



Netfonds Bank AS

API Gateway

Version 2.3 20180903

Pia Jakobsen
Netfonds Bank AS

CHANGE LOG

Issue	Date	Change
0.1	20150826	First draft
0.9	20150904	Final revision
1.0	20150908	Publication
1.01	20150908	P4: 8500 → 8405, P26: orders not available in beta.
1.1	20151026	New commands: index-members, winners, losers
1.2	20151210	4.2.6: Example of order operations
1.3	20151215	2.3 Two-factor authentication (BankID)
1.4	20160225	New commands: available, active-orders, trades
1.5	20160407	1) Subscribe to individual nftp-objects (new optional option :feed for subscribe, current-snapshot and unsubscribe) 2) added the NFTP object FUND. 3) Fixed some minor errors in this document.
1.6	20160616	New command: history. New NFTP object: HISTORY.
1.7	20160825	New command: list-account
1.8	20161128	New command: list-shortable New command: account-available New NFTP-object: extended-security New options for enter-order: short, check-only.
1.9	20170118	New NFTP object: TRADE-INFORMATION
2.0	20170127	The API now supports news messages (releases) New NFTP objects: RELEASE, EXTENDED-SPREAD New commands: subscribe-release, unsubscribe-release, spreads, list-securities
2.1	20170511	4.2.2: new order methods: stop-loss, invisible, iceberg, mid-spread, FOK, FAK and GTD orders.
2.2	20171023	Support for hierarchical users. New commands: change-user, who-am-i
2.3	20180903	Minor corrections: 2.1, 3.3.2.24 and 4.2.2

DISCLAIMER

The API and this document and its content are provided “as is”. While Netfonds Bank AS will try to ensure that the information is correct and up to date, Netfonds Bank AS is not liable for any damages or decisions made in reliance on it. The information herein is for guidance only.

TABLE OF CONTENTS

CHANGE LOG.....	2
DISCLAIMER.....	2
1. INTRODUCTION.....	5
2. ENABLEMENT AND CONNECTIVITY.....	5
2.1 SSL.....	5
2.2 ACCOUNTS WITH TWO-FACTOR AUTHENTICATION METHODS (BankID™).....	6
2.3 PUBLIC CHANNEL.....	6
2.4 PRIVATE CHANNEL.....	6
3. THE PROTOCOL.....	6
3.1 SYNTAX.....	6
3.2 CLIENT COMMANDS.....	7
3.3 SERVER RESPONSE.....	7
3.3.1 NFTP TYPES.....	9
3.3.2 NFTP OBJECTS.....	9
3.3.2.1 QUOTES (message type = 1).....	11
3.3.2.2 TRADE (message type = 3).....	12
3.3.2.3 EXTENDED-SPREAD (message type = 6).....	12
3.3.2.4 SECURITY (message type = 20).....	13
3.3.2.5 MARKET BY PRICE (message type = 22).....	14
3.3.2.6 EXCHANGE (message type = 24).....	15
3.3.2.7 RELEASE (message type = 25).....	16
3.3.2.8 HISTORY (message type = 27).....	17
3.3.2.9 AUTHORIZED (message type = 30).....	17
3.3.2.10 EXCHANGE-STATE (message type = 48).....	18
3.3.2.11 CUSTOMER-ORDER (message type = 53).....	18
3.3.2.12 CUSTOMER-DEAL (message type = 54).....	19
3.3.2.13 TRADER-AVAILABLE (message type = 62).....	20
3.3.2.14 USER-MESSAGE (message type = 70).....	21
3.3.2.15 EXTENDED-PAPER-SUMMARY (message type = 82).....	21
3.3.2.16 INDEX-WEIGHT (message type = 91).....	21
3.3.2.17 ERRMSG (message type = 99).....	22
3.3.2.18 EXTENDED-SECURITY (message type = 120).....	22
3.3.2.19 FUND (message type = 126).....	23
3.3.2.20 EXTENDED-AVAILABLE (message type = 131).....	25
3.3.2.21 HEARTBEAT (message type = 135).....	27
3.3.2.22 TRADE-INFORMATION (message type = 146).....	28
3.3.2.23 API-RESPONSE (message type = 193).....	28
3.3.2.24 API-ORDER-ENTRY (message type = 194).....	29
3.3.2.25 API-ORDER-CHANGE (message type = 195).....	29
3.3.2.26 POOL-CONTENT (message type = 311).....	29
3.4 EXCHANGES.....	31
4. THE API.....	32
4.1 THE PUBLIC CHANNEL.....	33
4.1.1 LOGIN.....	33
4.1.2 CURRENT-SNAPSHOT.....	33
4.1.3 CURRENT-QUOTES ¹	34
4.1.4 SUBSCRIBE.....	35

4.1.5 UNSUBSCRIBE.....	35
4.1.6 INDEX-MEMBERS.....	36
4.1.7 WINNERS.....	36
4.1.8 LOSERS.....	36
4.1.9 TRADES.....	37
4.1.10 HISTORY.....	37
4.1.11 LIST-SHORTABLE.....	38
4.1.12 SUBSCRIBE-RELEASE.....	38
4.1.13 UNSUBSCRIBE-RELEASE.....	39
4.1.14 SPREADS.....	39
4.1.15 LIST-SECURITIES.....	39
4.1.16 CHANGE-USER.....	39
4.1.17 WHO-AM-I.....	40
4.1.18 QUIT.....	40
4.2 THE PRIVATE CHANNEL.....	41
4.2.1 LOGIN.....	41
4.2.2 ENTER-ORDER.....	42
4.2.3 CANCEL-ORDER.....	42
4.2.4 CHANGE-ORDER.....	43
4.2.5 ACTIVE-ORDERS.....	43
4.2.6 AVAILABLE* (Obsolete).....	43
4.2.7 LIST-ACCOUNT.....	44
4.2.8 ACCOUNT-AVAILABLE.....	44
4.2.9 TEST-MODE.....	45
4.2.11 CHANGE-USER.....	46
4.2.12 WHO-AM-I.....	46
4.2.13 QUIT.....	46
4.2.13 EXAMPLE OF ORDER OPERATIONS.....	47
5. ERROR MESSAGES.....	48
6. RELEASE NOTES.....	49
6.1 NFAPI Version 2.0.....	49
7. FINAL WORDS.....	50

1. INTRODUCTION

The Netfonds API is designed for customers who need access to “raw” trade and security data for integration with their own software. This API is in some ways similar to our trading platform PrimeTrader but without the encased GUI. Clients may subscribe to various instruments and get instant real-time updates in return. Clients may also issue buy or sell orders which will be relayed immediately.

It is assumed that the reader has a reasonable understanding of protocol integration and some skills with software development. While the protocol and the API is made as simple as possible, it still requires a fair bit of technical understanding. Netfonds will not be able to assist you with the integration on the client side other than answering questions about the protocol. Netfonds is also free to terminate or temporarily halt your API-subscription at any time if your client doesn't meet the necessary standards.

2. ENABLEMENT AND CONNECTIVITY

To get access to the API, you must first sign the API agreement found on our web page here:

<https://www.netfonds.no/account/kill-form.php?type=api-order>

You must also have real-time subscriptions with full depths (level 2¹) on the exchanges you want to operate on. This can be ordered here:

<https://www.netfonds.no/account/subscribe.php>

Subscriptions are not necessary for testing connectivity and for exchanges that don't require subscriptions, like eg. NGM.

Once the API agreement is accepted, you are ready to connect. This is done by using SSL encapsulated TCP sockets to these addresses

Public channel (production)	nfapi.netfonds.no port 8400 - 8404
Private channel (production)	nfapi.netfonds.no port 8405

The public channel has 5 ports available. It is advisable that you pick one port at random to spread the load. They are otherwise completely identical.

2.1 SSL

The Secure Socket Layer is necessary for connecting to the API. This ensures that the data transmitted between the client and the server is encrypted. You need to check the documentation for your integration language on how to enable this. To test connectivity and the basics of the API protocol, you can use

```
% openssl s_client -connect nfapi.netfonds.no:<port>
```

This opens a “telnet” connection to the API where you can type in commands and see some results. OpenSSL™ is freely available on most machines and operating systems but other implementations will probably do just fine.

(¹ Level 1.5 on Oslo Stock Exchange (OSE) if you are *not* a professional actor)

2.2 ACCOUNTS WITH TWO-FACTOR AUTHENTICATION METHODS (BankID™)

If your account uses BankID or other token-based authentication methods, your client will need a one time password in addition to your normal credentials to log in. This is done by first logging in on our web pages (www.netfonds.no) and then click on “Temporary password” (“Engangspassord”) on the menu to the left. This gives you a one time password which can be used by your client.

2.3 PUBLIC CHANNEL

The public channel is for real time dissipation of securities, quotes and public trades. You subscribe to a pair of (ticker, exchange) to start the ticker feed. The feed data is (not limited to):

- Security information (Ticker, ISIN, last trade date, previous price, ...)
- Exchange information (Opening hours, next trade days, closed, open, ...)
- Public trades (Price, quantity, counterparties, ...)
- Quotes (Bid, offer, open price, high price, low price, last price, value, ...)
- Market by price (Order-book)

See below for a more thorough description of the data. The feed is continuous meaning that your client must loop through the data until you either unsubscribe from the ticker feed or log out. You can subscribe to several tickers during a session.

2.4 PRIVATE CHANNEL

The private channel is for your orders. Once connected you will get a order feed on all your open orders. This is also the channel where you place, amend or cancel your orders.

3. THE PROTOCOL

The protocol used for client communication is a “string based” protocol based on limited S-expressions. Although much less well known than eg. XML, it is an easy protocol to implement.

3.1 SYNTAX

The formal syntax of our (limited) S-expressions are:

```
value ::= <floating point> | <integer> | <string> | <keyword> | <symbol> |
      <list>

floating point ::= <integer>? '.' <integer>

integer ::= '-'? <positive integer>

positive integer ::= [0-9]+

string ::= '"' <character>* '"'

character ::= { anything but newline, '"' and '\' } | '\\" | '\\'
```

```

symbol ::= [A-Za-z][A-Za-z0-9]*
keyword ::= ':' <symbol>
list ::= '(' <value>* ')'

```

Some examples:

```

A string:           "Hello world"
An integer:         42
A floating point number: 3.141459
A symbol:           foo
A keyword:          :BAR
A list of something (foo :bar "Hello world" 42 (3.14159))

```

3.2 CLIENT COMMANDS

All commands from the client to the server (hereafter known as “client commands” or just “commands”) has the following syntax

```

command ::= '(' <symbol> <parameter> ')'
parameter ::= (<keyword> <value>)*

```

Examples of commands are:

```

(login :username "foo" :password "bar")
(subscribe :paper "TEL" :exchange "OSE")
(whoami)
(enter-order :paper "TEL" :exchange "OSE" :type "B" :quantity 100)

```

So in less formal terms, all commands are enclosed in parenthesis with the first symbol being the command name followed by key-value pairs of parameters. The order of the parameters is interchangeable.

The commands and their keywords are case-insensitive; upper and lower case can be used interchangeably. Strings are (obviously) case-sensitive; “foo” and “FOO” are not identical.

3.3 SERVER RESPONSE

The response from the server is a protocol called NFTP (Netfonds Transaction Protocol). It shares some commonality with the client protocol but is much more compact due to its use in high volume streaming of quotes.

```

NFTP ::= <sequence> ' ' <source>1) ' ' <message type> ' ' <values>
values ::= <fixed position values> <keyword values>
source ::= 'S' | 'A'
sequence ::= <positive integer>

```

```
message type ::= <positive integer>
fixed position values ::= <value> <fixed-position-values> | ?
keyword values ::= <symbol> <value> <keyword values> | ?
```

¹⁾ Only available in nftp version 1.1 and above.

Some examples:

```
1 99 SYNTAX-ERROR "Wrong format"
2 1 17933 OSE ? 103748 TEL od 4978 odt 164599 no 173
```

We'll get back to what this means in the next chapters but it's important to remember the following:

- For the client, strings and symbols are used interchangeably. "TEL" and TEL should be treated as synonyms. For the server, however, they are distinctly different types and can not be interchanged.
- Symbols are case-insensitive, strings are not.
- The question mark symbolizes an empty fixed position field.
- Empty keyword fields are left out.
- The symbol NIL is an empty field.
- The sequence number in field 1 is just for the clients to keep track of all the messages arriving from the servers. This number should always be an increment of one (otherwise a message has been lost). The client is otherwise free to do with it as it chooses.
- The server uses relaxed typing; a float may well be a number without the decimal; ie. 10.0 = 10 .
- A line terminated by a newline is always a single object.
- Dates are usually an integer on the format YYYYMMDD whereas time stamps are usually on the format HHMMSS but without the leading zero (ie. 73546 = 07:35:46).
- An integer can be an arbitrarily large number and is not limited to some word size.
- An NFTP object may contain fixed position values followed by keyword values, but never vice versa.
- The client shouldn't assume that the NFTP objects arrive in any particular order, that the feed doesn't contain any new objects or that the objects themselves are immutable (because they aren't). While the object values listed below are not likely to change, existing objects may be extended with additional fields. These will either come as an additional keyword field or as an extra positional field added to the end.
- The <source> field is available in nftp version 1.1 and above (see 4.1.1) to distinguish between synchronous and asynchronous messages. If the value is "S" it is a synchronous message whereas asynchronous messages are designated with an "A".

3.3.1 NFTP TYPES

Integer	Any number without a decimal.
Float	Any number with a decimal.
Time	An integer with the current time as HHMMSS but without any leading zeros (1 = 00:00:01 and 73546 = 07:35:46).
Date	An integer with current date encoded as YYYYMMDD.
Datetime	A string with date and time encoded as YYYYMMDDTHHMMSS.
String	A string with or without quotes. You can assume that everything that isn't one of the above types is a string, ie TEL and "TEL" are equal.
List	A list of something, ie. (A ₁ A ₂ ... A _n).

3.3.2 NFTP OBJECTS

1	QUOTES
3	TRADE
6	EXTENDED-SPREAD
20	SECURITY
22	MARKET-BY-PRICE
24	EXCHANGE
25	RELEASE
27	HISTORY
30	AUTHORIZED
48	EXCHANGE-STATE
53	CUSTOMER-ORDER
54	CUSTOMER-DEAL
62	TRADER-AVAILABLE
70	USER-MESSAGE
82	EXTENDED-PAPER-SUMMARY
91	INDEX-WEIGHT
99	ERROR MESSAGE
120	EXTENDED-SECURITY
126	FUND
131	EXTENDED-AVAILABLE
135	HEARTBEAT

146	TRADE-INFORMATION
193	API-RESPONSE
194	API-ORDER-ENTRY
195	API-ORDER-CHANGE
311	POOL-CONTENT

3.3.2.1 QUOTES (message type = 1)

This is the primary price information object. Provides information about last price, high price, current bid and offer, etc.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 1
3	Sequence	Integer	A system wide sequence number for this object.
4	Exchange	String	The exchange this quote is coming from
5	Board	Integer	N/A
6	Time	Time	The time this quote arrived at Netfonds
7	Paper	String	The ticker for this quote
b	Bid	Float	The current bid for this quote
bd	Bid depth	Integer	The bid depth for this quote
bdt	Bid depth total	Integer	The total bid depth for this quote
nb	Bid number	Float	The bid number for this quote
o	Offer	Float	The current offer
od	Offer depth	Integer	The offer depth
odt	Offer depth total	Integer	Total offer depth
no	Offer number	Float	The offer number
open	Open price	Float	The open price
hi	High	Float	Highest trade price today
lo	Low	Float	Lowest trade price today
l	Last	Float	Last trade price
c	Change	Float	Change price today
cp	Change %	Float	Change percent today
vo	Volume	Float	Volume
va	Value	Float	Value
nt	Trades	Float	Number of trades today
st	Status	String	Status, empty or one of: CL: The book is closed, no orders accepted. ST: The book is starting up but still closed. PRT: Pretrade. C: Book is open and trading is happening TH: Book is open but trading is halted A: Auctions OA: Open auctions

			CA: Close auctions POT: Post trade
i	Quote ID	String	ID of this quote

3.3.2.2 TRADE (message type = 3)

Trades are public trades executed on the exchange.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 3
3	Sequence	Integer	A system wide sequence number for this object.
4	Exchange	String	The exchange this quote is coming from
5	Board	Integer	The board (trading unit).
6	Time	Time	The time this trade arrived at Netfonds.
7	Paper	String	The ticker for this trade
8	Trade time	Time	The time this trade was done
9	Quantity	Integer	The trade quantity
10	Price	Float	Hammer price for the trade
11	Source	String	Source of the trade
12	Buyer	String	The buyer party
13	Seller	String	The seller party
14	Change since last	Float	Change in price since last
15	Initiator	String	Initiator of the trade

3.3.2.3 EXTENDED-SPREAD (message type = 6)

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 6
3	Sequence	Integer	A system wide sequence number for this object.
4	Exchange	String	The exchange this security is on

5	Time	Time	Time of entry
6	Paper	String	The ticker for this security
7	Bid	Float	Bid
8	Bid-depth	Integer	Bid depth
9	Bid depth total	Integer	Total bid depth
10	Offer	Float	Offer
11	Offer depth	Integer	Offer depth
12	Offer depth total	Integer	Total offer depth
13	Indexp	String	Not used

3.3.2.4 SECURITY (message type = 20)

Security provides basic information about the security such as ISIN, what type of instrument it is and so on.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 20
3	Sequence	Integer	A system wide sequence number for this object.
4	Exchange	String	The exchange this security is on
5	Time	Time	Timestamp
6	Paper	String	The ticker for this security
7	Name	String	Name of the instrument
8	ISIN	String	The ISIN for this security
9	Lot	Integer	The lot size
10	Last trade date	Date	The date (YYYYDDMM) when it was last traded
11	Previous price	Float	Yesterday's closing price.
12	Tradable size	Integer	Tradable size
13	Currency	String	The currency this security is in
14	Security type	String	The type of security, one of: AC: American call AP: American put B: Bonds BC: Binary call BO: Bonds

			BP: Binary put EN: Energy EC: European call ET: Exchange traded notes EO: Binary call EP: European put ER: Subscription rights EU: Binary put F: Funds FO: Forwards FS: Future spread FU: Futures FX: Currency cross I: Index IN: Interest S: Stock O: Option W: Warrants PM: Precious metals XF: Exchange traded funds CT: Certificate WT: Turbo warrant FM: Mini future WK: Knockout warrant
15	Sector	String	Depends on the exchange; for NGM it shows whether the security is traded on NGMN, NGMS, NGMFI or NGMDK, for funds it shows who the fund manager is.
16	Primary paper	String	Depends on the exchange, normally it shows the overlying paper for an underlying security.
17	Primary exchange	String	If primary paper is not empty, shows the exchange for the primary paper.
18	Value factor	Float	The value factor for this security
19	Status	String	The security status; Y for tradable, N for not tradable.

3.3.2.5 MARKET BY PRICE (message type = 22)

This is the order-book for the instrument.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 22
3	Sequence	Integer	A system wide sequence number for this object.

4	Exchange	String	The exchange this order book is coming from
5	Board	Integer	The board (trading unit).
6	Time	Time	The time this order book arrived at Netfonds
7	Paper	String	The ticker for this order book
8	Bids	List of <bid, depth, orders> triplets.	List of bid, depth, orders triplets for each position in the orderbook, ie. (164.6 3727 7 164.5 9174 12 ...) which shows that the first position has a bid for 164.6, depth of 3727 and 7 orders, position #2 has bid=164.5, depth=9174 and orders=12, and so on.
9	Offers	List of <offer, depth, orders> triplets.	Same as Bids but with offers.

3.3.2.6 EXCHANGE (message type = 24)

EXCHANGE provides some basic information about the exchange.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 24
3	Sequence	Integer	A system wide sequence number for this object.
4	Exchange	String	The exchange short-name
5	Name	String	Exchange long name
6	Open time	Time	When the exchange opens
7	Close time	Time	When the exchange closes
8	Pre-trade time	Time	When pre-trade period begins
9	Post-trade time	Time	When post-trade period begins
10	Offset from Oslo	Integer	Hours offset from Oslo time
11	Allow order change	String	Y if exchange allows order changes
12	Area	String	The exchange area
13	Max valid	Integer	Internal use only.
14	Max valid stop	Integer	Internal use only.
15	Next trade days	List of Date	The next days this exchange is open for trading
16	Allow stop loss	String	Y if exchange allows stop loss orders
17	Allow hidden	String	Y if exchange allows hidden orders
18	Allow direct routing	String	Y if exchange allows direct routing.

3.3.2.7 RELEASE (message type = 25)

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 25
3	Sequence	Integer	A system wide sequence number
4	Distributor	String	The distributor of this release
5	Time	Time	The time of this release
p	Papers	List	A list of tickers, if applicable, this release applies to.
e	Exchanges	List	A list of exchanges, if applicable, this release applies to.
ty	Release type	String	Type of release
to	Topic	String	Topic of release
id	Id	String	The distributors id for this release.
rt	Received time	Datetime	When the release was first received.
pt	Published time	Datetime	When the release was first published.
cn	Company name	String	The company name, if applicable, this release applies to
c	Country	String	The country, if applicable, this release applies to.
ci	City	String	The city, if applicable, this release applies to.
is	ISIN	String	The ISIN, if applicable, this release applies to.
eu	External URLs	List	A list of direct URL links to this release.
cu	Company URL	String	An URL link to the company this release applies to.
h	Heading	String	The heading of this release.
in	Ingress	String	The ingress.
b	Body	String	The body.
f	Footer	String	The footer.
co	Copyright	String	Copyright of this release.
ca	Category	String	Release category.
fm	Format	String	The format of the release texts.
la	Language	String	The language of the release texts.

3.3.2.8 HISTORY (message type = 27)

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 27
3	Sequence	Integer	A system wide sequence number
4	Exchange	String	The exchange this security is on.
5	Date	Date	History date.
6	Paper	String	Ticker
7	Open	Float	The opening price.
8	High	Float	The highest trade price.
9	Low	Float	The lowest trade price.
10	Close	Float	The closing price.
11	Volume	Integer	The trade volume.
12	Value	Integer	The trade value.

3.3.2.9 AUTHORIZED (message type = 30)

Authorized tells you if your are authorized to log in.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 30
3	Yes or No	String	Y if authorized, N of not.
4	Text	String	A welcome string
5	Realtime	String	Y if real-time feed, N if delayed.
6	Level	Integer	A unique ID for each client.
7	Login-id	String	Your user name

3.3.2.10 EXCHANGE-STATE (message type = 48)

Exchange state says something about the state of the exchange. Note that not all exchanges provide this information.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 48
3	Sequence	Integer	A system wide sequence number for this object.
4	Exchange	String	The exchange short-name
5	Time	Time	Timestamp of message
6	Date	Time	Datestamp of message
7	State	String	Exchange state, one of CLOSED: Exchange is closed. STARTUP: Exchange is starting up. PRE-TRADE: Pre-trade session. OPEN: The exchange is open. HALTED: Trading is halted. AUCTION: Auction has begun. OPEN-AUCTION: Startup auction. CLOSE-AUCTION: Closing auctions. POST-TRADE: Post trade session.

3.3.2.11 CUSTOMER-ORDER (message type = 53)

Customer order provides information about the actual order placed on the exchange. Only active (open) orders are sent to the client. The definition of an *active order* is an order with `action-status = "A"` and `status <> "C"`. A *filled order* is when `filled-number = order-number`.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 53
3	Sequence	Integer	Internal sequence number
4	Do odd-lotp	String	Y if odd-lot order, N otherwise
5	Id	Integer	The order ID
6	Paper	String	The ticker
7	Order-type	String	B if buy order, S if sell order.

8	Order-number	Float	Order quantity
9	Order-limit	Integer	Limit price (0 = market order)
10	Action status	String	A: Active order. D: Order is complete. C: Order is canceled P: Order is passive W: Order is awaiting activation
11	Customer	String	Your user-name
12	Filled number	Integer	Quantity filled
13	Trigger price	Float	The trigger price for the order.
14	Directp	String	Y if this is a direct order, N otherwise.
15	Status	String	U: New order P: Order is complete and ticketed D: Order is routed C: Order is canceled N: Order is not executed
16	Hidden-number	Float	How much of the quantity is hidden.
17	Exchange	String	The exchange for this order.
18	Order-time	Datetime	The time and date the order was entered.
19	Valid until	Date	Expiration date
20	Current part	Integer	The current part of the order.
21	Order method	String	Order method: FOK: Fill Or Kill order. FAK: Fill And Kill order. M: Multi-order I: Invisible N: Normal.
22	Entered by	String	Who entered the order (user-name)
23	Exec-paper	String	Only used if the paper is traded on a parallell exchange.
24	Exec-exchange	String	Only used if the paper is traded on a parallell exchange.

3.3.2.12 CUSTOMER-DEAL (message type = 54)

CUSTOMER-DEAL is the actual order-trade. One order may result in several deals.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client

2	Message type	Integer	Always 54
3	Sequence	Integer	Internal sequence
4	Deal ID	Integer	The deal ID
5	Order ID	Integer	The order ID this deal is associated with
6	Part	Integer	The deal part
7	Deal number	Integer	The deal quantity
8	Deal price	Float	The price of the deal
9	Entry time	Time	When the deal was made (time)
10	Notifiedp	String	Y if the customer has been notified about the deal, N or ? Otherwise.
11	Customer	String	Who owns the deal (user-name).
12	Order-type	String	B if a buy order, S if a sell order.
13	Paper	String	Ticker
14	Exchange	String	Exchange

3.3.2.13 TRADER-AVAILABLE (message type = 62)

TRADER-AVAILABLE shows the asset value of your balance.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 62
3	Sequence	Integer	Internal sequence
4	Customer	String	User-name
5	Update time	Time	Time updated
6	Money	Float	How much money the account has.
7	Long pos	Float	Accumulated long positions
8	Short pos	Float	Accumulated short positions
9	Required funds	Float	Required funds available.
10	Net available	Float	Net available
11	Active buy value	Float	The buy value of all your active orders.

3.3.2.14 USER-MESSAGE (message type = 70)

USER-MESSAGE provides an informational message to the client.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 70
3	Message type	String	The message type (can be anything)
4	Message	String	The message

3.3.2.15 EXTENDED-PAPER-SUMMARY (message type = 82)

EXTENDED-PAPER-SUMMARY gives a condensed summary of a paper.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 82
3	Paper	String	The ticker
4	Last	Float	The last traded value
5	Change percent	Float	The price change in percent
6	Value	Float	The value of the share
7	Name	String	The full name of the share
8	Displayed precision	Integer	The number of precision digits
9	Bid	Float	N/A
10	Offer	Float	N/A

3.3.2.16 INDEX-WEIGHT (message type = 91)

INDEX-WEIGHT gives information about the underlying paper of an index.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 91
3	Sequence	Integer	System wide sequence number
3	Exchange	String	The exchange of the paper
4	Time	Datestamp	Date last updated
5	Paper	String	The ticker
6	Index paper	String	The index-ticker
7	Index exchange	String	The exchange of the index
8	Weight	Float	The weight this paper has on the index (or zero if unknown)
9	Multiplier	Integer	Index weight multiplier.
10	Price adjustment	Float	Adjusted price.

3.3.2.17 ERRMSG (message type = 99)

ERRMSG signals an error. This is one of the two synchronous responses you always get (the other being API-RESPONSE). See section 5 for more information about errors and error messages.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 99
3	Error class	String	The error class (see section 5)
4	Error message	String	Error message text (see section 5)

3.3.2.18 EXTENDED-SECURITY (message type = 120)

EXTENDED-SECURITY is similar to SECURITY but with some additional fields.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client

2	Message type	Integer	Always 120
3	Sequence	Integer	System wide sequence number.
4	Exchange	String	Exchange
5	Time	Time	Time of last update
6	Paper	String	Ticker
7	Name	String	Name of security
8	ISIN	String	ISIN of security
9	Lot	Integer	Lot size
10	Last trade date	Date	Last traded date
11	Previous price	Float	Closing price
12	Tradeable size	Integer	Tradeable size
13	Currency	String	Currency this instrument is traded in
14	Security type	String	See SECURITY for a list of security types
15	Sector	String	See SECURITY
16	Existsp	String	For internal use
17	Tradeablep	String	Y if security is tradeable, N if not
18	Primary paper	String	The primary paper for this security
19	Primary exchange	String	The primary exchange for this security
20	Secondaries	List	Secondary papers for this security
21	Value factor	Float	The value factor of this security
22	URL	String	The URL to any extra information about this security
23	Issuer	String	The issuer of this security
24	Expiration date	Date	The expiration date for this security
25	Status	String	Y if tradeable, N if not.
26	Minimum trade value	Float	Minimum trade value
27	Last update date	Date	The date this security was last updated

3.3.2.19 FUND (message type = 126)

FUND gives some useful benchmarks and statistics for funds (instruments that are on exchange “FOND”).

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client

2	Message type	Integer	Always 126
3	Sequence	Integer	System wide sequence number
4	Exchange	String	The exchange
5	Paper	String	Ticker
fid	Fund manage id	String	Fund manager ID
fna	Fund manager name	String	The name of the fund manager
rg	Root group	String	Which root group the fond belongs to
ft	Fund group	String	Which group of fund this is
fa	Fund area	String	Which area the fund belongs to (Aksjefond, Obligasjonsfond, ...)
b	Benchmark	String	This funds benchmark index
mssc	Max sales charge	Float	Maximum sales charge
mrf	Max redemption fee	Float	Maximum redemption fee
mf	Management fee	Float	Management fee
ma	Minimum amount	Float	Minimum investment amount
mac	Currency of minimum amount.	String	Currency of minimum investment amount
vff	VFF classified?	String	Y if fund is VFF-classified, N if not.
cre	Credit risk?	String	Y if fund carries a credit risk, N if not.
cru	Currency risk?	String	Y if fund carries a currency risk, N if not.
air	Max interest rate risk	Float	Maximum interest rate risk.
iir	Min interest rate risk	Float	Minimum interest rate risk.
y	Yield	Float	The funds yield.
md	Modified duration	Float	Modified duration
ps	Profit share?	String	Y if the fund shares its profits, N if not.
tcpu	Total cost in percent, unshared	Float	Total cost in percent unshared.
tcps	Total cost in percent, shared.	Float	Total cost in percent shared.
tcd	Total cost date	Date	Total cost date.
r	Returns	List of (A B) where A is a string and B is a float.	The funds expected ROI over a period.
ry	Return years	List of (A B) where A is an integer and B is a float.	How much the funds expected return (B) over a period of years (A).
avr	Average returns	List of (A B)	The average return (B) of the fund over a

		where A is an integer and B is a float	period of years (A).
anr	Annualized returns	List of (A B) where A is an integer and B is a float.	The previous annualized return B of year <current year> - A.
v	Volatiles	List of (A B) where A is an integer and B is a float.	List of volatiles B of year <current year> - A.
sr	Sharpe ratios	List of (A B) where A is an integer and B is a float.	List of sharpe ratios B of year <current year> - A.
dr	Differential returns	List of (A B) where A is an integer and B is a float.	List of differentials B of year <current year> - A.
rv	Relative volatilities	List of (A B) where A is an integer and B is a float.	List of relative volatilities B of year <current year> - A.
ir	Information ratios	List of (A B) where A is an integer and B is a float.	List of information ratios B of year <current year> - A.

3.3.2.20 EXTENDED-AVAILABLE (message type = 131)

EXTENDED-AVAILABLE shows your current account overview.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 131
3	Sequence	Integer	N/A
4	Customer	String	Your user-name
ab	Active buys	Float	Qty of your active buy orders in this paper.
abv	Active buy value	Float	Total value of all your active buy orders.
asl	Allow stop loss	String	Y if stop-loss-order is allowed, N otherwise.

as	Active sells	Float	Qty of all your active sell orders in this paper.
b	Borrowed	Float	Qty of your borrowed shares in this paper.
cabv	Counting active buy value	Float	For internal use.
cc	Customer currency	String	Your primary currency
cr	Currency rate	Float	Currency rate of our primary currency to NOK.
crf	Concentration risk	Float	Concentration risk (see http://www.netfonds.no/ekfaktor.php)
db	Done buys	Float	Qty of done buys in this paper today.
dbv	Done buy value	Float	Total value of done buys today.
ds	Done sells	Float	Qty of done sells in this paper today.
dsv	Done sell value	Float	Total value of done sells today.
en	Exchange name	String	Exchange name.
fn	Full name	String	Your full name.
fr	Futures realized	Integer	Futures realized.
fv	Futures value	Float	The value of your realized futures.
ic	Intraday credit	Float	Your intraday credit.
ip	In portfolio	Float	Qty of this paper in your portfolio.
l	Limit	Float	Closest correct limit for this paper.
len	Lent	Integer	Lent shares in this paper.
lkd	Locked	Integer	Locked shares in this paper.
lkm	LK-money	Float	Money in your "life-insurance" account (if your account is a life-insurance account).
lot	Lot	Integer	The lot size of this paper.
lpv	Long position value	Float	Value of your long position in this paper.
m	Money	Float	The sum of money in your account.
mb	Max buy	Float	Maximum number you can buy without credit.
mbcv	Max buy credit value	Float	Maximum amount for which to buy with credit (for this paper).
mbl	Max buy lot	Integer	Maximum lot you can buy.
mbv	Max buy value	Float	Maximum amount for which you can buy without credit (for this paper).
mc	Max credit	Float	Your maximum credit.
mi	Min invisible	Integer	Minimum invisible order for this paper.
ms	Max sell	Float	Maximum number of shares you can sell without borrowing.
msl	Max sell long	Float	Maximum number of shares to sell without going short.
mss	Max sell short	Float	Maximum number of shares to sell if you

			borrow shares.
na	Net assets	Float	Your net available assets, ie. assets + done buys – done sells.
net	Net available	Float	Net available equity.
pc	Paper currency	String	The currency of this paper.
pf	Paper factor	Float	The paper factor of this paper.
psv	Previous share value	Float	Previous total share value including long and short.
rf	Required funds	Float	Your required equity.
rff	RF-factor	Float	Concentration risk factor.
rfp	Required funds percent	Float	Required funds in percent.
p	Paper	String	The ticker for this security.
e	Exchange	String	The exchange for this security.
sfr	Sum futures realized	Integer	Sum of your realized futures.
sn	Security name	String	The name of this security.
spv	Short position value	Float	Total value of shorted shares.
tsv	Total share value	Float	Total share value including long and shorts.
t	Time	Time	Time of this data.
ut	User type	String	Your user class: L = Credit T = Trader F = Managed D = Plus customer

3.3.2.21 HEARTBEAT (message type = 135)

HEARTBEAT is a status message sent to all the clients every minute (approx). It may be useful if you want to ensure that the connection is still up, otherwise it can just be ignored.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 135
3	Sequence	Integer	System wide sequence number
4	Date	Date	Current date
5	Time	Time	Current time
6	Text	String	SIHB: Server initiated heartbeat

			CIHB: Client initiated heartbeat
--	--	--	----------------------------------

3.3.2.22 TRADE-INFORMATION (message type = 146)

TRADE-INFORMATION is sent in conjunction with TRADE (message type = 3) on exchanges where not all trade data are available at the time of the trade. TRADE-INFORMATION depends on the exchange. It is normally sent after trades on OSE but not on eg. NGM. The client should treat TRADE-INFORMATION as a delta-operator for the TRADE object.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 146
3	Sequence	Integer	System wide sequence number
4	Exchange	String	The exchange
5	Board	Integer	The board (trading unit).
6	Time	Time	The time this trade arrived at Netfonds.
7	Paper	String	The ticker for this trade
8	Trade time	Time	The time this trade was done
9	Quantity	Integer	The trade quantity
10	Price	Float	Hammer price for the trade
12	Buyer	String	The buyer party
13	Seller	String	The seller party

3.3.2.23 API-RESPONSE (message type = 193)

API-RESPONSE is sent after each command to signal that the command was successful and that the server is ready for the next command. Your client must always wait for API-RESPONSE or ERRMSG before continuing.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 193
3	Message	String	A message

3.3.2.24 API-ORDER-ENTRY (message type = 194)

API-ORDER-ENTRY is sent in reply to the `enter-order` command. The Order ID is the id of the order. This ID can be used to track the status of the order.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 194
3	Order ID	Integer	The order ID (or 0 if <code>:check-only</code> was "Y" and the check was successful)

3.3.2.25 API-ORDER-CHANGE (message type = 195)

API-ORDER-CHANGE is sent in reply to the `change-order` and `delete-order` commands and contains information about the changed order.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each client
2	Message type	Integer	Always 195
3	Status	String	A: Order was amended D: Order was routed to the exchange
4	Partial	String	T: The order was partially deleted or changed. NIL: Change/delete was fully executed.
5	Remaining	Integer	The remaining part of the order if <code>partial = T</code>
6	Done	Integer	The done part of the order if <code>partial = T</code>
7	Limit	Float	The new limit of a limit change
8	Volume	Integer	The new quantity of a volume order change

3.3.2.26 POOL-CONTENT (message type = 311)

Shows the content of the pool of shortable instruments.

Position/ keyword	Name	Type	Description
1	Sequence	Integer	An incremental sequence number for each

			client
2	Message type	Integer	Always 311
3	Exchange	String	The exchange.
4	Paper	String	The ticker.
5	Name	String	Instrument name.

3.4 EXCHANGES

These are our operative stock exchanges as of writing. Please refer to our web pages for the latest information.

A	Amex
BTSE	BATS Europe
DJI	Dow Jones indexes
FOND	Funds
FXSB	Currencies (non-tradeable)
FXSX	Tradeable currencies
GTIS	Commodities
MERK	Merkur
MFEX	Mutual funds exchange
N	NYSE
NBB	Netfonds Bulletin Board
NGM	Nordic Growth Market
O	NASDAQ
OAX	Oslo Axess
OMFE	Options (Norway)
OMSW	Options (Sweden)
OSE	Oslo Stock Exchange
OTC	Oslo OTC

4. THE API

Communication with the API is both synchronously and asynchronously depending on the command. When you send a command, you will always get a synchronous response back which you should wait for before sending the next command. This response is either an API-RESPONSE or an ERRMSG. An API-RESPONSE says that the command was successful whereas the ERRMSG says that something went wrong. Here's an example:

```
(current-quotes :paper "FOO" :exchange "OSE")  
2275 99 SIMPLE-ERROR "Instrument FOO.OSE does not exist or is inactive"
```

However if we use a real ticker we get the following:

```
(current-quote :paper "TEL" :exchange "OSE")  
2283 1 50147 OSE ? 155410 TEL b 165. bd 10751 bdt 213571 nb 192 o 165.2  
od 8350 odt 182663 no 178 open 165.3 hi 166.5 lo 164.5 l 165. c 1.8 cp  
1.1 vo 1008981 va 166815551. nt 2563 st C  
2284 193 OK
```

An asynchronous command is a command that continues to produce output even after it was finished. The subscribe command is an example of this:

```
(subscribe :paper "TEL" :exchange "OSE")  
<lots of quotes and trades objects>  
2284 193 OK  
<continuous quotes and trade feed for TEL.OSE>
```

You still get an API-RESPONSE which tells the client that it can continue to send the next command.

4.1 THE PUBLIC CHANNEL

The public channel accepts the following commands: (M is a mandatory parameter, O is optional):

4.1.1 LOGIN

→	(login :username <string> :password <string> :otp <string> :nftp-version <float>)	M M O O	- Your username - Your password - A one time password if your account uses two factor authentication (see 2.2) - A number indicating which nftp-version your client supports (default = 0)
↵	AUTHORIZED API-RESPONSE		

Authenticates the client. The username and the password must be a registered username/password and the account must have API access. Example:

```
(login :username "foo" :password "bar")
```

```
1 30 Y "Logged in" ? ? ?
```

```
2 193 OK
```

As of version 1.9 there is a limit of 5000 logins per day. This is to prevent clients from flooding the servers with logins.

4.1.2 CURRENT-SNAPSHOT

→	(current-snapshot :paper <string> :exchange <string> :feed <list of symbols>)	O M O	- Ticker - Exchange - A list consisting of SECURITY, QUOTES, FUND, MARKET-BY-PRICE, TRADE, EXCHANGE, EXCHANGE-STATE, EXTENDED-SECURITY
↵	SECURITY EXTENDED-SECURITY QUOTES FUND MARKET-BY-PRICE TRADE EXCHANGE EXCHANGE-STATE API-RESPONSE		

Retrieve the current snapshot for a ticker. This includes the security, quotes, market-by-price the last 500 trades, the exchange data and finally the api-response. Example:

```
(current-snapshot :paper "GTAB-B" :exchange "NGM")
3253 20 5 NGM 174156 "GTAB-B" "GLYCOREX TRANSPLANTATION B" SE0000524530 1
"20150827" 2.12 0 SEK S NEST ? ? 1 Y
3254 1 61 NGM ? 171148 "GTAB-B" b 2.03 bd 2987 bdt 33183 o 2.12 od 3000 odt
62224 open 2. hi 2.24 lo 2. 1 2.12 c 0.07 cp 3.414634 vo 25445 va 51824.51
nt 15
3255 22 66 NGM ? 180000 "GTAB-B" (2.03 2987 1 2.02 4001 1 2. 5000 1 1.97 10000 1
1.9 1295 2 1.82 1100 1 1.81 5000 1 1.8 1800 1 1.43 2000 1) (2.12 4050 2
2.13 2513 1 2.14 14999 1 2.23 6500 2 2.26 10336 1 2.44 2400 1 2.45 10000 1
2.5 2720 1 2.55 2000 1 2.8 6706 1)
3256 3 16 NGM ? 165135 "GTAB-B" 165135 100. 2.12 A AVA AVA 0.04 ?
3257 3 15 NGM ? 152056 "GTAB-B" 152056 1099. 2.08 A SWB AVA 0.05 ?
3271 24 237 NGM "Nordic Growth Marked" 90000 173000 80000 180000 0 Y SE 6 6
(20150828 20150831 20150901 20150902 20150903 20150904) Y Y Y
3272 48 31 NGM 180000 20150827 CLOSED
3273 193 OK
```

As of version 1.5, you may retrieve the snapshot for each individual NFTP object in case you're not interested in all the data. The new `:feed` parameter accepts a list of the NFTP objects you want to retrieve data for, ie.

```
(current-snapshot :paper "TEL" :exchange "OSE" :feed (quotes market-by-price))
```

which will only output quotes and the order book for TEL.OSE. Notice that the `:ticker`-parameter is also now optional. This is because some objects (ie. EXCHANGE and EXCHANGE-STATE) don't require a ticker. Should you however use an object which requires a ticker, you'll get an error message if `:ticker` is not provided.

Also note that the `:feed-list` is a list of symbols (no quotes and no commas in between).

4.1.3 CURRENT-QUOTES¹

→	(current-quotes :paper <string> :exchange <string>)	M M	- Ticker - Exchange
↵	QUOTES API-RESPONSE		

Retrieve the current quotes for a ticker. The response is a simple quotes object. Example:

```
(current-quotes :paper "GTAB-B" :exchange "NGM")
3279 1 61 NGM ? 171148 "GTAB-B" b 2.03 bd 2987 bdt 33183 o 2.12 od 3000 odt
62224 open 2. hi 2.24 lo 2. 1 2.12 c 0.07 cp 3.414634 vo 25445 va 51824.51
nt 15
3280 193 OK
```

¹This command does exactly the same thing as `(current-snapshot ... :feed (quotes))`.

4.1.4 SUBSCRIBE

→	(subscribe :paper <string> :exchange <string> :feed <list of symbols>)	O M O	- Ticker - Exchange - A list consisting of SECURITY, QUOTES FUND, MARKET-BY-PRICE, TRADE, EXCHANGE, EXCHANGE-STATE, EXTENDED-SECURITY
↵	SECURITY EXTENDED-SECURITY FOND QUOTES MARKET-BY-PRICE TRADE EXCHANGE EXCHANGE-STATE API-RESPONSE ...		

Asynchronously subscribe to the quotes feed. This command will start the streamer feed and will continue to stream quotes and other objects until you either unsubscribe from the feed or log out. You can still issue commands while receiving the feed. You may also subscribe to several feed successively. Apart from the streamer feed, this command is similar to the `current-snapshot` command.

As of version 1.5, you may subscribe to individual NFTP objects in case you're not interested in all the data. The new `:feed` parameter accepts a list of the NFTP objects you want to subscribe to, ie.

```
(subscribe :paper "TEL" :exchange "OSE" :feed (quotes market-by-price))
```

which will only output quotes and the order book for TEL.OSE. Notice that the `:ticker`-parameter is also now optional. This is because some objects (ie. EXCHANGE and EXCHANGE-STATE) don't require a ticker. Should you however subscribe to an object which requires a ticker, you'll get an error message if `:ticker` is not provided.

Also note that the `:feed`-list is a list of symbols (no quotes and no commas in between). If `:feed` is not provided, this command will subscribe to all the objects, just as before.

4.1.5 UNSUBSCRIBE

→	(unsubscribe :paper <string> :exchange <string> :feed <symbol>)	O M O	- Ticker - Exchange - The NFTP object to unsubscribe.
↵	API-RESPONSE		

Unsubscribe from a previously subscribed feed. Example:

```
(unsubscribe :paper "GTAB-B" :exchange "NGM")
```

```
3317 193 OK
```

If `:feed` is given, unsubscribe from only that object, otherwise unsubscribe from the entire feed associated with `:paper` and `:exchange`.

4.1.6 INDEX-MEMBERS

→	(index-members :index <string> :exchange <string>)	M M	- Index name - Exchange
↵	INDEX-WEIGHT ... API-RESPONSE		

Get the underlying index members for an index. Example:

```
(index-members :index "OAX" :exchange "OSE")
6 91 ? OSE 20151026 ABT OAX OSE 0.892 6891218 17.2
7 91 ? OSE 20151026 AKPS OAX OSE 16.7212 12107901 183.5
8 91 ? OSE 20151026 ALNG OAX OSE 2.6988 67788874 5.29
9 91 ? OSE 20151026 APCL OAX OSE 1.2403 96946379 1.7
...
36 193 OK
```

4.1.7 WINNERS

→	(winners :exchange <string> :levels <integer>)	M O	- Exchange - Number of entries (default = 10)
↵	EXTERNAL-PAPER-SUMMARY ... API-RESPONSE		

List the intraday top winners. If :levels is omitted, list the top ten. Example:

```
(winners :exchange "OSE")
58 82 EMAS 1.37 19.13 939021.55 "EMAS Offshore" 2 ? ?
59 82 REACH 2.16 13.09 13115.28 "Reach Subsea" 2 ? ?
60 82 IDEX 9.13 13. 28420456.51 IDEX 2 ? ?
61 82 TIDE 21.9 7.88 11071.8 Tide 2 ? ?
...
97 193 OK
```

4.1.8 LOSERS

→	(losers :exchange <string> :levels <integer>)	M O	- Ticker - Number of entries (default = 10)
↵	EXTERNAL-PAPER-SUMMARY		

...	API-RESPONSE		
-----	--------------	--	--

Same as winners except that this command lists the top losers.

4.1.9 TRADES

→	(trades :paper <string> :exchange <string> :date <integer>)	M - Ticker M - Exchange O - Date (YYYYMMDD) (Default = intraday)
↵	TRADE ... API-RESPONSE	

List all public trades for this instrument at date :date. If :date is omitted, use today's date. Example:

```
(trades :paper "GTAB-B" :exchange "NGM")
2777 3 36771192 NGM ? 103536 "GTAB-B" 103536 140. 2.3 A NON NON ? ?
2778 3 60376854 NGM ? 120444 "GTAB-B" 120444 209. 2.3 A AVA NON ? ?
2779 3 61449754 NGM ? 120951 "GTAB-B" 120951 64. 2.3 A SHB NON ? ?
2780 3 71244829 NGM ? 125712 "GTAB-B" 125712 1000. 2.3 A ENS NON ? ?
2781 3 87990519 NGM ? 141535 "GTAB-B" 141535 3000. 2.21 A NON AVA ? ?
2782 193 OK
```

4.1.10 HISTORY

→	(history :paper <string> :exchange <string> :from-date <integer> :to-date <integer>)	M - Ticker M - Exchange M - From date (YYYYMMDD) O - To date (YYYYMMDD, default=today).
↵	HISTORY ... API-RESPONSE	

List the history values (open, close, low, high, ...) for an instrument between dates :from-date and :to-date. Example:

```
(history :paper "TEL" :exchange "OSE" :from-date 20160601 :to-date 20160603)
4 27 ? OSE 20160601 TEL 138.9 140.2 137.9 138.6 1985211 275197405
5 27 ? OSE 20160602 TEL 138.9 139.6 137.8 138.6 960389 133113657
6 27 ? OSE 20160603 TEL 139.4 139.5 136.4 136.4 2395091 330802957
7 193 OK
```

4.1.11 LIST-SHORTABLE

→	(list-shortable :exchange <string>)	O	List shortables only on <exchange>
↵	POOL-CONTENT ... API-RESPONSE		

Lists our pool of shortable instruments. Example:

```
(list-shortable)
3552 311 OAX ABT "Aqua Bio Technology"
3553 311 OAX AEGA Aega
3554 311 OAX ALNG "Awilco LNG"
3555 311 OAX APCL "African Petroleum Corporation"
3556 311 OAX AURLPG "Aurora LPG Holding"
3557 311 OAX AWDR "Awilco Drilling"
3558 311 OAX BXPL "Badger Explorer"
3559 311 OAX EAM "EAM Solar"
...
4726 193 OK
```

4.1.12 SUBSCRIBE-RELEASE

→	(subscribe-release :distributor <string>)	M	The release distributor
↵	RELEASE ... API-RESPONSE RELEASE ...		

Subscribes to a feed of news-messages from a distributor. The distributors are at the time of writing:

iMarkedet	TDN	DJ	DJF	Direkt
OBI	GlobeNewswire	DJNB	Hugin	Thomson Reuters ONE
OMX	NFMF	Netfonds	Waymaker	Cision
Hegnar	BeQuoted	Aktietorget		

The news-payload (header, body, footer) of these messages can be pure text, HTML, XML or any other format. It is up to the client to parse these as they see fit.

Some distributors (iMarkedet, TDN, DJ, DJF, Direkt) are not free of charge and may require additional subscriptions. This can be ordered from our web pages.

Clients may subscribe to several distributors in one session.

4.1.13 UNSUBSCRIBE-RELEASE

→	(unsubscribe-release :distributor <string>)	M	The release distributor
↵	API-RESPONSE		

Unsubscribes from a news feed from distributor.

4.1.14 SPREADS

→	(spreads :paper <string> :exchange <string> :date <date> :from-clock <time> :to-clock <time>)	M M O O O	Ticker Exchange Date (def=todays date) From time (def=000000) To time (def=now clock)
↵	EXTENDED-SPREAD ... API-RESPONSE		

List all the spreads for a given paper at a certain date and time. Similar to the “posdump.php” function on our web page.

4.1.15 LIST-SECURITIES

→	(list-securities :exchange <string>)	M	Exchange
↵	SECURITY ... API-RESPONSE		

List all the active securities on exchange.

4.1.16 CHANGE-USER

→	(change-user :username <string>)	O	A User-name
↵	AUTHORIZED API-RESPONSE		

Change the owner of your active session to :username. :Username must be direct child-account in your hierarchy. If :username is omitted, change back to the root account. Your child-account will inherit all the

product-attributes from the root-account for the remainder of the session, meaning that you can stream everything your root account is allowed to stream.

4.1.17 WHO-AM-I

→	(who-am-i)		
↵	AUTHORIZED API-RESPONSE		

Shows which user you are logged in as.

4.1.18 QUIT

→	(quit)		
---	--------	--	--

Gracefully terminate your connection and exit from the server.

4.2 THE PRIVATE CHANNEL

The private channel is for sending orders and receiving your order updates. Immediately after a successful login you will start receiving updates on your open orders. Orders submitted here will first be validated by our order engine so any restrictions you normally have also applies here. It goes without saying that order-submitting clients should be written with extreme care as valid orders will be accepted and immediately sent to the exchange for execution.

4.2.1 LOGIN

→	(login :username <string> :password <string> :otp <string>)	M M O	- You username - Your password - A one time password if your account uses two factor authentication (see 2.2)
↵	AUTHORIZED CUSTOMER-ORDER CUSTOMER-DEAL API-RESPONSE ORDER DEAL ...		

Authenticates the client. The user-name and the password must be a registered user-name and the account must have API access. After a successful login, you will first receive all your open orders and their corresponding deals (executed trades), then an API-RESPONSE and then a successive feed on ORDERS and DEALS. An example:

```
(login :username "foo" :password "bar")
```

```
1 53 1561 Y 12345678 STL B 1000 1 A foo 1000 0 N D 0 OSE "20150102T091124"
20150914 2 N foo ? ?
```

```
2 54 1562 25313132 12345678 1 500 27.3 "20150102T091125" Y foo B STL OSE
```

```
3 54 1563 25313133 12345678 1 500 27.3 "20150102T091125" Y foo B STL OSE
```

```
90 30 Y "Logged in" ? 4 foo
```

```
91 193 OK
```

```
1 53 1561 Y 12345679 TEL B 500 1 A foo 500 0 N D 0 OSE "20150102T091124"
20150914 2 N foo ? ?
```

```
2 54 1562 25313132 12345678 1 500 37.3 "20150102T091125" Y foo B TEL OSE
```

4.2.2 ENTER-ORDER

→	(enter-order :paper <string> :exchange <string> :type <string> :quantity <integer> :limit <float> :short <string> :check-only <string> :trigger-price <float> :hidden-number <integer> :order-method <string> :valid-until <integer>)	M M M M M M O O O O O O	- The ticker - The exchange - "B" for Buy, "S" for sell - Quantity - Limit price (0 = market order) - "Y" if this is a short order (default = "N") - "Y" if you only want to check the validity. - Stop loss price. - Hidden quantity. - Order method. - Expiration date (YYYYMMDD) (default = day order).
↵	API-ORDER-ENTRY API-RESPONSE		

Send an order to the exchange. If `:limit` is 0 the order is a market-order. If the order is a short order, use `:short "Y"`. If you only want to check the validity of the order, use `:check-only "Y"`. This will only verify the order and not send it to the exchange. If `:check-only` is "Y" and the check is successful, an API-ORDER-ENTRY with "Order ID" = 0 will be returned. Otherwise an API-ERROR will be returned.

The option `:trigger-price` allows the order to rest in our (Netfonds') order engine until the trigger price is met, at which point it will be sent to the exchange with the designated limit. Example: TEL.OSE is currently traded at 133 and you want to sell if the price goes down to 125 (stop-loss). Your order would then set `:trigger-price` to 125 and `:limit` to 0 (market order).

The option `:hidden-number` allows you to partially hide your order (iceberg-orders), if the exchange supports it. The option `:order-method` specifies what kind of order this is. Valid values are:

INVISIBLE	The order is an invisible order.
MID-SPREAD	The order should be placed at mid spread.
FILL-OR-KILL	Either immediately fill the entire order or cancel it.
FILL-AND-KILL	Immediately fill as much as possible and cancel the rest.

The option `:valid-until` sets an expiration date for your order (GTD-orders). The default is today (day orders).

4.2.3 CANCEL-ORDER

→	(cancel-order :orderid <integer>)	M	- The order-id
↵	API-ORDER-CHANGE API-RESPONSE		

Cancel an existing order on the exchange unless it has already been executed. Partial trades are not cancelled.

4.2.4 CHANGE-ORDER

→	(change-order :orderid <integer> :quantity <integer> :limit <float>)	M O O	- The order-id - Quantity to change - Limit to change
↵	API-ORDER-CHANGE API-RESPONSE		

Change an existing order. Either :limit or :quantity or both must be provided; if only :limit is provided, change just the limit, if only :quantity is provided change only the quantity and if both are provided, change both.

4.2.5 ACTIVE-ORDERS

→	(active-orders)		
↵	CUSTOMER-ORDER CUSTOMER-DEAL ... API-RESPONSE		

List your active orders and deals. This function doesn't take any parameters. Example:

(active-orders)

```
4 53 1 Y 12733715 STL B 1001 1 A foobar 1000 0 N D 0 OSE "20150102T091124"
20160407 2 N foobar ? ?
5 54 2 15310132 12733715 1 500 27.3 "20150102T091125" Y foobar B STL OSE
6 54 3 15310133 12733715 1 500 27.3 "20150102T091125" Y foobar B STL OSE
7 53 4 Y 13844790 NHY B 13625 0.01 A foobar 10482 0 N D 0 OSE "20160218T161955"
20160411 4 N foobar ? ?
8 54 5 17156870 13844790 2 192 0.83 "20160219T092430" Y foobar B NHY OSE
9 54 6 17182076 13844790 3 2550 0.83 "20160224T142623" Y foobar B NHY OSE
95 193 OK
```

4.2.6 AVAILABLE* (Obsolete)

→	(available)		
↵	TRADER-AVAILABLE API-RESPONSE		

Lists your available assets as of right now. Example:

(available)

104 62 92 gazonk 144909 3701575.36 1691.4 80584.45 144201.2 ? 187.93

105 193 OK

(* This command has been obsoleted by ACCOUNT-AVAILABLE and is only retained for backward compatibility).

4.2.7 LIST-ACCOUNT

→	(list-account :paper <string> :exchange <string>)	O	- The ticker
		O	- The exchange
↵	EXTENDED-AVAILABLE API-RESPONSE		

Lists your account overview in addition to some other useful information. If :paper and/or :exchange is included, list only the lines that match exchange and/or ticker. Example:

(list-account)

4 131 ? test ab 0 abv 0 asl Y as 0 b 0 cabv 0 cc NOK cr 0.9772 db 0 dbv 0 ds 0
dsv 0 en "Nordic Growth Marked" fn Test fr 0 fv 0 ic ? ip 6 l 0.36 len 0 lkd 0
lkm ? lot 1 lpv 49.91 m -223627.27 mb 0 mbcv 0 mbl 0 mbv -223627.27 mc 223627.27
mi 0 ms 6 msl 6 mss 6 na T net -223593.36 pc ? pf 1 psv ? rf 16. rff 1.21 rfp
100 p "IMHE-MTF-B" NGM sfr 0 sn "Implementa Hebe B" spv 0 tsv 183.9 t 152521 ut
L

5 131 ? test ab 0 abv 0 asl ? as 0 b 0 cabv 0 cc NOK cr 0.9772 db 0 dbv 0 ds 0
dsv 0 en "Mutual Funds Exchange" fn Test fr 0 fv 0 ic ? ip 0.0898 l 90.77 len 0
lkd 0 lkm ? lot 1 lpv 49.91 m -223627.27 mb 0 mbcv 0 mbl 0 mbv-223627.27 mc
223627.27 mi 4000000 ms 0.0898 msl 0.0898 mss 0.0898 na T net -223593.36 pc ? pf
1 psv ? rf 16. rff .21 rfp 100 p FI4000124250SEK e MFEX sfr 0 sn "Ålandsbanken
FI: Kina Aktie Placeringsfond" spv 0 tsv 183.9 t 15252 ut L

6 193 OK

4.2.8 ACCOUNT-AVAILABLE

→	(account-available)		
↵	EXTENDED-AVAILABLE API-RESPONSE		

List your accumulated available account assets. Example:

(account-available)

7 131 ? ? ab 0 abv 0. asl ? as 0 b 0 cabv ? cc ? cr 1 crf ? db 0 dbv 128800. ds

```

0 dsv 465212.9 en ? fn "gazonk" fr 0 fv 0 ic ? ip 0 l ? len 0 lkd 0 lkm ? lot 1
lpv ? m 18238.29 mb ? mbcv ? mbl ? mbv 354651.19 mc ? mi ? ms 0 msl ? mss ? na T
net ? pc ? pf 1 psv ? rf ? rff ? rfp ? p ? e ? sfr 0 sn "" spv 0 tsv 299885.7 t
150931 ut D

```

which shows that you have bought for 128800.0 today, sold for 465212.9, you have 18238.29 deposited in your account, you can buy for 354651.19 and the total share value is 299885.7. Of this you can infer that

- Non-posted account value is $465212.9 \text{ (dsv)} + 0 \text{ (sfr)} - 128800.0 \text{ (dbv)} = 465912.9$
- Sum cash/credit is $465912.9 + 18238.29 \text{ (m)} = 484151.19$
- Total value is $484151.19 + 299885.7 \text{ (tsv)} = 784036.89$

4.2.9 TEST-MODE

→	(test-mode :mode <symbol>)	O	- One of :OFF :ERROR or :OK. :OFF turns off test mode, :OK simulates a successful response and :ERROR simulates a failure.
↵	USER-MESSAGE API-RESPONSE		

When developing a client, use this command to set the current session in test mode. In test-mode, enter-order, cancel-order and change-order will not be sent to the exchange but instead return with a simulated response. Examples:

```

(test-mode :mode :error)
51 70 "TEST-MODE" ERROR
52 193 OK

(cancel-order :orderid 234323)
62 99 ORDER-FAILED "Cancel rejected: ORDER-CUSTOMER-MISMATCH"

(test-mode :mode :ok)
64 70 "TEST-MODE" OK
65 193 OK

(cancel-order :orderid 234323)
66 195 D 10 20 30 ? ?
67 193 OK

```

4.2.11 CHANGE-USER

→	(change-user :username <string>)	O	A User-name
↵	AUTHORIZED API-RESPONSE		

Change the owner of your active session to :username. :Username must be a direct child-user in your hierarchy. If :username is omitted, change back to the parent account. All orders will be committed to your changed user.

4.2.12 WHO-AM-I

See 4.1.17

4.2.13 QUIT

See 4.1.18

4.2.13 EXAMPLE OF ORDER OPERATIONS

Buy 2 STL.OSE @ 122 (No trade, price is too high)

```
(enter-order :paper "STL" :exchange "OSE" :type "B" :quantity 2 :limit 122.0)
58 194 13617139
59 193 OK
60 53 3 Y 13617139 STL B 2 122 A NetfondsTest 0 0 N U 0 OSE "20151208T113957"
20151208 1 N NetfondsTest ? ?
```

Cancel previous order

```
(cancel-order :orderid 13617139)
63 195 D ? 2 0 ? ?
64 193 OK
66 53 4 Y 13617139 STL B 2 122 C NetfondsTest 0 0 N U 0 OSE "20151208T113957"
20151208 1 N NetfondsTest ? ?
```

Buy 2 STL.OSE @ market price

```
(enter-order :paper "STL" :exchange "OSE" :type "B" :quantity 2 :limit 0)
203 194 13617932
204 193 OK
205 53 7 Y 13617932 STL B 2 0 A NetfondsTest 0 0 N U 0 OSE "20151208T135211"
20151208 1 N NetfondsTest ? ?
220 54 8 16784860 13617932 1 2 121.8 "20151208T140631" N NetfondsTest B STL OSE
221 53 9 Y 13617932 STL B 2 0 A NetfondsTest 2 0 N D 0 OSE "20151208T135211"
20151208 1 N NetfondsTest ? ?
```

Sell previous order @ market price

```
(enter-order :paper "STL" :exchange "OSE" :type "S" :quantity 2 :limit 0)
228 194 13618104
229 193 OK
230 53 10 Y 13618104 STL S 2 0 A NetfondsTest 0 0 N U 0 OSE "20151208T141052"
20151208 1 N NetfondsTest ? ?
232 54 11 16785052 13618104 1 2 121.5 "20151208T141117" N NetfondsTest S STL OSE
233 53 12 Y 13618104 STL S 2 0 A NetfondsTest 2 0 N D 0 OSE "20151208T141052"
20151208 1 N NetfondsTest ? ?
```

5. ERROR MESSAGES

The error messages are mostly self-explanatory. They are included here for reference only. The list is by no means complete.

Class	Explanation																																						
SYNTAX-ERROR	Your command was not well-formed, contained an illegal character or could not be parsed properly. Check the spelling, quotes or the parentheses.																																						
NOT-AUTHORIZED	Your client failed to log in, the username or password is wrong or you don't have the necessary permissions.																																						
NOT-AVAILABLE	Something was not available.																																						
ORDER-FAILED	<table border="0"> <tr> <td>:PLEASE-SLOW-DOWN</td> <td>You are entering orders too fast.</td> </tr> <tr> <td>:INACTIVE CUSTOMER</td> <td>User is inactive</td> </tr> <tr> <td>:USER-DENIED-TRADE</td> <td>User is denied trade</td> </tr> <tr> <td>:SECURITY-NOT-TRADEABLE</td> <td>Security is not tradeable</td> </tr> <tr> <td>:UNKNOWN-CUSTOMER</td> <td>Unknown customer</td> </tr> <tr> <td>:METHOD-REQUIRES-OPEN-EXCHANGE</td> <td>Exchange is not open</td> </tr> <tr> <td>:STOP-LOSS-NOT-SUPPORTED</td> <td>Exchange does not support stoploss.</td> </tr> <tr> <td>:X-ING-ORDER</td> <td>Crossing order not allowed</td> </tr> <tr> <td>:NO-SUCH-ORDER</td> <td>Order id not found</td> </tr> <tr> <td>:EXCHANGE-IS-CLOSED</td> <td>Exchange is closed</td> </tr> <tr> <td>:ORDER-NOT-FILLED</td> <td>Order was not filled</td> </tr> <tr> <td>:ORDER-NOT-ACTIVE</td> <td>Order is not active</td> </tr> <tr> <td>:ORDER-DOES-NOT-EXIST</td> <td>Order id not found</td> </tr> <tr> <td>:NEW-VOLUME-IS-ODD-LOT</td> <td>A quantity change resulted in an odd-lot order.</td> </tr> <tr> <td>:NEW-VOLUME-EXCEEDS-AVAILABLE</td> <td>New quantity is too large.</td> </tr> <tr> <td>:AUTOGIRO-CREDIT-LIMIT-FOR-NEW-VOLUME</td> <td>Your auto-giro limit is exceeded.</td> </tr> <tr> <td>:NOT-ENOUGH-AVAILABLE-FOR-NEW-VOLUME</td> <td>New quantity exceeds your available quantity.</td> </tr> <tr> <td>:NEW-VOLUME-LESS-THAN-DONE-VOLUME</td> <td>You are trying to cap a quantity below an already filled quantity.</td> </tr> <tr> <td>:NEW-LIMIT-OFF-BOUNDS</td> <td>Limit change is too large.</td> </tr> </table>	:PLEASE-SLOW-DOWN	You are entering orders too fast.	:INACTIVE CUSTOMER	User is inactive	:USER-DENIED-TRADE	User is denied trade	:SECURITY-NOT-TRADEABLE	Security is not tradeable	:UNKNOWN-CUSTOMER	Unknown customer	:METHOD-REQUIRES-OPEN-EXCHANGE	Exchange is not open	:STOP-LOSS-NOT-SUPPORTED	Exchange does not support stoploss.	:X-ING-ORDER	Crossing order not allowed	:NO-SUCH-ORDER	Order id not found	:EXCHANGE-IS-CLOSED	Exchange is closed	:ORDER-NOT-FILLED	Order was not filled	:ORDER-NOT-ACTIVE	Order is not active	:ORDER-DOES-NOT-EXIST	Order id not found	:NEW-VOLUME-IS-ODD-LOT	A quantity change resulted in an odd-lot order.	:NEW-VOLUME-EXCEEDS-AVAILABLE	New quantity is too large.	:AUTOGIRO-CREDIT-LIMIT-FOR-NEW-VOLUME	Your auto-giro limit is exceeded.	:NOT-ENOUGH-AVAILABLE-FOR-NEW-VOLUME	New quantity exceeds your available quantity.	:NEW-VOLUME-LESS-THAN-DONE-VOLUME	You are trying to cap a quantity below an already filled quantity.	:NEW-LIMIT-OFF-BOUNDS	Limit change is too large.
:PLEASE-SLOW-DOWN	You are entering orders too fast.																																						
:INACTIVE CUSTOMER	User is inactive																																						
:USER-DENIED-TRADE	User is denied trade																																						
:SECURITY-NOT-TRADEABLE	Security is not tradeable																																						
:UNKNOWN-CUSTOMER	Unknown customer																																						
:METHOD-REQUIRES-OPEN-EXCHANGE	Exchange is not open																																						
:STOP-LOSS-NOT-SUPPORTED	Exchange does not support stoploss.																																						
:X-ING-ORDER	Crossing order not allowed																																						
:NO-SUCH-ORDER	Order id not found																																						
:EXCHANGE-IS-CLOSED	Exchange is closed																																						
:ORDER-NOT-FILLED	Order was not filled																																						
:ORDER-NOT-ACTIVE	Order is not active																																						
:ORDER-DOES-NOT-EXIST	Order id not found																																						
:NEW-VOLUME-IS-ODD-LOT	A quantity change resulted in an odd-lot order.																																						
:NEW-VOLUME-EXCEEDS-AVAILABLE	New quantity is too large.																																						
:AUTOGIRO-CREDIT-LIMIT-FOR-NEW-VOLUME	Your auto-giro limit is exceeded.																																						
:NOT-ENOUGH-AVAILABLE-FOR-NEW-VOLUME	New quantity exceeds your available quantity.																																						
:NEW-VOLUME-LESS-THAN-DONE-VOLUME	You are trying to cap a quantity below an already filled quantity.																																						
:NEW-LIMIT-OFF-BOUNDS	Limit change is too large.																																						
INTERNAL-ERROR	Some internal malfunction.																																						
SIMPLE-ERROR	General error not caught by the exception handler.																																						

6. RELEASE NOTES

6.1 NFAPI Version 2.0

- **News messages.** Due to popular demand, clients may now subscribe to news from all our distributors. The messages will arrive in a feed similar to the security feed.
- Our (very rudimentary) "**Python example client**" has been rewritten to be more like a standalone library. It also supports **native Python classes of NFTP objects** and is finally compatible with **Python 3** and above.
- The **NFTP protocol has been extended** with an additional parameter (in position 2) called "source". Since this parameter will break all the existing clients, it is only available when you apply ":nftp-version 1.1" as an extra parameter to the login-command. The rationale behind this extension was the inherent limit of the protocol to distinguish between synchronous and asynchronous messages, since they all come from the same socket. This made it difficult for clients to implement RPC functions that returns data instead of feeding all the data through a callback handler. The new Python library makes use this, where you can say eg.

```
winners = api.winners("OSE")
```

and get a list of winners in return. Check out more examples in the **NFAPI-Python-lib.zip** package.

7. FINAL WORDS

Your client should be well tested before putting it in charge of your assets. Remember that accepted orders will go straight to the exchange for execution. Use the `(test-mode)` to test order entry. Also, try to limit your use of resources and subscribe only to the instruments you actually need.

All the servers will reboot every day at 03:00. This will throw out any connected clients.

Bug-reports, suggestions and recommendations on how to improve the API are most welcome!