Account SDK

- **AccountList GetAccountList() throw (...)**

This function returns list of accounts associated with a user.

- **CAccountBalanceResponse GetAccountBalance(CString strAccountId) throw (...)**

This API is returning the current account balance details for a specified account.

Type	Name	Default	Description
CString	strAccountId		Account number which you want to know balance.

- **CAccountPositionsResponse GetAccountPositions (CAccountPositionsRequest accountPositionsRequest) throw (...)**

This API will return positions held in the account if it is successful.

```
class CAccountPositionsRequest
{
private:
    CString m_AccountId;
    CString m_Count;
    CString m_Marker;
    CString m_Symbol;
    CString m_TypeCode;
    CString m_CallPut;
    double m_StrikePrice;
    CString m_ExpYear;
    CString m_ExpMonth;
    CString m_ExpDay;
};
```

Type	Name	Default	Description
CString	strAccountId		Account number which you want to know balance.
CString	m_Count	25	Returns the number of positions in the account. This is optional parameter.
CString	m_Marker		Additional positions in the account if there are more than 25 positions. This is optional parameter.
CString	m_Symbol		The market trading symbol for the stock being bought or sold.
CString	m_TypeCode		Type of derivative that the position reflects. There are two options for the derivative. EQ - Equity. OPTN - Option INDX - Index MF -Mutual Fund FI - Fixed Income (bond)

CString	m_CallPut		Option parameter that allows the owner the right to either buy or sell an option. Options are: CALL and PUT Note: For Type code OPTN, this is a required field.
double	m_StrikePrice		The price at which an option can be exercised. Note: For Type code OPTN, this is a required field.
CString	m_ExpYear		The year the option will expire, as specified in the option contract. For Type code OPTN, this is a required field.
CString	m_ExpMonth		The month the option will expire, as specified in the option contract. For Type code OPTN, this is a required field.
CString	m_ExpDay		The day the option will expire, as specified in the option contract. For Type code OPTN, this is a required field.

- **AlertList GetAlerts() throw (...)**

This function is returning a list of account and trading alerts.

- **CAlertDetailsResponse GetAlertDetails(int nAlertID) throw (...)**

This function is returning the details of a specific alert.

Type	Name	Default	Description
int	nAlertID		Alert id number.

- **CDeleteAlertResponse DeleteAlert(int nAlertID)throw (...)**

This function delete alter from account.

Type	Name	Default	Description
int	nAlertID		Alert id number.

Code snippets

How to call GetAccountPositions function?

```
CClientDetails clientDetails;

clientDetails.SetConsumerKey(“b011789d5cd732734bc69dedff8e263b”);
clientDetails.SetConsumerSecret(“98b373d1e5e9c411ef52d35419bb01bb”);
clientDetails.SetToken(“uiiGNIOe5EZn93dbZq0a9mGyp7QLsN%2BisYnpfrR8Vr4%3D“);
clientDetails.SetTokenSecret(“ndFgwZXFmSu3f81krleYfRUerZYiC%2FJxevp3n2R6MvA%3D“);

CAccountSDK ObjAccountSDK(clientDetails);

CAccountPositionsRequest accountPositionsRequestObj;

accountPositionsRequestObj.SetAccountId(“12345676”);
accountPositionsRequestObj.SetCount(24);

CAccountPositionsResponse accountPositions;
accountPositions = ObjAccountSDK.GetAccountPositions( accountPositionsRequestObj );
```

Market SDK

- **COptionChainResponse GetOptionChains(COptionChainRequest OptionChainRequestObj)throw (...)**

Function will return a list of option chains associated with a specific underlie if it is successful.

```
class COptionChainRequest
{
    CString m_ExpirationDay;
    CString m_ExpirationMonth;
    CString m_ExpirationYear;
    CString m_ChainType;
    CString m_SkipAdjusted;
    CString m_Underlier;
};
```

Type	Name	Default	Description
CString	m_ExpirationDay		The day the option will expire, as Specified in the option contract.
CString	m_ExpirationMonth		The month the option will expire, as Specified in the option contract.
CString	m_ExpirationYear		The year the option will expire, as Specified in the option contract.
CString	m_ChainType	CALL	The type of option chain. Options are: • CALL • PUT • CALLPUT
CString	m_SkipAdjusted	TRUE	Value that either shows or does not show adjusted options. Adjusted options are defined as options that have undergone a change and the option contract has been amended to modify this change. Options are: • TRUE • FALSE
CString	m_Underlier		The market trading symbol for the stock being bought or sold.

- **COptionExpireDateResponse GetOptionExpireDate(COptionExpireDateRequestWrappe OptionExpireDateRequestObj) throw (...)**

Function will return a return list of expiration dates for options.

```
class COptionExpireDateRequestWrappe
{
      CString m_symbol;
}
```

Type	Name	Default	Description
CString	m_symbol		The symbol of the underlying security or index for the option.

- **CProductLookupResponse GetProductLookup (CProductLookupRequest ProductLookupRequestObj) throw (...)**

Function wills returns matching stock information for a specified query string

```
class CProductLookupRequest
{
      CString m_Company;
      CString m_Type;
};
```

Type	Name	Default	Description
CString	m_Company		The name of the company.
CString	m_type		The type of security. Options are: • EQ - Equity • MF - Mutual Fund

- **CQuoteResponse GetQuote(CQuoteReqeust QuoteReqeustObj) throw (...)**

Function wills return the latest market information for a symbol

```
class CQuoteReqeust
{
      StringList m_SymbolList;
      CString m_DetailFlag ;
};
```

Type	Name	Default	Description
StringList	m_SymbolList		The market trading symbol for the stock being bought or sold. This is a list of symbols which can be any of the following: • EQ - Equity • OPTN- Option Note: The OPTN symbol has the following format (with six components separated by a colon): underlier:year:month:day: optiontype:strikePrice
CString	m_DetailFlag		Determines the market fields returned from a quote request. Fields

			returned from each option are as follows: • ALL • FUNDAMENTAL • INTRADAY • OPTIONS • WEEK_52

Code snippets

How to call GetOptionChanin function?

```
CClientDetails clientDetails;

clientDetails.SetConsumerKey(“b011789d5cd732734bc69dedff8e263b”);
clientDetails.SetConsumerSecret(“98b373d1e5e9c411ef52d35419bb01bb”);
clientDetails.SetToken(“uiiGNIOe5EZn93dbZq0a9mGyp7QLsN%2BisYnpfrR8Vr4%3D“);
clientDetails.SetTokenSecret(“ndFgwZXFmSu3f81krleYfRUerZYiC%2FJxevp3n2R6MvA%3D“);

CMarketSDK ObjMarketSDK(clientDetails);

COptionChainRequest OptionChainRequest;
COptionChainResponse OptionChainResponse;

OptionChainRequest.SetUnderlier(“GOOG”);
OptionChainRequest.SetChainType(CALL);
OptionChainRequest.SetExpirationDay(17);
OptionChainRequest.SetExpirationMonth(9);
OptionChainRequest.SetExpirationYear(2011) ;

OptionChainResponse = ObjMarketSDK.GetOptionChains(OptionChainRequest);
```

Order SDK

- **COrderListResponse GetOrderList(COrderListRequest OrderListRequestObj)**

Function wills return a list of open orders or orders updated today.

```
class COrderListRequest
{
	CString m_AccountId;
	CString m_Marker;
	CString m_Count;
};
```

Type	Name	Default	Description
CString	m_AccountId		The 8-digit account number assigned to the user.
CString	m_Marker		Parameter that designates additional orders in the account if there are more than the requested number of orders.
CString	m_Count		Parameter that returns the number of list of open orders or orders updated today. If this is not specified, the default value is 25.

- **CEquityOrderResponse GetPreviewEquityOrder(CEquityOrderRequest EquityOrderRequestObj) throw (...)**

Preview an equity order function to check for errors and view the estimated commission and total cost or proceeds.

```
class CRequestBase
{
      CString      m_AccountId;
      CString      m_ClientOrderId;
      double       m_LimitPrice;
      long long    m_PreviewId;
      double       m_StopPrice;
};

class CBasicOrderRequest : public CRequestBase
{
      CString      m_AllOrNone;
      long long    m_Quantity;
      CString      m_ReserveOrder;
      long long    m_ReserveQuantity;
};

class CEquityOrderRequest : public CBasicOrderRequest
{
      CString      m_symbol;
      CString      m_orderAction;
      CString      m_priceType;
      CString      m_routingDestination;
      CString      m_marketSession;
      CString      m_orderTerm;
};
```

Type	**Name**	**Default**	**Description**
CString	m_AccountId		The 8-digit account number assigned to the user.
CString	m_ClientOrderId		String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: required if priceType is LIMIT or STOP_LIMIT
long long	m_PreviewId		Not required.
double	m_StopPrice		The price at which a share is to be bought or sold if specified in a limit order. Note: required if priceType is STOP or STOP_LIMIT

CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE Note: This is case insensitive.
long long	m_Quantity		The number of shares (stock) specified in the equity order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is TRUE.
CString	m_symbol		The market trading symbol for the share being bought or sold.
CString	m_orderAction		User-specified action that instructs the broker what action to perform. Options are: • BUY • SELL • BUY_TO_COVER • SELL_SHORT
CString	m_priceType		The type of pricing specified in the equity order. Options are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
CString	m_routingDestination		The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO • ARCA • NSDQ • NYSE
CString	m_marketSession		Session when the equity order will be place. There are two options for this parameter: • REGULAR • EXTENDED
CString	m_orderTerm		The length of time an equity order is enforced. Options are: • GOOD_UNTIL_CANCEL

			• GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL(Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **CEquityOrderResponse GetPlaceEquityorder (CEquityOrderRequest EquityOrderRequestObj) throw (...)**

Function submits an equity order request.

```
class CRequestBase
{
      CString      m_AccountId;
      CString      m_ClientOrderId;
      double       m_LimitPrice;
      long long    m_PreviewId;
      double       m_StopPrice;
};


class CBasicOrderRequest : public CRequestBase
{
      CString      m_AllOrNone;
      long long    m_Quantity;
      CString      m_ReserveOrder;
      long long    m_ReserveQuantity;
};


class CEquityOrderRequest : public CBasicOrderRequest
{
      CString      m_symbol;
      CString      m_orderAction;
      CString      m_priceType;
      CString      m_routingDestination;
      CString      m_marketSession;
      CString      m_orderTerm;
};
```

Type	Name	Default	Description
CString	m_AccountId		The 8-digit account number assigned to the user.
CString	m_ClientOrderId		String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: required if priceType is

			LIMIT or STOP_LIMIT
long long	m_PreviewId		Numeric value associated with an order that allows users to ensure the same parameters for the preview order are input for the actual order
double	m_StopPrice		The price at which a share is to be bought or sold if specified in a limit order. Note: required if priceType is STOP or STOP_LIMIT
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE Note: This is case insensitive.
long long	m_Quantity		The number of shares (stock) specified in the equity order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is TRUE.
CString	m_symbol		The market trading symbol for the share being bought or sold.
CString	m_orderAction		User-specified action that instructs the broker what action to perform. Options are: • BUY • SELL • BUY_TO_COVER • SELL_SHORT
CString	m_priceType		The type of pricing specified in the equity order. Options are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
CString	m_routingDestination		The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO

			• ARCA • NSDQ • NYSE
CString	m_marketSession		Session when the equity order will be place. There are two options for this parameter: • REGULAR • EXTENDED
CString	m_orderTerm		The length of time an equity order is enforced. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL(Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **CChangeEquityOrderResponse GetPreviewChangeEquityOrder(CChangeEquityOrderRequest ChangeEquityOrderRequestObj) throw (...)**

Function is changed Preview equity order.

```
class CChangeOrderBase
{
      CString             m_AccountId;
      long long           m_OrderNum;
      CString             m_ClientOrderId;
      double              m_LimitPrice;
      long long           m_PreviewId;
      double              m_StopPrice;
      CString             m_AllOrNone;
      CString             m_Quantity;
      CString             m_ReserveOrder;
      long long           m_ReserveQuantity;
};


class CChangeEquityOrderRequest : public CChangeOrderBase
{
      CString      m_PriceType;
      CString      m_OrderTerm;
};
```

Type	Name	Default	Description
CString	m_AccountId		The 8-digit account number assigned to the user.
long long	m_OrderNum		Number assigned to the user for a request.
CString	m_ClientOrderId		Not required.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or

			sell the stock when it reaches a specified price. Note: required if priceType is LIMIT or STOP_LIMIT
long long	m_PreviewId		Not required.
double	m_StopPrice		The price at which a share is to be bought or sold if specified in a limit order. Note: required if priceType is STOP or STOP_LIMIT
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE Note: This is case insensitive.
CString	m_Quantity		The number of shares (stock) specified in the equity order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is true
CString	m_PriceType		The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
CString	m_OrderTerm		The length of time an equity order is enforced. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **CChangeEquityOrderResponse GetPlaceChangeEquityOrder (CChangeEquityOrderRequest ChangeEquityOrderRequestObj) throw (...)**

Function executes a change equity order request.

```
class CChangeOrderBase
{
      CString            m_AccountId;
      long long          m_OrderNum;
      CString            m_ClientOrderId;
      double             m_LimitPrice;
      long long          m_PreviewId;
      double             m_StopPrice;
      CString            m_AllOrNone;
      CString            m_Quantity;
      CString            m_ReserveOrder;
      long long          m_ReserveQuantity;
};


class CChangeEquityOrderRequest : public CChangeOrderBase
{
      CString      m_PriceType;
      CString      m_OrderTerm;
};
```

Type	Name	Default	Description
CString	m_AccountId		The 8-digit account number assigned to the user.
long long	m_OrderNum		Number assigned to the user for a request.
CString	m_ClientOrderId		Not required.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: required if priceType is LIMIT or STOP_LIMIT
long long	m_PreviewId		Not required.
double	m_StopPrice		The price at which a share is to be bought or sold if specified in a limit order. Note: required if priceType is STOP or STOP_LIMIT
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE Note: This is case insensitive.
CString	m_Quantity		The number of shares (stock) specified in the equity order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are:

			• TRUE • FALSE Note: This is case insensitive
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is true
CString	m_PriceType		The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
CString	m_OrderTerm		The length of time an equity order is enforced. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **COptionOrderResponse GetPreviewOptionOrder(COptionOrderRequest OptionOrderRequestObj) throw (...)**

Function previews an option order.

```
class CRequestBase
{
      CString      m_AccountId;
      CString      m_ClientOrderId;
      double       m_LimitPrice;
      long long    m_PreviewId;
      double       m_StopPrice;
};


class CBasicOrderRequest : public CRequestBase
{
      CString      m_AllOrNone;
      long long    m_Quantity;
      CString      m_ReserveOrder;
      long long    m_ReserveQuantity;
};


class COptionSymbol
{
      CString             m_Symbol;
```

```
       CString            m_CallOrPut;
       double             m_StrikePrice;
       long long          m_ExpirationYear;
       long long          m_ExpirationMonth;
       long long          m_ExpirationDay;
};


class COptionOrderRequest : public CBasicOrderRequest
{
       double                    m_StopLimitPrice;
       COptionSymbol      m_SymbolInfo;
       CString                   m_OrderAction;
       CString                   m_PriceType;
       CString                   m_RoutingDestination;
       CString                   m_OrderTerm;
};
```

Type	**Name**	**Default**	**Description**
CString	m_AccountId		The 8-digit account number assigned to the user.
CString	m_ClientOrderId		Not required.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: This value exists in response for LIMIT orders.
long long	m_PreviewId		Not Required.
double	m_StopPrice		The price at which a share is to be bought or sold if specified in a limit order. Note: This value exists in response for STOP orders.
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_Quantity		The number of contracts specified in the option order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of contracts in the market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_ReserveQuantity		The number of shares displayed for a reserve order.

			Note: reserveQuantity is required if reserveOrder is true
double	m_StopLimitPrice		The price for the stop and limit sides of the order. Note: This value exists in response for STOP_LIMIT orders.
CString	m_Symbol		The market trading symbol for the option underlier.
CString	m_CallOrPut		Value that allows the owner of an option the right to either buy or sell an option. Options are: • CALL • PUT
double	m_StrikePrice		The price at which an option is exercised.
long long	m_ExpirationYear		The year the option will expire, as specified in the option contract.
long long	m_ExpirationMonth		The month the option will expire, as specified in the option contract.
long long	m_ExpirationDay		The day the option will expire, as specified in the option contract.
CString	m_OrderAction		User-specified action that instructs the broker what action to perform. Options are: • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
CString	m_PriceType		The type of pricing specified in the equity order. Options are: • MARKET • STOP • LIMIT • STOP_LIMIT
CString	m_RoutingDestination		The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO • AMEX • BOX • CBOE • ISE • NOM • NYSE • PHX
CString	m_OrderTerm		The length of time an option order is enforced. Options are: • GOOD_UNTIL_CANCEL

			• GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **COptionOrderResponse GetPlaceOptionOrder(COptionOrderRequest OptionOrderRequestObj) throw (...)**

Function submits an option order request.

```
class CRequestBase
{
       CString      m_AccountId;
       CString      m_ClientOrderId;
       double       m_LimitPrice;
       long long    m_PreviewId;
       double       m_StopPrice;
};


class CBasicOrderRequest : public CRequestBase
{
       CString      m_AllOrNone;
       long long    m_Quantity;
       CString      m_ReserveOrder;
       long long    m_ReserveQuantity;
};


class COptionSymbol
{
       CString            m_Symbol;
       CString            m_CallOrPut;
       double             m_StrikePrice;
       long long          m_ExpirationYear;
       long long          m_ExpirationMonth;
       long long          m_ExpirationDay;
};


class COptionOrderRequest : public CBasicOrderRequest
{
       double                    m_StopLimitPrice;
       COptionSymbol      m_SymbolInfo;
       CString                   m_OrderAction;
       CString                   m_PriceType;
       CString                   m_RoutingDestination;
       CString                   m_OrderTerm;
};
```

Type	**Name**	**Default**	**Description**
CString	m_AccountId		The 8-digit account number assigned to the user.
CString	m_ClientOrderId		String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: This value exists in response for LIMIT orders.
long long	m_PreviewId		When using PreviewOptionOrder to obtain information on cost, commission, and other data, the preview ID is provided.
double	m_StopPrice		The price at which a share is to be bought or sold if specified in a limit order. Note: This value exists in response for STOP orders.
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_Quantity		The number of contracts specified in the option order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of contracts in the market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is true
double	m_StopLimitPrice		The price for the stop and limit sides of the order. Note: This value exists in response for STOP_LIMIT orders.
CString	m_Symbol		The market trading symbol for the option underlier.
CString	m_CallOrPut		Value that allows the owner of an option the right to either buy or sell an option. Options are:

			• CALL • PUT
double	m_StrikePrice		The price at which an option is exercised.
long long	m_ExpirationYear		The year the option will expire, as specified in the option contract.
long long	m_ExpirationMonth		The month the option will expire, as specified in the option contract.
long long	m_ExpirationDay		The day the option will expire, as specified in the option contract.
CString	m_OrderAction		User-specified action that instructs the broker what action to perform. Options are: • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
CString	m_PriceType		The type of pricing specified in the equity order. Options are: • MARKET • STOP • LIMIT • STOP_LIMIT
CString	m_RoutingDestination		The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO • AMEX • BOX • CBOE • ISE • NOM • NYSE • PHX
CString	m_OrderTerm		The length of time an option order is enforced. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **CChangeOptionOrderResponse GetPreviewChangeOptionOrder(CChangeOptionOrderRequest ChangeOptionOrderRequestObj) throw (...)**

Function changes preview option order request.

```
class CChangeOrderBase
{
      CString      m_AccountId;
      long long    m_OrderNum;
      CString      m_ClientOrderId;
      double       m_LimitPrice;
      long long    m_PreviewId;
      double       m_StopPrice;
      CString      m_AllOrNone;
      CString      m_Quantity;
      CString      m_ReserveOrder;
      long long    m_ReserveQuantity;
};

class CChangeOptionOrderRequest : public CChangeOrderBase
{
      double m_StopLimitPrice;
      CString m_PriceType;
      CString m_OrderTerm;
};
```

Type	**Name**	**Default**	**Description**
CString	m_AccountId		The 8-digit account number assigned to the user.
long long	m_OrderNum		Number assigned to the user for a request.
CString	m_ClientOrderId		String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: required if priceType is LIMIT
long long	m_PreviewId		
double	m_StopPrice		The price at which a contract is to be bought or sold if specified in a limit order. Note: required if priceType is STOP
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE
CString	m_Quantity		The number of shares (stock) specified in the equity order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of shares (reserve quantity) in the

			market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive.
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is True
double	m_StopLimitPrice		The price for the stop and limit sides of the order. Note: required if priceType is STOP_LIMIT.
CString	m_PriceType		The type of pricing specified in the option order. Options are: • MARKET • STOP • LIMIT • STOP_LIMIT
CString	m_OrderTerm		The length of time an equity order is enforced. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **CChangeOptionOrderResponse GetPlaceChangeOptionOrder(CChangeOptionOrderRequest ChangeOptionOrderRequestObj) throw (...)**

Function executes a change option order request

```
class CChangeOrderBase
{
      CString      m_AccountId;
      long long    m_OrderNum;
      CString      m_ClientOrderId;
      double       m_LimitPrice;
      long long    m_PreviewId;
      double       m_StopPrice;
      CString      m_AllOrNone;
      CString      m_Quantity;
      CString      m_ReserveOrder;
      long long    m_ReserveQuantity;
};

class CChangeOptionOrderRequest : public CChangeOrderBase
{
      double m_StopLimitPrice;
      CString m_PriceType;
```

```
	CString m_OrderTerm;
};
```

Type	Name	Default	Description
CString	m_AccountId		The 8-digit account number assigned to the user.
long long	m_OrderNum		Number assigned to the user for a request.
CString	m_ClientOrderId		String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
double	m_LimitPrice		The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price. Note: required if priceType is LIMIT
long long	m_PreviewId		When using PreviewOptionOrder to obtain information on cost, commission, and other data, the preview ID is provided.
double	m_StopPrice		The price at which a contract is to be bought or sold if specified in a limit order. Note: required if priceType is STOP
CString	m_AllOrNone		For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE
CString	m_Quantity		The number of shares (stock) specified in the equity order to be bought or sold.
CString	m_ReserveOrder		Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are: • TRUE • FALSE Note: This is case insensitive.
long long	m_ReserveQuantity		The number of shares displayed for a reserve order. Note: reserveQuantity is required if reserveOrder is True
double	m_StopLimitPrice		The price for the stop and limit sides of the order. Note: required if priceType is STOP_LIMIT.
CString	m_PriceType		The type of pricing specified in the option order. Options are:

			• MARKET • STOP • LIMIT • STOP_LIMIT
CString	m_OrderTerm		The length of time an equity order is enforced. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)

- **CCancelOrderResponse GetCancelOrder(CCancelOrderRequest CancelOrderRequestObj) throw (...)**

Function submits a cancel request for the indicated order.

```
class CChangeOrderBase
{
      CString      m_AccountId;
      long long    m_OrderNum;
      CString      m_ClientOrderId;
      double       m_LimitPrice;
      long long    m_PreviewId;
      double       m_StopPrice;
      CString      m_AllOrNone;
      CString      m_Quantity;
      CString      m_ReserveOrder;
      long long    m_ReserveQuantity;
};


class CChangeOptionOrderRequest : public CChangeOrderBase
{
      double       m_StopLimitPrice;
      CString      m_PriceType;
      CString      m_OrderTerm;
};
```

Type	**Name**	**Default**	**Description**
CString	m_AccountId		The 8-digit account number assigned to the user.
long long	m_OrderNum		Number assigned to the user for a request
CString	m_ClientOrderId		Not Required.
double	m_LimitPrice		Not Required.
long long	m_PreviewId		Not Required.
double	m_StopPrice		Not Required.
CString	m_AllOrNone		Not Required.
CString	m_Quantity		Not Required.
CString	m_ReserveOrder		Not Required.

long long	m_ReserveQuantity		Not Required.
double	m_StopLimitPrice		Not Required.
CString	m_PriceType		Not Required.
CString	m_OrderTerm		Not Required.

Code Snippets

How to call GetOrderList function?

```
CClientDetails clientDetails;

clientDetails.SetConsumerKey(“b011789d5cd732734bc69dedff8e263b”);
clientDetails.SetConsumerSecret(“98b373d1e5e9c411ef52d35419bb01bb”);
clientDetails.SetToken(“uiiGNIOe5EZn93dbZq0a9mGyp7QLsN%2BisYnpfrR8Vr4%3D“);
clientDetails.SetTokenSecret(“ndFgwZXFmSu3f81krleYfRUerZYiC%2FJxevp3n2R6MvA%3D“);

COrderSDK ObjOrderSDK(clientDetails);

COrderListRequest OrderListRequest;
COrderListResponse OrderListResponse;

OrderListRequest.SetAccountId(“12345678”);
OrderListRequest.SetCount(10);
OrderListResponse = ObjOrderSDK.GetOrderList(OrderListRequest);
```

How to call GetPreviewEquityOrder function?

```
CClientDetails clientDetails;

clientDetails.SetConsumerKey(“b011789d5cd732734bc69dedff8e263b”);
clientDetails.SetConsumerSecret(“98b373d1e5e9c411ef52d35419bb01bb”);
clientDetails.SetToken(“uiiGNIOe5EZn93dbZq0a9mGyp7QLsN%2BisYnpfrR8Vr4%3D“);
clientDetails.SetTokenSecret(“ndFgwZXFmSu3f81krleYfRUerZYiC%2FJxevp3n2R6MvA%3D“);

COrderSDK ObjOrderSDK(clientDetails);

CEquityOrderRequest EquityOrderRequest;
CEquityOrderResponse EquityOrderResponse;

EquityOrderRequest.SetAccountId("83406334");
EquityOrderRequest.SetLimitPrice(540) ;
EquityOrderRequest.SetStopPrice(540) ;
EquityOrderRequest.SetLimitPrice(540) ;

EquityOrderRequest.SetQuantity(500) ;
EquityOrderRequest.SetSymbol("GOOG");
EquityOrderRequest.SetOrderAction("BUY");
EquityOrderRequest.SetPriceType("STOP");
EquityOrderRequest.SetMarketSession("REGULAR") ;
EquityOrderRequest.SetOrderTerm("GOOD_FOR_DAY");
EquityOrderRequest.SetClientOrderId("random123456") ;

EquityOrderResponse = ObjOrderSDK.GetPreviewEquityOrder(EquityOrderRequest);
```

How to call GetCancelOrder function?

```
CClientDetails clientDetails;

clientDetails.SetConsumerKey(“b011789d5cd732734bc69dedff8e263b”);
clientDetails.SetConsumerSecret(“98b373d1e5e9c411ef52d35419bb01bb”);
clientDetails.SetToken(“uiiGNIOe5EZn93dbZq0a9mGyp7QLsN%2BisYnpfrR8Vr4%3D“);
clientDetails.SetTokenSecret(“ndFgwZXFmSu3f81krleYfRUerZYiC%2FJxevp3n2R6MvA%3D“);

COrderSDK ObjOrderSDK(clientDetails);

CCancelOrderRequest CancelOrderRequest;
CCancelOrderResponse CancelOrderResponse;

CancelOrderRequest.SetAccountId(OrderInputObj.m_EditBox1);
CancelOrderRequest.SetOrderNum(atol(OrderInputObj.m_EditBox2)) ;

CancelOrderResponse = ObjOrderSDK.GetCancelOrder(CancelOrderRequest);
```

Common Exception for all SDK functions.

```
class CExceptionSDK
{
	int m_nHTTPErrorCode;
	int m_nErrorCode;
	CString m_strErrorMsg;
};

class CError
{
	int m_nHTTPErrorCode;
	int m_nErrorCode;
	CString m_strErrorMsg;
};
```

Type	Name	Default	Description
Int	m_nHTTPErrorCode		Http error response code.
Int	m_nErrorCode		SDK Error code. ERROR_ENVIRONMENT_EMPTY 1001 ERROR_COMSUMER_KEY_EMPTY 1002 ERROR_COMSUMER_SECRET_EMPTY 1003 ERROR_REQUEST_TOKEN_EMPTY 1004 ERROR_REQUEST_TOKEN_SECRET_EMPTY 1005 ERROR_ACCESS_TOKEN_EMPTY 1006 ERROR_ACCESS_TOKEN_SECRET_EMPTY 1007 ERROR_VERIFIER_EMPTY 1008 ERROR_URL_EMPTY 1009 ERROR_POST_PARAM_EMPTY 1010 ERROR_ACCOUNT_ID_EMPTY 2000 ERROR_NO_OF_POSITION_EMPTY 2001 ERROR_ALERT_ID__EMPTY 2002 ERROR_XML_PARSING 5000 ERROR_SYMBOL_EMPTY 3000 ERROR_COMPANY_NAME_EMPTY 3001 ERROR_TYPE_EMPTY 3002 ERROR_SYSMBOL_KEY_EMPTY 3003

			ERROR_MISSING_REQUIRED_FIELD 5000
CString	m_strErrorMsg		Error message.