E*TRADE FINANCIAL

API Reference Guide

Release 1.3

April 2011

Table of Contents

Chapter 5: Push Notifications

Glossary

Appendix A: Error Messages

Chapter 1: Conceptual Overview

Revision History

Table 1-1 API Reference Guide Version History

Version	Date
1.0	April 1, 2010
1.1	October 10, 2010
1.2	February 1, 2011
1.3	March 31, 2011

Introduction

The E*TRADE API Platform allows customers and developers to leverage E*TRADE's extensive market data offerings, order routing capabilities, and other services to enable users to develop their own platforms, as well as provide third-party developers the capability to create applications that direct trades to E*TRADE.

This API implementation allows E*TRADE customers currently using a third-party trading platform to send trade orders to E*TRADE directly from the platform, as well as develop their own trading solutions to send and retrieve account information.

Using the E*TRADE API, a client application can:

- authenticate E*TRADE customers to the E*TRADE authentication servers.
- manage trades and orders, including placing a new order, modifying an existing order, cancelling an order, or retrieving the status of an order.

API Agreement

A developer or customer can access the E*TRADE API provided that they have accepted the terms on the API Agreement. This agreement is a legally-binding contract between E*TRADE and a user specifying the terms and conditions for using the E*TRADE API. If a user or developer attempts to log into their account from a third-party application that has not been validated and verified, and the API Agreement not accepted, they are redirected to the API Agreement page before being allowed to continue logging into their account.

If a developer wants to create a custom platform that can be distributed to users, the following conditions must be met:

- the user must be verified.
- the platform must undergo a review.
- when the review is completed, if the platform is approved, a Consumer Key is issued.

  Note: If a user wants to create their own trading platform which will not be distributed, the user goes through the typical authentication process but is not required to have the platform undergo a review by E*TRADE. A Consumer Key is issued for that specific platform, therefore, the platform and its Consumer Key can't be distributed to other users because the Consumer Key is tied to the UserID.

Chapter 2: E*TRADE API Authentication

Introduction

The E*TRADE API platform uses the OAuth open authentication protocol to manage user access and authentication, as well as the protection of user credentials. Before discussing how user authentication and login is managed, it is important to understand the actors used in the OAuth workflow that are responsible to not only granting user access to the platform, but also provide third-party vendor access to user information without exposing sensitive user credentials and information.

For more information on the OAuth protocol and specification, go to the following website:

http://oauth.net/core/1.0a/

The three actors used in OAuth are:

- **Consumer** - a website or application that uses OAuth to access the Service Provider on behalf of a user.
- **User** - an individual who has an account with the Service Provider.
- **Service Provider** - a service provider that allows access via OAuth.

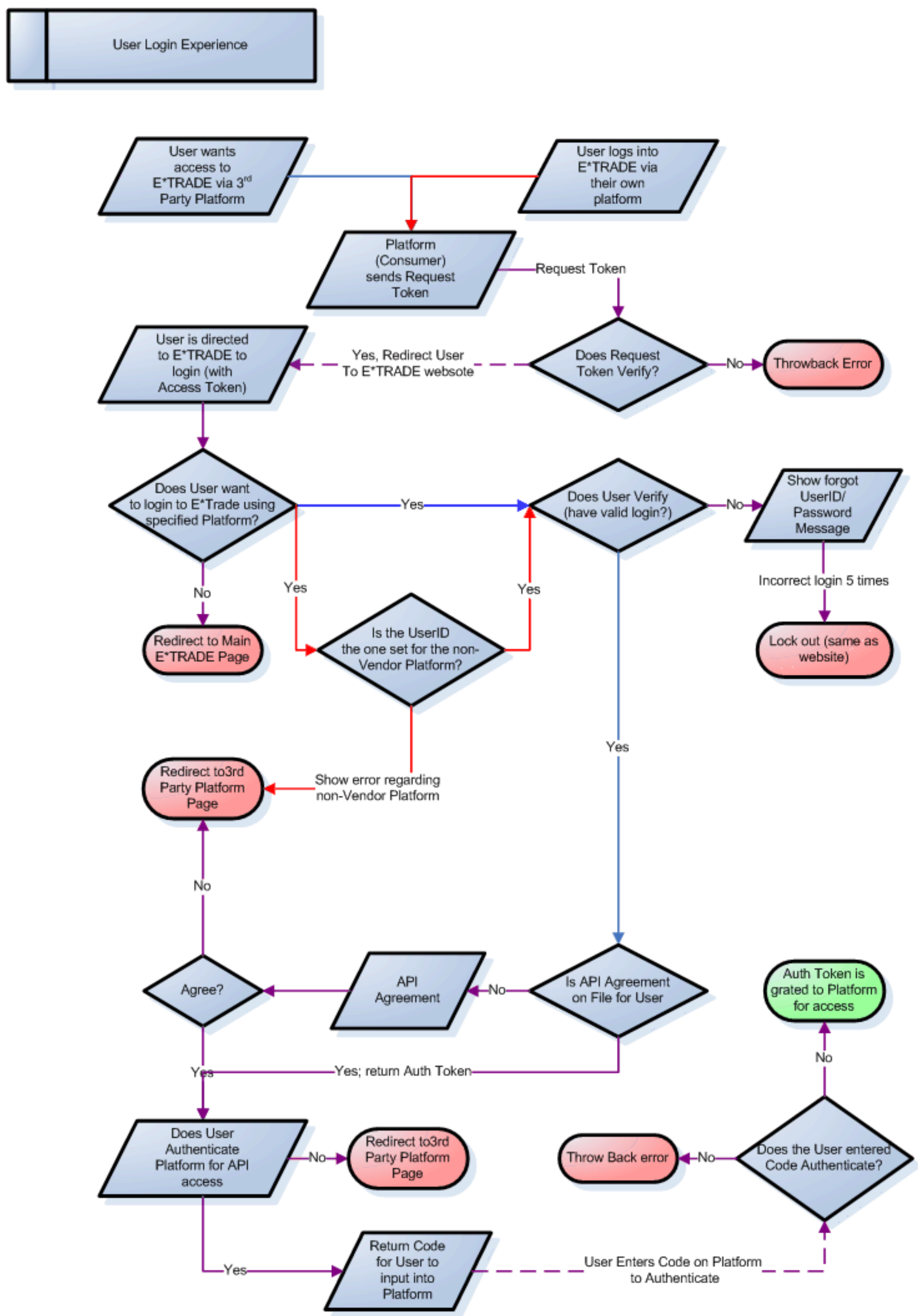
OAuth was selected as the authentication protocol for the E*TRADE API Platform because third-party sites/applications can process transactions on behalf of an E*TRADE customer without sharing user credentials.

Authentication is split into three distinct steps:

1. The third-party website/application is authenticated with E*TRADE.
2. The E*TRADE customer is authenticated with E*TRADE.
3. The customer grants authorization to third-party website/application.

Figure 2-1 illustrates the OAuth business logic workflow for user authentication.

Figure 2-1 OAuth Authentication Login Workflow



OAuth Tokens

Request Token

URL

https://etws.etrade.com/oauth/request_token

Parameters

Table 2-1 Request Token Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp with each request.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_callback	oob (out of band).

Authorizing the consumer application

After getting the request token, the consumer application must then redirect the user to the E*TRADE authorization page where the user can authorize the application. After the consumer application has been authorized, the user is given a **verification code**, which the user then enters to allow the consumer application to request an access token.

Note: The Request token is only valid for 5 minutes.

The URL to the authorization page is:

https://us.etrade.com/e/etws/authorize?key=<consumer_key>&token=<request_token>

Access Token

URL

https://etws.etrade.com/oauth/access_token

Parameters

Table 2-2 Access Token Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp with each request.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_token	The Consumer's Request Token to be exchanged for an Access Token.
oauth_verifier	The code received by the user to authenticate with the third-party application.

Renew Access Token

URL

https://etws.etrade.com/oauth/renew_access_token

Parameters

Table 2-3 Renew Access Token Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp with each request.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_token	The Consumer's Request Token to be exchanged for an Access Token.

Revoke Access Token

URL

https://etws.etrade.com/oauth/revoke_access_token

Parameters

Table 2-4 Revoke Access Token Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp with each request.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_token	The Consumer's Access Token.

Request Valid Consumer Key

To request a consumer key for development, please send a secure message (https://us.etrade.com/e/t/accounts/sccreatemsg) (**Subject**: *Technical Issues*, **Topic**: *E*TRADE API*) with a short description of what you are looking to do. You will be directed to review an API Agreement and a Consumer Key will be generated that will allow only your userID to be used with the developed platform. If you are looking to allow other users to connect to E*TRADE using the developed platform, an additional request will need to be made to allow for multiple users with the supplied Consumer Key.

Key Expiration

Key expiration is an important part of the registration approval process and is used to set a specific amount of time that a consumer key will be active and valid for a user. Typically, a key is valid for up to one year. The following steps are taken when setting a valid expiration date for the key:

1. The Consumer submits a request to gain access to the consumer keys.
2. This request is then submitted for approval.
3. Once the key is approved, the consumer key is set to Active and an expiration date is specified.

Authentication Process

The OAuth authentication process consists of a number of actions undertaken by the Consumer, User, and Service Provider. In the Authentication Process, please note that other than the consumer key, the authentication process is the same for both production and sandbox environments; however depending on the key used, only production or sandbox resources can be accessed.

This business logic and flow is described below.

The Consumer requests a token from the Service Provider.

The Consumer obtains an unauthorized Request Token by asking the Service Provider to issue a token. This Request Token is used to receive a user's approval and is then used later to obtain an Access Token.

The Consumer then sends an HTTP request to the Service Provider, requesting access to a user's resources. The token URL for this request is *https://etws.etrade.com/oauth/request_token*.

This request must adhere to the Service Provider's specifications and needs to include the following parameters listed in Table 2-5.

Table 2-5 Request Token HTTP Request Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp with each request.
oauth_version	Optional parameter used to identify the OAuth version being used.
oauth_callback	oob (out of band).

The Service Provider issues a Token and Token Secret to the Consumer

Once the Service Provider has verified the signature and consumer key, a Request Token and Token Secret is issued to the Consumer. This token is unable to be exchanged for an Access Token until the user has granted access to the Consumer. Table 2-6 lists the parameters included in the Service Provider's response to the Consumer request.

Table 2-6 Service Provider Return Values

Parameter	Description
oauth_token	The Request Token issued to the Consumer.
oauth_token_secret	The Token Secret issued to the Consumer.

The Consumer directs the User to the Service Provider

In order for the Consumer to exchange the Request Token for an Access Token, the User must first grant access to the Consumer.

The URL that the user is directed to once they have the token and key is:

https://us.etrade.com/e/t/etws/authorize?key=<consumer_key>&token=<request_token

To accomplish this, the User is directed to the Service Provider, along with an HTTP request with the following parameters listed in Table 2-7.

Table 2-7 Service Provider Request Parameters

Parameter	Description
oauth_token	The Request Token issued from the Service Provider.
consumer_key	The value used by the Consumer to identify itself to the Service Provider.

The Service Provider authenticates the User and Obtains Approval

The Service Provider must verify the user's identity before authorizing a Request Token from the Consumer. The Service Provider performs the following actions:

1. Verify the user's identity through user login or other authentication process as shown in Figure 2-2.

   Note: Make sure to enter the verification code received from a successful login back to the originating platform.

Figure 2-2 E*TRADE Login Page

2. Provide information to the User about the Consumer requesting access, including the resources that will be provided to the Consumer.

3. The User must then choose to either grant or deny access to the Consumer in the Grant Access page (Figure 2-3). If the User grants access to the Consumer, the Service Provider will grant access to the Consumer. If the User denies access, the Service Provider MUST NOT make modules available to the Consumer.

   Note: Make sure to enter the verification code received from a successful login back to the originating platform.

Figure 2-3 E*TRADE Grant Access Page

4. A verification code is provided to the user in order to authenticate the Consumer to the Service Provider.

The User enters the verification code at the Consumer application

After the E*TRADEserver has authorized the user, the user must then copy the access code generated by the system and then enter this unique verification code in order to access a user's resources.

> **Note:** Make sure to enter the verification code received from a successful login back to the originating platform.

Figure 2-4 Authorization Complete Page

The Consumer requests an Access Token

The Request Token and Token Secret must now be exchanged for an Access Token and Token Secret.

The relative url for the token activation is:

https://etws.etrade.com/oauth/access_token

To request this Access Token, an HTTP request must be sent to the Service Provider's Access Token URL (*https://etws.etrade.com/oauth/access_token*) and consist of the following parameters listed in Table 2-8.

Table 2-8 Access Token Request Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_token	The Consumer's Request Token to be exchanged for an Access Token.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp for each request.
oauth_version	Optional parameter used to identify the OAuth version being used.
oauth_verifier	The code received by the user to authenticate with the third-party application.
oauth_callback	oob (out of band)

The Service Provider grants an Access Token

The Service Provider then ensures that the request signature has been verified, the Request Token has never been exchanged for an Access Token before, and the Request Token matches the Consumer Key, and the verification code received by the user must match the Request Token.

If the Service Provider is able to verify these conditions, an Access Token and Token Secret can then be issued to the Consumer with the following parameters listed in Table 2-9.

Table 2-9 Access Token Parameters

Parameters	Description
oauth_token	The Access Token issued by the Service Provider for use by the Consumer.
oauth_token_secret	The Token Secret issued with the Access Token by the Service Provider for use by the Consumer.

The Consumer is granted access to the User's resources

After receiving the Access Token and Token Secret, the Consumer is then able to access resources on the User's behalf. To access these resources, a request must be made to the Service Provider with the following parameters listed in Table 2-10.

Table 2-10 User Resource Request Parameters

Parameter	Description
oauth_consumer_key	The value used by the Consumer to identify itself to the Service Provider.
oauth_token	The Access Token issued by the Service Provider.
oauth_signature_method	The signature method used by the Consumer to sign the request. The options are: • HMAC-SHA1
oauth_signature	Signing the request with shared secret and token secret using the specified oauth_signature_method.
oauth_timestamp	The date/time value used to designate the date and time a request was made.
oauth_nonce	A randomly-generated unique value included with the timestamp with each request.
oauth_version	Optional parameter used to identify the OAuth version being used.

E*TRADE Authentication User Scenario

Third-party Application Authentication

Prior to a user logging into E*TRADE, the consumer application must be verified and authenticated as valid. The Consumer will send an HTTP request for a Request Token. If the Consumer Key sent by the application indicates the application is invalid or has expired, the following error message is displayed:

This application is not authorized to work with the E*TRADE API, you may be using an invalid or expired version of the software.

User Authentication

Before a user can connect to the E*TRADE API, they must first login to their E*TRADE account and authorize the Consumer application. The following process is used to verify a user's access to the API:

1. The user is directed to the **Login** page on the E*TRADE website so they can log into their account.

2. Once authenticated, an **API Agreement** page (Figure 2-5) is displayed if the customer has not already agreed to abide by the terms and conditions of the agreement. If the customer has already agreed to the terms of the API Agreement, then the customer is directed to a page to verify the application being used is valid.

Figure 2-5 E*TRADE API Agreement Page

3. On the **Grant Access** page (see Figure 2-3), the user is shown the application name and the modules the application will be able to access (for example, submit and review orders, retrieve account information, retrieve market data, etc.).

   a. By clicking on the **I Agree** button, the user acknowledges the application and grants the application access to user resources. The user is then sent to complete the login process.

   b. If the user selects **Close**, the login process is cancelled.

4. In order to complete authorization, the E*TRADE server generates an verification code that the user must enter into the original third-party application to exchange the Request Token for an Access Token to establish the connection between the application and E*TRADE.

Chapter 3: Rate Limiting

Introduction

The E*TRADE API Platform incorporates rate limiting technologies to limit the number of incoming requests. The need to implement these processes is based on both traffic management, as well as security. Without limiting the number of requests over the E*TRADE network, the network is not only vulnerable to Denial of Service (DOS) attacks from malicious users and hackers, but can also cause network to become congested due to too many requests being processed.

The process of rate limiting (also known as “throttling”) is a control mechanism used to manage the number of requests accepted. An Administrator sets a rate limit for the network (order, market, account) modules, and then after this threshold has been specified, requests within the limit are accepted. If the incoming request rate exceeds the specified data (request) rate set by the Administrator, the excess requests are dropped.

The way that rate limiting is used on the E*TRADE network is:

- **Policing** - the process by which excess requests are discarded.

Rate Limits

E*TRADE has implemented specific limits on the number of incoming requests per module per second, per user for every Consumer. The following limits have been set for the platform:

- **Order Services**- 2 incoming requests per second per user
- **Account Services**- 2 incoming requests per second per user
- **Market Data Services**- 4 incoming requests per second per user
- **System Notifications** - 2 incoming system notification requests per second are processed

Note: If the number (rate) of incoming requests for a module exceed the rate limit, these (the excess) requests are dropped and the following message is displayed: **"Rate limit exceeded, please wait."**

Note: Please note additional processing may impact the above thresholds.

Sample Request

```
statuses/rest/
limits?oauth_consumer_key=45e56f05d7fa1021e2c81d3c7a1d2e06Z7YVANJzqlp00SZR8HXqhEL1r
gDORhsYet5UR1C1tlk%3D&module=accounts
```

Sample Response

```
<RateLimitStatus>
<consumerKey>45e56f05d7fa1021e2c81d3c7a1d2e06</consumerKey>
<limitIntervalInSeconds>3600</limitIntervalInSeconds>
<requestsLimit>7000</requestsLimit>
<requestsRemaining>7000</requestsRemaining>
<resetTime>03-20-2010 01:37:32 GMT</resetTime>
<resetTimeEpochSeconds>1269049052</resetTimeEpochSeconds>
<userId>26084935</userid>
</rateLimitStatus>
```

RateLimitStatus REST service

As mentioned above, E*TRADE has set limits on the number of valid incoming requests that can be received per module per second. If you want to view the number of requests that can be sent to the service before the current interval expires, there is a specific Web Service that can be used to retrieve these details.

The E*TRADE RateLimit Status REST web service can be accessed at the following URI:

statuses/rest/limits

Parameters

When accessing this URI to retrieve rate limit details, the parameters listed in Table 3-1 must be entered.

Table 3-1 RateLimitStatus Parameters

Parameter	Data Type	Required/Optional	Description
oauth_consumer_key	integer	Required	The value used by the Consumer to identify itself to the Service Provider.
oauth_token	integer	Required	The Request Token issued by the Service Provider.
module	string	Required	The grouping of services that the rate limit applies to. Options are: • accounts • market • order • notifications

XML Output

The following elements listed in Table 3-2 are returned with the XML response:

Table 3-2 XML Output Elements

Element	Data Type	Description
oauth_consumer_key	string	The consumer key.
limitIntervalInSeconds	integer	The time in seconds interval for which these limits apply.
requestLimit	integer	The number of requests that will be allowed in the specified time interval.
requestsRemaining	integer	The number of request that the user will be able to send in the specified time interval.
resetTime	integer	The time when the interval will be reset.
resetTimeEpochSeconds	integer	The time when the interval will be reset in epoch seconds.

Sample Response


```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<RateLimitStatus>
<consumerKey>d85df5b910e3cb47b2a4cf26c20229ae</consumerKey>
<limitIntervalInSeconds>3600</limitIntervalInSeconds>
<requestsLimit>6500</requestsLimit>
<requestsRemaining>6500</requestsRemaining>
<resetTime>02-25-2010 22:16:58 GMT</resetTime>
<resetTimeEpochSeconds>1267136218</resetTimeEpochSeconds>
</RateLimitStatus>
```

Chapter 4: API Resources- REST

Introduction

The E*TRADE API consists of a number of REST resources that third-party developers and E*TRADE users can use to interact with E*TRADE. This chapter includes specific information about each resource, but also provides sample requests and responses so users can understand the business logic of each resource.

Each resource consists of the following information:

- **Description** - a brief description of the resource.
- **URL and Sandbox URL** - URLs showing where the user can go to test their applications prior to implementing them into Production.
- **Parameters (Request & Response)** - tables showing the name, data type, whether the parameter is required or option, and a brief description of the parameter with any options needed for the parameter.
- **Sample Request** - a sample request displaying the parameters passed by the user.
- **Sample Response** - a sample response displaying the values returned to the user. This also includes JSON responses.
- **Notes** - lists any additional important information about the resource.
- **See Also** - list of related REST resources.

REST Resources

The following sections make up this chapter:

- "Accounts."
- "Markets."
- "Order Services."
- "Notifications."

Accounts

The Accounts module of the E*TRADE API allows users to review specific account information for an account, such as balances, and current positions.

The following resources make up the Accounts section of this chapter:

- `accountlist`
- `accountbalance`
- `accountpositions`
- `alerts`
- `alertdetails`
- `deletealert`

accountlist

Description - return a list of accounts associated with a user.

URL - *https://etws.etrade.com/accounts/rest/accountlist*

Sandbox URL - *https://etwssandbox.etrade.com/accounts/sandbox/rest/accountlist*

HTTP Method - GET

Sample Request

This sample request shows the REST-based syntax required to retrieve a list of accounts for a specific account ID.

```
/accounts/rest/accountlist
```

Sample Response

When a user requests a list of accounts associated with a user ID, the response values returned to the user show account information, including the type of account, the account ID, margin level, the net value of the account, and the type of registration for the account.


```
<AccountListResponse>
   <Account>
      <accountDesc>INDIVIDUAL</accountDesc>
      <accountId>83353535</accountId>
      <marginLevel>MARGIN</marginLevel>
      <netAccountValue>-663.20</netAccountValue>
      <registrationType>INDIVIDUAL</registrationType>
   </Account>
</AccountListResponse>
```


Sample JSON Response


```
{"json.accountListResponse":{"response":{"accountDesc":"INDIVIDUAL","accountId":833
10056,"marginLevel":"MARGIN","netAccountValue":"109999998764.17","registrationType"
:"INDIVIDUAL"}}}
```

Parameters

Request Parameters - *None*

Table 4-1 accountlist Return Values

Parameter	Data Type	Description
accountDesc	string	A text description of the user account.
accountID	string	The 8-digit account number assigned to the user.
marginLevel	enum	The level of the margin account. Options are • CASH • MARGIN
netAccountValue	double	The total value of the account (cash + stocks, options, mutual funds, bonds, etc).
registrationType	string	The type of user account. Some examples of user account types are • INDIVIDUAL • JOINT • CORPORATE • IRA • ESTATE • TRUST

Notes

See Also

`accountbalance`, `accountpositions`

accountbalance

Description - return the current account balance details for a specified account.

URL - *https://etws.etrade.com/accounts/rest/accountbalance/{AccountId}*

Sandbox URL - *https://etwssandbox.etrade.com/accounts/sandbox/rest/accountbalance/{AccountId}*

HTTP Method - GET (make global change throughout document)

Sample Request

This sample request shows the REST-based syntax required to retrieve the account balance for a specific account ID.

```
/accounts/rest/accountbalance/83353535
```

Sample Response

When a user requests the account balance for an account associated with a user ID, the response values returned to the user show account information, including the account ID, type, option level, account balance, margin account balance, any DT non-marginable securities and options for the account, the DT balance, and any marginable securities and options for the account.


```
<AccountBalanceResponse>
   <accountId>83353535</accountId>
   <accountType>MARGIN</accountType>
   <optionLevel>LEVEL_4</optionLevel>
   <accountBalance>
      <cashAvailableForWithdrawal>0.00</cashAvailableForWithdrawal>
      <netAccountValue>-663.20</netAccountValue>
      <netCash>-740.450013</netCash>
      <sweepDepositAmount>0.00</sweepDepositAmount>
      <totalLongValue>122.23</totalLongValue>
      <totalSecuritiesMktValue>-454.22</totalSecuritiesMktValue>
   </accountBalance>
   <marginAccountBalance>
      <dtBalance>
         <dtCashBalance>0.00</dtCashBalance>
         <dtMarginBalance>0.00</dtMarginBalance>
         <dtMarginableSecurities>-4183.73</dtMarginableSecurities>
         <dtNonMarginableSecuritiesAndOptions>-2714.88</
dtNonMarginableSecuritiesAndOptions>
```

```
         <dtStatus>QUALIFIED_4_X</dtStatus>
</dtBalance>
      <fedCall>-105.49</fedCall>
      <marginBalance>-740.45</marginBalance>
      <marginBalanceWithdrawal>-2714.88</marginBalanceWithdrawal>
      <marginEquity>-714.88</marginEquity>
      <marginableSecurities>-4183.73</marginableSecurities>
      <maxAvailableForWithdrawal>-2714.88</maxAvailableForWithdrawal>
      <minEquityCall>-25663.20</minEquityCall>
      <nonMarginableSecuritiesAndOptions>-2714.88</
nonMarginableSecuritiesAndOptions>
      <totalShortValue>-576.45</totalShortValue>
      <shortReserve>531.47</shortReserve>
   </marginAccountBalance>
</AccountBalanceResponse>
```


Sample JSON Response


```
{"json.accountBalanceResponse":{"accountId":83310056,"accountType":"MARGIN","option
Level":"LEVEL_4","accountBalance":{"cashAvailableForWithdrawal":"109999998743.24","
fundsWithheldFromWithdrawal":"0.00","netAccountValue":"109999998764.39","netCash":"
109999998743.24","sweepDepositAmount":"99999999999.00","totalLongValue":21.15,"tota
lSecuritiesMktValue":21.15},"marginAccountBalance":{"marginBalance":"9999998744.24"
,"marginBalanceWithdrawal":"109999998743.24","marginEquity":"109999998764.39","marg
inEquityPct":"100.00","marginableSecurities":"219999997516.47","maxAvailableForWith
drawal":"109999998743.24","nonMarginableSecuritiesAndOptions":"109999998743.24","sh
ortReserve":"0.00"}}
```

Parameters

Table 4-2 accountbalance Request Parameters

Parameter	Data Type	Required/ Optional	Description
accountId	string	Required	The 8-digit account number assigned to the user.

Table 4-3 accountbalance Return Values

Parameter	Data Type	Description
accountID	integer	The 8-digit account number assigned to the user.
accountType	enum	The type of user account. Options are: • CASH • MARGIN
optionLevel	enum	The option level of the account. Levels are from 1-4 • LEVEL 1 - write covered calls • LEVEL 2 - write covered calls, purchases • LEVEL 3 - write covered calls, purchases, spreads, uncovered puts, uncovered equity/index calls, and uncovered index puts (this requires a margin account) • LEVEL 4 - writes covered calls, purchases, spreads, uncovered puts, uncovered equity/index calls, and uncovered index puts (this requires a margin account).
accountBalance	custom	The amount of money in the account.
The following values are returned to the user:		
cashAvailableFor Withdrawal	double	The amount of cash available for withdrawal.
cashCall	double	The amount of cash or equivalent needed to meet the cash call.
fundsWithheldFromPurchaseP ower	double	The total amount of funds withheld from the purchasing power.
fundsWithheldFromWithdraw al	double	The total amount of funds withheld from withdrawal.
netAccountValue	double	The account value minus any margin debt.
netCash	double	The net cash amount held in the account.
sweepDepositAmount	double	The amount held in the sweep deposit account.
totalLongValue	double	The total value of securities that are held long in the account.
totalSecuritiesMktValue	double	The total value of the securities held in the account.

Parameter	Data Type	Description
marginAccountBalance	custom*	The total amount of buying power a user has.
The following values are returned to the user:		
fedCall	double	The amount of cash or equivalent needed to meet the cash call.
marginBalance	double	The margin buying power for the account.
marginBalanceWithdrawal	double	The amount you can withdraw using the margin balance within the account.
marginEquity	double	The amount of equity the margin balance is based on.
marginEquityPct	double	The ratio of margin debt to the total value of the equity in the account.
marginableSecurities	double	The amount of marginable securities held in the account.
maxAvailableForWithdrawal	double	The total amount (cash + margin) available for withdrawal from the account.
minEquityCall	double	The amount of equity that is needed to deposit in order to meet the minimum amount of equity required for the account.
nonMarginableSecuritiesAndOptions	double	The securities and options that are not marginable.
totalShortValue	double	The total value of securities that are held short in the account.
shortReserve	double	The amount reserved to cover the short positions in the account.
dtBalance	dtBalance*	The daytrading balance in the account.
The following values are returned for dtBalance:		
dtCashBalance	double	The cash available for investment in a day trading account.
dtMarginBalance	double	The margin available for investment in a day trading account.
dtMarginableSecurities	double	The amount of marginable securities held in a day trading account.
dtNonMarginableSecuritiesAndOptions	double	The amount of non-marginable securities and options held in a day trading account.

Parameter	Data Type	Description
dtStatus	enum	The current status of a day trading account. Options are: • NOT_QUALIFIED • QUALIFIED_4X • MARGIN_CALL_2XD • MARGIN_CALL_1XD • MIN_EQUITY_CALL_1XK • QUALIFIED_RESTRICTION • CASH_ONLY
cashAccountBalance	custom*	The cash balance in the account.
The following values are returned to the user:		
cashAvailableForInvestment	double	The total amount of cash available for investment in the account.
cashBalance	double	The amount of cash in the account.
settledCashForInvestment	double	The amount of settled cash in the account.
unSettledCashForInvestment	double	The amount of unsettled cash in the account.

Notes

See Also

`accountlist`, `accountpositions`

accountpositions

Description - Return positions held in the account.

URL - *https://etws.etrade.com/accounts/rest/accountpositions/{AccountId}*

Sandbox URL - *https://etwssandbox.etrade.com/accounts/sandbox/rest/accountpositions/{AccountId}*

HTTP Method - GET

Sample Request

When a user requests a list of account positions associated with a user ID, the response values returned to the user show account information, including the account ID, count, information about each account position for the account.

```
/account/rest/accountpositions/83353535
```

Sample Response


```
<AccountPositionsResponse>
   <accountId>83353535</accountId>
   <count>4</count>
   <marker/>
   <AccountPositions>
      <AccountPosition>
         <costBasis>3.96</costBasis>
         <description>CITIGROUP INC COM</description>
         <longOrShort>LONG</longOrShort>
         <marginable>true</marginable>
         <productId>
            <symbol>C</symbol>
            <typeCode>EQ</typeCode>
         </productId>
         <qty>10</qty>
         <marketValue>39.60</marketValue>
      </AccountPosition>
      <AccountPosition>
         <costBasis>25.87</costBasis>
         <description>CISCO SYS INC COM</description>
         <longOrShort>LONG</longOrShort>
         <marginable>true</marginable>
         <productId>
            <symbol>CSCO</symbol>
            <typeCode>EQ</typeCode>
         </productId>
         <qty>3</qty>
         <marketValue>77.62</marketValue>
      </AccountPosition>
      <AccountPosition>
         <costBasis>1.67</costBasis>
         <description>E TRADE FINANCIAL CORP COM</description>
         <longOrShort>LONG</longOrShort>
         <marginable>true</marginable>
         <productId>
            <symbol>ETFC</symbol>
            <typeCode>EQ</typeCode>
         </productId>
         <qty>3</qty>
         <marketValue>5.01</marketValue>
      </AccountPosition>
      <AccountPosition>
         <costBasis>576.45</costBasis>
```

```
         <description>GOOGLE INC CL A</description>
         <longOrShort>SHORT</longOrShort>
         <marginable>false</marginable>
         <productId>
            <symbol>GOOG</symbol>
<typeCode>EQ</typeCode>
         </productId>
         <qty>-1</qty>
         <marketValue>-576.45</marketValue>
      </AccountPosition>
   </AccountPositions>
</AccountPositionsResponse>
```


Sample JSON Response


```
{"json.accountPositionsResponse":{"accountId":83310056,"count":1,"marker":"","respo
nse":{"costBasis":0,"description":"CISCO SYS INC
COM","longOrShort":"LONG","marginable":true,"productId":{"symbol":"CSCO","typeCode"
:"EQ"},"qty":1,"marketValue":21.15}}}
```

Parameters

Table 4-4 accountpositions Request Parameters

Parameter	Data Type	Required/ Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
count	integer (1-25)	Optional	Parameter that returns the number of positions in the account. Default value is 25.
marker	string	Optional	Parameter that designates additional positions in the account if there are more than 25 position (26+).
symbol	string	Optional	The market trading symbol for the stock being bought or sold.
typeCode	string	Optional	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 Investment (bond)
callPut	enum	Optional	Option parameter that allows the owner the right to either buy or sell an option. Options are: • CALL • PUT **Note**: For Type code OPTN, this is a required field.
strikePrice	double	Optional	The price at which an option can be exercised. **Note**: For Type code OPTN, this is a required field.
expYear	integer (4-digits)	Optional	The year the option will expire, as specified in the option contract. **Note**: For Type code OPTN, this is a required field.

Table 4-4 accountpositions Request Parameters

Parameter	Data Type	Required/ Optional	Description
expMonth	integer (1-12)	Optional	The month the option will expire, as specified in the option contract. **Note**: For Type code OPTN, this is a required field.
expDay	integer (1-31)	Optional	The day the option will expire, as specified in the option contract. **Note**: For Type code OPTN, this is a required field.

Table 4-5 accountpositions Return Values

Parameter	Data Type	Description
accountId	8-digit integer	The 8-digit account number assigned to the user.
count	integer (1-25)	Parameter that returns the number of positions in the account.
marker	string	Parameter that designates additional positions in the account if there are more than 25 position (26+).
costBasis	double	The total cost you paid for this position.
description	string	Text description of the position (i.e. company name)
longOrShort	enum	Options are: • LONG • SHORT
marginable	boolean	Value that specifies whether the position will count toward your margin.
quantity	double	The number of account positions for the account.
marketValue	double	The value of a position based on your quantity and previous close price.
productId	custom	Returns fields that describe the position. Return values can include:
The following fields are returned for productId		
symbol	string	The market trading symbol for the option.
typeCode	enum	The type of derivative. Options are: • OPTN
callPut	enum	Specifies whether options are being bought (call) or sold (put), as specified in the option order. Options are: • CALL • PUT
strikePrice	double	The strike price for the option, as specified in the option contract.
expYear	4-digit integer	The year the option will expire, as specified in the option contract.
expMonth	integer (1-12)	The month the option will expire, as specified in the option contract.
expDay	integer (1-31)	The day of the month the option will expire, as specified in the option contract.

Notes

- If the typeCode selected is OPTN, then *callPut*, *strikePrice*, *expYear*,*expMonth,* and *expDay* are mandatory.

See Also

`accountlist`, `accountbalance`

alerts

Description - return a list of account and trading alerts.

URL - *https://etws.etrade.com/accounts/rest/alerts*

Sandbox URL - *https://etwssandbox.etrade.com/accounts/sandbox/rest/alerts*

HTTP Method - GET

Sample Request

This sample request shows the syntax required to return a list of alerts associated with an account.

```
/accounts/rest/alerts
```

Sample Response

When a user requests a list of alerts associated with an account, the response values returned to the user show information about the alert, including date and time for the alert, the ID associated with the alert, a subject line describing the alert, and a read flag which states whether the flag is read or unread.


```
<AlertsResponse>
   <Alerts>
      <Alert>
         <dateTime>1266968868</dateTime>
         <alertId>332</alertId>
         <subject>Buy 1 ETFC  Executed @ $1</subject>
         <readFlag>UNREAD</readFlag>
         <symbol/>
      </Alert>
      <Alert>
         <dateTime>1266438507</dateTime>
         <alertId>331</alertId>
         <subject>Buy 1 CSCO  Executed @ $100</subject>
         <readFlag>UNREAD</readFlag>
         <symbol/>
      </Alert>
```

```
<Alert>
   <dateTime>1266376877</dateTime>
   <alertId>329</alertId>
   <subject>Buy 1 ETFC  Expired</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1266376876</dateTime>
   <alertId>330</alertId>
   <subject>Buy 100 SPY  Expired</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1266376874</dateTime>
   <alertId>328</alertId>
   <subject>Buy 100 AAPL  Expired</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1265739660</dateTime>
   <alertId>327</alertId>
   <subject>Buy 1 ETFC  Cancelled</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1264179013</dateTime>
   <alertId>326</alertId>
   <subject>Buy 90 ETFC  Cancelled</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1264130514</dateTime>
   <alertId>324</alertId>
   <subject>Buy 1 ETFC  Expired</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1264130503</dateTime>
   <alertId>323</alertId>
   <subject>Buy 1 ETFC  Expired</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1264130494</dateTime>
   <alertId>325</alertId>
   <subject>Buy 1 CSCO  Expired</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
```

```
<Alert>
   <dateTime>1264019544</dateTime>
   <alertId>322</alertId>
   <subject>Buy 2 C  Executed @ $8.25</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1264019285</dateTime>
   <alertId>321</alertId>
   <subject>Buy 2 CSCO  Executed @ $100</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
<Alert>
   <dateTime>1264018791</dateTime>
   <alertId>320</alertId>
   <subject>Buy 2 ETFC  Executed @ $1</subject>
   <readFlag>UNREAD</readFlag>
   <symbol/>
</Alert>
```


Sample JSON Response


```
{"json.getAlertsResponse":{"response":{"dateTime":1296590336,"alertId":448,"subject
":"Attention Brokerage Customer","readFlag":"UNREAD","symbol":""}}}
```

Parameters

Request Parameter - *None*

Table 4-6 alerts Return Values

Parameter	Data Type	Description
dateTime	long (UTC)	The date and time the alert was issued.
alertID	integer	Numerical value assigned to an alert.
subject	string	The subject of the alert.
readFlag	enum	Details whether the alert has been read by the user. Options are: • READ • UNREAD
symbol	string	The market trading symbol for the stock being bought or sold.

Notes

See Also

`alertdetails`, `deletealert`

alertdetails

Description - return the details of a specific alert.

URL - *https://etws.etrade.com/accounts/rest/alerts/{alertID}*

Sandbox URL - *https://etwssandbox.etrade.com/accounts/sandbox/rest/alerts/{AlertID}*

HTTP Method - GET

Sample Request

This sample request shows the syntax required to return details of a specific alert.

```
/accounts/rest/alerts/321
```

Sample Response

When a user requests details about a specific alert, the response values returned to the user show information about the alert, including date and time for when the alert was created, the ID associated with the alert, text describing what the alert will state to the user when triggered, a read date, and a subject line describing the alert.


```
<AlertDetailsResponse>
   <createDate>1264019285</createDate>
   <alertId>321</alertId>
   <msgText>Account: xxxx-3535 Your Day buy order for 2 CSCO at market price  was
executed at $100.  See order # 388 for details.</msgText>
   <readDate>0</readDate>
   <subject>Buy 2 CSCO  Executed @ $100</subject>
</AlertDetailsResponse>
   </Alerts>
</AlertsResponse>
```


Sample JSON Response


```
{"json.getAlertDetailsResponse":{"createDate":1296590336,"alertId":448,"msgText":"\
n\n NEW YORK (MarketWatch) Yields on 10-year Treasury notes , which move
inversely\nto prices, fell just 1 basis point in early trading Tuesday, to 3.33%. A
basis\npoint is one one-hundreth of a percentage point, while yields move inversely
to\nprices.\n\n\n\n\n","readDate":0,"subject":"Attention Brokerage Customer"}}
```

Parameters

Table 4-7 alertdetails Request Parameters

Parameter	Data Type	Required/ Optional	Description
alertId	integer	Required	Numerical value assigned to an alert.

Table 4-8 alertdetails Return Values

Parameter	Data Type	Description
createDate	long (UTC)	The date that the alert was created.
alertID	integer	Numerical value assigned to an alert.
msgText	string	The text displayed in the alert. This is the message that is displayed when the alert is displayed.
readDate	long (UTC)	The date the alert was read by the user.
subject	string	The subject of the alert.

Notes

- The first request for details for a specific alert will have 0 as the read date.

See Also

`alerts`, `deletealert`

deletealert

Description - delete an indicated alert.

URL - *https://etws.etrade.com/accounts/rest/alerts/{AlertID}*

Sandbox URL - *https://etwssandbox.etrade.com/accounts/sandbox/rest/alerts/{AlertID}*

HTTP Method - DELETE

Sample Request

This sample request shows the REST syntax required to delete an alert associated with an account.

```
/accounts/rest/alerts/321
```

Sample Response

When a user requests an alert be deleted from an account, the response value returned to the user is a result of the request, which can be either success or failure.

```
<DeleteAlertResponse>
   <result>SUCCESS</result>
</DeleteAlertResponse>
```

Sample JSON Response

```
{"json.deleteAlertResponse":{"result":"SUCCESS"}}
```

Parameters

Table 4-9 deletealert Request Parameters

Parameter	Data Type	Required/Optional	Description
alertId	integer	Required	Numerical value assigned to an alert.

Table 4-10 deletealert Return Values

Parameter	Data Type	Description
result	string	Text description of whether the action was performed successfully or not.

Notes

See Also

`alerts`, `alertdetails`

Markets

The Markets module of the E*TRADE API consists of REST resources that allow a user to retrieve information about an index, stock, or option.

The following resources make up the Markets section of this chapter:

- `optionchains`
- `optionexpiredates`
- `productlookup`
- `quote`

optionchains

Description - return a list of option chains associated with a specific underlier.

URL - *https://etws.etrade.com/market/rest/optionchains*

Sandbox URL - *https://etwssandbox.etrade.com/market/sandbox/rest/optionchains*

HTTP Method - GET

Sample Request

This sample request shows the syntax required for a user to request a list of option chains for a specific underlier for an account.

```
/market/rest/optionchains?
expirationMonth=04&expirationYear=2010&chainType=PUT&skipAdjusted=true&underlier=ET
FC
```

Sample Response

When a user requests a list of option chains for an underlier associated with an account, response values returned to the user include the option pair count, information about that option chain pair, and the symbol for the option pair.


```
OptionChainResponse>
   <optionPairCount>9</optionPairCount>
   <OptionChainPair>
      <callCount>0</callCount>
      <pairType>PUTONLY</pairType>
      <put>
         <OptionChain>
            <rootSymbol>EUS</rootSymbol>
            <expireDate>
               <day>17</day>
               <month>4</month>
               <year>2010</year>
            </expireDate>
            <product>
               <exchangeCode>CINC</exchangeCode>
               <symbol>EUS Apr 17 '10 $1 Put</symbol>
               <typeCode>OPTN</typeCode>
            </product>
            <strikePrice>1.000000</strikePrice>
         </OptionChain>
      </put>
      <putCount>1</putCount>
   </OptionChainPair>
<OptionChainPair>
      <callCount>0</callCount>
      <pairType>PUTONLY</pairType>
      <put>
         <OptionChain>
            <rootSymbol>EUS</rootSymbol>
            <expireDate>
               <day>17</day>
               <month>4</month>
               <year>2010</year>
            </expireDate>
            <product>
               <exchangeCode>CINC</exchangeCode>
               <symbol>EUS Apr 17 '10 $3 Put</symbol>
               <typeCode>OPTN</typeCode>
```

```
            </product>
            <strikePrice>3.000000</strikePrice>
         </OptionChain>
      </put>
      <putCount>1</putCount>
   </OptionChainPair>
      <callCount>0</callCount>
      <pairType>PUTONLY</pairType>
      <put>
         <OptionChain>
            <rootSymbol>EUS</rootSymbol>
            <expireDate>
               <day>17</day>
               <month>4</month>
               <year>2010</year>
            </expireDate>
            <product>
               <exchangeCode>CINC</exchangeCode>
               <symbol>EUS Apr 17 '10 $5 Put</symbol>
               <typeCode>OPTN</typeCode>
            </product>
            <strikePrice>5.000000</strikePrice>
         </OptionChain>
      </put>
      <putCount>1</putCount>
   </OptionChainPair>
   <symbol>ETFC</symbol>
</OptionChainResponse
```


Sample JSON Response

This JSON response shows the option chains for March 2011 EFTC chains,


```
{"optionChainResponse":{"optionPairCount":12,"optionPairs":[{"call":{"rootSymbol":"
ETFC","expireDate":{"day":19,"month":3,"year":2011,"expiryType":"MONTHLY"},"product
":{"exchangeCode":"CINC","symbol":"ETFC Mar 19 '11 $11
Call","typeCode":"OPTN"},"strikePrice":"11.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $12
Call","typeCode":"OPTN"},"strikePrice":"12.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $13
Call","typeCode":"OPTN"},"strikePrice":"13.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $14
Call","typeCode":"OPTN"},"strikePrice":"14.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $15
Call","typeCode":"OPTN"},"strikePrice":"15.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
```

```
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $16
Call","typeCode":"OPTN"},"strikePrice":"16.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $17
Call","typeCode":"OPTN"},"strikePrice":"17.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $18
Call","typeCode":"OPTN"},"strikePrice":"18.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $19
Call","typeCode":"OPTN"},"strikePrice":"19.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $20
Call","typeCode":"OPTN"},"strikePrice":"20.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $21
Call","typeCode":"OPTN"},"strikePrice":"21.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0},{"call":{"rootSymbol":"ETFC","expireDate":{"day":19,"month":3,"y
ear":2011,"expiryType":"MONTHLY"},"product":{"exchangeCode":"CINC","symbol":"ETFC
Mar 19 '11 $22
Call","typeCode":"OPTN"},"strikePrice":"22.000000"},"callCount":1,"pairType":"CALLO
NLY","putCount":0}],"symbol":"ETFC"}}
```


Parameters

Table 4-11 optionchains Request Parameters

Parameter	Data Type	Required/ Optional	Description
expirationMonth	integer (1-12)	Required	The month the option will expire, as specified in the option contract.
expirationYear	4-digit integer	Required	The year the option will expire, as specified in the option contract.
chainType	enum	Required	The type of option chain. Options are: • CALL • PUT • CALLPUT

Parameter	Data Type	Required/ Optional	Description
underlier	string	Required	The market trading symbol for the stock being bought or sold.
skipAdjusted	boolean	Optional	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 ben amended to modify this change. Options are: • TRUE (default) • FALSE

Table 4-12 optionchains Return Values

Parameter	Data Type	Description
optionPairCount	integer	The number of option pairs returned.
symbol	string	The market trading symbol for the option.
optionPairs	OptionChainPair*	The list of option chains.
***Under each optionChain Pair the following values are returned:**		
call	OptionChain*	A list of call objects.
callCount	integer	The number of call objects.
put	OptionChain*	A list of put objects.
putCount	integer	The number of put objects.
pairType	enum	The type of option pair. Options are: • CALLONLY • PUTONLY • SPREADS • CALLPUT
***For each OptionChain, the following values are returned:**		
rootSymbol	string	The underlier for the option however, for adjusted options, the option symbol may have a different symbol.
expireDate	expireDate*	The expiration date (year, month, day) that the option contract will expire.
productId	productId*	The product ID.
strikePrice	double	The strike price for the option, as specified in the option contract.
For expireDate, the following values are included in the response:		
day	integer (1-31)	The day of the month the option will expire, as specified in the option contract.
month	string (1-12)	The month of the year the option will expire, as specified in the option contract
year	integer (4-digit integer)	The year the option will expire, as specified in the option contract.
For ProductId, the following values are returned.		
symbol	string	The market trading symbol for the option.
typeCode	enum	The type of derivative. Options are: • OPTN
exchangeCode	enum	The primary Exchange where the option is traded. Options are:

Notes

See Also

`productlookup`, `quote`, `optionexpiredates`

optionexpiredates

Description - return list of expiration dates for options.

URL - *https://etws.etrade.com/market/rest/optionexpiredate*

Sandbox URL - *https://etwssandbox.etrade.com/market/sandbox/rest/optionexpiredate*

HTTP Method - GET

Sample Request

This sample request shows the syntax required for a REST request to return the expiration dates for a specific underlier.

```
/market/rest/optionexpiredate?underlier=GOOG
```

Sample Response

Once a user requests the expiration dates for an option, the values returned to the user include each expiration date for each option (day, month, and year).

```
<OptionExpireDateGetResponse>
   <ExpirationDate>
      <day>20</day>
      <month>3</month>
      <year>2010</year>
   </ExpirationDate>
   <ExpirationDate>
      <day>17</day>
      <month>4</month>
      <year>2010</year>
   </ExpirationDate>
   <ExpirationDate>
      <day>19</day>
```

```
      <month>6</month>
      <year>2010</year>
   </ExpirationDate>
   <ExpirationDate>
      <day>18</day>
      <month>9</month>
      <year>2010</year>
   </ExpirationDate>
   <ExpirationDate>
      <day>22</day>
      <month>1</month>
      <year>2011</year>
   </ExpirationDate>
   <ExpirationDate>
      <day>21</day>
      <month>1</month>
      <year>2012</year>
   </ExpirationDate>
</OptionExpireDateGetResponse>
```


Sample JSON Response


```
{"optionExpireDateGetResponse":{"expireDates":[{"day":19,"month":2,"year":2011,"exp
iryType":"MONTHLY"},{"day":19,"month":3,"year":2011,"expiryType":"MONTHLY"},{"day":
18,"month":6,"year":2011,"expiryType":"MONTHLY"},{"day":17,"month":9,"year":2011,"e
xpiryType":"MONTHLY"},{"day":21,"month":1,"year":2012,"expiryType":"MONTHLY"},{"day
":19,"month":1,"year":2013,"expiryType":"MONTHLY"}]}}
```

Parameters

Table 4-13 optionexpiredates Request Parameters

Parameter	Data Type	Required/ Optional	Description
underlier	string	Required	The symbol of the underlying security or index for the option.

Table 4-14 optionexpiredates Return Values

Return Value	Data Type	Description
day	integer (1-31)	The day of the month the option will expire, as specified in the option contract.
month	integer (1-12)	The month of the year the option will expire, as specified in the option contract.
year	integer (4-digit)	The year the option will expire, as specified in the option contract

Notes

See Also

`optionchains`, `productlookup`

productlookup

Description - returns matching stock information for a specified query string.

URL - *https://etws.etrade.com/market/rest/productlookup*

Sandbox URL - *https://etwssandbox.etrade.com/market/sandbox/rest/productlookup*

HTTP Method - GET

Sample Request

This sample request shows the REST-based syntax required to retrieve information about Cisco Systems stock.

```
/market/rest/productlookup?company=cisco&type=EQ
```

Sample Response

When a user requests information about a specific stock, the values returned to the user include the company name, the exchange where the stock is listed, the type of security, and the stock symbol.


```
<ProductLookupResponse>
   <Product>
      <companyName>CISCO SYS INC COM</companyName>
      <exchange>NASDAQ NM</exchange>
      <securityType>EQUITY</securityType>
      <symbol>CSCO</symbol>
   </Product>
   <Product>
      <companyName>FRANCISCO INDS INC COM</companyName>
      <exchange>OTC</exchange>
      <securityType>EQUITY</securityType>
      <symbol>FRAN</symbol>
   </Product>
</ProductLookupResponse>
```


Sample JSON Response


```
{"productLookupResponse":{"productList":[{"companyName":"CISCO SYS INC
COM","exchange":"NASDAQ
NM","securityType":"EQ","symbol":"CSCO"},{"companyName":"FRANCISCO INDS INC
COM","exchange":"OTC","securityType":"EQ","symbol":"FRAN"},{"companyName":"SAN
FRANCISCO CO","exchange":"OTC","securityType":"EQ","symbol":"SFHC"}]}}
```

Parameters

Table 4-15 productlookup Request Parameters

Parameter	Data Type	Required/ Optional	Description
company	string	Required	The name of the company.
type	enum	Required	The type of security. Options are: • EQ - Equity • MF - Mutual Fund

Table 4-16 productlookup Return Values

Parameter	Data Type	Description
companyName	string	The name of the company.
exchange	string	The Exchange where the company is listed.
securityType	enum	The type of security for the company. Options are • EQ - Equity • MF - Mutual Fund
symbol	string	The market trading symbol for the stock being bought or sold.

Notes

See Also

`quote`, `optionchains`, `optionchains`

quote

Description - return the latest market information for a symbol.

URL - *https://etws.etrade.com/market/rest/quote/{symbol, comma-separated list}*

Sandbox URL - *https://etwssandbox.etrade.com/market/sandbox/rest/quote/{symbol}*

HTTP Method - GET

Sample Request

This sample request shows the REST-based syntax required to retrieve market information for Google stock.

```
/market/rest/quote/GOOG,EFTC?detailFlag=ALL
```

Sample Response

Once the user requests stock information for Google, all market information for the stock is displayed.


```
<QuoteResponse>
   <QuoteData>
      <all>
         <adjNonAdjFlag>false</adjNonAdjFlag>
         <annualDividend>0.0</annualDividend>
         <ask>579.4</ask>
         <askExchange>I</askExchange>
         <askSize>100</askSize>
         <askTime>18:49:23 EST 03-12-2010</askTime>
         <beta>0.92</beta>
         <bid>578.79</bid>
         <bidExchange>Q</bidExchange>
         <bidSize>100</bidSize>
         <bidTime>18:49:23 EST 03-12-2010</bidTime>
         <companyName>GOOGLE INC</companyName>
         <daysToExpiration>0</daysToExpiration>
         <dirLast>D</dirLast>
         <dividend>0.0</dividend>
         <eps>20.41</eps>
         <estEarnings>22.754</estEarnings>
         <exDivDate/>
         <exchgLastTrade>NASDAQ</exchgLastTrade>
         <fsi>N</fsi>
         <high>588.275</high>
```

```
            <high52>629.51</high52>
            <highAsk>590.0</highAsk>
            <highBid>588.65</highBid>
            <lastTrade>578.78</lastTrade>
            <low>578.78</low>
            <low52>294.25</low52>
            <lowAsk>578.79</lowAsk>
            <lowBid>572.21</lowBid>
            <numTrades>18037</numTrades>
            <open>587.7</open>
            <openInterest>0</openInterest>
            <optionStyle/>
            <optionUnderlier/>
            <prevClose>581.14</prevClose>
            <prevDayVolume>1550127</prevDayVolume>
            <primaryExchange>Q</primaryExchange>
            <symbolDesc>GOOGLE INC</symbolDesc>
            <todayClose>579.54</todayClose>
            <totalVolume>2755015</totalVolume>
            <upc>0</upc>
            <volume10Day>2010351</volume10Day>
         </all>
         <dateTime>18:47:21 EST 03-12-2010</dateTime>
         <product>
            <symbol>GOOG</symbol>
            <type>EQ</type>
         </product>
      </QuoteData>
      <QuoteData>
         <all>
            <adjNonAdjFlag>false</adjNonAdjFlag>
            <annualDividend>0.0</annualDividend>
            <ask>1.65</ask>
            <askExchange>Q</askExchange>
            <askSize>2000</askSize>
            <askTime>18:50:42 EST 03-12-2010</askTime>
            <beta>1.9</beta>
            <bid>1.64</bid>
            <bidExchange>Q</bidExchange>
            <bidSize>30300</bidSize>
            <bidTime>18:50:42 EST 03-12-2010</bidTime>
            <companyName>E*TRADE FINANCIAL CORP</companyName>
            <daysToExpiration>0</daysToExpiration>
            <dirLast>D</dirLast>
            <dividend>0.0</dividend>
            <eps>-1.18</eps>
            <estEarnings>-0.841</estEarnings>
            <exDivDate/>
            <exchgLastTrade>Pacific</exchgLastTrade>
            <fsi>N</fsi>
            <high>1.67</high>
            <high52>2.9</high52>
            <highAsk>1.79</highAsk>
            <highBid>1.67</highBid>
            <lastTrade>1.64</lastTrade>
            <low>1.64</low>
```

```
        <low52>0.59</low52>
        <lowAsk>1.65</lowAsk>
        <lowBid>1.58</lowBid>
        <numTrades>14559</numTrades>
        <open>1.67</open>
        <openInterest>0</openInterest>
        <optionStyle/>
        <optionUnderlier/>
        <prevClose>1.67</prevClose>
        <prevDayVolume>134914923</prevDayVolume>
        <primaryExchange>Q</primaryExchange>
        <symbolDesc>E*TRADE FINANCIAL CORP</symbolDesc>
        <todayClose>1.65</todayClose>
        <totalVolume>23968286</totalVolume>
        <upc>0</upc>
        <volume10Day>50286144</volume10Day>
     </all>
     <dateTime>19:03:27 EST 03-12-2010</dateTime>
     <product>
        <symbol>ETFC</symbol>
        <type>EQ</type>
     </product>
  </QuoteData>
</QuoteResponse>
```


Sample JSON Repsonse

detailFlag = ALL


```
HTTP Response:
{"quoteResponse":{"quoteData":{"all":{"adjNonAdjFlag":false,"annualDividend":0,"ask
":603.94,"askExchange":"Q","askSize":100,"askTime":"14:35:42 EST 01-04-
2011","bid":603.75,"bidExchange":"P","bidSize":100,"bidTime":"14:35:42 EST 01-04-
2011","chgClose":-0.4900000000000091,"chgClosePrcn":-0.08,"companyName":"GOOGLE
INC","daysToExpiration":0,"dirLast":"U","dividend":0,"eps":24.63,"estEarnings":33.6
42,"exDivDate":"","exchgLastTrade":"","fsi":"N","high":606.18,"high52":630.85,"high
Ask":640,"highBid":606.79,"lastTrade":603.86,"low":600.12,"low52":433.63,"lowAsk":6
00.12,"lowBid":449.47,"numTrades":10590,"open":606.4,"openInterest":0,"optionStyle"
:"","optionUnderlier":"","prevClose":604.35,"prevDayVolume":2366286,"primaryExchang
e":"Q ","symbolDesc":"GOOGLE
INC","todayClose":0,"totalVolume":1412611,"upc":0,"volume10Day":1436450},"dateTime"
:"14:35:30 EST 01-04-2011","product":{"symbol":"GOOG","type":"EQ","exchange":"Q
"}}}}
```


FUNDAMENTAL:


```
HTTP Response: {"quoteResponse":{"quoteData":{"dateTime":"14:51:22 EST 01-04-
2011","fundamental":{"companyName":"GOOGLE
INC","eps":24.63,"estEarnings":33.642,"high52":630.85,"lastTrade":602.43,"low52":43
3.63,"symbolDesc":"GOOGLE
INC","volume10Day":1436450},"product":{"symbol":"GOOG","type":"EQ","exchange":"Q
"}}}}
```

INTRADAY:


```
HTTP Response: {"quoteResponse":{"quoteData":{"dateTime":"14:52:03 EST 01-04-
2011","intraday":{"ask":602.56,"bid":602.45,"chgClose":-
1.8999999999999773,"chgClosePrcn":-0.31,"companyName":"GOOGLE
INC","high":606.18,"lastTrade":602.45,"low":600.12,"totalVolume":1465065},"product"
:{"symbol":"GOOG","type":"EQ","exchange":"Q "}}}}
```


OPTIONS:


```
HTTP Response: {"quoteResponse":{"quoteData":{"dateTime":"14:52:52 EST 01-04-
2011","option":{"ask":602.73,"askSize":100,"bid":602.57,"bidSize":300,"companyName"
:"GOOGLE
INC","daysToExpiration":0,"lastTrade":602.7299,"openInterest":0},"product":{"symbol
":"GOOG","type":"EQ","exchange":"Q "}}}}
```


WEEK_52:


```
HTTP Response: {"quoteResponse":{"quoteData":{"dateTime":"14:54:41 EST 01-04-
2011","product":{"symbol":"GOOG","type":"EQ","exchange":"Q
"},"week52":{"annualDividend":0,"companyName":"GOOGLE
INC","high52":630.85,"lastTrade":602.6,"low52":433.63,"perf12Months":97,"prevClose"
:604.35,"symbolDesc":"GOOGLE INC","totalVolume":1469699}}}}
```

Parameters

Table 4-17 quotes Request Parameters

Parameter	Data Type	Required/ Optional	Description
symbol	string	Required	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*
detailFlag	enum	Optional	Determines the market fields returned from a quote request. Fields returned from each option are as follows: • ALL • FUNDAMENTAL • INTRADAY • OPTIONS • WEEK_52
Return Value	**Data Type**	**Description**	
adjNonAdjFlag	boolean	Displays whether an option has been adjusted due to a corporate action (e.g. a dividend or stock split).	
annualDividend	double	The cash amount paid per share over the past year.	
ask	double	The current ask price for a security.	
askExchange	string	A code* representing the exchange reporting the ask price.	
askSize	long	The number of shares or contracts offered by a broker/ dealer at the ask price.	
askTime	string	The time at which the ask was placed.	
beta	double	A measure of a stock's volatility relative to the primary market index.	
bid	double	The current bid price for a security.	
bidExchange	string	A code* representing the exchange reporting the bid price.	
bidSize	long	The number of shares or contracts offered at the bid price.	
bidTime	string	The time at which the bid was placed.	
companyName	string	The name of the company or mutual fund (shows up to 40 characters).	

Parameter	Data Type	Required/ Optional	Description
daysToExpiration	long		Number of days before the option expires.
dirLast	string		Indicates whether the current price is higher or lower than the price of the most recent trade.
dividend	double		The cash amount per share of the latest dividend.
eps	double		Earnings per share on a rolling basis (applies to stocks only).
estEarnings	double		Projected earnings per share for the next fiscal year (applies to stocks only).
exDivDate	string		The date on which shareholders were entitled to receive the latest dividend.
exchgLastTrade	string		A code* representing the exchange of last trade.
fsi	string		Financial Status Indicator - Indicates whether a Nasdaq-listed issuer has failed to submit its regulatory filings on a timely basis, failed to meet continuing listing standards, and/or filed for bankruptcy. Codes are: • D - Deficient • E - Delinquent • Q - Bankrupt • N - Normal • G - Deficient and Bankrupt • H - Deficient and Delinquent • J - Delinquent and Bankrupt • K - Deficient, Delinquent, and Bankrupt
high	double		The highest price at which a security has traded during the current day.
high52	double		The highest price at which a security has traded during the past year (52 weeks, this is a lifetime high for options).
highAsk	double		The highest ask price for the current trading day.
highBid	double		The highest bid price for the current trading day.
lastTrade	double		The price of the most recent trade in a security.
low	double		The lowest price at which a security has traded during the current day.
low52	double		The lowest price at which a security has traded during the past year (52 weeks this is a lifetime high for options).
lowAsk	double		The lowest ask price for the current trading day.
lowBid	double		The lowest bid price for the current trading day.
numTrades	long		Number of transactions involving buying a security from another entity.

Parameter	Data Type	Required/ Optional	Description
open	double		The price of a security at the current day's market open.
openInterest	long		The total number of options or futures contracts that are not closed or delivered on a particular day.
optionStyle	string		American or European style option.
optionUnderlier	string		The symbol for the underlying security of a particular option.
prevClose	double		The official price at the close on the previous trading day.
prevDayVolume	long		The final volume from the previous market session.
primaryExchange	string		A code* indicating the exchange on which the security is primarily listed. The following Exchange codes can be selected: • A - American Stock Exchange • G - Amex Emerging Company Marketplace • AP - Archipelago • Z - BATS • M - Chicago (Midwest) Stock Exchange • CO - Chicago Board Options Exchange • C - Cincinnati Stock Exchange • GP - GovPX Bonds • I - International Securities Exchange (Options) • IS - Island • XT - Market XT • V - Nasdaq Bulletin Board - Trades Only • F - NASDAQ Mutual Fund & Money Market Fund • Q - Nasdaq National Market System (NMS) • B - NASDAQ OMX BX • U - Nasdaq OTC Bulletin Board (Pink Sheet Stocks) • S - Nasdaq Small Cap • T - Nasdaq Trades in Listed Stocks • N - New York Stock Exchange • R - New York Stock Exchange Trade Reporting Facility (TRF) • P - Pacific Stock Exchange • P - Pacific Stock Exchange - Tier I • PT - Pacific Stock Exchange - Tier II • X - Philadelphia Stock Exchange • RD - REDbook

Parameter	Data Type	Required/ Optional	Description
symbolDesc	string		A description of the security – e.g. company name or option description.
todayClosed	double		The price at the close of the regular trading session for the current day.
totalVolume	long		Total number of shares or contracts exchanging hands.
upc	long		Uniform Practice Code - identifies specific FINRA advisories detailing unusual circumstances (e.g. extremely large dividends, when-issued settlement dates, and worthless securities).
volume10Day	long		Ten-day average trading volume for the security (total volume for the security over the last 10 trading days divided by 10)
If FUNDAMENTAL is selected, the following fields are returned:			
beta	double		A measure of a stock's volatility relative to the primary market index.
eps	double		Earnings per share on a rolling basis (applies to stocks only).
estEarnings	double		Projected earnings per share for the next fiscal year (applies to stocks only).
high52	double		The highest price at which a security has traded during the past year (52 weeks, this is a lifetime high for options).
lastTrade	double		The price of the most recent trade in a security.
low52	double		The lowest price at which a security has traded during the past year (52 weeks this is a lifetime high for options).
symbolDesc	string		A description of the security – e.g. company name or option description.
volume10Day	long		Ten-day average trading volume for the security (total volume for the security over the last 10 trading days divided by 10)
If INTRADAY is selected, the following fields are returned:			
ask	double		The current ask price for a security.
bid	double		The current bid price for a security.
chgCloseD	double		The dollar change of the last price from the previous close
chgClosePrcn	double		The percentage change of the last price from the previous close.
high	double		The highest price at which a security has traded during the current day.
lastTrade	double		The price of the most recent trade in a security.

Parameter	Data Type	Required/ Optional	Description
low	double		The lowest price at which a security has traded during the current day.
totalVolume	long		Total number of shares or contracts exchanging hands.
IF OPTN is selected, the following fields are returned:			
ask	double		The current ask price for a security.
askSize	long		The number of shares or contracts offered by a broker/ dealer at the ask price.
bid	double		The current bid price for a security.
bidSize	long		The number of shares or contracts offered at the bid price.
daysToExpiration	long		Number of days before the option expires.
lastTrade	double		The price of the most recent trade in a security.
openInterest	long		The total number of options or futures contracts that are not closed or delivered on a particular day.
If WEEK_52 is selected, the following fields are returned:			
annualDividend	double		The cash amount paid per share over the past year
high	double		The highest price at which a security has traded during the current day.
lastTrade	double		The price of the most recent trade in a security.
low	double		The lowest price at which a security has traded during the current day.
prevClose	double		The official price at the close on the previous trading day.
symbolDesc	string		A description of the security – e.g. company name or option description.
totalVolume	long		Total number of shares or contracts exchanging hands.

Table 4-18 quotes Return Values

Parameter	Data Type	Description
adjNonAdjFlag	boolean	Flag that determines whether you are looking at an adjusted option
annualDividend	double	The annual dividend for the stock.
ask	double	The best offer price for a stock.
askExchange	string	The exchange where this best ask price is being listed.
dateTime	long (UTC)	The date and time the quote request was processed by the system.
symbol	string	The market trading symbol for the stock in the quote.
type	enum	The type of stock in the quote. There are two options • EQ • OPTN

Notes

See Also

`productlookup`, `optionchains`

Order Services

The Order Services Module of the E*TRADE API allows users to perform a number of order operations for both equities and options. These operations include placing, modifying, and canceling an order.

The following endpoints make up the Orders/Trading section of this chapter:

- `orderlist`
- `previewequityorder`
- `placeequityorder`
- `previewchangeequityorder`
- `placechangeequityorder`
- `previewoptionorder`
- `placeoptionorder`
- `previewchangeoptionorder`
- `placechangeoptionorder`
- `cancelorder`

orderlist

Description - return a list of open orders or orders updated today.

URL - *https://etws.etrade.com/order/rest/orderlist/{accountId}*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/orderlist/{accountId}*

HTTP Method - GET

Sample Request

This sample request shows the REST-based syntax required to retrieve a list of open orders for an account.

```
orders/rest/orderlist/(add acount ID here)
```

Sample Response

The sample response below shows the values returned to the user when a request has been made for a list of all open orders. The information returned to the user includes the number of open orders, a marker for the order, and details about each specific open order.


```
<GetOrderListResponse>
   <orderListResponse>
      <count>3</count>
      <marker>1270023312|296</marker>
      <orderDetails>
         <OrderDetails>
            <order>
               <orderId>320</orderId>
               <orderPlacedTime>1270036641807</orderPlacedTime>
               <orderValue>1068.77</orderValue>
               <orderStatus>OPEN</orderStatus>
               <orderType>BUY_WRITES</orderType>
               <orderTerm>GOOD_FOR_DAY</orderTerm>
               <priceType>MARKET</priceType>
               <limitPrice>0</limitPrice>
               <stopPrice>0</stopPrice>
               <legDetails>
                  <LegDetails>
                     <legNumber>1</legNumber>
                     <symbolInfo>
                        <symbol>DELL</symbol>
                     </symbolInfo>
```

```
            <symbolDescription>DELL INC COM</symbolDescription>
            <orderAction>BUY</orderAction>
            <orderedQuantity>100</orderedQuantity>
            <filledQuantity>0</filledQuantity>
            <executedPrice>0</executedPrice>
            <estimatedCommission>7.99</estimatedCommission>
            <estimatedFees>0</estimatedFees>
         </LegDetails>
         <LegDetails>
            <legNumber>2</legNumber>
            <symbolInfo>
               <symbol>DLY</symbol>
               <callPut>CALL</callPut>
               <expYear>2010</expYear>
               <expMonth>4</expMonth>
               <expDay>17</expDay>
               <strikePrice>10</strikePrice>
            </symbolInfo>
            <symbolDescription>DELL APR 10 Call</symbolDescription>
            <orderAction>SELL_OPEN</orderAction>
            <orderedQuantity>1</orderedQuantity>
            <filledQuantity>0</filledQuantity>
            <executedPrice>0</executedPrice>
            <estimatedCommission>8.74</estimatedCommission>
            <estimatedFees>0</estimatedFees>
         </LegDetails>
      </legDetails>
      <allOrNone>false</allOrNone>
   </order>
</OrderDetails>
<OrderDetails>
   <order>
      <orderId>317</orderId>
      <orderPlacedTime>1270031791517</orderPlacedTime>
      <orderExecutedTime>1270031819000</orderExecutedTime>
      <orderValue>242.61</orderValue>
      <orderStatus>EXECUTED</orderStatus>
      <orderType>EQ</orderType>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <priceType>MARKET</priceType>
      <limitPrice>0</limitPrice>
      <stopPrice>0</stopPrice>
      <legDetails>
         <LegDetails>
            <legNumber>1</legNumber>
            <symbolInfo>
               <symbol>MSFT</symbol>
            </symbolInfo>
            <symbolDescription>MICROSOFT CORP COM</symbolDescription>
            <orderAction>SELL_SHORT</orderAction>
            <orderedQuantity>10</orderedQuantity>
            <filledQuantity>10</filledQuantity>
            <executedPrice>1</executedPrice>
            <estimatedCommission>7.99</estimatedCommission>
            <estimatedFees>0</estimatedFees>
         </LegDetails>
```

```
            </legDetails>
            <allOrNone>false</allOrNone>
         </order>
      </OrderDetails>
      <OrderDetails>
         <order>
            <orderId>305</orderId>
            <orderPlacedTime>1270025557990</orderPlacedTime>
            <orderExecutedTime>1270025673000</orderExecutedTime>
            <orderValue>0</orderValue>
            <orderStatus>CANCELLED</orderStatus>
            <orderType>EQ</orderType>
            <orderTerm>GOOD_FOR_DAY</orderTerm>
            <priceType>MARKET</priceType>
            <limitPrice>0</limitPrice>
            <stopPrice>0</stopPrice>
            <legDetails>
               <LegDetails>
                  <legNumber>1</legNumber>
                  <symbolInfo>
                     <symbol>F</symbol>
                  </symbolInfo>
<symbolDescription>FORD MTR CO DEL COM PAR $0.01</symbolDescription>
<orderAction>SELL</orderAction>
                  <orderedQuantity>4</orderedQuantity>
                  <filledQuantity>0</filledQuantity>
                  <executedPrice>0</executedPrice>
                  <estimatedCommission>7.99</estimatedCommission>
                  <estimatedFees>0</estimatedFees>
               </LegDetails>
            </legDetails>
            <replacedByOrderId>306</replacedByOrderId>
            <allOrNone>false</allOrNone>
         </order>
      </OrderDetails>
   </orderDetails>
  </orderListResponse>
</GetOrderListResponse
```

Parameters

Table 4-19 orderlist Request Parameters

Parameter	Data Type	Required/ Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
marker	string	Optional	Parameter that designates additional orders in the account if there are more than the requested number of orders.
count	integer	Optional	Parameter that returns the number of list of open orders or orders updated today. If this is not specified, the default value is 25.

Table 4-20 orderlist Return Values

Parameter	Data Type	Description
count	integer	The number of list of open orders or orders updated today. If this is not specified, the default value is 25.
GroupOrder - custom data type		
groupOrderId	long	Number associated with a group order.
groupOrderType	enum	The type of group order specified in the response. Options are: • CONTINGENT • ONE_CANCELS_ALL • ONE_TRIGGERS_ALL • ONE_TRIGGERS_OCO • BRACKETED
cumulativeEstimatedCommission	double	The cumulative estimated commission for the group order,
cumulativeEstimatedFees	double	The cumulative estimated fee for the group order.
order - List of Orders (fields shown below)		
orderId	long	Number associated with the order.
orderPlacedTime	long (UTC)	The time the order was placed.
orderExecutedTime	long (UTC)	The time the order was executed.
orderValue	double	Total cost or proceeds, including commission.
orderStatus	enum	The current status of the order. Options are: • OPEN • EXECUTED • CANCELLED • CANCEL_REQUESTED • EXPIRED • REJECTED

Parameter	**Data Type**	**Description**
orderType	enum	The type of order being placed. Options are: • EQ • OPTN • ADVANCE_EQ • ADVANCE_OPTN • SPREADS • BUY_WRITES • CONTINGENT • ONE_CANCELS_ALL • ONE_TRIGGERS_ALL • ONE_TRIGGERS_OCO • BRACKETED
orderTerm	enum	The order term specified. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL • FILL_OR_KILL

Parameter	Data Type	Description
priceType	enum	The price type specified in the order. Options are: • MARKET • LIMIT • STOP • STOP_LIMIT • TRAILING_STOP_CNST_BY_LOWER_TRIGGER • UPPER_TRIGGER_BY_TRAILING_STOP_CNST • TRAILING_STOP_PRCT_BY_LOWER_TRIGGER • UPPER_TRIGGER_BY_TRAILING_STOP_PRCT • TRAILING_STOP_CNST • TRAILING_STOP_PRCT • HIDDEN_STOP • HIDDEN_STOP_BY_LOWER_TRIGGER • UPPER_TRIGGER_BY_HIDDEN_STOP • NET_DEBIT • NET_CREDIT • NET_EVEN • MARKET_ON_OPEN • MARKET_ON_CLOSE • LIMIT_ON_OPEN • LIMIT_ON_CLOSE
limitPrice	double	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
stopPrice	double	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
routingDestination	enum	Options are: • AUTO • ARCA • NSDQ • NYSE • AMEX • BOX • CBOE • ISE • NOM • PHX

Parameter	Data Type	Description
legDetails - A list of leg details (at least one entry will be present in each Order record, details are provided below)		
bracketLimitPrice	double	The user's lower trigger price on a buy and upper trigger price on a sell.
initialStopPrice	double	The initial stop price that was set when a trailing stop order was submitted.
trailPrice	double	The current trailing value. • For Trailing Stop $ orders, this is a fixed dollar amount. • For Trailing Stop%, this is the price reflected by the percentage selected.
triggerPrice	double	The price that an advanced order will trigger. For example, if it's a $1 buy trailing stop, then the trigger price will $1 above the last price.
conditionPrice	double	The price that the condition of a conditional order is being compared against.
conditionSymbol	string	The symbol that the condition of a conditional order is competing against.
conditionType	enum	The type of comparison the condition of a conditional order is comparing against. Options are: • CONTINGENT_GTE • CONTINGENT_LTE
conditionFollowPrice	enum	The price of the condition of the conditional order being followed. Options are: • ASK • BID • LAST
replacedByOrderId	long	The order ID of the order that was replaced due to a change order request.
replacesOrderId	long	The order ID of the order that this order replaced due to a change order request.
allOrNone	boolean	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE
legNumber	long	The number associated with the portion of the order. For example, Buy Write orders will have two legs.
symbolDescription	string	Text description of the market symbol for the company.

Parameter	Data Type	Description
orderAction	enum	User-specified action that instructs the broker what action to perform. Options are: • BUY • SELL • BUY_TO_COVER • SELL_SHORT • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
orderedQuantity	double	The number of shares ordered.
filledQuantity	double	The number of shares actually filled in the order.
executedPrice	double	The price the stock was actually purchased for.
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedFees	double	The total amount of fees billed to the user to fill the order.
reserveQuantity	double	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is true
symbolInfo - custom		
symbol	string	The market trading symbol for the share being bought or sold.
callPut	enum	Specifies whether options are being bought (call) or sold (put), as specified in the option order. Options are: • CALL • PUT
expYear	4-digit integer	The year the option will expire, as specified in the option contract.
expMonth	integer (1-12)	The month the option will expire, as specified in the option contract.
expDay	integer (1-31)	The day of the month the option will expire, as specified in the option contract.
strikePrice	double	The strike price for the option specified in the order.

Notes

- For complex options, only the first leg will show the accurate status. All other legs will show status as “open.”

previewequityorder

Description - preview an equity order to check for errors and view the estimated commission and total cost or proceeds.

URL - *https://etws.etrade.com/order/rest/previewequityorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/previewequityorder*

HTTP Method - POST

Sample Request

This sample request shows the REST-based syntax required for previewing an equity order before it is submitted. (These parameters should be included in the request header)

```
<PreviewEquityOrder xmlns="http://order.etws.etrade.com">
        <EquityOrderRequest>
           <accountId>83495799</accountId>
           <limitPrice></limitPrice>
           <stopPrice>197</stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity>4</quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <symbol>IBM</symbol>
           <orderAction>BUY</orderAction>
           <priceType>STOP</priceType>
           <routingDestination></routingDestination>
           <marketSession>REGULAR</marketSession>
           <orderTerm>GOOD_FOR_DAY</orderTerm>
           <clientOrderId>random123456</clientOrderId>
        </EquityOrderRequest>
     </PreviewEquityOrder>
```

Sample Response

The sample response below shows information about the equity order before it has been placed, including the account ID, estimated commission to place the equity order, and the total amount required to place the equity order.

```
<PreviewEquityOrderResponse>
   <equityOrderResponse>
      <accountId>83495799</accountId>
```

```
      <allOrNone>false</allOrNone>
      <estimatedCommission>7.99</estimatedCommission>
      <estimatedTotalAmount>795.99</estimatedTotalAmount>
      <messageList>
         <message>
            <msgDesc>You have an existing open order for this security on the same
side of the market.  If you did not intend to place a second order for this security,
please modify your order now.</msgDesc>
            <msgCode>1042</msgCode>
         </message>
      </messageList>
      <previewTime>1269428745346</previewTime>
      <previewId>449548380022</previewId>
      <quantity>4</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <limitPrice>0</limitPrice>
      <stopPrice>197</stopPrice>
      <symbolDesc>INTERNATIONAL BUSINESS MACHS COM</symbolDesc>
      <symbol>IBM</symbol>
      <orderAction>BUY</orderAction>
      <priceType>STOP</priceType>
   </equityOrderResponse>
</PreviewEquityOrderResponse>
```

Parameters

Table 4-21 previewequityorder Request Parameters

Parameter	Data Type	Required/Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
symbol	string	Required	The market trading symbol for the share being bought or sold.
orderAction	enum	Required	User-specified action that instructs the broker what action to perform. Options are: • BUY • SELL • BUY_TO_COVER • SELL_SHORT
clientOrderId	string	Required	String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
limitPrice	double	Optional*	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
stopPrice	double	Optional*	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
allOrNone	boolean	Optional	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE **Note**: This is case insensitive.
quantity	integer	Required	The number of shares (stock) specified in the equity order to be bought or sold.

Parameter	Data Type	Required/Optional	Description
reserveOrder	boolean	Optional	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
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is TRUE.
priceType	enum	Required	The type of pricing specified in the equity order. Options are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
marketSession	enum	Required	Session when the equity order will be place. There are two options for this parameter: • REGULAR • EXTENDED
orderTerm	enum	Required	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)
routingDestination	enum	Optional	The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO • ARCA • NSDQ • NYSE

Table 4-22 previewequityorder Return Values

Parameter	Data Type	Description
accountId	integer	The 8-digit account number assigned to the user.
allOrNone	enum	Specifies whether the shares designated in the order must be purchased all at once, or not at all. Options are: • TRUE • FALSE **Note**: This is case-insensitive
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedTotalAmount	double	The cost or proceeds, including broker commission, billed to the user to perform the requested action.
msgDesc	string	Text description of the message displayed to the user.
msgCode	integer	Numeric value associated with a system message delivered to a user in the response.
previewTime	long (UTC)	The time the order was displayed to the user.
previewId	integer	Numeric value associated with an order that allows users to ensure the same parameters for the preview order are input for the actual order.
quantity	integer	The number of shares of stock that will be bought or sold specified in the order.
reserveOrder	boolean	Value specified by the use in the order. Options are • TRUE • FALSE
reserveQuantity	integer	The number of shares displayed for a reserve order.
orderTerm	enum	Value specified by the user in the order for the term of the order. Options are • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
limitPrice	double	The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price.
stopPrice	double	The price at which a share is to be bought or sold if specified in a limit order.
symbolDesc	string	Text description of the stock being bought or sold.
symbol	string	The market trading symbol for the stock being bought or sold.

Parameter	Data Type	Description
orderAction	enum	User-specified action that instructs the broker what action to perform. Options are: • BUY • SELL • BUY_TO_COVER • SELL_SHORT
orderTime	long (UTC)	The time the order was submitted.
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
routingDestination	enum	The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO • ARCA • NSDQ • NYSE

Notes

See Also

`placechangeequityorder`, `placeequityorder`, `previewchangeequityorder`

placeequityorder

Description - submit an equity order request.

URL - *https://etws.etrade.com/order/rest/placeequityorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/placeequityorder*

HTTP Method - POST

Sample Request

This sample request shows the user submitting a request for an equity order to be processed. (These parameters should be included in the request header)

```
<PlaceEquityOrder xmlns="http://order.etws.etrade.com">
        <EquityOrderRequest>
           <accountId>83495799</accountId>
           <clientOrderId>45</clientOrderId>
           <limitPrice>3</limitPrice>
           <previewId></previewId>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity>4</quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <symbol>ETFC</symbol>
           <orderAction>BUY</orderAction>
           <priceType>LIMIT</priceType>
           <routingDestination></routingDestination>
           <marketSession>REGULAR</marketSession>
           <orderTerm>GOOD_FOR_DAY</orderTerm>
        </EquityOrderRequest>
```

Sample Response

```
<PlaceEquityOrderResponse>
   <equityOrderResponse>
      <accountId>83495799</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>7.99</estimatedCommission>
      <estimatedTotalAmount>19.99</estimatedTotalAmount>
      <messageList>
         <message>
            <msgDesc>Your order was successfully entered during market hours.</
```

```
msgDesc>
            <msgCode>1026</msgCode>
         </message>
      </messageList>
      <orderNum>13</orderNum>
      <orderTime>1269423948429</orderTime>
      <quantity>4</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <limitPrice>3</limitPrice>
      <stopPrice>0</stopPrice>
      <symbolDesc>E TRADE FINANCIAL CORP COM</symbolDesc>
      <symbol>ETFC</symbol>
      <orderAction>BUY</orderAction>
      <priceType>LIMIT</priceType>
   </equityOrderResponse>
</PlaceEquityOrderResponse>
</PlaceEquityOrder>
```

Parameters

Table 4-23 placeequityorder Request Parameters

Parameter	Data Type	Required/Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
quantity	integer	Required	The number of shares (stock) specified in the equity order to be bought or sold.
symbol	string	Required	The market trading symbol for the stock being bought or sold.
orderAction	enum	Required	User-specified action that instructs the broker what action to perform. Options are: • BUY • SELL • BUY_TO_COVER • SELL_SHORT
priceType	enum	Required	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
marketSession	enum	Required	Session when the equity order will be place. There are two options for this parameter: • REGULAR • EXTENDED
previewId	integer	Required	Numeric value associated with an order that allows users to ensure the same parameters for the preview order are input for the actual order.
orderTerm	enum	Required	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)
clientOrderId	string	Required	String value assigned to the order by the client. This must be a unique value that is 20 characters or less.

Parameter	Data Type	Required/Optional	Description
limitPrice	double	Optional*	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
stopPrice	double	Optional*	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
allOrNone	boolean	Optional	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE **Note**: This is case insensitive.
reserveOrder	boolean	Optional	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
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is true
routingDestination	enum	Optional	The exchange where a user wants to send orders to be executed. Exchanges that can be selected are: • AUTO • ARCA • NSDQ • NYSE

Notes

See Also

`placechangeequityorder`, `previewchangeequityorder`, `previewequityorder`

previewchangeequityorder

Description - preview change equity order request.

URL - *https://etws.etrade.com/order/rest/previewchangeequityorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/previewchangeequityorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required to enable a user to preview changes to an existing equity order before it is placed and processed. (These parameters should be included in the request header)

```
<previewChangeEquityOrder xmlns="http://order.etws.etrade.com">
        <changeEquityOrderRequest>
           <accountId>83495799</accountId>
           <orderNum>16</orderNum>
           <limitPrice></limitPrice>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity></quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <priceType></priceType>
           <orderTerm>GOOD_UNTIL_CANCEL</orderTerm>
        </changeEquityOrderRequest>
</previewChangeEquityOrder>
```

Sample Response

Once a user has requested to view the changes to an equity order before it is submitted, values are returned to the user, including account ID, the estimate commission, the estimated total amount to be collected for the equity order, and details about the equity order.

```
<PreviewChangeEquityOrderResponse>
   <equityOrderResponse>
      <accountId>83495799</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>7.99</estimatedCommission>
      <estimatedTotalAmount>19.99</estimatedTotalAmount>
      <messageList/>
```

```
        <orderNum>0</orderNum>
        <orderTime>1269430328285</orderTime>
        <previewTime>1269430496444</previewTime>
        <previewId>449548414022</previewId>
        <quantity>4</quantity>
        <reserveOrder>false</reserveOrder>
        <reserveQuantity>0</reserveQuantity>
        <orderTerm>GOOD_UNTIL_CANCEL</orderTerm>
        <limitPrice>3</limitPrice>
        <symbolDesc>E TRADE FINANCIAL CORP COM</symbolDesc>
        <symbol>ETFC</symbol>
        <orderAction>BUY</orderAction>
        <priceType>LIMIT</priceType>
    </equityOrderResponse>
</PreviewChangeEquityOrderResponse>
```

Parameters

Table 4-24 previewchangeequityorder Request Parameters

Parameter	Data Type	Required/Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
orderNum	integer	Required	Number assigned to the user for a request.
limitPrice	double	Optional*	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
stopPrice	double	Optional*	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
allOrNone	enum	Optional	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE **Note**: This is case insensitive.
quantity	integer	Optional	The number of shares (stock) specified in the equity order to be bought or sold.
reserveOrder	boolean	Optional	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
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is true

Parameter	Data Type	Required/Optional	Description
priceType	enum	Optional	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
orderTerm	enum	Optional	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)

Table 4-25 previewchangeequityorder Return Values

Parameter	Data Type	Description
accountId	integer	The 8-digit account number assigned to the user.
allOrNone	boolean	Specifies whether the shares designated in the order must be purchased all at once, or not at all.
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedTotalAmount	double	The cost and proceeds, including broker commission, billed to the user to perform the requested action.
msgDesc	string	Text description of the message displayed to the user.
msgCode	integer	Numeric value associated with a system message delivered to a user in the response.
previewTime	long (UTC)	The time the preview order was shown to the user.
previewId	integer	Numeric value associated with an order that allows users to ensure the same parameters for the preview order are input for the actual order.
quantity	integer	The number of shares of stock that will be bought or sold specified in the order.
reserveOrder	boolean	Value specified by the use in the order. Options are • TRUE • FALSE
reserveQuantity	integer	The number of shares displayed for a reserve order.
orderTerm	enum	Value specified by the user in the order for the term of the order. Options are • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
limitPrice	double	The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price.
stopPrice	double	Stop price specified in the option order.
symbolDesc	integer	Text description of the market symbol for the company.
symbol	string	The market trading symbol for the stock being bought or sold.

Parameter	Data Type	Description
orderAction	enum	Order action type specified in the order. Types that can be displayed are: • BUY • SELL • BUY_OPEN • SELL_OPEN
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE

Notes

See Also

`placechangeequityorder`, `placeequityorder`, `previewequityorder`

placechangeequityorder

Description - execute a change equity order request.

URL - *https://etws.etrade.com/order/rest/placechangeequityorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/placechangeequityorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required for a user to execute a change equity order request. (These parameters should be included in the request header)

```
<placeChangeEquityOrder xmlns="http://order.etws.etrade.com">
        <changeEquityOrderRequest>
           <accountId>83495799</accountId>
           <orderNum>14</orderNum>
           <clientOrderId></clientOrderId>
           <limitPrice>5</limitPrice>
           <previewId></previewId>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity></quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <priceType>LIMIT</priceType>
           <orderTerm></orderTerm>
        </changeEquityOrderRequest>
</placeChangeEquityOrder>
```

Sample Response

When a user executes a change equity order request, values returned to the user include the account ID, the estimated commission for the equity order, the estimated total cost for the equity order, and details about the equity order being placed.

```
<PlaceChangeEquityOrderResponse>
   <equityOrderResponse>
      <accountId>83495799</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>7.99</estimatedCommission>
      <estimatedTotalAmount>27.99</estimatedTotalAmount>
      <messageList>
         <message>
            <msgDesc>Your order was successfully entered during market hours.</
msgDesc>
            <msgCode>1026</msgCode>
         </message>
      </messageList>
      <orderNum>15</orderNum>
      <orderTime>1269424985364</orderTime>
      <previewTime>0</previewTime>
      <previewId>0</previewId>
      <quantity>4</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <limitPrice>5</limitPrice>
      <symbolDesc>E TRADE FINANCIAL CORP COM</symbolDesc>
      <symbol>ETFC</symbol>
      <orderAction>BUY</orderAction>
      <priceType>LIMIT</priceType>
   </equityOrderResponse>
</PlaceChangeEquityOrderResponse>
```

Parameters

Table 4-26 placechangeequityorder Request Parameters

Parameter	Data Type	Required/ Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
orderNum	integer	Required	Number assigned to the user for a request.
limitPrice	double	Optional	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
stopPrice	double	Required*	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
allorNone	boolean	Optional	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE **Note**: This is case insensitive.
quantity	integer	Optional	The number of shares (stock) specified in the equity order to be bought or sold.
reserveOrder	boolean	Optional	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
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is true

Parameter	Data Type	Required/ Optional	Description
priceType	enum	Optional	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
OrderTerm	enum	Optional	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)

Table 4-27 placechangeequityorder Return Values

Parameter	Data Type	Description
accountId	integer	The 8-digit account number assigned to the user.
allOrNone	enum	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedTotalAmount	double	The cost or proceeds, including broker commission, billed to the user to perform the requested action.
msgDesc	string	Text description of the message displayed to the user.
msgCode	integer	Numeric value associated with a system message delivered to a user.
orderNum	integer	Number assigned by the system for the order.
orderTime	long (UTC)	The time the order was received and processed by the system.
quantity	integer	The number of shares of stock that will be bought or sold, as specified in the order.
reserveOrder	boolean	Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are: • TRUE • FALSE
reserveQuantity	integer	The number of shares displayed for a reserve order.
orderTerm	enum	The order term specified by the user in the order. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
limitPrice	integer	The limit price specified by the user in the order.
symbolDesc	string	Text description of the stock being bought or sold in the order.
symbol	string	The market trading symbol for the stock being bought or sold.

Parameter	Data Type	Description
orderAction	enum	Order action type specified in the order. Types that can be displayed are: • BUY • SELL • BUY_OPEN • SELL_OPEN
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE

Notes

See Also

`placeequityorder`, `previewchangeequityorder`, `placechangeequityorder`

previewoptionorder

Description - preview an option order.

URL - *https://etws.etrade.com/order/rest/previewoptionorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/previewoptionorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required for a user to preview the details of an option order before it is placed. (These parameters should be included in the request header)

```
<PreviewOptionOrder xmlns="http://order.etws.etrade.com">
        <OptionOrderRequest>
           <accountId>83420939</accountId>
           <limitPrice></limitPrice>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity>1</quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <symbolInfo>
              <symbol>IBM</symbol>
              <callOrPut>CALL</callOrPut>
              <strikePrice>115</strikePrice>
              <expirationYear>2010</expirationYear>
              <expirationMonth>4</expirationMonth>
              <expirationDay>17</expirationDay>
           </symbolInfo>
           <orderAction>BUY_OPEN</orderAction>
           <priceType>MARKET</priceType>
           <routingDestination></routingDestination>
           <marketSession></marketSession>
           <orderTerm>GOOD_FOR_DAY</orderTerm>
        </OptionOrderRequest>
</PreviewOptionOrder>
```

Sample Response

When a user executes a preview option order request, values returned to the user include the account ID, the estimated commission for the option order, the estimated total cost to submit the option order, and details about the option order being placed.

```
<PreviewOptionOrderResponse>
   <optionOrderResponse>
      <accountId>83420939</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>8.74</estimatedCommission>
      <estimatedTotalAmount>1463.78</estimatedTotalAmount>
      <previewTime>1269430877665</previewTime>
      <previewId>449548422022</previewId>
      <quantity>1</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <optionSymbol>
         <symbol>IBM</symbol>
         <callOrPut>CALL</callOrPut>
         <strikePrice>115.000000</strikePrice>
```

```
            <expirationYear>2010</expirationYear>
            <expirationMonth>4</expirationMonth>
            <expirationDay>17</expirationDay>
        </optionSymbol>
        <orderAction>BUY_OPEN</orderAction>
        <priceType>MARKET</priceType>
    </optionOrderResponse>
</PreviewOptionOrderResponse>
```

Parameters

Table 4-28 previewoptionorder Request Parameters

Parameter	Data Type	Required/Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
symbolInfo - a required custom data type with the following required fields:			
symbol	string	Required	The market trading symbol for the option underlier.
callOrPut	enum	Required	Value that allows the owner of an option the right to either buy or sell an option. Options are: • CALL • PUT
strikePrice	double	Required	The price at which an option is exercised.
expirationYear	4-digit integer	Required	The year the option will expire, as specified in the option contract.
expirationMonth	integer (1-12)	Required	The month the option will expire, as specified in the option contract.
expirationDay	integer (1-31)	Required	The day the option will expire, as specified in the option contract.
orderAction	enum	Required	User-specified action that instructs the broker what action to perform. Options are: • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
limitPrice	double	Optional*	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.
stopPrice	double	Optional*	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.

Parameter	**Data Type**	**Required/Optional**	**Description**
allOrNone	boolean	Optional	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE **Note**: This is case insensitive
quantity	integer	Required	The number of contracts specified in the option order to be bought or sold.
reserveOrder	boolean	Optional	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
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is true
stopLimitPrice	double	Optional*	The price for the stop and limit sides of the order. **Note**: This value exists in response for STOP_LIMIT orders.
priceType	enum	Required	The type of pricing specified in the equity order. Options are: • MARKET • STOP • LIMIT • STOP_LIMIT

Parameter	Data Type	Required/Optional	Description
orderTerm	enum	Required	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)
routingDestination	enum	Optional	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

Table 4-29 previewoptionorder Return Values

Parameter	Data Type	Description
accountId	8-digit integer	The 8-digit account number assigned to the user.
allOrNone	enum	Specifies whether the shares designated in the order must be purchased all at once, or not at all. Options are: • TRUE • FALSE
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedTotalAmount	double	The cost or proceeds, including broker commission, billed to the user to perform the requested action.
previewTime	long (UTC)	The time the preview order was displayed to the user.
previewId	integer	Numeric value associated with an order that allows users to ensure the same parameters for the preview order are input for the actual order.
quantity	integer	The number of shares of stock that will be bought or sold specified in the order.
reserveOrder	boolean	Value specified by the use in the order. Options are • TRUE • FALSE
orderTerm	enum	Value specified by the user in the order for the term of the order. Options are • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
optionSymbol- a required custom field with the following fields:		
symbol	string	The symbol for the stock specified in the order.
callOrPut	enum	Specifies whether options are being bought (call) or sold (put), as specified in the option order. Options are: • CALL • PUT
strikePrice	double	The strike price for the option specified in the order.
expirationYear	integer	The year the option will expire, as specified in the option contract.
expirationMonth	integer	The month the option will expire, as specified in the option contract.
expirationDay	integer	The day of the month the option will expire, as specified in the option contract.

Parameter	Data Type	Description
orderAction	enum	Order action type specified in the order. Types that can be displayed are: • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • STOP • LIMIT • STOP_LIMIT
routingDestination	enum	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

Notes

See Also

`placechangeoptionorder`, `placeoptionorder`, `previewchangeoptionorder`

placeoptionorder

Description - submit an option order request.

URL - *https://etws.etrade.com/order/rest/placeoptionorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/placeoptionorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required to place an option order. (These parameters should be included in the request header)

```
<PlaceOptionOrder xmlns="http://order.etws.etrade.com">
        <OptionOrderRequest>
           <accountId>83420939</accountId>
           <clientOrderId>259</clientOrderId>
           <limitPrice>10</limitPrice>
           <previewId></previewId>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity>1</quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <symbolInfo>
              <symbol>IBM</symbol>
              <callOrPut>CALL</callOrPut>
              <strikePrice>115</strikePrice>
              <expirationYear>2010</expirationYear>
              <expirationMonth>4</expirationMonth>
              <expirationDay>17</expirationDay>
           </symbolInfo>
           <orderAction>BUY_OPEN</orderAction>
           <priceType>LIMIT</priceType>
           <routingDestination></routingDestination>
           <marketSession></marketSession>
           <orderTerm>GOOD_FOR_DAY</orderTerm>
        </OptionOrderRequest>
</PlaceOptionOrder>
```

Sample Response

When a user placed an option order, values returned to the user include the account ID, the estimated commission for the option order, the estimated total cost for the option order, and details about the option order being placed.


```
<PlaceOptionOrderResponse>
   <optionOrderResponse>
      <accountId>83420939</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>8.74</estimatedCommission>
      <estimatedTotalAmount>1008.78</estimatedTotalAmount>
      <messageList>
         <message>
            <msgDesc>Your order was successfully entered during market hours.</
msgDesc>
            <msgCode>1026</msgCode>
         </message>
      </messageList>
      <orderNum>257</orderNum>
      <orderTime>1269430628956</orderTime>
      <quantity>1</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <limitPrice>10</limitPrice>
      <optionSymbol>
         <symbol>IBM</symbol>
         <callOrPut>CALL</callOrPut>
         <strikePrice>115.000000</strikePrice>
         <expirationYear>2010</expirationYear>
         <expirationMonth>4</expirationMonth>
         <expirationDay>17</expirationDay>
      </optionSymbol>
      <orderAction>BUY_OPEN</orderAction>
      <priceType>LIMIT</priceType>
   </optionOrderResponse>
</PlaceOptionOrderResponse>
```

Parameters

Table 4-30 placeoptionorder Request Parameters

Parameter	Data Type	Required/Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
quantity	integer	Required	The number of contracts specified in the option order to be bought or sold.
symbol	string	Required	The market trading symbol for the contract being bought or sold.
callOrPut	enum	Required	Value that allows the owner of an option the right to either buy or sell an option.
strikePrice	double	Required	The price at which an option is exercised.
expirationYear	4-digit integer	Required	The year the option will expire, as specified in the option contract.
expirationMonth	integer (1-12)	Required	The month the option will expire, as specified in the option contract.
expirationDay	integer (1-31)	Required	The day the option will expire, as specified in the option contract.
orderAction		Required	Order action type specified in the option order. Types that can be displayed are: • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
priceType	enum	Required	The type of pricing specified in the option order. Options are: • MARKET. • LIMIT • STOP • STOP_LIMIT
orderTerm	enum	Required	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)
clientOrderId	string	Required	String value assigned to the order by the client. This must be a unique value that is 20 characters or less.

Parameter	Data Type	Required/Optional	Description
limitPrice	double	Optional*	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.
previewId	string	Optional	When using PreviewOptionOrder to obtain information on cost, commission, and other data, the preview ID is provided.
stopPrice	double	Optional*	The price at which stock is to be bought or sold if specified in a limit order. **Note**: This value exists in response for STOP orders.
allOrNone	enum	Optional	For all account types, this designates that all shares must be executed at once. This is similar to FILL_OR_KILL.
reserveOrder	boolean	Optional	Displays only a certain number of contracts in the market, even though the order is actually much larger.
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is TRUE.
stopLimitPrice	double	Optional*	The price for the stop and limit sides of the order. **Note**: This value exists in response for STOP_LIMIT orders.
routingDestination	enum	Optional	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

Table 4-31 placeoptionorder Return Values

Parameter	Data Type	Description
accountId	8-digit integer	The 8-digit account number assigned to the user.
allOrNone	enum	Specifies whether the contracts designated in the order must be purchased all at once, or not at all. Options are: • TRUE • FALSE
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedTotalAmount	double	The cost or proceeds, including broker commission, billed to the user to perform the requested action.
orderTime	long (UTC)	The time the order was submitted.
quantity	integer	The number of contracts of options that will be bought or sold specified in the order.
reserveOrder	boolean	Value specified by the use in the order. Options are • TRUE • FALSE
orderTerm	enum	Value specified by the user in the order for the term of the order. Options are • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
optionSymbol- a required custom field with the following fields:		
symbol	string	The symbol for the option underlier specified in the order.
callOrPut	enum	Specifies whether options are being bought (call) or sold (put), as specified in the option order. Options are: • CALL • PUT
strikePrice	double	The strike price for the option specified in the order.
expirationYear	integer	The year the option will expire, as specified in the option contract.
expirationMonth	integer	The month the option will expire, as specified in the option contract.
expirationDay	integer	The day of the month the option will expire, as specified in the option contract.

Parameter	Data Type	Description
orderAction	enum	Order action type specified in the option order. Types that can be displayed are: • BUY_OPEN • SELL_OPEN • BUY_CLOSE • SELL_CLOSE
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • STOP • LIMIT • STOP_LIMIT

Notes

See Also

`previewoptionorder`, `previewchangeoptionorder`, `placechangeoptionorder`

previewchangeoptionorder

Description - preview change option order request.

URL - *https://etws.etrade.com/order/rest/previewchangeoptionorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/previewchangeoptionorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required for a user to preview changes made to an existing option order. (These parameters should be included in the request header)

```
<previewChangeOptionOrder xmlns="http://order.etws.etrade.com">
        <changeOptionOrderRequest>
           <accountId>83420939</accountId>
           <orderNum>258</orderNum>
           <limitPrice>25</limitPrice>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity>4</quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <priceType>LIMIT</priceType>
           <orderTerm>GOOD_FOR_DAY</orderTerm>
        </changeOptionOrderRequest>
</previewChangeOptionOrder>
```

Sample Response

When a user executes a preview change option order request, values returned to the user include the account ID, the estimated commission for the option order, the estimated total cost to place the option order, and details about the option order being placed.

```
<PreviewChangeOptionOrderResponse>
   <optionOrderResponse>
      <accountId>83420939</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>10.99</estimatedCommission>
      <estimatedTotalAmount>10056.046</estimatedTotalAmount>
      <messageList/>
      <orderNum>0</orderNum>
```

```
      <orderTime>1269431139424</orderTime>
      <previewTime>1269432162454</previewTime>
      <previewId>449548438022</previewId>
      <quantity>4</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <limitPrice>25</limitPrice>
      <optionSymbol>
         <symbol>IBM</symbol>
         <callOrPut>CALL</callOrPut>
         <strikePrice>175.000000</strikePrice>
         <expirationYear>2010</expirationYear>
         <expirationMonth>10</expirationMonth>
         <expirationDay>16</expirationDay>
      </optionSymbol>
      <orderAction>BUY_OPEN</orderAction>
      <priceType>LIMIT</priceType>
   </optionOrderResponse>
</PreviewChangeOptionOrderResponse>
```

Parameters

Table 4-32 previewchangeoptionorder Request Parameters

Parameter	Data Type	Required/Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
orderNum	integer	Required	Number assigned to the user for a request.
clientOrderId	string	Optional	String value assigned to the order by the client. This must be a unique value that is 20 characters or less.
limitPrice	double	Optional*	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
stopLimitPrice	double	Optional*	The price for the stop and limit sides of the order. **Note**: required if priceType is STOP_LIMIT.
stopPrice	double	Optional*	The price at which a contract is to be bought or sold if specified in a limit order. **Note**: required if priceType is STOP
allOrNone	boolean	Optional	For all account types, this designates that all shares must be executed at once. Options are: • TRUE • FALSE
quantity	integer	Optional	The number of shares (stock) specified in the equity order to be bought or sold.
reserveOrder	boolean	Optional	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.

Parameter	Data Type	Required/Optional	Description
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note:** reserveQuantity is required if reserveOrder is True
priceType	enum	Optional	The type of pricing specified in the option order. Options are: • MARKET • STOP • LIMIT • STOP_LIMIT
orderTerm	enum	Optional	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)

Table 4-33 previewchangeoptionorder Return Values

Parameter	Data Type	Description
accountId	integer	The 8-digit account number assigned to the user.
allOrNone	boolean	Specifies whether the shares designated in the order must be purchased all at once, or not at all.
estimatedCommission	double	The cost billed to the user to perform the requested action.
estimatedTotalAmount	double	The cost and proceeds, including broker commission, billed to the user to perform the requested action.
orderNum	integer	Numeric value assigned to the order.
orderTime	long (UTC)	The time the order was submitted.
previewTime	long (UTC)	The time the preview order was shown to the user.
previewId	integer	Numeric value associated with an order that allows users to ensure the same parameters for the preview order are input for the actual order.
quantity	integer	The number of shares of stock that will be bought or sold specified in the order.
reserveOrder	boolean	Value specified by the use in the order. Options are • TRUE • FALSE
reserveQuantity	integer	The number of shares displayed for a reserve order.
orderTerm	enum	Value specified by the user in the order for the term of the order. Options are • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
routingDestination	enum	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

Parameter	**Data Type**	**Description**
orderAction	enum	Order action type specified in the order. Types that can be displayed are: • BUY • SELL • BUY_OPEN • SELL_OPEN
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE

Notes

See Also

`placechangeoptionorder`, `placeoptionorder`, `previewoptionorder`

placechangeoptionorder

Description - execute a change option order request.

URL - *https://etws.etrade.com/order/rest/placechangeoptionorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/placechangeoptionorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required for a user to place a change option order request. (These parameters should be included in the request header)

```
<placeChangeOptionOrder xmlns="http://order.etws.etrade.com">
        <changeOptionOrderRequest>
           <accountId>83420939</accountId>
           <orderNum>258</orderNum>
           <clientOrderId></clientOrderId>
           <limitPrice>10</limitPrice>
           <previewId></previewId>
           <stopPrice></stopPrice>
           <stopLimitPrice></stopLimitPrice>
           <allOrNone></allOrNone>
           <quantity>2</quantity>
           <reserveOrder></reserveOrder>
           <reserveQuantity></reserveQuantity>
           <priceType>LIMIT</priceType>
           <orderTerm>GOOD_FOR_DAY</orderTerm>
        </changeOptionOrderRequest>
</placeChangeOptionOrder>
```

Sample Response

```
<PlaceChangeOptionOrderResponse>
   <optionOrderResponse>
      <accountId>83420939</accountId>
      <allOrNone>false</allOrNone>
      <estimatedCommission>9.49</estimatedCommission>
      <estimatedTotalAmount>2054.53</estimatedTotalAmount>
      <messageList>
         <message>
            <msgDesc>Your order was successfully entered during market hours.</
msgDesc>
            <msgCode>1026</msgCode>
         </message>
      </messageList>
      <orderNum>260</orderNum>
      <orderTime>1269431139424</orderTime>
      <previewTime>0</previewTime>
      <previewId>0</previewId>
      <quantity>2</quantity>
      <reserveOrder>false</reserveOrder>
      <reserveQuantity>0</reserveQuantity>
      <orderTerm>GOOD_FOR_DAY</orderTerm>
      <limitPrice>10</limitPrice>
      <optionSymbol>
         <symbol>IBM</symbol>
         <callOrPut>CALL</callOrPut>
         <strikePrice>175.000000</strikePrice>
         <expirationYear>2010</expirationYear>
         <expirationMonth>10</expirationMonth>
         <expirationDay>16</expirationDay>
```

```
        </optionSymbol>
        <orderAction>BUY_OPEN</orderAction>
        <priceType>LIMIT</priceType>
     </optionOrderResponse>
  </PlaceChangeOptionOrderResponse>
```

Parameters

Table 4-34 placechangeoptionorder Request Parameters

Parameter	Data Type	Required/ Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
orderNum	integer	Required	Number assigned to the user for a request.
limitPrice	double	Optional	The price at which to buy an option at or below a certain price, or sell the option when it reaches a specified price. **Note**: required if priceType is LIMIT
previewId	integer	Optional	When using PreviewOptionOrder to obtain information on cost, commission, and other data, the preview ID is provided.
stopPrice	double	Optional*	The price at which a share is to be bought or sold if specified in a limit order. **Note**: required if priceType is STOP
allorNone	enum	Optional	For all account types, this designates that all contracts must be executed at once. Options are: • TRUE • FALSE **Note**: This is case insensitive.
quantity	integer	Optional	The number of options specified in the contract to be bought or sold.
reserveOrder	boolean	Optional	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
reserveQuantity	integer	Optional*	The number of shares displayed for a reserve order. **Note**: reserveQuantity is required if reserveOrder is TRUE.
stopLimitPrice	double	Optional*	The price for the stop and limit sides of the order. **Note**: required if priceType is STOP_LIMIT.

Parameter	Data Type	Required/ Optional	Description
priceType	enum	Optional	The type of pricing specified in the equity order. Options are: • MARKET • STOP • LIMIT • STOP_LIMIT
OrderTerm	enum	Optional	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)

Table 4-35 placechangeoptionorder Return Values

Parameter	Data Type	Description
accountId	integer	The 8-digit account number assigned to the user.
allOrNone	enum	Specifies whether the shares designated in the order must be purchased all at once, or not at all.
estimated Commission	double	The cost billed to the user to perform the requested action.
estimatedTotaAmount	double	The cost, including broker commission, billed to the user to perform the requested action.
msgDesc	string	Text description of the message displayed to the user.
orderNum	integer	Number assigned by the system for the order.
orderTime	long (UTC)	The time the order was received and processed by the system.
previewTime	long (UTC)	The time the order was displayed to the user.
previewId	integer	When using PreviewEquityOrder to obtain information on cost, commission, and other data, the preview ID is provided. This allows users to ensure the same parameters for the Preview Order are input for the actual order.
quantity	integer	The number of shares of stock that will be bought or sold, as specified in the order.
orderTerm	enum	The order term specified by the user in the order. Options are: • GOOD_UNTIL_CANCEL • GOOD_FOR_DAY • IMMEDIATE_OR_CANCEL (Only for Limit Orders) • FILL_OR_KILL (Only for Limit Orders)
routingDestination	enum	The exchange where a user wants to send orders to be executed. Users will want to perform this action if they believe they can get a better order fill by directing their order to a specific exchange, rather than relying on the automatic order routing system. Exchanges that can be selected are: • AUTO • AMEX • BOX • CBOE • ISE • NOM • NYSE • PHX
optionSymbol - a required custom field with the following fields: symbol, callOrPut, strikePrice, expirationYear, expirationMonth, expirationDay		

Parameter	Data Type	Description
symbol	string	The market trading symbol for the stock being bought or sold.
callOrPut	enum	Specifies whether options are being bought (call) or sold (put), as specified in the option order. Options are: • CALL • PUT
strikePrice	double	The strike price for the option specified in the order.
expirationYear	4-digit integer	The year the option will expire, as specified in the option contract.
expirationMonth	integer (1-12)	The month the option will expire, as specified in the option contract.
expirationDay	integer (1-31)	The day of the month the option will expire, as specified in the option contract.
orderAction	enum	Order action type specified in the order. Types that can be displayed are: • BUY • SELL • BUY_OPEN • SELL_OPEN
priceType	enum	The price type specified in the order. Types that can be displayed are: • MARKET • LIMIT • STOP • STOP_LIMIT • MARKET_ON_CLOSE
reserveOrder	boolean	Displays only a certain number of shares (reserve quantity) in the market, even though the order is actually much larger. Options are: • TRUE • FALSE
limitPrice	double	The price at which to buy a share at or below a certain price, or sell the stock when it reaches a specified price.
stopPrice	double	The price at which a share is to be bought or sold if specified in a limit order.

Notes

See Also

`placeoptionorder`, `previewchangeoptionorder`, `previewoptionorder`

cancelorder

Description - submit a cancel request for the indicated order.

URL - *https://etws.etrade.com/order/rest/cancelorder*

Sandbox URL - *https://etwssandbox.etrade.com/order/sandbox/rest/cancelorder*

HTTP Method - POST

Sample Request

The sample request below shows the REST-based syntax required for a user to cancel an order. (These parameters should be included in the request header)

```
<cancelOrder xmlns="http://order.etws.etrade.com">
        <cancelOrderRequest>
           <accountId>83420939</accountId>
           <orderNum>262</orderNum>
        </cancelOrderRequest>
</cancelOrder>
```

Sample Response

When a user cancels an existing order, values returned to the user to confirm the order has been cancelled include account ID, the order number, the time the order was cancelled, and a confirmation message stating that the cancel request was placed successfully.

```
<CancelOrderResponse>
   <cancelResponse>
      <accountId>83420939</accountId>
      <orderNum>262</orderNum>
      <cancelTime>1269434848649</cancelTime>
      <resultMessage>Cancel Request Placed Successfully</resultMessage>
   </cancelResponse>
</CancelOrderResponse>
```

Parameters

Table 4-36 cancelorder Request Parameters

Parameter	Data Type	Required/ Optional	Description
accountId	8-digit integer	Required	The 8-digit account number assigned to the user.
orderNum	long	Required	Number assigned to the user for a request.

Table 4-37 cancelorder Return Values

Parameter	Data Type	Description
accountId	8-digit integer	The 8-digit account number assigned to the user.
orderNum	long	Number assigned to the user for a request.
cancelTime	long (UTC)	The time the cancel order action was submitted.
resultMessage	string	Text description of whether the action was performed successfully or not.

Notes

See Also

`placechangeequityorder`, `placechangeoptionorder`, `placeequityorder`, `placeoptionorder`

Notifications

The E*TRADE API contains a system message returned to user that allows the user to retrieve a list of messages in the system. The following REST resource makes up the Notifications section of this chapter:

- `getmessagelist`

getmessagelist

Description - return a list of active system messages specified by a user.

URL - *https://etws.etrade.com/notification/rest/getmessagelist*

Sandbox URL - *https://etwssanbox.etrade.com/notification/sandbox/rest/getmessagelist*

HTTP Method - GET

Sample Request

The sample request below shows the REST-based syntax required for a user to retrieve a list of messages in the system.

```
notification/rest/getmessagelist
```

Sample Response

Once a user has made a request to retrieve a list of messages in the system, the values returned to the user include each message in the system with a description of the message and a subject line. Note that messages can include alert messages and info messages.

```
<getMessageListResponse xmlns="http://notification.etws.etrade.com/">
   <globalMessages>
      <alertMessages>
         <count>1</count>
      </alertMessages>
      <infoMessages>
         <count>2</count>
         <message>
            <description>UATTESTMESSAGE1</description>
            <subject>TESTTESTTEST</subject>
         </message>
```

```
        <message>
           <description>UATTESTMESSAGE</description>
           <subject>TESTTEST</subject>
        </message>
     </infoMessages>
     <noticeMessages>
        <count>1</count>
        <message>
           <description>UATTTESTMESSAGE1</description>
           <subject>TESTTEST13TT</subject>
        </message>
     </noticeMessages>
  </globalMessages>
  <platformMessages>
     <alertMessages>
        <count>1</count>
        <message>
           <description>Test Verification</description>
           <subject>Vinay</subject>
        </message>
     </alertMessages>
     <consumerKey>45e56f05d7fa1021e2c81d3c7a1d2e06</consumerKey>
     <infoMessages>
        <count>2</count>
        <message>
           <description>UATTESTTESTTESTMESSAGE1</description>
           <subject>TESTTESTTEST123TEST</subject>
        </message>
        <message>
           <description>UATTESTTTESTMESSAGE1</description>
           <subject>TESTTESTT123TEST</subject>
        </message>
     </infoMessages>
     <noticeMessages>
        <count>4</count>
        <message>
           <description>EXPLORATORY TESTING NEW</description>
           <subject>EXPLORATORY TESTING UPDATED</subject>
        </message>
        <message>
           <description>test message</description>
           <subject>JNHG</subject>
        </message>
        <message>
           <description>Test Platform message</description>
           <subject>test Platform message</subject>
        </message>
        <message>
           <description>UPDATEDTONEW</description>
           <subject>UPDATED</subject>
        </message>
     </noticeMessages>
  </platformMessages>
</getMessageListResponse>
```

Parameters

Table 4-38 getmessagelist Request Parameters

Parameter	Data Type	Required	Description
messageType	enum	Optional	Determines who can view the message. Options are: • GLOBAL - everyone • PLATFORM - specific to consumer key.
messageTier	enum	Optional	Denotes the priority of the message. Options are: • INFO -Low • NOTICE - Medium • ALERT - High
oauth_consumer_key	string	Required	The platform identifier.

Table 4-39 getmessagelist Return Values

Parameter	Description
count	The number of messages returned in the response.
description	A text description of the message.
subject	The header of the message.

Notes

See Also

`getmessagelist`

HTTP Status Codes

The following table shows the HTTP status codes and, if appropriate, additional subcodes, common to all resources. In addition to the codes, the table shows the accompanying status messages. Note that some of the same status codes may have different messages. The status message correlates with the specific resource.

Table 4-40 HTTP Status Codes

<table>
<tr><th>Code</th><th>Subcode</th><th>Applies To</th><th>Status Message</th></tr>
<tr><td>400</td><td>53</td><td>All Market resources</td><td>No data found.</td></tr>
<tr><td>400</td><td>1000 - 1050</td><td>All Market resources</td><td>Invalid [parameter]/Input has some type of problem.</td></tr>
<tr><td>400</td><td>9999</td><td>All Market resources</td><td>The service you requested is not available at this time. Please try again later.</td></tr>
<tr><td>400</td><td></td><td>All Accounts resources</td><td>Invalid [parameter]/Input has some type of problem.</td></tr>
<tr><td>400</td><td></td><td>All Order resources</td><td>Errors found.</td></tr>
<tr><td>401</td><td></td><td>All resources</td><td>The following status messages can be displayed:
• oauth parameters absent
• realm mismatch
• oauth_version mismatch
• invalid request token
• invalid signature method
• invalid consumer key
• invalid signature
• invalid access token
• invalid nonce</td></tr>
<tr><td>404</td><td>53 or 11</td><td>All Accounts resources</td><td>No data found.</td></tr>
<tr><td>404</td><td>1040</td><td>All Accounts resources</td><td>Invalid alert.</td></tr>
<tr><td>404</td><td>1050 or 1000</td><td>All Accounts resources</td><td>No input Account Ids.</td></tr>
<tr><td>500</td><td>9999</td><td>All Accounts resources</td><td>The service you requested is not available at this time. Please try again later.</td></tr>
</table>

Chapter 5: Notifications

Introduction

E*TRADE has implemented a push notification application to allow real-time market data and information to be published to clients using a Comet web architecture. The CometD web architecture allows web servers to send information to a browser without requiring the browser to make a request for the data. There are two methods that can be used by a web server to “push” data to a client:

- Polling
- Long Polling

Although these options are available in Comet, the E*TRADE web server uses long polling to send data to the client. In long polling, the data request sleeps on the server until data arrives or a timeout occurs, instead of sending an empty response. Once data arrives, the sleeping request becomes active again and sends a complete response to the client. The connection for the request is re-established as soon as the response is sent to a client or if a request times out, thereby significantly reducing latency.

> **Note:** If a customer logs in twice and uses same cookies/session, then CometD will using polling.

CometD can implement user interactions for web clients using Ajax and a server-push technique called Comet. To define the protocol used to exchange information between the client and server, a specification called Bayeux defines the interoperability aspects of this implementation. CometD implements the Bayeux protocol in both Java and Javascript; however, E*TRADE uses Javascript for its server-push method.

For more information on both the Java and Javascript implementations, see:

Javascript: *http://cometd.org/documentation/cometd-javascript*

Java: *http://cometd.org/documentation/cometd-java*

The following table lists the CometD implementations and clients

Table 5-1 CometD Bayeux Implementations

Language	Client
javascript	cometd-javascript
java	cometd-java
perl	
python	

For more information on the CometD web architecture, refer to the following documentation:

http://cometd.org/documentation/cometd

Push Notifications

Push notifications allow users to receive real-time notifications of order status and other order details and data.

The following return requirements are used for push notifications:

- For each order update only the order status for that order will be pushed.
- Order status will be pushed for accounts of the user's that they have requested
- The return will include order details, as shown in Table 1.

Table 5-2 Push Notification Return Order Details

Data Field	Description
Account Number	Account Number.
Order Number	Sequential order number, unique within account.
Order Status	Order status: open, pending, executed, cancelled, cancel-requested, etc.
Symbol	Security symbol.

Table 5-2 Push Notification Return Order Details

Data Field	Description
Order	Options are: BUY SELL SHORT COVER BUY_TO_OPEN SELL_TO_CLOSE SELL_TO_OPEN BUY_TO_CLOSE
Quantity	Quantity ordered. For mutual funds, can be $ value or “All I Own.”
Quantity Filled	Quantity that has been executed.
Price Type	Options are: • Market • Market_On_Close • Limit • Stop • Stop_Limit
Price	Price for Price Type
Event Type	See *Push Events* listed below.

Push Events

Table 2 lists Push Events used as part of Push Notifications.

Table 5-3 Push Events

Event	Description
Live	Used for orders that have been accepted by the Exchange.
Partial Execution	for orders that have been partially executed (e.g. the status is live but the quantity executed has changed)
Completed	for orders that have been filled
Cancelled	for orders that have been cancelled
Killed	for orders that [were partially filled but] have been cancelled by the Exchange
Rejected	for orders that have been rejected
Open	Submitted orders waiting on a condition to execute (for trailing stops, etc)

Complex options are treated differently in Push Notifications, with the following options available:

- For Buy Writes the order status update will show both legs of the orders, e.g. for partial executions the quantity filled for both the stock and the option will be pushed.
- Each leg of the order will still be pushed as the update is returned from the market, the legs can be pushed separately.
- For Spreads the order status update will show both legs of the orders e.g. for partial executions the quantity filled for both legs will be pushed.
- Each leg of the order will still be pushed as the update is returned from the market, the legs can be pushed seperately.

Streaming

The E*TRADE Streaming Framework provides real-time market data and information to customers. Data and information that can be “pushed” to a customer can include:

- Market index information (NYSE, NSDQ, etc.)
- Quote information (GOOG, AAPL, MSFT)
- Company news and information

The E*TRADE Streaming infrastructure implements a publish/subscribe and point-to-point model which allows a client to subscribe to multiple channels in order to receive new data when it is published. In this model, the client is not required to send a request to the server, but rather, wait for new data to be published to the channel, and the client does not need to download any thick Java clients or plug-ins because HTTP-based streaming is utilized to leverage CometD browser technology and the Bayeux protocol.

HTTP streaming can best be described as a method of utilizing HTTP connections to provide low-latency connections between web servers and browsers. Using a streaming method known as “long polling,” a web server “pushes” new data to clients, which can be a browser or standalone Java program. Long polling allows a request to remain on a server until new data arrives or a timeout limit is reached. Once new data arrives, the dormant AJAX request becomes active and sends the complete response to the client, which in turn causes the connection to be re-established.

HTTP Streaming

Streaming is based on the CometD server-push technology and its related Bayeux specification to define user interactions for web clients using Ajax calls. By defining the protocol used to exchange information between client and server, standard APIs for both clients and servers can be developed based on either Java or Javascript (E*TRADE's streaming framework utilizes Javascript). Before going into greater detail about the specifics of the E*TRADE streaming framework, it is important to understand the underlying technologies used to push data to a browser.

Streaming Technologies

The E*TRADE Streaming Framework takes advantage of two core web services technologies to publish content to channels:

- AJAX
- CometD (w/ Bayeux Protocol)

AJAX

Ajax can be defined as a broad group of web technologies used to implement a web application which can then communicate with a server in the background without interfering with the state of a web page. Ajax technologies include:

- HTML or XML, and CSS – used to present information on a page
- DOM – used to display and interact with data
- JSON (Javascript Object Notation) – used for data interchanges
- Javascript – used to bring all these technologies together

Residing on the client side, Ajax is most often used to create interactive web applications. In E*TRADE's use of Ajax, the client browser calls a specific web page to retrieve data so the page being currently displayed is updated with the latest information. For example, an Ajax call may be made to a company web page to ensure the user viewing the web page has access to the latest information about that company

CometD

E*TRADE uses Comet to push data to a browser. In order to initiate communication with the server, a Bayeux "handshake" is used, which negotiates the type of long poll transport that should be used. The client is then informed, with detailed timings of the long poll requests. One of the advantages of using long polling transport is that long polling-based CometD interactions are simple to debug and ensure that the HTTP connection is not broken, allowing developers a simple method to view long polling statistics and results. Also, because CometD supports industry-standard web technologies like Jetty, Tomcat, Weblogic, Websphere and others, developers have various options when developing web applications. CometD also provides a publication-subscription model which allows users to subscribe to specific channels where information is published. Using the publish() command, information can be published to a channel as long as a handshake has been initiated. There are two channels used in the streaming framework.

Client-side steps for streaming

The CometD web architecture is based on the Bayeux protocol which allows the client to publish, subscribe and un-subscribe tasks to a Bayeux enabled server.

Java Script Cometd client implementation

The cometd.init (URI) requires a URI *https://etwspush.etrade.com/apistream/cometd/oauth/* as its first argument and oauth header as the second argument. This call directs all Bayeux communication to the comet enabled server.

Sample

```
_cometd.init({url: _cometUrl, requestHeaders: oauthHeader});

Where _cometUrl = "https://etwspush.etrade.com/apistream/cometd/oauth/" and
oauthHeader = OAuth Authorization header

The program listings below show examples of how it is accomplished this
_cometd = $.cometd; // Uses the default Comet object
_cometUrl = cometUrl;

_metaSubscriptions = [];
$.each(_metaSubscriptions, function(index, subscription)
{
_cometd.removeListener(subscription);
});
_metaSubscriptions.push(_cometd.addListener('/meta/handshake', this,
_metaHandshake));
_metaSubscriptions.push(_cometd.addListener('/meta/connect', this,
_metaConnect));
var oauthHeader= {Authorization: header}
_cometd.init({url: _cometUrl,
requestHeaders: oauthHeader});
```

The following sample call back function is used for a handshake and connect:

Sample

```
function _metaHandshake(message)
{
  _handshook = message.successful;
  _connected = false;
}
function _metaConnect(message)
{
        var wasConnected = _connected;
        _connected = message.successful;
        if (wasConnected)
        {
            if (_connected)
            {
                // Normal operation, a long poll that reconnects
            }
            else
            {
                // Disconnected
            }
        }
        else
        {
            if (_connected)
            {
            // Connected
            baseSubscribe();
            if(_firstConnection){
            _cometd.publish("/service/etws/join", { type:'join'});
            }else{
            _cometd.publish("/service/etws/join", { type:'re-connect'});

            }
                _firstConnection = false;
            }
            else
            {
                // Could not connect
            }
        }
    }
```

Sample

```
function baseSubscribe(){
  _cometd.startBatch();
channel = _cometd.subscribe("/etws/error", this, error);
channel = _cometd.subscribe("/service/etws/error", this, error);
channel = _cometd.subscribe("/service/etws/orderupdate", this, orderUpdate);
        _cometd.endBatch();
}
```

The following is a sample callback for the *error* and *accountupdate* channels:

Sample

```
    function error(message)
    {
//Handle error
    }
    function orderUpdate(message)
    {
//Handle  order status update message
    }
```

Channels

Subscribing To A Channel

In order to receive notifications, alerts, and other data pushed to a client, a client should subscribe to the following channels:

- */etws/error (not used by server side)*
- */service/etws/error (not used by server side)*
- */service/etws/redirect (not used by server side)*
- */service/etws/accountupdate*

Publishing To A Channel

The client also needs to publish to */service/etws/join* with either of the following types listed in Table 5-4.

Table 5-4 Channel Publishing Types

Type	Description
join	Sample `cometd.publish("/service/etws/join", { type:'join'});`
reconnect	Sample `cometd.publish("/service/etws/join", { type:'re-connect'});`

Adding/Subscribing To An Account

To add/subscribe an account, the client needs to publish to */service/etws/accountlisten*. The client then needs to send the account information with the name accounts and value as single account or a comma-separated list of accounts

Sample

```
cometd.publish("/service/etws/accountlisten", {accounts:123,456,789});
cometd.publish("/service/etws/accountlisten", {accounts:123,456,789,123,456,789});
```

Removing/Unsubscribing An Account

To remove/unsubscribe an account, the client needs to publish to */service/etws/accountunlisten*. The client then needs to send the account information with the name accounts and value as single account or a the comma separated list of accounts.

Sample

```
cometd.publish("/service/etws/accountunlisten", {accounts:123,456,789});
cometd.publish("/service/etws/accountunlisten", {accounts:123,456,789,123,456,789})
```

Trailing Stops and Conditional Orders

For trailing stops and conditional order, an order status is sent if those orders are triggered and executed. If a trailing stop is triggered, an order is sent to the market to execute (triggered orders are market orders) and the push notification alerts the user once that order is executed (whether fully or a portion of the shares).

When orders are cancelled, for example 1-Cancels-All orders, or when a user cancels a conditional order, the order is cancelled within the E*TRADE system and not at the exchange. Thus, no update message is sent.

Note that Live and Open are not valid statuses. The only push events are when there is an execution (full or partial) and cancellations and rejections from the exchange.

Glossary

Access Token - Token provided to the Consumer by the Service Provider after the Request Token has been validated and verified.

Authentication - The process of authorizing a user and granting access to system resources.

Buy - Buys the number of shares specified in an order.

Buy To Cover - Buys the required number of shares needed to cover a user's position if the user is "shorting."

Consumer Key - A string value used by a consumer to identify itself to a service provider.

Equity Order - A request to a broker to either buy or sell equities.

Exchange - The exchange where a user wants to send orders to be executed.

Extended - A market session time that designates the exchange is open from 8am to 8pm. These exchanges manage orders pre-market and post-market.

Fill or Kill - Fills an equity or option order completely or "kills" the order so it isn't divided into multiple lots of being filled over a period of time.

Good For Day - Fills the order only for the specific day noted in an equity or option order.

Good Until Cancel - Fills the order until the user cancels an equity or option order.

Immediate or Cancel - Fills the order in full immediately or cancels an equity or option order.

Limit Price - The price at which to buy a share at or below a certain price, or sell the share when it reaches a specified price.

OAuth - Authentication protocol used to manage user access and authentication.

Option Order - A request for a broker to either buy or sell options.

Rate Limiting - The process of limiting the number of requests process by the E*TRADE server.

Regular - A market session time that designates that the exchange is open from 9:30am to 4:00 EST. This is the typical time when orders are executed.

Request Token - Token issued to a Consumer after the consumer key and signature have been verified by the Service Provider.

Reserve Order - The number of shares displayed in the market, even though the order is much larger.

Reserve Quantity - The number of shares displayed for a reserve order.

Sell - Sells the number of shares specified in an order.

Sell Short - Sells the required number of shares needed to ensure the user gains more money from the sale of borrowed shares.

Stop Limit Price - The price for the stop and limit sides of an order.

Stop Price - The price at which a share is to be bought or sold if specified in a limit order.

Strike Price - The price at which a stock option can be exercised.

Symbol - The market trading symbol for a stock.

Token Secret - A unique string value associated with a Request or Access Token. This Token Secret must match the Token.

Appendix A: Error Messages

Introduction

Account Services Error Messages

Table 1 lists the various Account Services Error Messages that may be returned to the user by the system.

Table 1 Account Services Error Messages

Error Code	Error Message
53	No Data Found.
11	No Data Found.
600	Alert has been deleted.
1000	Invalid Account Number: Null value passed for account no.
1001	Invalid Account Number: Invalid length.
1002	Invalid Account Number: Invalid format.
1004	Invalid Symbol.
1005	Invalid expiration date.
1006	Invalid Strike Price. Its either missing or has invalid value.
1007	Invalid Call/Put. Its either missing or has invalid value. Allowed values are CALL or PUT.
1008	Invalid Type Code. Its either missing or has invalid value. Allowed values are EQ, OPTN, INDX, MF, FI.
1009	Invalid Count. Count value should be between 1 and 25.
1010	Invalid marker. Maximum allowed length is 32 chars.

Table 1 Account Services Error Messages

Error Code	Error Message
1011	No positions matching the symbol you have entered can be found.
1012	Specified account does not have any positions.
1013	Invalid User Id: Null value passed for UserId.
1014	Invalid User Id: Invalid length.
1015	Invalid User Id: Invalid format.
1016	Invalid Account no: Negative value passed for account no.
1017	Invalid Alert Id.
1018	Alert cannot be deleted. Please try again later.
1019	Symbol is missing. Please specify symbol along with type code.
1020	Account Number does not exist for user.
1040	Invalid Alert.
6	Invalid parameters passed. Please verify input values.
9999	The service you requested is not available at this time. Please try again.

Market Services Error Messages

Table 2 lists the various Market Services error messages returned to a user by the system.

Table 2 Market Services Error Messages

Error Code	Error Message
53	No Data Found.
1002	Invalid ReqId Format.
1003	Invalid Expiration Date.
1004	Invalid Spread Date.
1005	Invalid Price Type.
1006	Invalid Strike Price.
1008	Invalid Symbol.
1010	Invalid UserId.
1011	Invalid ReqId Format.
1012	Invalid Option Type.
1013	Invalid Security Type.
1014	The request has no symbols.
1015	A request may contain up to 25 quotes.
1016	The user has not signed the real time quote agreement.
1017	The symbol specified is invalid.
1018	The company or symbol specified is invalid.
1019	The input data is invalid. Please verify.
1020	Invalid company name.
1021	Invalid add adjusted flag.
9999	The service you requested is not available at this time. Please try again.

Order Services Messages

Table 3 lists the various Order Services error messages returned to a user by the system.

Table 3 Order Services Error Messages

Error Code	Error Message
1003	This does not appear to be a valid stock symbol. Please make sure that you have entered the symbol correctly.
1005	This symbol is not recognized. Please make sure that you have entered the symbol correctly.
1010	There is currently a 500 share or 10% of volume maximum share quantity allowed in your account. Please adjust the share quantity you entered, or contact Customer Service at 1-800-ETRADE-1 (1-800-387-2331) for assistance.
1028	You have an existing open order for this security on the same side of the market.
1039	You have entered a number that exceeds the number of shares you hold in the security you want to trade. Please go back and enter a number either equal to or less than the number of shares you hold.
1041	Your sell short order cannot be processed. Either you do not have a margin account or short sales in this security are not allowed.
1044	Your sell short order cannot be processed. Short sales in this security are not allowed, as we were unable to borrow the shares.
1046	The security for which you have placed an order is on the E*TRADE Securities restricted list.
1051	E*TRADE allows only limit buy orders on Bulletin Board securities. Please go back and enter a valid limit price.
1068	Because the market is now closed, this order will not be reviewed until the morning prior to market open on next regular trading day.
1100	Null value specified.
1101	Invalid format.
1102	Invalid account specified for the user.
1103	Invalid count specified. Count should be between 0 and 25.
1104	Invalid markers. Please provide either beginMarker or endMarker not both.
1105	Invalid Account no: Negative value passed for account no.
1106	Invalid User Id: Null value passed for UserId.
1107	Invalid User Id: Invalid format.
2019	The symbol you entered does not appear to be valid. Please make sure you've entered the symbol correctly.
2027	This account is not approved for options trading.
2078	Orders entered via the Web site must be for whole shares. Your order has been rounded down to the nearest whole number.
2086	This security is not eligible for new advanced orders at this time.

Error Code	Error Message
2093	This symbol does not appear to be a stock symbol. Please make sure that you have entered the valid symbol correctly.
2097	Important! NYSE Rule 431 Notice: Because you are currently subject to a day trading minimum equity call, your account will be restricted to cash only transactions for 90 days if this day trade executes.
2098	This order to purchase securities, if accepted and executed, will be paid for out of proceeds from the previous sale of securities in your account, which is to settle on or before the date by which payment for this purchase transaction is due. If you subsequently enter an order to sell these securities before they are fully paid for, your account may be subject to a 90-day restriction under federal securities regulations.
2099	Because these securities were purchased with the proceeds of the sale of another security in your account, the order to sell these securities, if executed prior to the settlement date of the previous sale, may subject your account to a 90-day restriction under federal securities regulations.
3002	“This account is not approved for Level 2 or 3 options trading.
3003	“Please note that this option contract does not equal 100 shares per contract.
3006	Naked index options are not permitted at E*TRADE Securities LLC.
3007	You are not approved for naked calls.
3008	This account is not approved for Level 3 options trading.
3011	We cannot accept this order until your sell order for underlying stock is canceled at your current option trading level.
3015	This account is not approved for Level 3 options trading.
3023	“The ratio of contracts for this order is not one-to-one.
3024	The ratio of shares to contracts for this order is not one-to-one.
7046	You may not place a Sell Short, Market on Close order.
36103	The ratio of shares to contracts for this order is not one-to-one.
36104	No Records Found.
36105	The ratio of contracts for this order is not one-to-one.
7508	This order cannot be accepted because a Reserve Order can be routed to ECNs only. Please select ARCA, INET, or NSDQ for the market center.
100	Only GTC and Day orders are allowed with All or None.
200	All Or None, Quantity should be greater than equal to 300.
300	Only Limit and Stop Limit orders are allowed with All or None.
400	Select either Reserve Order and All or None.
500	The Quantity for Reserve Order should be greater than or equal to 1000.
600	For Reserve Order the Order should be a LIMIT Order.
700	Invalid Reserve Quantity.
800	Invalid Client OrderId.
900	Invalid Preview Id.

Error Code	Error Message
520	Invalid Routing Destination Specified.
530	Invalid AON/Order Term combination.
39999	For sandbox testing - order number is inconsistent with security type.
40000	Order number is either missing or invalid. Please specify valid order number.
630000	Sent too many requests at the same time.
41000	This order cannot be changed using this Application, it can only be cancelled. To change please log into the E*TRADE website.
640000	Invalid expiration date. Please check the input values for expirationYear, expirationMonth and expirationDay.
600000	Please specify Option Type (Call/Put).
610000	Invalid strike price specified.
620000	Invalid expiration year specified.
650000	Invalid All Or None specified.
660000	Invalid Reserve order specified.

System Messages

Table 4 lists the various System error messages returned to a user by the system.

Table 4 System Error Messages

Error Code	Error Message
3000	Failed to get messages.
4000	Invalid Consumer Key.
5000	Enter the subject of the message.
6000	Enter the message description.
7000	Message Id is missing or invalid Message Id entered.
8000	Message Tier is missing or invalid Message Tier entered.
9000	Message Type is missing or invalid Message Type entered.
10000	Create By is missing.
11000	Update By is missing.
12000	Status is missing or invalid Status entered.
13000	Global messages cannot have same subject.
14000	Platform messages cannot have same subject.
15000	No Data Found for given Message Id.