WebSocket API

Version 2.3

Contents

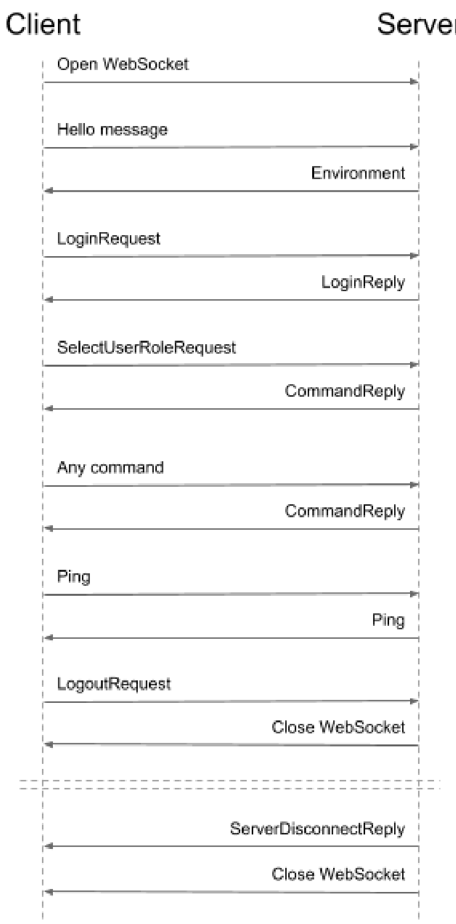
Revision history

Version	Description	UAT Release
1.0	First version of document.	30.11.2018
1.1	Added **ProtoVersion** field to messages. Added Logout Request message.	11.01.2018
1.2	Added **Server Disconnect** message.	08.02.2019
2.0	Modified versioning scheme, removed **ProtoVersion** and added **VersionMajor**, **VersionMinor** to **Hello Request** and **Environment Reply**. Simplified format of Optional fields. Changed meaning of password field in **Login Request**. Added **Subscribe Firms**, **Firms Update** messages. Added **Request Account States**, **Account States Update** messages.	07.03.2019
2.1	Added **ActiveSessions** field to JSON Client Info. Added **Change Account Request** and **Change Firm Request**.	05.04.2019
2.2	Added **Change Login Request** message. Added **Subscribe Trading Timeout Settings** and **Change Trading Timeout Settings** requests. Added **Trading Timeout Settings Snapshot** message. Added **Subscribe Audit Operations** and **Audit Operations Update** messages.	17.05.2019
2.3	Added **Subscribe Trades** request and **Trades Update** message. Added **Subscribe Orders** request and **Order States Update** message. Added **Subscribe Instruments** request and **Instruments Update** message.	21.06.2019

Introduction

This document describes how to use NTPro WebSocket API. You can use WebSocket API to get your trading data, for example your account's position, or to manage NTPro settings. The schema below describes the flow of WebSocket messages.

To keep connection in open state client should send any message to server every 10 seconds. You can send ping message if you have no any payload messages. If you do not get any messages from server during 10 seconds you should close current connection and try to establish a new one.



Versioning

Protocol has version, that consists of 2 integer numbers: major and minor part, like 2.0. All clients who have same major and less or equal minor version as on server are accepted to connect.

We notify clients two weeks before changes are applied to PROD environment. On UAT changes are available one week before PROD.

Changing minor version of the protocol means that data models might be expanded by new fields, new requests and responses could be added. But neither any model field nor requests and responses could be removed or modified.

Changing major version means incompatible changes in protocol.

WebSocket API messages

All messages has common JSON format.

```
{
  "Type": MessageType,
  "Msg": Message
}
```

There are three message types:

1. Ping request message
2. Ping reply messages
3. Data message.

Ping request

If you get “Ping request” message from server then you should send “Ping reply” back.

Field	Type	Comment
Type	Int32	Always equals 1
Msg	String	Current Unix UTC Time (ms ticks)

Example:

```
{"Type": 1, "Msg": "1542899051463"}
```

Ping reply

Field	Type	Comment
Type	Int32	Always equals 2
Msg	String	Current Unix UTC Time (ms ticks)

Example:

```
{"Type": 2, "Msg": "1542899051463" }
```

Data message

Field	Type	Comment
Type	Int32	Always set as 3
Msg	String	JSON serialized payload message

Example:

```
{
  "Type": 3,
  "Msg": "{\"RequestId\":1,\"OriginalRequestId\":1,\"RequestType\":\"Hello\",\"Message\":{}}"
}
```

Messages

All payload messages are sent to server inside Data message structure. All payload messages have the following format.

Field	Type	Comment
RequestId	Int64	Unique request identifier. Actually it is sequence number for messages. Every time the message sends, you must increment the sequence number for outgoing message.
OriginalRequestId	Int64	For requests it normally coincides with RequestId, for replies it references to the original request.

Field	Type	Comment
RequestType	String	String represents the request type ("*Hello*", "*LoginRequest*", "*SelectUserRoleRequest*", "*SubscribePositions*" etc.)
Message	JSON	Request/Reply message

Example:

```
{
  "RequestId": 2,
  "OriginalRequestId": 2,
  "RequestType": "LoginRequest",
  "Message": {
    "UserName": "managerwstest",
    "Password": "password",
    "ClientAppType": 1
  }
}
```

Message special data types

JSON DateTime

Field	Type	Comment
count	Int64	Milliseconds since started Unix epoch

Example:

```
{"count": 1542153600000}
```

JSON KeyValuePair<KeyType, ValueType>

Field	Type
key	KeyType
value	ValueType

Example:

```
{"key": "USD", value: "1"}
```

Optional Type

Means that object may have no meaningfull value for this property and then it wouldn't appear in JSON.

Payload messages

All payload messages are sent as part of Data message.

Hello Request

RequestType: Hello

Field	Type	Comment
VersionMajor	Int32	Major version of protocol, supported by client
VersionMinor	Int32	Minor version of protocol, supported by client

Example:

```
{
  "VersionMajor": 2,
  "VersionMinor": 0
}
```

You should start session with “Hello” message. After you send “Hello” message you will get back “Environment” message.

Environment Reply

RequestType: Environment

Field	Type	Comment
Type	Int32	For UAT is 1, for PROD is 2
VersionMajor	Int32	Major version of server protocol
VersionMinor	Int32	Minor version of server protocol

Example:

```
{
  "Type": 1,
  "VersionMajor": 2,
  "VersionMinor": 0
}
```

Login Request

RequestType: LoginRequest

Field	Type	Comment
UserName	String	User name
Password	String	String representing a plain text password
ClientAppType	Int32	Always equals 1

Example:

```
{
  "UserName": "TestWSManager",
  "Password": "ClientPassword",
  "ClientAppType": 1
}
```

After you send “Login Request” message you will get back “Login Reply” message.

Login Reply

RequestType: LoginReply

Field	Type	Comment
Reply	JSON Command Reply	
AvailableRoles	Array of Int32	Always equals `[2]`
DefaultRole	Int32	Always equals 2

Example:

```
{
  "Reply": {
    "Result": 0,
    "Message": "Login succeeded"
  },
  "AvailableRoles": [2],
  "DefaultRole": 2
}
```

Select User Role Request

RequestType: SelectUserRoleRequest

Field	Type	Comment
SelectedRole	Int32	Always equals 2

Example:

```
{
  "SelectedRole": 2
}
```

After you send “Select User Role Request” you get “Command Reply” message.

Logout Request

RequestType: LogoutRequest

Example:

```
{ }
```

After you send “Logout Request” message server will close your session and will not send any message.

Command Reply

RequestType: CommandReply

Field	Type	Comment
Result	Int32	0 - Success 1 - Error
Message	String	Text message

“CommandReply” is the common reply message for the different commands.

Server Disconnect

RequestType: ServerDisconnect

Field	Type	Comment
Reason	Int32	0 - Heartbeat timeout expired 1 - Login is closed 2 - Password is changed 3 - Domain is changed 4 - No permissions 5 - Access filters was changed 6 - Server is stopped 1000 - Client version is not supported
Text	String	Text message

On server disconnect you will get “ServerDisconnect” message.

Subscribe Positions

RequestType: SubscribePositions

There are either AccountId or PortfolioId has to be set in the request.

Field	Type	Comment
AccountId	Optional Int64	Account identifier
PortfolioId	Optional Int64	Portfolio identifier

Example:

```
{
  "AccountId": %Account Id%,
  "PortfolioId": %Portfolio Id%
}
```

After “Subscribe Positions” you will get “Command reply” message and then you will get position snapshots. You will get position snapshots each time position is changed on server. You should subscribe once for each account or portfolio.

If you want to stop getting snapshots you should send unsubscribe message.

Unsubscribe

RequestType: Unsubscribe_%SubscriptionRequestType%

For example, to unsubscribe from positions **RequestType**: Unsubscribe_SubscribePositions.

Field	Type	Comment
SubscriptionId	Int64	RequestId of subscription message

Example:

```
{
  "SubscriptionId": 4
}
```

Use this unsubscribe message to unsubscribe from any subscription.

Position Snapshot

RequestType: PositionSnapshot

Field	Type	Comment
AccountId	Optional Int64	AccountId from Subscribe message
PortfolioId	Optional Int64	PortfolioId from Subscribe message
SpotQuotum	JSON Position Snapshot	Position for spot instruments
ForwardQuotum	JSON Position Snapshot	Position for forward instruments
SpotLimit	Optional Double	Spot limit from account settings
ForwardLimit	Optional Double	Forward limit from account settings
SpotProfit	Double	Spot Profit Loss in USD
ForwardProfit	Double	Forward Profit Loss in USD
SpotOpenPosition	Optional Double	Spot Open Position in USD
ForwardOpenPosition	Optional Double	Forward Open Position in USD
SpotPercentLimitUsage	Optional Double	Percent of spot limit utilization
ForwardPercentLimitUsage	Optional Double	Percent of forward limit utilization
AccountCurrency	String	Principal settings, same for all accounts of principal
CheckSpotLimit	Bool	Account settings of limit checking
CheckForwardLimit	Bool	Account settings of limit checking

JSON Position Snapshot

Field	Type	Comment
AssetPositions	Array of JSON Asset Position	Positions for each asset

Field	Type	Comment
MakersOpenPositionsInfo	Array of KeyValuePair<String, JSON Maker Position Info> Keys are maker codes	Limit value and limit usage for each maker code

JSON Maker Position Info

Field	Type	Comment
MakerOpenPosition	Optional Double	Value of open position in principal main currency (usual USD)
MakerPercentLimitUsage	Optional Double	Percent of limit utilization

JSON Asset Position

Field	Type	Comment
Asset	String	Asset
Amount	Double	Value of open position
AccountCurrencyAmount	Double	Value of open position in principal main currency (usual USD).
Date	Optional JSON DateTime	If empty then it's total position across all dates.
MakerCode	String	If empty then it's total position across all makers

Example:

```
{
  "AccountId": %Account Id%,
  "SpotQuotum": {
    "AssetPositions": [{
      "Asset": "RUB",
      "Amount": 45350.0,
      "AccountCurrencyAmount": -685.83,
      "Date": {"count": 1541635200000},
      "MakerCode": ""
    }, ... ],
    "MakersOpenPositionsInfo": [{
      "key": "ECNMAKER",
      "value": {
        "MakerOpenPosition": 5500685.83,
        "MakerPercentLimitUsage": 11.00137166
      }
    }, {
      "key": "EXECUTORMAKER",
      "value": {
        "MakerOpenPosition": 599609.35,
        "MakerPercentLimitUsage": 1.1992187
      }
    }]
  },
  "ForwardQuotum": {
    "AssetPositions": [{
      "Asset": "USD",
      "Amount": -1600000.0,
      "AccountCurrencyAmount": 1600000.0,
      "Date": {"count": 1542153600000},
      "MakerCode": ""
    }, ... ],
    "MakersOpenPositionsInfo": [{
      "key": "ECNMAKER",
      "value": {
        "MakerOpenPosition": 0.0,
        "MakerPercentLimitUsage": 0.0
      }
    }, {
      "key": "EXECUTORMAKER",
      "value": {
        "MakerOpenPosition": 1603622.59,
        "MakerPercentLimitUsage": 0.0
      }
    }]
  },
  "SpotLimit": 50000000.0,
  "ForwardLimit": 0.0,
  "SpotProfit": 5597048.29,
  "ForwardProfit": 17849.72,
  "SpotOpenPosition": 6100295.18,
  "ForwardOpenPosition": 1603622.59,
  "SpotPercentLimitUsage": 12.20059036,
  "ForwardPercentLimitUsage": 0.0,
  "AccountCurrency": "USD",
  "CheckSpotLimit": true,
  "CheckForwardLimit": false
}
```

Subscribe Interest Rates

RequestType: SubscribeInterestRates

Example:

```
{ }
```

After “Subscribe Interest Rates” you will get command reply and interest rate update.

Interest Rates Update

RequestType: InterestRatesUpdate

Field	Type
InterestRates	Array of JSON Interest Rate

JSON Interest Rate

Field	Type	Comment
Id	Int64	Identifier of interest rate item. If you use this structure to add new item then set this field as 0 (zero)
AssetId	Int64	Identifier of assets
AssetName	String	Asset name
BasePeriod	Int32	1 - O/N 2 - T/N 4 - 1W 5 - 2W 6 - 1M 7 - 2M 8 - 3M 9 - 6M 10 - 9M 11 - 1Y
BidRate	Double	Bid rate
AskRate	Double	Ask rate
OwnerFirmId	Int64	Always equal your principal id

Example:

```
{
  "InterestRates": [{
    "Id": 219,
    "AssetId": 4,
    "AssetName": "AUD",
    "BasePeriod": 1,
    "BidRate": 1.24,
    "AskRate": 1.63,
    "OwnerFirmId": %Your Firm Id%
  }, {
    "Id": 220,
    "AssetId": 4,
    "AssetName": "AUD",
    "BasePeriod": 2,
    "BidRate": 1.24,
    "AskRate": 1.63,
    "OwnerFirmId": %Your Firm Id%
  }, ... ]
}
```

Change Interest Rate Request

RequestType: ChangeInterestRateRequest

Field	Type	Comment
Action	Int32	1 - Add 2 - Change 3 - Delete
InterestRate	JSON Interest Rate	Interest rate item which should be changed

Example:

```
{
  "Action": 2,
  "InterestRate": {
    "Id": 219,
    "AssetId": 4,
    "AssetName": "AUD",
    "BasePeriod": 1,
    "BidRate": 1.24,
    "AskRate": 1.63,
    "OwnerFirmId": %Your Firm Id%
  }
}
```

After “Change Interest Rate Request” you should get command reply.

Subscribe Firms

RequestType: SubscribeFirms

Example:

```
{}
```

After “Subscribe Firms” you will get command reply and Firms update. You will get Firms update each time Firms are changed on server. If you want to stop getting Firms update you should send unsubscribe message

Firms Update

RequestType: FirmsUpdate

Field	Type
Firms	Array of JSON Firm

JSON Firm

Field	Type	Comment
Id	Int64	Identifier of Firm
Name	String	Firm name
Type	Int32	0 - Principal 1 - Client
OwnerFirmId	Optional Int64	Identifier of Principal Firm
SpotLimit	Int64	Firm Spot Limit
Domain	String	Firm Domain
MainCurrency	Optional Int32	Identifier of client subsystem currency. Usually "1" - USD
ResponsibleLoginId	Optional Int64	Responsible LoginId

Example:

```
{
  "Firms": [{
    "Id": 1,
    "Name": "Firm1",
    "Type": 0,
    "SpotLimit": "999999999999999",
    "Domain": "F1",
    "MainCurrency": 1,
    "ResponsibleLoginId": 1
  }, ... ]
}
```

Change Firm Request

RequestType: ChangeFirmRequest

Field	Type
Firm	JSON Firm Change Fields

JSON Firm Change Fields

Field	Type	Comment
Id	Int64	Identifier of Firm
Name	String	Firm name
SpotLimit	Int64	Firm Spot Limit
Domain	String	Firm Domain

Example:

```
{
  "Firm": {
    "Id": 100,
    "Name": "Firm",
    "SpotLimit": 1000000,
    "Domain": "DOMAIN"
  }
}
```

After “Change Firm Request” you should get command reply.

Request Account States

RequestType: RequestAccountStates

Example:

```
{}
```

After “Request Account States” you will get command reply and Accounts States. You will get Account States Update each time Accounts or Clients are changed on server. If you want to stop getting Account States Update you should send unsubscribe message.

Account States Update

RequestType: AccountStatesUpdate

Field	Type
AccountStates	Array of KeyValuePair<JSON Account Info, Array of JSON Client Info>

JSON Account Info

Field	Type	Comment
Id	Int64	Identifier of account
Name	String	Name of account
Permissions	Int32	Account permissions 0 - Open 1 - Indicative 2 - Closed
SpotLimit	JSON Limit Rule	Account spot limit

Field	Type	Comment
SpotDayLimit	JSON Limit Rule	Account spot day limit
ForwardLimit	JSON Limit Rule	Account forward limit
Code	String	Account bank code
FirmId	Int64	Firm of account
OwnerFirmId	Int64	Owner firm of client subsystem
LocationId	Optional Int32	1 - Moscow 2 - London
QuoterGroupId	Optional Int64	Quoter group identifier
ExecutorGroupId	Optional Int64	Executor group identifier
External	Bool	Indicates whether account is external (true) or internal (false)
PbCommission	Optional Double	The value of Prime Brokerage Commission (USD per mio)
LeverageGroupId	Optional Int64	Leverage group identifier
UseFokOrders	Bool	Indicated whether Full amount order are supported for account
DirectSourceIds	Array of Int64	List of direct sources identifiers

JSON Limit Rule

Field	Type	Comment
Limit	Double	Account limit
LimitRule	Int32	0 - Net 1 - Gross
PositionMarketType	Int32	0 - Spot 1 - Forward 2 - SpotDay
CheckLimit	Bool	Indicates whether limit should be checked to accept the client’s order

JSON Client Info

Field	Type	Comment
Id	Int64	Identifier of client
AccountId	Optional Int64	Identifier of client’s account

Field	Type	Comment
Type	Int32	0 - Ui client 1 - Fixmarket 2 - Fixorder 5 - Administrator 6 - System 8 - StopLoss 9 - Quoter 10 - AutoFill 11 - Executor 12 - Manager 14 - LP 15 - SystemTrader 17 - CrazyCoverer 18 - B2bCoverer 19 - PkCoverer 21 - PmCoverer 22 - ExternalClient 23 - Stp
StpType	Optional Int32	0 - Client Stp 1 - Principal Stp 2 - Primebank Stp
Permissions	Int32	Client permissions 0 - Open 1 - Indicative 2 - Closed
Name	String	Client name
UseStopLossMaxSlippage	Bool	Indicates whether Stop loss max slippage is applicable for the client
UseTradingTimeout	Bool	Indicates whether Trading timeout is applicable for the client
FixSenderCompId	Optional String	SenderCompID for fix client
FixTargetCompId	Optional String	TargetCompID for fix client
FixIncomingConnectionIpList	Optional String	Allowed IPs list for fix client
OrderRequestsDelay	Int64	Order request timeout for fix client in microseconds
FirmId	Int64	Identifier of client’s firm
ServiceId	Optional Int32	Service identifier
FixQuoteBookSendDelay	Optional Int64	Quotes sending timeout for fixmarket client in microseconds
DomainName	String	Client full domain name
SourceId	Optional Int64	Source identifier for fix gate clients
UseOrderPriceExecution	Bool	Indicated whether Use order price execution feature is applicable for the client

Field	Type	Comment
UseFixCounterparty	Bool	Indicated whether Use fix counterparty is set for the client (for external clients)
FixCounterparty	String	Fix counterparty code (for external clients)
UseFixCounterTrader	Bool	Indicated whether Use fix counter trader is set for the client (for external clients)
FixCounterTrader	String	Fix counter trader code (for external clients)
BtbUseCoverPrice	Bool	Indicated whether Btb Use cover price feature is applicable for the client
FixSessionType	Optional Int32	0 - Streaming 1 - Rfs
ActiveSessions	Array of KeyValuePair<Int32, Int32>	Indicates number of active sessions. Key: 0 - Desktop Terminal or Fix Session 1 - Mobile Terminal

Example:

```
{
  "AccountStates": [{
    "key": {
      "Id": 1,
      "Name": "TRADING_AF",
      "Permissions": 0,
      "SpotLimit": {
        "Limit": 0.0,
        "LimitRule": 0,
        "PositionMarketType": 0,
        "CheckLimit": true
      },
      "SpotDayLimit": {
        "Limit": 0.0,
        "LimitRule": 0,
        "PositionMarketType": 2,
        "CheckLimit": false
      },
      "ForwardLimit": {
        "Limit": 0.0,
        "LimitRule": 0,
        "PositionMarketType": 1,
        "CheckLimit": false
      },
      "Code": "TRADING",
      "FirmId": 1,
      "OwnerFirmId": 1,
      "External": false,
      "UseFokOrders": false,
      "DirectSourceIds": []
    },
    "value": [{
      "Id": 1000,
      "AccountId": 1,
      "Type": 10,
      "Permissions": 0,
      "Name": "TRADING_AF",
      "UseStopLossMaxSlippage": false,
      "UseTradingTimeout": false,
      "OrderRequestsDelay": 0,
      "ShowTradeCounterparty": false,
      "FirmId": 1,
      "OwnerFirmId": 1,
      "DomainName": "TRADING_AF@TRADING",
      "UseOrderPriceExecution": false,
      "UseFixCounterparty": false,
      "FixCounterparty": "",
      "UseFixCounterTrader": false,
      "FixCounterTrader": "",
      "BtbUseCoverPrice": false,
      "ActiveSessions": []
    }, ... ]
  }, ... ]
}
```


Change Account Request

RequestType: ChangeAccountRequest

Field	Type
Account	JSON Account Change Fields

JSON Account Change Fields

Field	Type	Comment
Id	Int64	Identifier of account
Name	String	Name of account
Permissions	Int32	Account permissions 0 - Open 1 - Indicative 2 - Closed
SpotLimit	JSON Limit Rule	Account spot limit
SpotDayLimit	JSON Limit Rule	Account spot day limit
ForwardLimit	JSON Limit Rule	Account forward limit
Code	String	Account bank code
PbCommission	Optional Double	The value of Prime Brokerage Commission (USD per mio)
UseFokOrders	Bool	Indicated whether Full amount order are supported for account

Example:


```
{
  "Account": {
    "Id": 100,
    "Name": "Account",
    "Permissions": 1,
    "SpotLimit": {
      "Limit": 0.0,
      "LimitRule": 0,
      "PositionMarketType": 0,
      "CheckLimit": true
    },
    "SpotDayLimit": {
      "Limit": 0.0,
      "LimitRule": 0,
      "PositionMarketType": 2,
      "CheckLimit": false
    },
    "ForwardLimit": {
      "Limit": 0.0,
      "LimitRule": 0,
      "PositionMarketType": 1,
      "CheckLimit": false
    },
    "Code": "CODE",
    "PbCommission": 1.5,
    "UseFokOrders": true
  }
}
```


After “Change Account Request” you should get command reply.

Change Login Request

RequestType: ChangeLoginRequest

Field	Type
Login	JSON Client Settings

JSON Client Settings

Field	Type	Comment
Id	Int64	Identifier of client
Permissions	Int32	Client permissions 0 - Open 1 - Indicative 2 - Closed
UseTradingTimeout	Bool	Indicates whether Trading timeout is applicable for the client

Example:

```
{
  "Login": {
    "Id": 1,
    "Permissions": 0,
    "UseTradingTimeout": false,
  }
}
```

After “Change Login Request” you should get command reply.

Subscribe Trading Timeout Settings

RequestType: SubscribeTradingTimeoutSettings

Example:

```
{}
```

After “Subscribe TradingTimeoutSettings” you will get command reply and TradingTimeoutSettingsSnapshot message. You will get updates each time Trading Timeout Settings are changed on server. If you want to stop getting updates you should send unsubscribe message.

Trading Timeout Settings Snapshot

RequestType: TradingTimeoutSettingsSnapshot

Field	Type
TradingTimeoutEntries	Array of JSON TradingTimeoutSettings

JSON TradingTimeoutSettings

Field	Type	Comment
Id	Optional Int64	Identifier of Trading Timeout Settings: in TradingTimeoutSettingsSnapshot: is always present in ChangeTradingTimeoutSettingsRequest: is missing for new entries
InstrumentBasisId	Int64	Identifier of Instrument Basis
IsEnabled	Bool	Indicated whether Trading Timeout Setting is enabled
OwnerFirmId	Int64	Identifier of Owner Firm
PlatformId	Int64	Identifier of Platform
TimeoutInMicrosecs	Int64	Timeout in microseconds

Example:

```
{
    "TradingTimeoutEntries": [
        {
            "Id": 1,
            "InstrumentBasisId": 1,
            "IsEnabled": true,
            "OwnerFirmId": 1,
            "PlatformId": 1,
            "TimeoutInMicrosecs": 5000000
        },
        {
            "Id": 2,
            "InstrumentBasisId": 2,
            "IsEnabled": true,
            "OwnerFirmId": 1,
            "PlatformId": 1,
            "TimeoutInMicrosecs": 5000000
        },
        {
            "Id": 3,
            "InstrumentBasisId": 3,
            "IsEnabled": true,
            "OwnerFirmId": 1,
            "PlatformId": 1,
            "TimeoutInMicrosecs": 5000000
        }
    ]
}
```

Change Trading Timeout Settings Request

RequestType: ChangeTradingTimeoutSettingsRequest

Field	Type	Comment
Action	Int32	1 - Add 2 - Change 3 - Delete
TradingTimeoutSetting	JSON TradingTimeoutSettings	Trading Timeout Settings item which should be changed

Example for add:

```
{
  "Action": 1,
  "TradingTimeoutSetting": {
    "InstrumentBasisId": 37,
    "IsEnabled": true,
    "TimeoutInMicrosecs": 1000001,
    "OwnerFirmId": 1,
    "PlatformId": 1
  }
}
```

Example for change:

```
{
  "Action": 2,
  "TradingTimeoutSetting": {
    "Id": 32,
    "InstrumentBasisId": 37,
    "IsEnabled": true,
    "TimeoutInMicrosecs": 1000002,
    "OwnerFirmId": 1,
    "PlatformId": 1
  }
}
```

Example for delete (you have to send full structure for delete request too):

```
{
  "Action": 3,
  "TradingTimeoutSetting": {
    "Id": 32,
    "InstrumentBasisId": 37,
    "IsEnabled": true,
    "TimeoutInMicrosecs": 1000002,
    "OwnerFirmId": 1,
    "PlatformId": 1
  }
}
```

After “Change TradingTimeoutSettings Request” you should get command reply.

Subscribe Audit Operations

RequestType: SubscribeAuditOperations

Field	Type
TimestampFrom	Optional JSON DateTime
TimestampTo	Optional JSON DateTime

Example:

```
{
    "TimestampFrom": {"count": 1557308651788}
}
```

After “Subscribe Audit Operations” you will get command reply and Audit Operations update. You will get Audit Operations update each time data is changed on server. If you want to stop getting Audit operations update you should send unsubscribe message

Audit Operations Update

RequestType: AuditOperationsUpdate

Field	Type	Comment
Data	Array of JSON Audit Operation	
IsNextDataPending	Bool	Server breaks up big messages (> 100 operations) into smaller ones. This flag will be True until server has sent last message from package

Audit Operation

Field	Type	Comment
Id	Int64	Identifier of Audit Operation
Timestamp	JSON DateTime	Timestamp
Category	Int32	Data category 1 - LoginLogout 2 - ControlButton 3 - Account 4 - QuoterGroups 5 - ExecutorGroups 6 - AggregatorGroups 7 - Instruments 8 - Permissions 9 - LPSources 10 - Firms 11 - LeverageGroups 12 - StopLossSettings 13 - InterestRates 14 - PositionManager 17 - OrderQuoterRules 18 - Portfolio 19 - Login 20 - AutoLoad 21 - TradingTimeoutSettings 22 - SyntheticGroups 23 - AccessGroups 25 - PasswordSettings 26 - Tags 27 - AccountSourceSettings
LoginId	Optional Int64	Identifier of Login

Field	Type	Comment
OwnerFirmId	Int64	Identifier of Owner Firm
Action	Int32	1 - Add 2 - Change 3 - Delete
NewState	String	New State
OldState	String	Old State
Result	Bool	Indicates whether Audit Operation Update is success or failed
ComponentName	String	Name of Component

Example:


```
{
    "Data": [
        {
            "Id": 7421825,
            "Timestamp": {"count": 1557820594692}
            "Category": 1,
            "LoginId": 2,
            "OwnerFirmId": 1,
            "Action": 1,
            "NewState": "Login;  Error: Invalid password; AppType: Desktop; ... ",
            "OldState": "",
            "Result": false,
            "ComponentName": "UiServerWorker0",
        },
        {
            "Id": 7421826,
            "Timestamp": {"count": 1557820597730}
            "Category": 2,
            "LoginId": 2,
            "OwnerFirmId": 1,
            "Action": 2,
            "NewState": "Button: Moex Md; Status: Started; Location: Moscow; ",
            "OldState": "Button: Moex Md; Status: Stopped; Location: Moscow; ",
            "Result": true,
            "ComponentName": "TradingControl",
        }
    ],
    "IsNextDataPending": false
}
```


Subscribe Trades

RequestType: SubscribeTrades

Field	Type
StartTime	Optional JSON DateTime
EndTime	Optional JSON DateTime
ShowPrimeBankTrades	Bool

Example:

```
{
    "StartTime": {"count": 1557308651788},
    "ShowPrimeBankTrades": False
}
```

After “Subscribe Trades” you will get command reply and Trades update. You will get Trades update each time data is changed on server. If you want to stop getting Trades updates you should send unsubscribe message

Trades Update

RequestType: TradesUpdate

Field	Type	Comment
Trades	Array of JSON Trade	
IsNextDataPending	Bool	Server breaks up big messages (> 100 operations) into smaller ones. This flag will be True until server has sent last message from package

Trade

Field	Type	Comment
Account	String	Trader account's name
Comment	Optional String	Additional information
CommissionPriceShift	Optional Double	Comission added to price
CounterPartyAccount	String	Name of counterparty account
CounterPartyCode	String	Bank code of counterparty
CounterPartyTrader	String	Name of counterparty trader
CounterQuantity	Double	Deal's volume in quoted currency
Side	Int32	Deal side in relation to dealer (principal) 0 - Buy 1 - Sell
EcnCounterpartyLoginId	Optional Int64	Identifier of real counterparty login on ECN platform
External	Bool	Indicates whether one of the deal sides is related to external account
ExternalClientId	Optional String	Identifier of external source's client
ExternalExecutionId	Optional String	Identifier of external source's deal
ExternalOrderId	Optional String	Identifier of external source's order
FirstRate	Optional Double	Swap near leg price
Id	Int64	Identifier of deal

Field	Type	Comment
InitiatorLogin	String	Deal initiator's login
InitiatorLoginId	Int64	Identifier of deal initiator's login
InstrumentBasisId	Int64	Identifier of instrument basis
IsBaseQuantity	Bool	Indicates whether deal's volume was set in base currency
IsCancelation	Bool	Indicates whether this deal is cancelation (cancelation deal has inversed side in relation to cancelled deal)
LastOperationTime	Optional JSON DateTime	Actual date and time of deal in system. It is may be differ from Timestamp if deal has been maked manually
MakerCode	String	Maker code
OperationIdChainSize	Int64	Sequence number of current deal in order execution related deals chain
OperationIdGuid	String	Identifier of an order execution related deals chain
OrderId	String	Identifier of order which deal is related to
OwnerFirmName	String	Principal, that trader's firm belongs to
PlatformId	Int64	Identifier of platform
Price	Double	Deal price or swap price (for swap trades)
Quantity	Double	Deal volume in base currency
ReferenceId	Optional String	Associated deal identificator or cancelled deal identifier in the case when deal is a cancelation
ResponsibleLogin	Optional String	Responsible person on principal side
SecondCounterQuantity	Optional Double	Swap far leg volume in quoted currency
SecondQuantity	Optional Double	Swap far leg volume in base currency
SecondValueDate	Optional JSON DateTime	Swap far leg value date
SendToStp	Bool	Indicates whether deal details were sent to client's STP client
SourceId	Int64	Identifier of liquidity source
SourceName	String	Liquidity source name
Timestamp	JSON DateTime	Date and time of deal
Trader	String	Trader's name
ValueDate	JSON DateTime	Deal's value date

Example:

```
{
    "Type": 3,
    "Msg": {
        "OriginalRequestId": 4,
        "RequestId": 7,
        "RequestType": "TradesUpdate",
        "Message": {
            "IsNextDataPending": false,
            "Trades": [
                {
                    "Account": "Account",
                    "Comment": "",
                    "CounterPartyAccount": "ac1",
                    "CounterPartyCode": "ABC",
                    "CounterPartyTrader": "login",
                    "CounterQuantity": 6700000,
                    "Direction": 0,
                    "External": true,
                    "Id": 14361747,
                    "InitiatorLogin": "login",
                    "InitiatorLoginId": 1100108,
                    "InstrumentBasisId": 418,
                    "IsBaseQuantity": true,
                    "IsCancelation": false,
                    "LastOperionTime": {"count": 1558680620910},
                    "MakerCode": "MAKER",
                    "OperationIdChainSize": 0,
                    "OperationIdGuid": "9aa27ba4-27ca-4b03-9ec4-234800a37a21",
                    "OrderId": "dc9400de-0826-48db-b347-a74418a8c775",
                    "OwnerFirmName": "FirmName",
                    "PlatformId": 31,
                    "Price": 67,
                    "Quantity": 100000,
                    "SendToStp": true,
                    "SourceId": 2,
                    "SourceName": "Executor",
                    "Timestamp": {"count": 1558680282280},
                    "Trader": "Trader",
                    "ValueDate": {"count": 1558656000000}
                }
            ]
        }
    }
}
```


Subscribe Orders

RequestType: SubscribeOrders

Field	Type
AccountId	Optional Int64
StartTime	Optional JSON DateTime
EndTime	Optional JSON DateTime
OnlyActive	Bool

Example:

```
{
    "StartTime": {"count": 1559043536607},
    "OnlyActive": true,
}
```

After “Subscribe Orders” you will get command reply and Order States update. You will get Order States update each time data is changed on server. If you want to stop getting Order States updates you should send unsubscribe message

Order States Update

RequestType: OrderStatesUpdate

Field	Type	Comment
OrderStates	Array of JSON Order State	
IsNextDataPending	Bool	Server breaks up big messages (> 100 operations) into smaller ones. This flag will be True until server has sent last message from package

Order

Field	Type	Comment
Id	Optional String	Identifier of order
ExternalOrderId	Optional String	Identifier of external source's order
InstrumentBasisId	Int64	Identifier of instrument basis
PlatformId	Int64	Identifier of platform
SourceId	Int64	Identifier of liquidity source
Side	Int32	Order side 0 - Buy 1 - Sell
TimeInForce	Int32	Indicates how long the order remains in effect 0 - Gtc 1 - Ioc
Type	Int32	Type of order 0 - Limit 1 - Market 2 - MarketLimit 4 - StopLoss 5 - Oco
TraderId	Int64	Trader's login identifier
TraderName	Optional String	Trader's login name
TraderAccountName	Optional String	Trader's account name
TraderFirmName	Optional String	Firm name, that trader belongs to
TraderOwnerFirmName	Optional String	Principal, that trader's firm belongs to

Field	Type	Comment
FullAmount	Bool	Indicates whether full amount order filling is enabled
ValueDate	Optional JSON DateTime	Order's value date
FirstRate	Optional Double	Original first rate, set for swap only
ExternalClientId	Optional String	Client's identifier from external site
IsBaseQuantity	Bool	Indicates whether orders's volume was set in base currency
InitiatorLoginId	Int64	Identifier of order initiator's login
InitiatorLogin	String	Name of initiator's login
OcoRelatedOrderId	Optional String	Identifier of the associated order in case of OCO (One Cancel Other)
OperationId	String	Identifier of operation
OperationIdChainSize	Int64	Sequence number of current deal in order execution related deals chain

Order State

Field	Type	Comment
Order	JSON Order	Immutable part of order
Type	Int32	Current state of order 0 - Active 1 - Canceled 2 - Rejected 3 - Filled
Quantity	Double	Order's quantity
Price	Optional Double	Original order's price, not set for market orders
AvgFillPrice	Optional Double	Average price of filled order's amount
OrderMessage	Optional String	Order's message in case of errors
OrderCreationTime	JSON DateTime	Order's creation time
LastChangeTime	JSON DateTime	Order's last change time
SourceId	Int64	Identifier of liquidity source
SourceName	String	Name of liquidity source
Comment	Optional String	Additional information
ExpirationTime	Optional JSON DateTime	Order's expiration time

Example:

```
{
    "OrderStates": [
        {
            "Order": {
                "FullAmount": false,
                "Id": "97d1031d-ed70-4c3e-af83-ebdcbbba6700",
                "InitiatorLogin": "login",
                "InitiatorLoginId": 1100351,
                "InstrumentBasisId": 25,
                "IsBaseQuantity": true,
                "OperationId": "14cfcb7d-c5e9-425e-ae37-faa1a148d5b2",
                "OperationIdChainSize": 0,
                "TraderAccountName": "AccountName",
                "TraderFirmName": "FirmName",
                "TraderOwnerFirmName": "OwnerFirmName",
                "TraderId": 1100351,
                "TraderName": "login",
                "PlatformId": 20,
                "Side": 0,
                "SourceId": 1342,
                "TimeInForce": 0,
                "Type": 0
            },
            "Type": 0,
            "Quantity": 100000,
            "Price": 108.541,
            "AvgFillPrice": 108.541,
            "OrderCreationTime": { "count": 1559042855938 },
            "LastChangeTime": { "count": 1559116410200 },
            "SourceId": 1342,
            "SourceName": "Executor_Mobile1",
        }
    ]
}
```

Subscribe Instruments

RequestType: SubscribeInstruments

Example:

```
{}
```

After “Subscribe Instruments” you will get command reply and Instruments update. You will get Instruments update each time data is changed on server. If you want to stop getting Instruments updates you should send unsubscribe message

Instruments Update

RequestType: InstrumentsUpdate

Field	Type	Comment
Instruments	Array of JSON Instrument	

Field	Type	Comment
IsNextDataPending	Bool	Server breaks up big messages (> 100 operations) into smaller ones. This flag will be True until server has sent last message from package

Instrument

Instrument object is a combination of immutable InstrumentBasis part and some fields related to certain platform. So primary unique key for Instrument is combination of InstrumentBasisId and PlatformId.

Field	Type	Comment
InstrumentBasisId	Int64	Identifier of instrument basis
Name	String	Full name (with tenor) of instrument
BaseAssetId	Int64	Identifier of base asset
BaseAsset	String	Name of base asset
BaseAssetType	Int32	Type of base asset 0 - Forex 1 - Crypto
QuoteAssetId	Int64	Identifier of quote asset
QuoteAsset	String	Name of quote asset
QuoteAssetType	Int32	Type of quote asset 0 - Forex 1 - Crypto

Field	Type	Comment
Tenor	Int32	Instrument's tenor 0 - TOD 1 - TOM 2 - SPOT 3 - OUTRIGHT 4 - 1W 5 - 2W 6 - 1M 7 - 2M 8 - 3M 9 - 6M 10 - 9M 11 - 1Y 12 - O/N 13 - T/N 14 - JAN 15 - FEB 16 - MAR 17 - APR 18 - MAY 19 - JUN 20 - JUL 21 - AUG 22 - SEP 23 - OCT 24 - NOV 25 - DEC
PointSize	Double	Amount of pip
Type	Int32	Type of instrument 0 - Exchange 1 - Swap 2 - Settlement 3 - Forward 4 - Futures
PlatformId	Int64	Identifier of platform
PriceStep	Double	Indicates accuracy of the price setting
IsIndicative	Bool	Indicates whether instrument if indicative
OrderQuantityStep	Double	Step of order's quantity
OrderCounterQuantityStep	Optional Double	Step of order's counter quantity
MinOrderQuantity	Double	Minimal size of order's quantity
MinOrderCounterQuantity	Optional Double	Minimal size of order's counter quantity

Example:

```
{
    "Instruments": [
        {
            "BaseAsset": "USD",
            "BaseAssetId": 1,
            "BaseAssetType": 0,
            "InstrumentBasisId": 50,
            "IsIndicative": false,
            "MinOrderQuantity": 1,
            "OrderQuantityStep": 1,
            "PlatformId": 204,
            "PointSize": 0.01,
            "PriceStep": 0.001,
            "QuoteAsset": "JPY",
            "QuoteAssetId": 7,
            "QuoteAssetType": 0,
            "Name": "USD/JPY_TOM",
            "Tenor": 1,
            "Type": 0
        }
    ]
}
```