Trading API – Orion Lite Python Language

(Detailed understanding and Implementation guide)

Date of document: 30 Aug, 2021
Version: 2.0 – Draft

Prepared by: Technology department

Contents

1. LIBRARIES, ORIONLITEPY AND HEADER DETAILS

- Libraries needs to Install
  - pycryptodome { pip install pycryptodome }
- **How to use OrionLitePy Module**
  - **from OrionLitePy import OrionLitePy**

    For Sending Request. { e.g :- OrionLitePy.LoginRequest (UesrID, Password) }
  - **from OrionLitePy. OrionLitePy import event**

    For Catching events. { e.g :- event.LoginResponse += Login }
- **Data Type Used**

Type
String
Int
Float

- **Message Header Response**

In every Response user will get Message Header in following structure.

Name	Type	Description
MessageCode	Int	Contains the message number for the messages received or sent. A message is invalid without this.
MessageLength	Int	Specifies the length of the entire message including the length of the Message Header.
ChannelID	Int	1 (one)
ClientID	String	Logged in ID from which message is coming.
TimeStamp	Int	Time at which message is generated in terms of seconds from 01-01-1980 00:00:00 hours.
ErrorCode	Int	Outbound field. Always set to 0(zero) in request.
ErroMessageLength	Int	Outbound field. Always set to 0(zero) in request. In response if any value is there in this field then Message length will be increased by this and error text will come at the end of packet of this length.
NumberOfDecimals	Int	Always set to 2 in request. In response usually value will be 2 but some cases it may be different.
Source	String	Always set to ‘L’.
ClientDispatcherID	Int	0(zero)

2. LIST OF API

- Login API
- Change Password
- Forgot Password
- Order Management System (OMS)
  - Orders
  - Trade
  - New Order
  - Modify Order
  - Cancel Order
  - Trade Confirmation
- Reports
  - Order Download
  - Trade Download
  - Net Position
  - Margin Report
  - Error
- Broadcast
  - Subscribing Quote
  - Un-Subscribing Quote
  - Subscribing Depth
  - Un-Subscribing Depth
- Headers
- Responses

3. LOGIN

Login Request

OrionLitePy.LoginRequest (UserID , Password)

Name	Type	Description
UserID	String	ID provided by MOSL.
Password	String	Password of the ID.

Login Response

Message Code – 2. {event.LoginResponse += LoginResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
UserID	String	ID provided by MOSL.
UserType	Int	0 – Client 1 – Dealer
PwdValidityDays	Int	Days left for password to expire.
LastLogonDateTime	Int	Last logged in date time in terms of seconds from 01-01-1980 00:00:00 hours.
FirstTimeLogin	String	‘Y’ if user is login for first time.
PwdExpired	String	‘Y’ if password is expired.
SessionNo	Int	Reserved field.
ExchangeRights	Int	Reserved field.
MaxScripCount	Int	Reserved field.
SleepBetweenOrders	Int	Reserved field.
AvailableVersion	Int	Reserved field.

Change Password Request

OrionLitePy.ChangedPasswordRequest (UserID, Password, NewPassword)

Name	Type	Description
UserID	String	ID provided by MOSL.
Password	String	Old Password of the ID.
NewPassword	String	New Password of the ID.

Change Password Response

Message Code – 51. {event.ChangePasswordResponse += ChangePasswordResponseFunc }

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
PwdValidityDays	Int	New password is valid for this many days.

Forgot Password Request

OrionLitePy.ForgotPasswordRequest (UserID, EmailorPhone)

Name	Type	Description
UserID	String	ID provided by MOSL.
EmailOrMobile	String	Old Password of the ID.

Forgot Password Response

Message Code – 3 (three). {event. ForgotPasswordResponse += ForgotPasswordResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
UserID	String	ID provided by MOSL.
EmailOrMobile	String	Old Password of the ID.

4. OMS – ORDER MANAGEMENT SYSTEM

- **Order Management (OMS):** Field-wise Structure

- **Order**

In Order related Response user will get Order Response in following structure.

Name	Type	Description
ExchangeCode	Int	For MCX commodity, set to 6.
ScripCode	Int	Code of the instrument as per exchange.
OrderNo	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ClientCode	String	ID of client for which order is being placed.
BuyOrSell	String	‘B’ – Buy ‘S’ - Sell
DisclosedQty	Int	It should be >= 0 and in Lots.
TotalQtyRemaining	Int	It contains total qty remaining for the order. For fresh order request it should be same as OrderQty.
OrderQty	Int	It should be > 0 and in Lots.
QtyTradedToday	Int	It contains qty traded today for the order.
Price	Int	It should be >= 0. For market order it should be 0.
TriggerPrice	Int	It is mandatory for stop loss order else it will be 0.
GoodTillDate	Int	Date up to which order should be retained in terms of seconds from 01-01-1980 00:00:00 hours.
EntryDateTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
MinFillQty	Int	0 (zero)
LastModifiedTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
OrderInitiatorID	String	Contains logged in ID from which order is generated. Set it at the time of fresh order entry and send same value in case of modification/cancellation.
Remarks	String	Reserved field.
OrderType	String	‘C’
BookType	String	‘P’ – Stop loss order ‘L’ – Limit Order ‘M’ – Market Order

OrderLife	String	'D' – Day order 'I' – IOC order 'G' – GTD order 'C' – GTC order
ExchangeDispatcherID	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ProductCode	Int	1
ProductFlag	String	' ' – One space
TerminalID	String	Logged in ID which is doing the current activity.
UniqueOrderID	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ExpiryDate	Int	Expiry date of scrip in terms of seconds from 01-01-1980 00:00:00 hours.
Symbol	String	Symbol or short name of scrip.
Series	String	'XX'
InstrumentName	String	'FUTCOM'
StrikePrice	Int	-1
CALevel	Int	0
BunchSeqNo	Int	0
AMOOrder	String	'N'
AMOOrderValidated	String	'N'
MPlusBlockedAmount	float	0
ParticipantType	String	'N'
UserRefText	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
OrderStatus	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ParticipantCode	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
PanNumber	String	This field will get populated on server, by default send it blank.
AlgoId	Int	By default send it 0
AlgoCategory	Int	By default send it 0

Trade

In Trade Response user will get Trade Response in following Structure.

Name	Type	Description
ExchangeCode	Int	For MCX commodity, it is 6.
OrderNo	String	Order number of corresponding order to which this trade belongs.
TerminalID	String	Logged in ID which is doing the current activity.
OrderInitiatorID	String	Contains logged in ID from which order is generated.
ScripCode	Int	Code of the instrument as per exchange.
ClientCode	String	ID of client for which order is being placed.
BuyOrSell	String	'B' – Buy 'S' - Sell
OrderQty	Int	It will be in Lots.
DisclosedQty	Int	It will be in Lots.
QtyRemaining	Int	It contains total qty remaining for the order
TradeNo	String	Trade number provided by exchange.
TradeQty	Int	Trade qty in Lots.
TradePrice	Int	Price at which trade is executed.
TradeDateTime	Int	Time at which it got traded in exchange.
OrderType	String	'C'
BookType	String	'P' – Stop loss order 'L' – Limit Order 'M' – Market Order
OrderLife	String	'D' – Day order 'I' – IOC order 'G' – GTD order 'C' – GTC order
ProductCode	Int	1
ProductFlag	String	' ' – One space
UniqueOrderID	String	Unique Identification number generated by system.
ExchangeDispatcherID	Int	For internal use.

New Order Request:

- OrionLitePy.NewOrder (UserID, ExchangeCode, ScripCode, OrderNo, ClientCode, BuyOrSell, DisclosedQty, TotalQtyRemaining, OrderQty, QtyTradedToday, Price, TriggerPrice, GoodTillDate, EntryDateTime, MinFillQty, LastModifiedTime, OrderInitiatorID, Remarks,OrderType, BookType,OrderLife, ExchangeDispatcherID, ProductCode, ProductFlag, TerminalID, UniqueOrderID, ExpiryDate, Symbol, Series, InstrumentName, StrikePrice, CALevel, BunchSeqNo, AMOorder, AMOorderValidated, MPlusBlockedAmt, ParticipantType, UserRefText, OrderStatus, ParticipantCode ,PANnumber, AlgoID, AlgoCategory)

Name	Type	Description
UserID	String	ID Provided by MOSL.
ExchangeCode	Int	For MCX commodity, set to 6.
ScripCode	Int	Code of the instrument as per exchange.
OrderNo	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ClientCode	String	ID of client for which order is being placed.
BuyOrSell	String	‘B’ – Buy ‘S’ - Sell
DisclosedQty	Int	It should be >= 0 and in Lots.
TotalQtyRemaining	Int	It contains total qty remaining for the order. For fresh order request it should be same as OrderQty.
OrderQty	Int	It should be > 0 and in Lots.
QtyTradedToday	Int	It contains qty traded today for the order.
Price	Int	It should be >= 0. For market order it should be 0.
TriggerPrice	Int	It is mandatory for stop loss order else it will be 0.
GoodTillDate	Int	Date up to which order should be retained in terms of seconds from 01-01-1980 00:00:00 hours.
EntryDateTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
MinFillQty	Int	0 (zero)
LastModifiedTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
OrderInitiatorID	String	Contains logged in ID from which order is generated. Set it at the time of fresh order entry and send same value in case of modification/cancellation.
Remarks	String	Reserved field.
OrderType	String	‘C’
BookType	String	‘P’ – Stop loss order ‘L’ – Limit Order ‘M’ – Market Order
OrderLife	String	‘D’ – Day order ‘I’ – IOC order ‘G’ – GTD order

		‘C’ – GTC order
ExchangeDispatcherID	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ProductCode	Int	1
ProductFlag	String	‘ ’ – One space
TerminalID	String	Logged in ID which is doing the current activity.
UniqueOrderID	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ExpiryDate	Int	Expiry date of scrip in terms of seconds from 01-01-1980 00:00:00 hours.
Symbol	String	Symbol or short name of scrip.
Series	String	‘XX’
InstrumentName	String	‘FUTCOM’
StrikePrice	Int	-1
CALevel	Int	0
BunchSeqNo	Int	0
AMOOrder	String	‘N’
AMOOrderValidated	String	‘N’
MPlusBlockedAmount	Float	0
ParticipantType	String	‘N’
UserRefText	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
OrderStatus	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ParticipantCode	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
PanNumber	String	This field will get populated on server, by default send it blank.
AlgoId	Int	By default send it 0
AlgoCategory	Int	By default send it 0

New Order Response:

Message Code = 13. “Order Confirm”. {event.OrderNewResponse += OrderNewResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 16. “Order Error”. {event. OrderErrorResponse += OrderErrorResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 19. “Order Reject”. {event. OrderRejectResponse += OrderRejectResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Modify Order Request:

- OrionLitePy.ModifyOrder
(UserID, ExchangeCode, ScripCode, OrderNo, ClientCode, BuyOrSell, DisclosedQty, TotalQtyRemaining, OrderQty, QtyTradedToday, Price, TriggerPrice, GoodTillDate, EntryDateTime, MinFillQty, LastModifiedTime, OrderInitiatorID, Remarks,OrderType, BookType,OrderLife, ExchangeDispatcherID, ProductCode, ProductFlag, TerminalID, UniqueOrderID, ExpiryDate, Symbol, Series, InstrumentName, StrikePrice, CALevel, BunchSeqNo, AMOorder, AMOorderValidated, MPlusBlockedAmt, ParticipantType, UserRefText, OrderStatus, ParticipantCode ,PANnumber, AlgoID, AlgoCategory)

Name	Type	Description
UserID	String	ID Provided by MOSL.
ExchangeCode	Int	For MCX commodity, set to 6.
ScripCode	Int	Code of the instrument as per exchange.
OrderNo	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ClientCode	String	ID of client for which order is being placed.
BuyOrSell	String	'B' – Buy 'S' - Sell
DisclosedQty	Int	It should be >= 0 and in Lots.
TotalQtyRemaining	Int	It contains total qty remaining for the order. For fresh order request it should be same as OrderQty.
OrderQty	Int	It should be > 0 and in Lots.
QtyTradedToday	Int	It contains qty traded today for the order.
Price	Int	It should be >= 0. For market order it should be 0.
TriggerPrice	Int	It is mandatory for stop loss order else it will be 0.
GoodTillDate	Int	Date up to which order should be retained in terms of seconds from 01-01-1980 00:00:00 hours.
EntryDateTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
MinFillQty	Int	0 (zero)
LastModifiedTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
OrderInitiatorID	String	Contains logged in ID from which order is generated. Set it at the time of fresh order entry and send same value in case of modification/cancellation.
Remarks	String	Reserved field.
OrderType	String	'C'
BookType	String	'P' – Stop loss order 'L' – Limit Order 'M' – Market Order

OrderLife	String	'D' – Day order 'I' – IOC order 'G' – GTD order 'C' – GTC order
ExchangeDispatcherID	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ProductCode	Int	1
ProductFlag	String	' ' – One space
TerminalID	String	Logged in ID which is doing the current activity.
UniqueOrderID	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ExpiryDate	Int	Expiry date of scrip in terms of seconds from 01-01-1980 00:00:00 hours.
Symbol	String	Symbol or short name of scrip.
Series	String	'XX'
InstrumentName	String	'FUTCOM'
StrikePrice	Int	-1
CALevel	Int	0
BunchSeqNo	Int	0
AMOOrder	String	'N'
AMOOrderValidated	String	'N'
MPlusBlockedAmount	Float	0
ParticipantType	String	'N'
UserRefText	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
OrderStatus	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ParticipantCode	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
PanNumber	String	This field will get populated on server, by default send it blank.
AlgoId	Int	By default send it 0
AlgoCategory	Int	By default send it 0

Modify Order Response:

Message Code = 14. “Order Modify Confirm”. {event.OrderModifiedResponse += OMRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 16. “Order Error”. {event.OrderErrorResponse =+ OrderErrorResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 20. “Order Modify Reject”. {event.OrderModifyRejectResponse += OMRRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Cancel Order Request:

- OrionLitePy.CancelOrder (UserID, ExchangeCode, ScripCode, OrderNo, ClientCode, BuyOrSell, DisclosedQty, TotalQtyRemaining, OrderQty, QtyTradedToday, Price, TriggerPrice, GoodTillDate, EntryDateTime, MinFillQty, LastModifiedTime, OrderInitiatorID, Remarks,OrderType, BookType,OrderLife, ExchangeDispatcherID, ProductCode, ProductFlag, TerminalID, UniqueOrderID, ExpiryDate, Symbol, Series, InstrumentName, StrikePrice, CALevel, BunchSeqNo, AMOorder, AMOorderValidated, MPlusBlockedAmt, ParticipantType, UserRefText, OrderStatus, ParticipantCode ,PANnumber, AlgoID, AlgoCategory)

Name	Type	Description
UserID	String	ID Provided by MOSL.
ExchangeCode	Int	For MCX commodity, set to 6.
ScripCode	Int	Code of the instrument as per exchange.
OrderNo	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ClientCode	String	ID of client for which order is being placed.
BuyOrSell	String	'B' – Buy 'S' - Sell
DisclosedQty	Int	It should be >= 0 and in Lots.
TotalQtyRemaining	Int	It contains total qty remaining for the order. For fresh order request it should be same as OrderQty.
OrderQty	Int	It should be > 0 and in Lots.
QtyTradedToday	Int	It contains qty traded today for the order.
Price	Int	It should be >= 0. For market order it should be 0.
TriggerPrice	Int	It is mandatory for stop loss order else it will be 0.
GoodTillDate	Int	Date up to which order should be retained in terms of seconds from 01-01-1980 00:00:00 hours.
EntryDateTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
MinFillQty	Int	0 (zero)
LastModifiedTime	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
OrderInitiatorID	String	Contains logged in ID from which order is generated. Set it at the time of fresh order entry and send same value in case of modification/cancellation.
Remarks	String	Reserved field.
OrderType	String	'C'
BookType	String	'P' – Stop loss order 'L' – Limit Order 'M' – Market Order
OrderLife	String	'D' – Day order 'I' – IOC order 'G' – GTD order

		'C' – GTC order
ExchangeDispatcherID	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ProductCode	Int	1
ProductFlag	String	' ' – One space
TerminalID	String	Logged in ID which is doing the current activity.
UniqueOrderID	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
ExpiryDate	Int	Expiry date of scrip in terms of seconds from 01-01-1980 00:00:00 hours.
Symbol	String	Symbol or short name of scrip.
Series	String	'XX'
InstrumentName	String	'FUTCOM'
StrikePrice	Int	-1
CALevel	Int	0
BunchSeqNo	Int	0
AMOOrder	String	'N'
AMOOrderValidated	String	'N'
MPlusBlockedAmount	Float	0
ParticipantType	String	'N'
UserRefText	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
OrderStatus	Int	Send 0 in case of fresh order request and send same value in case of modification/cancellation.
ParticipantCode	String	Send blank in case of fresh order request and send same value in case of modification/cancellation.
PanNumber	String	This field will get populated on server, by default send it blank.
AlgoId	Int	By default send it 0
AlgoCategory	Int	By default send it 0

Cancel Order Response:

Message Code = 15. “Order Cancel Confirm”. {event.OrderCancelResponse += OCRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 16. “Order Error”. {event.OrderErrorResponse += OrderErrorResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 21. “Order Cancel Reject”. {event.OrderCancelRejectResponse += OCRRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Trade Confirm Response:

Message code = 34. “Trade Confirm”. {event.TradeResponse += TradeResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Trade	Trade	This is a nested structure. (Refer Trade)

5. REPORTS

Order Download Request

OrionLitePy.OrderDownloadRequest (UserID, DownloadType, MemberCode)

Name	Type	Description
UserID	String	ID provided by MOSL.
DownloadType	String	“M”.
MemberCode	String	Logged in ID

Order Download Response

Message Code = 45. “Start Download”. {event.StartOrderDownloadResponse += SODRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)

Message Code = 32. “Orders”. {event.MultipleOrdersResponse += MORFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Order	Order	This is a nested structure. (Refer Order)

Message Code = 46. “End Download”. {event.EndOrderDownloadResponse += EORFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)

Trade Download Request

OrionLitePy.TradeDownloadRequest (UserID, DownloadType, MemberCode)

Name	Type	Description
UserID	String	ID provided by MOSL.
DownloadType	String	"M".
MemberCode	String	Logged in ID

Trade Download Response

Message Code = 47. "Start Download". {event.StartTradeDownloadResponse += STDRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)

Message Code = 33. "Trades". {event.MultipleTradesResponse += MTRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)
Trade	Trade	This is a nested structure. (Refer Trade)

Message Code = 48. "End Download". {event.EndTradeDownloadResponse += ETDRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)

Net Position Request

OrionLitePy.NetPositionRequest (UserID, ClientCode, ExchangeCode, ScripCode)

Name	Type	Description
UserID	String	ID Provided by MOSL.
ClientCode	String	Code of the client whose report is required.
ExchangeCode	Int	Exchange Code.
ScripCode	Int	Scrip Code.

Net Position Response

Message Code = 38. “Start Download”. {event.StartNetPositionResponse += SNPRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure.

Message Code = 39. “Net Positions”. {event.ReportNetPositionResponse += RNPRFunc}

Name	Type	Description
ScripCode	Int	
ScripName	String	
ExchangeCode	Int	
MemberCode	String	
ProductCode	Int	
BotQty	Int	
BotAmount	Float	
SoldQty	Int	
SoldAmount	Float	
DayBotQty	Int	
DayBotAmount	Float	
DaySoldQty	Int	
DaySoldAmount	Float	
DayBotOrderQty	Int	
DayBotOrderAmount	Float	
DaySoldOrderQty	Int	
DaySoldOrderAmount	Float	
LTP	Float	
MarkToMarket	Float	
BPL	Float	
ActualBotAmount	Float	
ActualSoldAmount	Float	
ActualBotQty	Int	
ActualSoldQty	Int	
ActualGainloss	Double	
RMSSellingQty	Int	

Message Code = 40. “End Download”. {event.EndNetPositionResponse += ENPRFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure.

Margin Report Request

OrionLitePy.MarginReportRequest(UserID, ClientCode, IsEquity)

Name	Type	Description
UserID	String	ID Provided by MOSL.
ClientCode	String	Code of the client whose report is required.
IsEquity	String	‘Y’ – For Equity report. ‘N’ – For commodity report.

Margin Report Response

Message Code = 36. {event.MarginResponse += MarginResponseFunc}

Name	Type	Description
MessageHeader	Message Header	Nested structure.
IsEquity	Int	‘1’ – For Equity report. ‘0’ – For commodity report.
ClientCode	String	Code of the client.
LedgerBalance	Float	Cash field
InstrumentCapital	Float	0
InitialMarginPaid	Float	0
StockCapital	Float	Non cash field
AdditionalCapital	Float	Non cash field
EquityMarginUsed	Float	0
EquityM2MLoss	Float	0
EquityBPL	Float	0
DerivativeSpanMargin	Float	MCX Span margin blocked
DerivativeExposureMargin	Float	MCX Exposure margin blocked
DerivativeM2MLoss	Float	MCX M2M Loss.
DerivativeBPL	Float	MCX booked profit/loss.
OptionSegment	Float	0
WithDrawnAmount	Float	Amount withdrawn today.
MaxWithDrawalAmt	Float	0
FundTransferAmount	Float	Cash field.
MFIPOAmount	Float	0
AvailableFundsForMFIPO	Float	0

AvailableFundsforWithDraw	Float	0
AvailableFundsforCash	Float	0
AvailableFundsforFO	Float	Available fund for MCX
AvailableFundsforCurrency	Float	Available fund for NCDEX
AdditionalCapitalFO	Float	0
AdditionalCapitalCurrency	Float	0
StockCapitalFO	Float	Non cash field
StockCapitalCurrency	Float	0
StockCapitalDelivery	Float	0
CreditLimit	Float	0
CurrencySpanMargin	Float	NCDEX Span margin blocked
CurrencyExposureMargin	Float	NCDEX Exposure margin blocked
CurrencyM2MLoss	Float	NCDEX M2M Loss.
CurrencyBPL	Float	NCDEX booked profit/loss.
UnRealisedAmount	Float	UnRealized cheque amount
MaxUnrealisedAmount	Float	Max Realised cheque Amount
MFEquityAmount	Float	0
NetOptionPremiumFO	Float	0
NetOptionPremiumCurrency	Float	0
CreditOptionAmountFO	Float	0
CreditOptionAmountCurrency	Float	0
Derivative Special Margin	Float	0
Derivative Additional Margin	Float	0
Currency Special Margin	Float	0
Currency Additional Margin	Float	0
AvailableFundsforCommodity	Float	0
StockCapitalCommodity	Float	0
AdditionalCapitalCommodity	Float	0
CommoditySpanMargin	Float	0
CommodityExposureMargin	Float	0
CommoditySpecialMargin	Float	0
CommodityAdditionalMargin	Float	0
CommodityM2MLoss	Float	0
CommodityBPL	Float	0
NetOptionPremiumCommodity	Float	0
CreditOptionAmountCommodity	Float	0
SLBMMarginUsed	Float	0
SLBMBPL	Float	0
SLBMM2MLoss	Float	0
MtfMargin	Float	0
MtfLoss	Float	0

Brokerage	Double	0

Error Response

For Other Message Codes structure is as Follows.
{event.ErrorResponse += ErrorResponseFunc}

Name	Type	Description
MessageHeader	Message Header	This is a nested structure. (Refer Header)

6. BROADCAST

Broadcast Response

- **For Subscribing Quote**

OrionLitePy.SubQuoteRequest (ExchangeCode, ScripCode)

Name	Type	Description
ExchangeCode	Int	Exchange Code.
ScripCode	String	Scrip Code

- **For Unsubscribing Quote**

OrionLitePy.UnsubQuoteRequest (ExchangeCode, ScripCode)

Name	Type	Description
ExchangeCode	Int	Exchange Code.
ScripCode	String	Scrip Code

- **For Subscribing Depth**

OrionLitePy.SubDepthRequest (ExchangeCode, ScripCode)

Name	Type	Description
ExchangeCode	Int	Exchange Code.
ScripCode	String	Scrip Code

- **For Unsubscribing Depth**

OrionLitePy.UnsubDepthRequest (ExchangeCode, ScripCode)

Name	Type	Description
ExchangeCode	Int	Exchange Code.
ScripCode	String	Scrip Code

7. HEADERS

- In every response user will get structure of headers as follows.

Header 1

Name	Type
SOF	Int
MsgLen	Int
MsgType	Int

Header 2

Name	Type
Service	Int
Exch	Int
PacketType	Int
PacketLen	Int
ScripCodeLen	Int
ScripCode	String

8. RESPONSES

*** - This field only available when value of '<u>TodayCloseInd</u>' field is 1.**

<u>Quote Responses.</u> { event.QuoteResponse += QuoteResponseFunc}
<u>Packet Type</u> = 3.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
High	Int
Low	Int
Open	Int
TimeStamp	Int
TodayCloseInd	Int
TodaysClose*	Int

<u>Packet Type</u> = 12.

Name	Type
Header1	Message Header
Header2	Message Header
OpenInterest	Int

<u>Packet Type</u> = 21.

Name	Type
Header1	Message Header
Header2	Message Header
LCL	Int
UCL	Int

Packet Type = 22.

Name	Type
Header1	Message Header
Header2	Message Header
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
TotalBuy	Int
TotalSell	Int

Packet Type = 22. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
TotalBuy	Float
TotalSell	Float

Packet Type = 23.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 27.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 33.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
Total Buy	Int
Total Sell	Int
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

<u>Packet Type</u> = 33. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
Total Buy	Float
Total Sell	Float
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

<u>Packet Type</u> = 35.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
Total Buy	Int
Total Sell	Int
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 35. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
Total Buy	Float
Total Sell	Float
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

<u>Packet Type</u> = 37.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
Total Buy	Int
Total Sell	Int
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

<u>Packet Type</u> = 37. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
Total Buy	Float
Total Sell	Float
Lower	Int
Upper	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 38.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 39.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
Total Buy	Int
Total Sell	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 39. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
Total Buy	Float
Total Sell	Float
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int

Packet Type = 41.

Name	Type
Header1	Message Header
Header2	Message Header
Total Buy	Int
Total Sell	Int

Packet Type = 41. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
Total Buy	Int
Total Sell	Int

Packet Type = 42.

Name	Type
Header1	Message Header
Header2	Message Header
LCL	Int
UCL	Int
YearHigh	Int
YearLow	Int

Packet Type = 49.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
Total Buy	Int
Total Sell	Int
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int
PrevHigh	Int
PrevLow	Int
Prevttq	Int
PrevOpen	Int

Packet Type = 49. (For **MCX** Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
TTQ	Int
High	Int
Low	Int
Open	Int
Wap	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
Total Buy	Float
Total Sell	Float
TimeStamp	Int
LTQ	Int
TodayCloseInd	Int
TodaysClose*	Int
PrevHigh	Int
PrevLow	Int
Prevttq	Int
PrevOpen	Int

Packet Type = 55.

Name	Type
Header1	Message Header
Header2	Message Header
TradedVolume	Int
TotalTradedQty	Int
TotalBuy	Int
TotalSell	Int
OpenPrice	Int
HighPrice	Int
LowPrice	Int
LTP	
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
TimeStamp	Int

Packet Type = 63.

Name	Type
Header1	Message Header
Header2	Message Header
LTP	Int
Close	Int
High	Int
Low	Int
Open	Int
TimeStamp	Int
TodayCloseInd	Int
TodaysClose*	Int
PrevDayHigh	Int
PrevDayLow	Int

Depth Response. { event.DepthResponse}

Packet Type = 4.

Name	Type
Header1	Message Header
Header2	Message Header
RecordCount	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
BuyOrders	Int
SellOrders	Int
TimeStamp	Int
TotalBuy	Int
TotalSell	Int

Packet Type = 4. (For MCX Exchange)

Name	Type
Header1	Message Header
Header2	Message Header
RecordCount	Int
BuyPrice	Int
BuyQty	Int
SellPrice	Int
SellQty	Int
BuyOrders	Int
SellOrders	Int
TimeStamp	Int
TotalBuy	Float
TotalSell	Float

-----------------------------------End of document------------------------------