit to view data of that API.

Initial Setup

Logging into EMS xAPI Python Sample Application

To log in to the EMS xAPI Python Sample Application.

1. Click **Connect**.

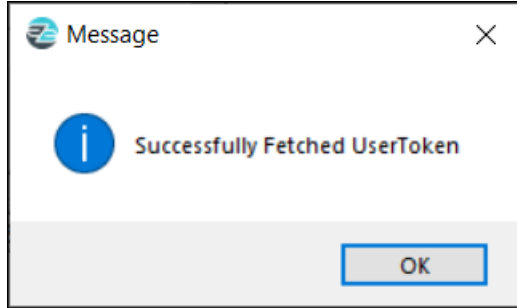


2. Fill in your login credentials.



Note: Contact your SS&C Eze client service representative for any issues related to login.

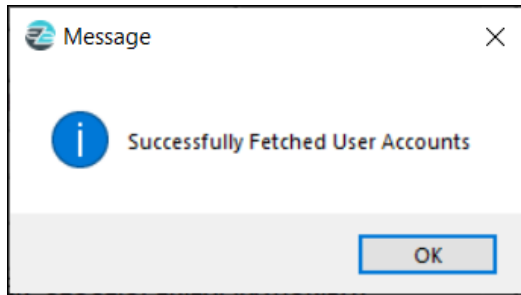
3. On successful login you will receive a user token, as shown below.



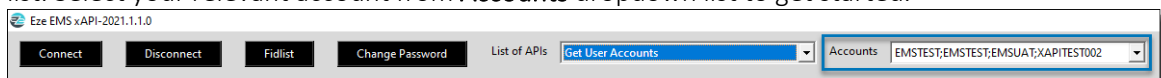
Get User Accounts

Follow the steps below to fetch your list of user accounts.

1. Click **List of APIs** dropdown and select **Get User Accounts** API and click **Submit** to fetch your available accounts.



2. On successful submission you can see the account details populated in the **Accounts** dropdown list. Select your relevant account from **Accounts** dropdown list to get started.



You have now successfully logged in on EMS xAPI Python sample application with specific user account.

Placing a Single Order

Follow the steps below to place a single order.

1. To place an order, fill in the **Symbol, Side, Quantity, Route** etc., as shown below.

The screenshot shows the Eze EMS xAPI-2021.1.1.0 application window. At the top, there are buttons for 'Connect', 'Disconnect', 'Fidlist', and 'Change Password'. To the right, there is a 'List of APIs' dropdown menu and a 'Get User Accounts' button. The main area contains a form with the following fields:

Symbol	AAPL
Side	BUY
Quantity	1000
Route	DEMO
Staged	<input type="checkbox"/>
Claim Require	<input type="checkbox"/>
Trade Report Type	UserSubmitTradeReport
Ticket ID	
Order Tag	
Single Order Extended Fields	

Below the form, there is a text box with the example: `For ExtFields/OptFields Example: OPEN_CLOSE:OPEN;EXPIRATION:DAY`.

- a. **Symbol:** Valid ticker symbol. Example: AAPL, IBM or VOD.LSE etc.
 - b. **Side:** BUY, BUYMINUS, CROSS, CROSSSHORT, CROSSSHORTEXEMPT, NONE, SELL, SELLPLUS, SELLSHORT, SELLSHORTEXEMPT
 - c. **Quantity:** Value > 0
 - d. **Route:** Any permissioned route such as CITI-ALGO, BNP PAIRS etc.
 - e. **Staged:** TRUE or FALSE (Note: in order to send a staged order, this field becomes mandatory and has to be set as TRUE only)
 - f. **Claim Require:** TRUE or FALSE (Note: setting TRUE value envisages a user running Eze EMS who then claims the Order so it can switch from Pending to Live State)
 - g. **Trade Report Type:** UserSubmitTradeReport = 0; ForeignExecution = 1
 - h. **Ticket ID:** Ticket ID
 - i. **Order Tag:** Order Tag
 - j. **Single Order Extended Fields:** Refer the `fidlist.txt` file in [xAPIclient](#) sample files.
2. Select **Submit Single Order** in **List of APIs** dropdown list. Click **Submit**. A success message is displayed, as shown below.



Note: By default, the order is sent as a Market order. Refer [Limit Order](#) section to change order type.

- You can view the submitted order details in the **Order Blotter**. Select **SubscribeOrderInfo** from **List of APIs** dropdown and click **Submit**.

A screenshot of a window titled 'Order Blotter'. It contains a table with four columns: ORDER ID, ORDER TAG, SYMBOL, and SIDE. The table lists four orders, all with a 'BUY' side. The first two orders are for AAPL, and the next two are for IBM. The ORDER TAG column only has entries for the IBM orders.

ORDER ID	ORDER TAG	SYMBOL	SIDE
a25504fc-74-000k		AAPL	BUY
a25504fc-74-001g		AAPL	BUY
a86504fc-74-0002-4	20210310-0108-000040	IBM	BUY
a86504fc-74-0002	20210310-0108-000040	IBM	BUY

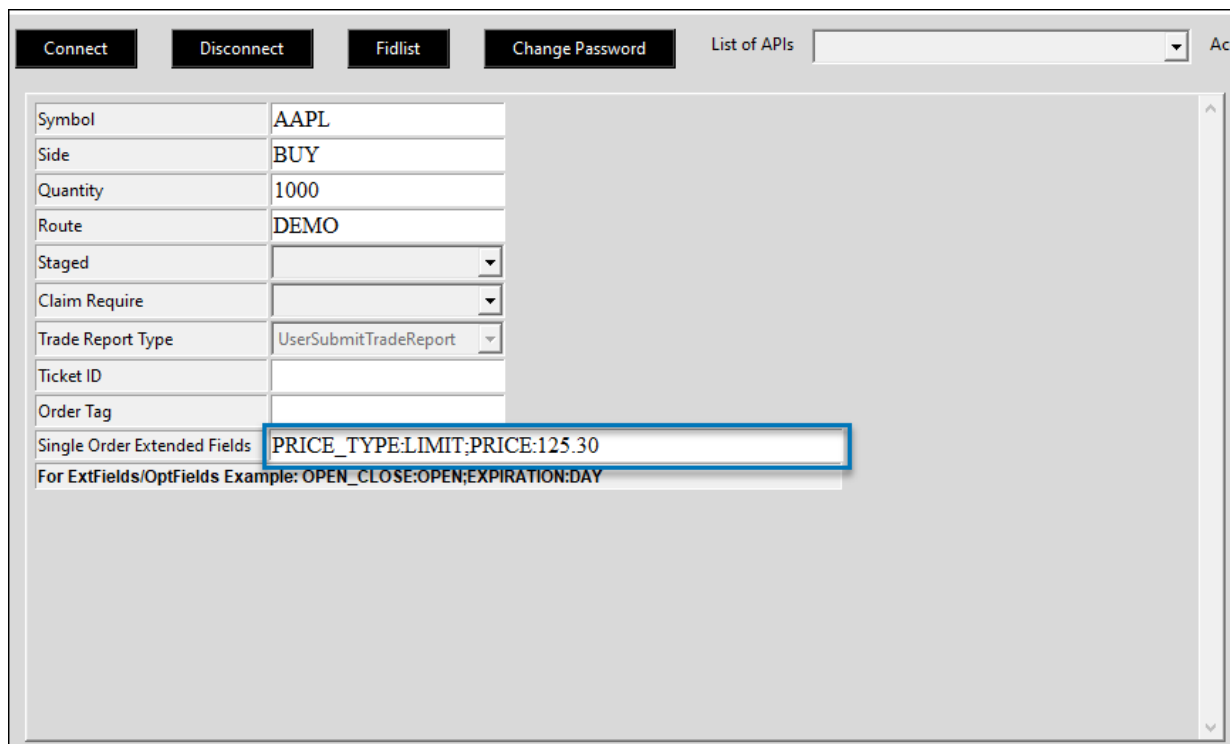
Placing a Limit Order

Follow the steps below to place a limit order.

To place a limit order, fill in the **Symbol**, **Side**, **Quantity**, **Route** etc. You must enter the **PRICE_TYPE** and **PRICE** in **Single Order Extended Fields** e.g., `PRICE_TYPE:LIMIT;PRICE:125.30`, as shown below. The valid values are **LIMIT** for limit order and **MARKET** for market order.



Note: The Price field is mandatory when **PRICE_TYPE** is specified as **LIMIT**.

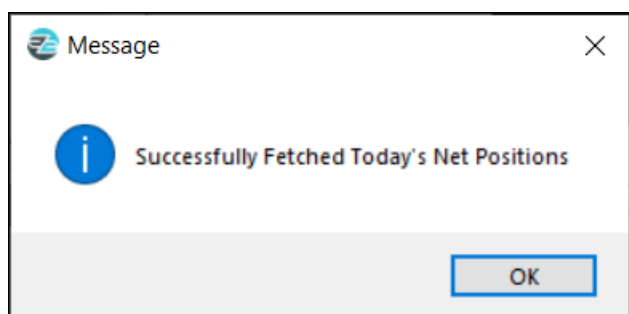


The screenshot shows a web application interface for EZE xAPI. At the top, there are buttons for 'Connect', 'Disconnect', 'Fidlist', and 'Change Password'. To the right is a 'List of APIs' dropdown menu. Below these is a form with the following fields:

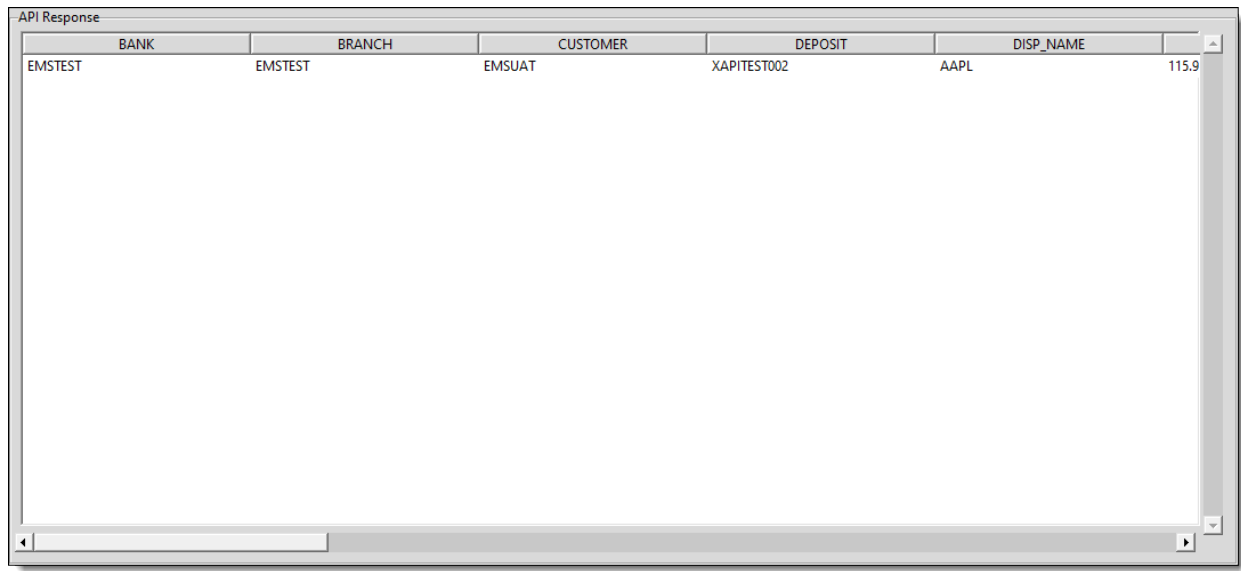
Symbol	AAPL
Side	BUY
Quantity	1000
Route	DEMO
Staged	<input type="checkbox"/>
Claim Require	<input type="checkbox"/>
Trade Report Type	UserSubmitTradeReport
Ticket ID	
Order Tag	
Single Order Extended Fields	PRICE_TYPE:LIMIT;PRICE:125.30
For ExtFields/OptFields Example: OPEN_CLOSE:OPEN;EXPIRATION:DAY	

Get Net Positions

To fetch today's net positions, select **Get Today's NetPositions** from **List of APIs** and click **Submit**. A success message is displayed, click **OK**.



You can view the order details on the **API Response** blotter, as shown below. Refer EMS xAPI Technical Documentation for API response details.



The screenshot shows a window titled "API Response" containing a table. The table has six columns: BANK, BRANCH, CUSTOMER, DEPOSIT, DISP_NAME, and a final column with the value 115.9. The data row contains the following values: EMSTEST, EMSTEST, EMSUAT, XAPITEST002, AAPL, and 115.9.

BANK	BRANCH	CUSTOMER	DEPOSIT	DISP_NAME	
EMSTEST	EMSTEST	EMSUAT	XAPITEST002	AAPL	115.9

Get Brokdown Positions

To fetch brokdown positions, select **Get Today's BrokdownPositions** from **List of APIs** and click **Submit**. A success message is displayed, click **OK**. You can view the details on the **API Response** blotter. Refer to the "EMS xAPI Technical Documentation" for API response details.

Get Todays Activity

To fetch todays activity, select **Get Today's Activity** from **List of APIs** and click **Submit**. A success message is displayed, click **OK**. You can view the details on the **API Response** blotter. Refer to the "EMS xAPI Technical Documentation" for API response details.

Subscribe Order Info

To subscribe for order info, select **Subscribe Order Info** from **List of APIs** and click **Submit**. A success message is displayed, click **OK**. You can view the details on the **API Response** blotter. Refer to the "EMS xAPI Technical Documentation" for API response details.