Trading Platform API Reference

Table of Contents

1 Introduction

Trading Platform
API Reference

© TD Ameritrade Think Tech
6940 Columbia Gateway Drive • Suite 200
Columbia, MD 21046-2788
Phone 443.539.2100 • Fax 240.568.5778

TD Ameritrade Trading Platform API Reference
Notice of Non-Liability

TD Ameritrade Think Tech and the authors assume no liability for errors, omissions, or damages, resulting from the use of this manual or information contained within this manual.

Market volatility, volume and system availability may delay account access and trade executions. Prices change quickly in fast market conditions, resulting in an execution price different from the quote displayed at order entry. If you experience difficulties, please contact Client Services, available 24 hours a day, 7 days a week (excluding market holidays).

1.1 Preface

This Document

This document, *Trading Platform API Reference*, describing the TD Ameritrade Trading Platform Application Programming Interface (TDAPI) and functions, is provided for use by third-party developers. It specifies an open, standard interface that can be implemented on any platform using any programming language.

Intended Audience

The document is written for programmers well versed in stock market research and trading and with expert knowledge of web application programming such as: XML Schema Definition (XSD) language, XML data parsing and Representational State Transfer (REST) architectural style. In addition, this release of the document defines a proprietary response data stream for some functions the receiving application must interpret.

Organization of This Document

Development Considerations presents high-level information about the purposes of the API, supported development platforms, and usage limits.

Programming Guide describes the architecture of the API request/response protocols and formats, security features, and data characteristics, such as character encodings, currency, date/time formats, and other information.

The reset of the sections describe individual TD Ameritrade request/response pairs for a variety of business functions.

See Appendix A: URL Encoding for additional information about how to encode parameter strings required by functions described in this API reference.

1.2 Typeface Conventions

Notational Conventions

Typefaces and text effects are used to identify text characteristics. These styles and the characteristics they imply are described below:

Typeface Conventions

FONT/STYLE	HOW USED
	Chapter Title
Arial	Used to denote major sections within a chapter
Arial	Normal text Description text in tables
Verdana	Literal examples in text Code samples
Verdana Bold	Literal values presented to user Variable and example data Titles within major sections of chapters
#Verdana Bold#	Variable value required from user, # is not passed in the URL
XXXXX	Comments in pseudo code
Garamond	Hyperlink to an external source email address

1.3 Revision History

REVISION #	DATE	DESCRIPTION
2.90	June 10,2010	DF - Added GTC to Complex Options and Complex Option orders to Edit
2.89	April 22, 2010	JM - Updated Snapshot Quote request - Now max 300 symbols per request
2.88	April 14, 2010	JM - fixed MultiLegOption docs.
2.87	March 8, 2010	JM - changed all references to root symbol to be underlying symbol
2.86	February 6, 2010	JM - Fixed BinaryOptionChain samples Fixed samples of new option symbols - they were shown with wrong date format
2.85	January 24, 2010	JM - Added BinaryOptionChain samples
2.84	November 19, 2009	JM - Modified ACCT_ACTIVITY * Added - ShortDescription * Added - SymbolUnderlying * Added - more possible values for ComplexOrderType JM - Changed - ACTIVES - subscription for OPTIONS changed JM - Added REACHED_SYMBOL_LIMIT error code for streaming JM - Updated BinaryOptionChain documentation - boolean values were defined incorrectly. Added range parameter.
2.83	October 28, 2009	RJ - Updated OptionChain, BinaryOptionChain and SavedOrders section Added advanced-margin tag to LogIn
2.82	August 28, 2009	RJ - Added MultiLegOptionTrade
2.81	July 28, 2009	RJ - Updated OrderStatus Added QuoteDelayed Added MarketOverview Added News Added FullStoryNews Added QuoteNews
2.80	May 20, 2009	JM - Added Symbol Lookup Added delimiters to end of STREAMER SERVER responses
2.79	May 08, 2009	JM - OrderStatus - changed Expiration to XAM and XPM for Extended AM and PM values (previous stuff was not working) JM - OrderStatus - Added underlying parameter and corresponding response field JM - Snapshot quote on options now has expiration-day
2.78	April 21, 2009	JM - EditOrder - removed moc from expire choices and updated validation rules for expire parameter.
2.77	April 15, 2009	JM - Removed some Service IDs from Streaming docs

2.76	April 01, 2009	JM - Updated BinaryOptionChain - quote info on underlying symbol
2.75	March 24, 2009	JM - ACCT_ACTIVITY - added complex option related fields
2.74	March 20, 2009	JM - Added docs for Streaming - OPTION - fields 20-27,31
2.73	March 11, 2009	JM - Added MOC to OrderStatus - SESSION tag docs
2.72	March 5, 2009	JM - Added BinaryOptionChain Added login-time to LogIn response Added - put-call tag for Quote response on Option symbols
2.71	Feb 20, 2009	JM - fixed OrderStatus definition - had reference to oe-strategy. Should be just "strategy"
2.70	January 29, 2009	JM - fixed docs for CHART DATA - SNAPSHOT History - 4 bytes in beginning of the record were not documented
2.69	January 26, 2009	JM - Fixed some STREAMER SERVER Response docs
2.68	December 8, 2008	JM - Fixed docs for ACCT_ACTIVITY - Special Instructions section
2.67	November 22, 2008	JM - Added SAVINGS to balances and positions
2.66	October 18, 2008	JM - Added docs for MONOPOLIZE streaming service ID
2.65	October 08, 2008	JM - Added History Sub-Type documentation
2.64	September 12, 2008	JM - Fixed LogIn response doc - timeout is in MINUTES, not seconds JM - Added Watchlist management commands
2.63	August 20, 2008	JM - Fixed ending delimiter in NEWS_HISTORY data response
2.62	August 18, 2008	JM - Added complex option chain parameters
2.61	August 08, 2008	JM - Added trailing-activation-price for OrderStatus docs
2.60	July 11, 2008	JM - Fixed News History docs
2.59	June 04, 2008	JM - Added Positions - suppressquotes parameter and QUOTE section documentation. Added SavedOrders and related functions. Added notes for Streaming Server symbol conversions under Streaming Data/Request Format Added accountid to StreamerInfo docs Added OPEN and LAST-TRADE-DATE to Index snapshot quotes Added executedDate on History records Fixed docs for NEWS_HISTORY Updated Streaming Data ACTIVES docs - better descriptions Added range parameter to OptionChain request
2.58	April 10, 2008	JM - Fixed streaming Heartbeat message definition - no FF and 0A delimiters on the end

2.57	April 08, 2008	JM - Added History command documentation (Transaction History for the account)
2.56	March 30, 2008	JM - Added Maintenance Requirement and Put/Call indicator on positions Added MessageKey command and corresponding streaming Account Activity request (not yet released) Revised Streamer Server Response docs
2.55	February 29, 2008	JM - Added CDI and display-name tags to LogIn and
2.54	February 04, 2008	JM - Updated StreamerInfo docs with session timeout info
2.53	January 17, 2008	JM - option chains - added some fields that were not documented
2.52	January 15, 2008	JM - updated Streaming Data docs - can now use POST to submit parameters
2.51	January 14, 2008	JM - fixed field descriptions for Streaming Chart Bars response
2.50	December 31, 2007	JM - Updated OrderStatus examples to show multiple partial fill situations
2.49	December 26, 2007	JM - Added lists of possible responses to OrderStatus for routes and status tags
2.48	December 06, 2007	JM - Added record terminators to various streaming data records
2.47	December 03, 2007	JM - Added PriceHistory and Volatility history documentation. Removed "compressed" references in Level II response
2.46	November 24, 2007	JM - Added New Level II datafeed documentation and updated references to the old one
2.45	October 24, 2007	JM - Cleaned up some issues with EditOrder and use of accountid outside of orderstring
2.44	October 19, 2007	JM - Corrected mixed up samples for various complex options commands
2.43	September 18, 2007	JM - added quantity2 to various complex option Trade commands. Added OptionChains Added ComboOptionTrade Added exchange related info to Login response
2.42	August 16, 2007	JM - added Complex Option and Conditional Order documentation
2.41	July 13, 2007	JM - fixed CHART Data (SNAPSHOT) doc. Was not documenting 6 bytes
2.40	June 18, 2007	JM - Rewrite for the "1.0" New XML release
2.30	March 12, 2007	Added session control documentation to the Programming Guide (Ch. 2) Added FILL related tag docs to OrderStatus section Added docs for more option specific Streamer datafeed fields (32 to 36)
2.11	February 5, 2007	Updated Login XML documentation Updated options parsing Updated Most Actives documentation Updated News data documentation

2.10	November 3, 2006	Added "source" as an input parameter to Quotes, Positions, Balances and Positions and Order Status API. Replaced Chapter 11 – Login API with XML format response. Response fields are mapped to previous fields extracted from HTML formatted response.
2.00	August 24, 2006	Maintenance revision (see Version 1.1.0 Changes)
	July 17, 2006	Initial publication

Questions and Suggestions

If you have questions or suggestions regarding this document, or need assistance implementing the API, please email us at API@TDAmeritrade.com

2 Development Considerations

2.1 General Considerations

A central tenet of Web services is the use of standardized protocols. By supporting this pervasive set of standards (REST for initiating requests and XML for results, layered on a foundation of standard Internet protocols), TD Ameritrade reduces the amount of work that a developer must do to integrate with the TD Ameritrade Trading Platform and allows the use of a wide variety of existing trading and research functions.

An important and useful property of REST is that the developer can use the web browser as a development and prototyping tool for functions returning XML streams. Without writing any code, the developer can enter a request URL into the browser address bar and click "Submit" (or equivalent) to make a request. The browser then creates an HTTPS GET request to the TDAPI server and displays the result. If the developer is using Internet Explorer, the XML data returned by TDAPI is displayed in a convenient and readable outline form. You can expand and collapse logical sections of the data by clicking "+" and "-"symbols displayed to the left of the actual data. You must have a valid account with TD Ameritrade and have successfully logged in before testing the URLs. Other formats cannot be tested in this manner.

As convenient as this process is, this is only the first step in the developer's development strategy. If the developer is building a downloadable client application or server-based application accessing the TD Ameritrade Trading Platform, he or she will ultimately write code to make the request and to process results. The exact way to do this is, of course, dependent upon the target operating system and programming language. Fortunately, almost every operating system and programming language has the ability to issue an HTTP GET request and to parse (process) the results. Quite often these capabilities are provided as part of a "toolkit" or other add-on. Although space within this document precludes an exhaustive treatment of this topic, here are some recommendations to get you started:

2.2 Language Specific HTTP Specs

LANGUAGE	GET	PARSE

Perl	LWP	XML::Simple or XML::XPath
PHP	PHP Interface to Curl	PHP Interface to Expat
VB.Net, C#	System.Net.WebRequest	System.Xml.XmlTextReader
VB6	WinInet	MSXML
Python	urllib	xmlparser
Java	java.net.URLConnection	JAXP

Resources

To support developers, TD Ameritrade provides the following resources:

- Updates to the API to support new or improved features,
- An online FAQ will provide answers to common development questions. Questions and answers from common support requests, and
- eMail support at API@TDAmeritrade.com
- developer message boards at http://apiforms.tdameritrade.com

2.3 Static vs. Dynamic Access

There is no defined limit on the number of requests allowed from a single user. Every attempt is made to service the requests but there is no guarantee regarding response time. Heavy volume may cause requests to be queued in a FIFO list for processing as resources become available. Monitoring tools are in place to observe traffic and will alert the support group if excessive request volume impacts the performance of servers. If the volume of requests is being generated by a specific application it will be restricted and the developer will be contacted by the support team to investigate the cause of the activity.

API users should consult with the TD Ameritrade Developer Relations team before deployment of a potentially high-volume application, to ensure that the package design does not exceed the server capacity and impact the client base. Application user requirements will be considered on a case by case basis and TDA will make adjustments to resources if justified.

Use streamer functions to provide real-time quote information. Do not use the single quote function with a high refresh rate to simulate real-time quotes.

2.4 Resources

To support developers, TD Ameritrade provides the following resources:

- Updates to the API to support new or improved features,
- A message board will provide answers to common development questions (http://apiforums.tdameritrade.com). Questions and answers from common support requests, and
- eMail support at api@tdameritrade.com

3 Programming Guide

TD Ameritrade Trading Platform API (TDAPI) allows the developer to access TD Ameritrade data and functionality through a Web site or Web-enabled application. TDAPI follows the standard Web services model: users of the service request data over HTTP (REST) and data is returned by the service as a formatted stream of text.

Throughout this programming guide, we will include examples to use with a browser to quickly and easily formulate calls to TDAPI and review results. We recommend an Internet-enabled computer to test TDAPI requests and immediately see the power and flexibility of TDAPI.

TDAPI can be incorporated in many different ways over a variety of Web sites and client applications. It provides an efficient method for accessing TD Ameritrade data for your clients and adding value to your products.

3.1 Overview of the TDAPI Model

Client applications must perform a user authentication dialog with TD Ameritrade's Trading Platform before issuing requests. The login process authenticates the user account and provides the level of service and feature access associated with the account.

After the end-user is authenticated and account parameters are returned, the client may initiate requests. Available requests are:

- Synchronous requests (request/response) – Snapshot Quotes, Balances and Positions, Order Status, Trading and Order Cancel - use HTTPS protocol where the request takes the form of an HTTPS GET or POST with parameters as required by the type of request. The response is returned using XML format,

- Asynchronous requests (request/streaming) – Level I and Level II Quotes (OPRA, NYSE BOOK, ADAP INET), Time & Sale, Option, News, News History, NYSE Chart, NASDAQ Chart, INDEX Chart, Actives and ACCOUNT ACTIVITY use HTTP protocol where the request takes the form of an HTTP GET or POST with parameters as required by the type of request. The request opens a connection that returns a continuous stream of data using Trading Platform Streaming format.

  NOTE: If you are using a POST, you must add two newline characters (ascii 10) to the end of the data being posted.

3.2 Session control

There are two methods of maintaining session control with the API: 1) with and 2) without cookies.

WITH COOKIES

Upon a successful LogIn Command, the server will set a cookie called JSESSIONID with the same value as the LogIn XML response tag session-id. Once that cookie is set, you can issue subsequent HTTPS commands without having to pass anything else in the URL, as long as you automatically handle the cookies. If you need to connection multiple sessions, you can store the session-id value from the login to each session and then set the JSESSIONID cookie accordingly prior to EVERY HTTPS call.

WITHOUT COOKIES

Upon a successful LogIn Command, the server will return XML result tag session-id. You then include jsessionid=#session-id# in every HTTPS URL call (NOT Including streaming API commands). The way you add the parameter is a bit different than the regular parameters. For example, to get quotes, the regular URL would be:
https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**>&symbol=<**#symbol#**>

With the **jsessionid** specified, it would be:
https://apis.tdameritrade.com/apps/100/Quote;**jsessionid**=<**#sessionid#>**?source=<**#sourceID#**>&symbol=<**#symbol#**>

NOTES:
1) the **jsessionid** parameter is added BEFORE the question mark and is separated from the path using a semicolon
2) the name of the cookie and parameter is **case sensitive**. It has to be UPPER CASE for COOKIE and lower case if in the URL
3) The value of the sessionid is also **case sensitive**. It has to be entered exactly as it was returned from LogIn

3.3 Source ID

Organizations are assigned a unique identifier to be passed as part of every request to the TD Ameritrade Trading Platform generated by the organization's software packages. The identifier is a string that indicates the company and product making the request. The support team at TD Ameritrade uses this information to analyze traffic from a source and isolate groups of data when a developer requests support. The business team aggregates data by source for statistical reporting.

For example, Get Rich Quick LLC (a hypothetical company) provides a real time charting, quote and trading software package, version 2.3, and a separate tool for back testing trading strategy models, version 3.4.3. TDA creates two identifiers, one for each package. The first identifier is XA and the second is XB.

The identifier is part of each request string and is referenced throughout this document.

If you have not received a source ID, please email us at API@TDAmeritrade.com

4 URL Encoding

Many of the request strings require that the URL parameters be encoded before transmission. Encoding is the process of substituting some characters in the request parameters for other characters. This enables passing of the arguments with characters that otherwise might be misinterpreted in the process. The background for encoding is based on RFC 1738: Uniform Resource Locators (URL) specification.

The specification for URLs (RFC 1738, Dec. '94) poses a problem, in that it limits the use of allowed characters in URLs to only a limited subset of the US-ASCII character set:

> "...Only alphanumerics [0-9a-zA-Z], the special characters "$-_.+!*'()," [not including the quotes - ed], and reserved characters are used for reserved purposes that may be used unencoded within a URL."

HTML, on the other hand, allows the entire range of the ISO-8859-1 (ISO-Latin) character set to be used in documents. HTML4 expands the allowable range to include all of the Unicode character set as well. In the case of non-ISO-8859-1 characters (characters above FF hex/255 decimal in the Unicode set), they just cannot be used in URLs because there is no safe way to specify character set information in the URL content [RFC2396.]

4.1 What needs to be encoded and why

ASCII Control characters

Why: These characters are not printable.

Characters: Includes the ISO-8859-1 (ISO-Latin) character ranges 00-1F hex (0-31 decimal) and 7F (127 decimal.)

Non-ASCII characters

Why: These are by definition not legal in URLs since they are not in the ASCII set.

Characters: Includes the entire "top half" of the ISO-Latin set 80-FF hex (128-255 decimal.)

"Reserved characters"

Why: URLs use some characters for special use in defining their syntax. When these characters are not used in their special role inside a URL, they need to be encoded.

CHARACTER	HEX	DECIMAL
Dollar ("$")	24	36
Ampersand ("&")	26	38
Plus ("+")	2B	43
Comma (",")	2C	44
Forward slash/Virgule ("/")	2F	47
Colon (":")	3A	58
Semi-colon (";")	3B	59

Equals ("=")	3D	61
Question mark ("?")	3F	63
'At' symbol ("@")	40	64

"Unsafe characters"

Why: Some characters present the *possibility* of being misunderstood within URLs for various reasons. These characters should also always be encoded.

<table>
<tr><th>CHARACTER</th><th>HEX</th><th>DECIMAL</th><th>WHY ENCODE?</th></tr>
<tr><td>Space</td><td>20</td><td>32</td><td>Significant sequences of spaces may be lost in some uses (especially multiple spaces)</td></tr>
<tr><td>Quotation marks
'Less Than' symbol ("<")
'Greater Than' symbol (">")</td><td>22
3C
3E</td><td>34
60
62</td><td>These characters are often used to delimit URLs in plain text.</td></tr>
<tr><td>'Pound' character ("#")</td><td>23</td><td>35</td><td>This is used in URLs to indicate where a fragment identifier (bookmarks/anchors in HTML) begins.</td></tr>
<tr><td>Percent character ("%")</td><td>25</td><td>37</td><td>This is used to URL encode/escape other characters, so it should itself also be encoded.</td></tr>
<tr><td>Misc. characters:
Left Curly Brace ("{")
Right Curly Brace ("}")
Vertical Bar/Pipe ("|")
Backslash ("\")
Caret ("^")
Tilde ("~")
Left Square Bracket ("[")
Right Square Bracket ("]")
Grave Accent ("`")</td><td>7B
7D
7C
5C
5E
7E
5B
5D
60</td><td>123
125
124
92
94
126
91
93
96</td><td>Some systems can possibly modify these characters.</td></tr>
</table>

4.2 How are characters URL Encoded?

URL encoding of a character consists of a "%" symbol, followed by the two-digit hexadecimal representation (case-insensitive) of the ISO-Latin code point for the character.

Example

Space = decimal code point 32 in the ISO-Latin set.

32 decimal = 20 in hexadecimal
The URL encoded representation will be "%20"

4.3 All Substitutions

URL encoding from %00 to %8F

ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE
æ	%00	0	%30	`	%60
	%01	1	%31	a	%61
	%02	2	%32	b	%62
	%03	3	%33	c	%63
	%04	4	%34	d	%64
	%05	5	%35	e	%65
	%06	6	%36	f	%66
	%07	7	%37	g	%67
backspace	%08	8	%38	h	%68
tab	%09	9	%39	i	%69
linefeed	%0A	:	%3a	j	%6a
	%0B	;	%3b	k	%6b
	%0C	<	%3c	l	%6c
c return	%0D	=	%3d	m	%6d
	%0E	>	%3e	n	%6e
	%0F	?	%3f	o	%6f
	%10	@	%40	p	%70
	%11	A	%41	q	%71
	%12	B	%42	r	%72
	%13	C	%43	s	%73
	%14	D	%44	t	%74
	%15	E	%45	u	%75
	%16	F	%46	v	%76
	%17	G	%47	w	%77
	%18	H	%48	x	%78
	%19	I	%49	y	%79
	%1A	J	%4A	z	%7A
	%1B	K	%4B	{	%7B
	%1C	L	%4C	|	%7C
	%1D	M	%4D	}	%7D
	%1E	N	%4E	~	%7E
	%1F	O	%4F		%7F

ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE
space	%20	P	%50	€	%80
!	!	Q	%51		%81
"	%22	R	%52	‚	%82
#	%23	S	%53	ƒ	%83
$	%24	T	%54	„	%84
%	%25	U	%55		%85
&	%26	V	%56	†	%86
'	%27	W	%57	‡	%87
(	%28	X	%58	ˆ	%88
)	%29	Y	%59	‰	%89
*	%2A	Z	%5A	Š	%8A
+	%2B	[	%5B	‹	%8B
,	%2C	\	%5C	Œ	%8C
-	%2D	]	%5D		%8d
.	%2E	^	%5E	Ž	%8E
/	%2F	_	%5F		%8F

URL encoding from %90 to %FF

ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE
	%90	À	%C0	ð	%F0
‘	%91	Á	%C1	ñ	%F1
’	%92	Â	%C2	ò	%F2
“	%93	Ã	%C3	ó	%F3
”	%94	Ä	%C4	ô	%F4
•	%95	Å	%C5	õ	%F5
–	%96	Æ	%C6	ö	%F6
—	%97	Ç	%C7	÷	%F7
˜	%98	È	%C8	ø	%F8
™	%99	É	%C9	ù	%F9
š	%9A	Ê	%CA	ú	%FA
›	%9B	Ë	%CB	û	%FB
œ	%9C	Ì	%CC	ü	%FC
	%9D	Í	%CD	ý	%FD
ž	%9E	Î	%CE	þ	%FE
Ÿ	%9F	Ï	%CF	ÿ	%FF
	%A0	Ð	%D0		

ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE	ASCII VALUE	URL-ENCODE
¡	%A1	Ñ	%D1		
¢	%A2	Ò	%D2		
£	%A3	Ó	%D3		
	%A4	Ô	%D4		
¥	%A5	Õ	%D5		
|	%A6	Ö	%D6		
§	%A7		%D7		
¨	%A8	Ø	%D8		
©	%A9	Ù	%D9		
ª	%AA	Ú	%DA		
«	%AB	Û	%DB		
¬	%AC	Ü	%DC		
	%AD	Ý	%DD		
®	%AE	Þ	%DE		
	%AF	ß	%DF		
°	%B0	à	%E0		
±	%B1	á	%E1		
²	%B2	â	%E2		
³	%B3	ã	%E3		
´	%B4	ä	%E4		
µ	%B5	å	%E5		
¶	%B6	æ	%E6		
·	%B7	ç	%E7		
¸	%B8	è	%E8		
¹	%B9	é	%E9		
º	%BA	ê	%EA		
»	%BB	ë	%EB		
¼	%BC	ì	%EC		
½	%BD	í	%ED		
¾	%BE	î	%EE		
¿	%BF	ï	%EF		

5 LogIn

Service provides user login capability through the authorization processes. A successful login returns fields required for additional access. Error messages are returned for failed logon attempts.

5.1 LogIn Request

The XML Login Request should be as a POST request.

GET REQUESTS ARE NOT SUPPORTED FOR LOGIN

Request URL

https://apis.tdameritrade.com/apps/100/LogIn?source=#sourceID#&version=#version number#

Parameters

The following parameters have to be POSTed:

userid=#**userid**#&password=#**password**#&source=#**sourceID**#&version=#**version number**#

NOTE that the source and version are repeated. They show up in both the URL parameters and in the POST data.

DO NOT INCLUDE Userid and Password in the URL

BIG NOTE: You MUST have "Content-Type" HTTP Header specified otherwise the login will fail:

Content-Type: application/x-www-form-urlencoded

5.2 LogIn Parameters

The login parameters specify username and password for the account being logged in, as well as the Source ID, assigned by TD Ameritrade and the version information about the program logging in.

Parameter names are always lower case.

LogIn Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
userid	The User ID used for entering the TDA web site	YES	The value will be assigned to you by TD AMERITRADE
password	The Password used for entering the TDA web site	YES	The value is assigned to the end-user and should be provided by the end user in order to login to the site
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application

version	The Version number of the client software application.	YES	version number of the client software application. e.g. version=1.0

5.3 LogIn Response

The response to the LogIn request will be in XML format.

General Structure of XML Response


```
<?xml version="1.0" ?>
<amtd>
<result></result>
<xml-log-in>
   <session-id></session-id>
   <user-id></user-id>
   <cdi></cdi>
   <timeout></timeout>
   <login-time></login-time>
   <associated-account-id></associated-account-id>
   <nyse-quotes></nyse-quotes>
   <nasdaq-quotes></nasdaq-quotes>
   <opra-quotes></opra-quotes>
   <amex-quotes></amex-quotes>
   <exchange-status></exchange-status>
   <accounts>
      <account>
         <account-id></account-id>
         <display-name></display-name>
         <cdi></cdi>
         <description></description>
         <associated-account></associated-account>
         <company></company>
         &lt;segment&gt;&lt;/segment&gt;
         <unified></unified>
         <preferences>
            <express-trading></express-trading>
            <option-direct-routing></option-direct-routing>
            <stock-direct-routing></stock-direct-routing>
         </preferences>
         <authorizations>
            <apex></apex>
            <level2></level2>
            <stock-trading></stock-trading>
            <margin-trading></margin-trading>
            <streaming-news></streaming-news>
            <option-trading></option-trading>
            <streamer></streamer>
            <advanced-margin></advanced-margin>
         </authorizations>
      </account>
```

```
        <account>
          ......
        </account>
    </accounts>
    </xml-log-in>
</amtd>
```

Login Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
xml-log-in	Complex	Container for the login information object
session-id	String	Session ID used for Session Control of the current login
user-id	String	User ID used to login for the account
cdi	String	User Level CD Domain ID (Internal Use)
timeout	Integer	The number of MINUTES of inactivity before the session is invalidated, requiring a new Login
login-time	String	Current Date/time, for example **2009-03-05 14:28:29 EST**. Can be used for time display adjustments relative to local computer time.
associated-account-id	String	The account ID of the main account associated with the current login. Only one account can be the "main" account
nyse-quotes	String	**realtime/delayed**
nasdaq-quotes	String	**realtime/delayed**
opra-quotes	String	**realtime/delayed**
amex-quotes	String	**realtime/delayed**
exchange-status	String	**professional/non-professional/unknown**
accounts	Complex	Container for the accounts information objects
account	Complex	Container for the individual account object
account-id	String	The account ID of the user's account
display-name	String	The display name for the account, same as shown on the TD Ameritrade web site
cdi	String	Account Level CD Domain ID (Internal Use)
description	String	User Friendly description or Name associated with the account
associated-account	String/Boolean	true/false - Indicates whether or not this account is the main account associated with the User ID/Login session
company	String	TDA internal code for the company the account is associated with. Will be needed for Streaming Quote requests
segment	String	TDA internal code for the segment the account is associated with. Will be needed for Streaming Quote requests

unified	String/Boolean	true/false - Indicates if the account is enabled for Unified Site. If not, then you will not be able to launch any URL commands with /u/ in them
preferences	Complex	Container for the Preferences for the given account
express-trading	String/Boolean	true/false - Indicates if the user has selected Express Trading option on the web site (does not affect API)
options-direct-routing	String/Boolean	true/false - Indicates if the account is enabled for direct routing options orders
stock-direct-routing	String/Boolean	true/false - Indicates if the account is enabled for direct routing stock orders
authorizations	Complex	Container for the Authorizations for the given account
apex	String/Boolean	true/false - denotes whether or not the account has APEX status
level2	String/Boolean	true/false - Indicates whether the account is authorized for Level 2 quotes (NASDAQ Level II)
stock-trading	String/Boolean	true/false - Indicates if the account is enabled for stock trading
option-trading	String	Indicates if the account is enabled for options trading. If so, then the type of permissions. Possible values are: **none** **long** **covered** **spread** **full**
margin-trading	String/Boolean	true/false - Indicates if the account is enabled for MARGIN trading. If false, then its a CASH account
streamer	String/Boolean	true/false - Indicates if the account is enabled for streaming data access
streaming-news	String/Boolean	true/false - Indicates if the account is enabled for streaming news
advanced-margin	String/Boolean	true/false - Indicates if the account will use a new middleware (AMX-TIBCO) for margin computation

5.4 Request/Response Examples

Request:

https://apis.tdameritrade.com/apps/100/LogIn?source=**#sourceID#**&version=**#version**
POST DATA
userid=**#userid#**&password=**#password#**&source=**#sourceID#**&version=**#version**

Response:


```
<?xml version="1.0" ?>
<amtd>
<result>OK</result>
<xml-log-in>
  <session-id>OK</session-id>
  <user-id>joeabandon13</user-id>
  <cdi>A000000000001111</cdi>
  <timeout>55</timeout>
```

```xml
<login-time>2009-03-05 14:28:29 EST</login-time>
<associated-account-id>392485381</associated-account-id>
<nyse-quotes>realtime</nyse-quotes>
<nasdaq-quotes>realtime</nasdaq-quotes>
<opra-quotes>realtime</opra-quotes>
<amex-quotes>realtime</amex-quotes>
<exchange-status>non-professional</exchange-status>
<accounts>
    <account>
        <account-id>123456789</account-id>
        <display-name>Joe Abandon</display-name>
        <cdi>A000000100001111</cdi>
        <description>Joe Abandon Main Account</description>
        <associated-account>true</associated-account>
        <company>AMER</company>
        &lt;segment&gt;UAMER&lt;/segment&gt;
        <unified>true</unified>
        <preferences>
            <express-trading>false</express-trading>
            <option-direct-routing>true</option-direct-routing>
            <stock-direct-routing>true</stock-direct-routing>
        </preferences>
        <authorizations>
            <apex>true</apex>
            <level2>true</level2>
            <stock-trading>true</stock-trading>
            <margin-trading>true</margin-trading>
            <streaming-news>true</streaming-news>
            <option-trading>long</option-trading>
            <streamer>true</streamer>
            <advanced-margin>true</advanced-margin>
        </authorizations>
    </account>
    <account>
        <account-id>123456789</account-id>
        <display-name>Joe Abandon2</display-name>
        <cdi>A000000100001111</cdi>
        <description>Jane Abandon Account</description>
        <associated-account>false</associated-account>
        <company>AMER</company>
        &lt;segment&gt;UAMER&lt;/segment&gt;
        <unified>true</unified>
        <preferences>
            <express-trading>true</express-trading>
            <option-direct-routing>true</option-direct-routing>
            <stock-direct-routing>true</stock-direct-routing>
        </preferences>
        <authorizations>
            <apex>true</apex>
            <level2>true</level2>
            <stock-trading>true</stock-trading>
            <margin-trading>true</margin-trading>
            <streaming-news>true</streaming-news>
```

```
                    <option-trading>long</option-trading>
                    <streamer>true</streamer>
                    <advanced-margin>true</advanced-margin>
                </authorizations>
            </account>
        </accounts>
        </xml-log-in>
</amtd>
```

5.5 Response Errors

Request:

https://apis.tdameritrade.com/apps/100/LogIn?userid=**#userid#**&password=**#password#**
&source=**#sourceID#**&version=**#version**

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Login Failed</error>
</amtd>
```

This will occur if you put all the parameters into the URL (use a GET to request the data). All LOGIN Requests must be POSTs.

If the any of the required parameters is missing or incorrect, the following generic error message is displayed:

Request:

https://apis.tdameritrade.com/apps/100/LogIn?source=**#sourceID#**&version=**#version**
POST DATA
userid=**#userid#**

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Unable to authorize access.</error>
</amtd>
```

6 LogOut

This call will invalidate the user session. It is a security feature that should be called when the user wants to stop their current session or close the client application.

Request URL

https://apis.tdameritrade.com/apps/100/LogOut?source=#sourceID#

Parameters

No parameters are needed other then the source.

General Structure of XML Response

The response to the LogOut request will be in XML format

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
</amtd>
```

Login Response Fields

XML Attribute Name	Type	Definitions
result	String	**LoggedOut**

Response:

```
<?xml version="1.0" ?>
<amtd>
<result>LoggedOut</result>
</amtd>
```

7 KeepAlive

If the login user is inactive for more than the timeout period, the session will expire and the client will need to login again. This servlet is specifically designed to refresh the login user session so that it does not expire.

Request URL

https://apis.tdameritrade.com/apps/KeepAlive?source=#sourceID#

Parameters

No parameters are needed other then the source

Response:

The response to the KeepAlive request will be a one world reply. Either "LoggedOn" or "InvalidSession" without html or xml formatting. The content type is text/plain.

8 StreamerInfo

This service provides the Streamer data request information specific to the associated logged in account. A successful StreamerInfo request returns fields required for access to streaming data requests. Error messages are returned if the account is not enabled for streaming data.

NOTE: The StreamerInfo parameters that are returned are good for 24 hours. You would need to re-submit the StreamerInfo request if you are running your software overnight.

ALSO NOTE: The info returned from StreamerInfo is account specific. The information is returned for the **associated account**, even if there are many accounts linked for the given login. Therefore, when using the information to request streaming data, you have to make sure to send the associated account #, company and segment parameters, regardless of which linked account may be the main one you are using.

8.1 StreamerInfo Request

Request URL

https://apis.tdameritrade.com/apps/100/StreamerInfo?source=<**#SourceID#**>

Parameters

The only parameter required is the source assigned by TD Ameritrade. The account id parameter is optional.

8.2 StreamerInfo Parameters

The only parameter required is the source assigned by TD Ameritrade. The account id parameter is optional.

StreamerInfo Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by TD Ameritrade.	YES	The value is assigned by TD Ameritrade to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned. Can be combined with any other parameter.	NO	One of the account IDs returned from the LogIn service. The default is the associated account

8.3 StreamerInfo Response

The response to the StreamerInfo request will be in XML format.

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
<result></result>
```

```
<streamer-info>
    <streamer-url></streamer-url>
    <token></token>
    <timestamp></timestamp>
    <cd-domain-id></cd-domain-id>
    <usergroup></usergroup>
    <access-level></access-level>
    <acl></acl>
    <app-id></app-id>
    <authorized></authorized>
    <error-msg></error-msg>
</streamer-info>
</amtd>
```

StreamerInfo Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
streamer-info	Complex	Container for StreamerInfo information object
streamer-url	String	the domain of the server that should be used for streaming data
token	String	Variable to be passed in the "token" parameter in all streamer requests
timestamp	Integer	Variable to be passed in the "timestamp" parameter in all streamer requests
cd-domain-id	String	Variable to be passed in the "cddomain" parameter in all streamer requests
usergroup	String	Variable to be passed in the "usergroup" parameter in all streamer requests
access-level	String	Variable to be passed in the "accesslevel" parameter in all streamer requests
acl	String	Variable to be passed in the "acl" parameter in all streamer requests
app-id	String	Variable to be passed in the "appid" parameter in all streamer requests **NOTE**: This is NOT the same as the Source ID assigned to the developer application.
authorized	String	Variable to be passed in the "authorized" parameter in all streamer requests
error-msg	String	Plain text of the error, in case one occurred
error	String	Error Code (currently only one – NS)

8.4 Request/Response Examples

Request:

https://apis.ameritrade.com/apps/100/StreamerInfo?source=**#sourceID#**

Response:


```
<?xml version="1.0" ?>
<amtd>
<result>OK</result>
<streamer-info>
    <streamer-url>ameritrade02.streamer.com</streamer-url>
    <token>1223749a1518c56c3b40c92123602a54a4d9ba2d</token>
    <timestamp>1170615023913</timestamp>
    <cd-domain-id>A000000018311352</cd-domain-id>
    <usergroup>ACCT</usergroup>
    <access-level>ACCT</access-level>
    <acl>DRESGKMANSOLFNQ2QSTETFTOTTUAUR</acl>
    <app-id>QT</app-id>
    <authorized>Y</authorized>
    <error-msg></error-msg>
</streamer-info>
</amtd>
```


8.5 Response Errors

Possible Errors

If the account is not enabled for streaming data, then user will get the following response:


```
<?xml version="1.0" ?>
<amtd>
  <result>FAIL</result>
  <error>NS</error>
  <streamer-info>
      <error-msg> User is not allowed to access Streamer.  </error-msg>
  </streamer-info>
</amtd>
```


NOTE: this would happen if the client is NON-Professional (or not yet declared) and one of the exchange agreements is not signed

9 MessageKey

This service provides user the Encrypted Token for the accounts that can then be monitored via Streamer ACCT_ACTIVITY request. A successful MessageKey request returns the key for the accounts that were requested in the parameters and required for access to streaming data requests. Error messages are returned if the specified accounts are not found or are not permitted for the currently logged in User ID.

NOTE: The MessageKey token that is returned is good for 24 hours. You would need to re-submit the MessageKey request if you are running your software overnight.

9.1 MessageKey Request

Request URL

https://apis.tdameritrade.com/apps/100/MessageKey?source=<**#SourceID#**>&accountid=<#AccountNumber#>

Parameters

The request must contain the source ID assigned by TD Ameritrade. You can also specify one or more **account** parameters to specify the accounts for which updates are to be requested and thus the for which the message key is to be build. Only accounts that are accessible for the logged in User ID can be specified.

If no account parameters are specified, the Message Key will be returned for all accounts accounts accessible by the logged in User ID.

MessageKey Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned Can be combined with any other parameter	NO	One of the account IDs returned from the LogIn service.

9.2 MessageKey Response

The response to the MessageKey request will be in XML format:

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
<result></result>
<message-key>
    <token></token>
    <timestamp></timestamp>
</message-key>
</amtd>
```

MessageKey Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
message-key	Complex	Container for MessageKey information object
token	String	Variable to be passed in the "token" parameter in all streamer requests for ACCT_ACTIVITY
timestamp	Integer	The time at which the token was generated. Used to track when a new token is needed since the toke n is good for 24 hours
error	String	Plain text error message

9.3 Request/Response Examples

Request:

https://apis.ameritrade.com/apps/100/MessageKey?source=**#sourceID#**&account=**#accountNumber1#**&account=**#accountNumber2#**

Response:

```
<?xml version="1.0" ?>
<amtd>
<result>OK</result>
<message-key>
    <token>1223749a1518c56c3b40c92123602a54a4d9ba2d</token>
    <timestamp>1170615023913</timestamp>
</message-key>
</amtd>
```

9.4 Response Errors

Possible Errors

If the account is not recognized or some other error occurred, then you would get the following response:

```
<?xml version="1.0" ?>
<amtd>
  <result>FAIL</result>
  <error>We were unable to process your request. Please try again later.</error>

</amtd>
```

10 Snapshot (XML) Quotes

Service provides detailed quote information for one or more symbols. Currently the API allows symbol types of Stocks, Options, Mutual Funds and Indexes. Quotes are real-time for accounts subscribed to this service; otherwise, quotes are delayed according to exchange rules.

NOTE: Use Streaming Data functions to provide real-time quote information whenever possible. Do not use the single quote function with a high refresh rate to simulate real-time quotes.

10.1 Quote Request

Request URL

https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**>&symbol=<**#symbol#**>

Parameters

The quote parameter contains one or more symbols in a comma or space separated list.

10.2 Quote Parameters

The quote parameter contains one or more symbols in a comma or space separated list. The TD AMERITRADE quote standards apply. Indices require a $ as the first character. The query string should be URL encoded

Quote Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
symbol	List of symbols for which the quotes should be returned	YES	Can be stock, option, index, or Mutual fund symbols. Standard TD AMERITRADE symbology applies. **NOTE: Max 300 symbols per request**

10.3 Quote Response

The response to the Quote request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
```

```
  <quote-list>
      <error></error>
      <quote>
        <error></error>
        <symbol></symbol>
        <description></description>
        <bid></bid>
        <ask></ask>
        <bid-ask-size></bid-ask-size>
        <last></last>
        <last-trade-size></last-trade-size>
        <last-trade-date></last-trade-date>
        <open></open>
        <high></high>
        <low></low>
        <close></close>
        <volume></volume>
        <year-high></year-high>
        <year-low></year-low>
        <real-time></real-time>
        <exchange></exchange>
        <asset-type></asset-type>
        <change></change>
        <change-percent></change-percent>
      </quote>
  </quote-list>
</amtd>
```

Response Fields

All response fields will be different depending on the Asset Type of the symbol being quoted. The fields for each specific asset type are documented at:

Stocks
Options
Indices
Mutual Funds

10.3.1 Quote Response - Stocks

Quote Response Fields for STOCKS

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
quote-list	Complex	Container for one or more <quote> objects
error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**." would be returned if the symbol was not found. The error tag exists within each level and returns a message specific to the layer.

quote	Complex	Container for detailed quote fields
error	String	
symbol	String	Symbol of the security being quoted. For example, DELL
description	String	Contains a description of the symbol. For example, "**DELL INC COM Status Alert: Delinquent**:" The string "**Status Alert:**" is appended to each description except for asset type "**F**" when an FSI flag is set on a symbol. Any text following this string specifies some trading status message.
bid	Double	The highest price anyone has declared that they are willing to pay for a security.
ask	Double	The lowest price anyone will offer to sell a security.
bid-ask-size	String	The size of the **bid/ask** shows how many shares the market has available at those prices. The value is displayed as "bid qty X ask qty". For example, **3900X1800**
last	Double	The price of the last trade for this symbol.
last-trade-size	Integer	The number of shares traded at the last price.
last-trade-date	String	The date and time of the last trade. For example, "**2006-06-30 14:36:27 EDT**".
open	Double	The price of the first trade at normal market open. Extended hours' trading is not reflected in this value.
high	Double	The highest price trade for the symbol during the normal trading session. If the session is still open this represents the highest price to the time of the quote.
low	Double	The lowest price trade for the symbol during the normal trading session. If the session is still open this represents the lowest price to the time of the quote.
close	Double	The price of the last trade for the symbol at the end of the previous trading session.
volume	Integer	The number of shares traded for the symbol.
year-high	Double	The highest price trade for the symbol in the past 52 weeks
year-low	Double	The lowest price trade for the symbol in the past 52 weeks
real-time	String	Indicates whether the quote information is real-time or delayed. **true** — Real-time quote if the account is subscribed to RT **false** — Delayed quote if the account not subscribed to RT
exchange	String	Stock exchange where security symbol is listed. For example, **NASDAQ, AMEX, NYSE, OTC, PACX**

asset-type	String	Type of asset the symbol represents (Equity in this case). It is a one character code, such as: **E** - Equity or ETF **F** - Mutual Fund **I** - Index. **O** - Option **B** - Bond
change	Double	The difference between the last trade price and the previous trading session's closing price. If the last trade price is greater than the previous session's closing price, the value is returned without a leading sign. If the last trade price is less than the previous session's closing price the value is returned with "-" as the leading character.
change-percent	String	Percent change from the last trade price compared to the previous session's closing price. For example, if the previous session closing price is 14.48 and the last trade price is 14.88, the change is 0.40 and the change percent is 2.76. The value is returned as 2.76%.

10.3.2 Quote Response - Options

Quote Response Fields for OPTIONS

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
quote-list	Complex	Container for one or more <quote> objects
error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**." would be returned if the symbol was not found. The error container exists within each level and returns a message specific to the layer.
quote	Complex	Container for detailed quote fields
symbol	String	Symbol of the security being quoted. For example, **AMTD_092210P12.5**
description	String	Contains a description of the symbol. For example, "**AMTD Sep 22 2010 12.5 Call**"
bid	Double	The highest price anyone has declared that they are willing to pay for a security.
ask	Double	The lowest price anyone will offer to sell a security.

bid-ask-size	String	The size of the **bid/ask** shows how many contracts the market has available at those prices. The value is displayed as "bid qty X ask qty". For example, **390X41**
last	Double	The price of the last trade for this symbol.
last-trade-size	Integer	The number of contracts traded at the last price.
last-trade-date	String	The date and time of the last trade. For example, "**2006-06-30 14:36:27 EDT**".
open	Double	The price of the first trade at normal market open.
high	Double	The highest price trade for the symbol during the normal trading session. If the session is still open this represents the highest price to the time of the quote.
low	Double	The lowest price trade for the symbol during the normal trading session. If the session is still open this represents the lowest price to the time of the quote.
close	Double	The price of the last trade for the symbol at the end of the previous trading session.
volume	Integer	The number of shares traded for the symbol.
strike-price	Double	
open-interest	Integer	
expiration-month	Integer	
expiration-day	Integer	
expiration-year	Integer	
real-time	String	Indicates whether the quote information is real-time or delayed. **true** — Real-time quote if the account is subscribed to RT **false** — Delayed quote if the account not subscribed to RT
exchange	String	Stock exchange where security symbol is listed. For example, **OPRA, AMEX**
asset-type	String	Type of asset the symbol represents (Option in this case). It is a one character code, such as: **E** - Equity or ETF **F** - Mutual Fund **I** - Index. **O** - Option **B** - Bond

change	Double	The difference between the last trade price and the previous trading session's closing price. If the last trade price is greater than the previous session's closing price, the value is returned without a leading sign. If the last trade price is less than the previous session's closing price the value is returned with "-" as the leading character.
change-percent	String	Percent change from the last trade price compared to the previous session's closing price. For example, if the previous session closing price is 14.48 and the last trade price is 14.88, the change is 0.40 and the change percent is 2.76. The value is returned as 2.76%.
underlying-symbol	String	
put-call	String	**C** or **P**
delta	Double	
gamma	Double	
theta	Double	
vega	Double	
rho	Double	
implied-volatility	Double	
days-to-expiration	Integer	
time-value-index	Double	
multiplier	Integer	

10.3.3 Quote Response - Indices

Quote Response Fields for INDICES

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
quote-list	Complex	Container for one or more <quote> objects

error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**." would be returned if the symbol was not found. The error container exists within each level and returns a message specific to the layer.
quote	Complex	Container for detailed quote fields
symbol	String	Symbol of the security being quoted. For example, **$SPX.X**
description	String	Contains a description of the symbol. For example, "**NEW S & P 500 INDEX**"
open	Double	The first quote for the session
high	Double	The highest price quote for the symbol during the normal trading session. If the session is still open this represents the highest price to the time of the quote.
low	Double	The lowest price quote for the symbol during the normal trading session. If the session is still open this represents the lowest price to the time of the quote.
last	Double	The price of the last quote for this symbol.
last-trade-date	String	The date and time of the last update. For example, "**2006-06-30 14:36:27 EDT**".
close	Double	The price of the last quote for the symbol at the end of the previous trading session.
volume	Integer	
year-high	Double	The highest price quote for the symbol in the past 52 weeks
year-low	Double	The lowest price quote for the symbol in the past 52 weeks
real-time	String	Indicates whether the quote information is real-time or delayed. **true** — Real-time quote if the account is subscribed to RT **false** — Delayed quote if the account not subscribed to RT
exchange	String	Stock exchange where security symbol is listed. For example, **NASDAQ, AMEX, NYSE**
asset-type	String	Type of asset the symbol represents (Index in this case). It is a one character code, such as: **E** - Equity or ETF **F** - Mutual Fund **I** - Index. **O** - Option **B** - Bond
change	Double	The difference between the last quote price and the previous trading session's closing price. If the last quote price is greater than the previous session's closing price, the value is returned without a leading sign. If the last quote price is less than the previous session's closing pric,e the value is returned with "-" as the leading character.

change-percent	String	Percent change from the last quote price compared to the previous session's closing price. For example, if the previous session closing price is 14.48 and the last quote price is 14.88, the change is 0.40 and the change percent is 2.76. The value is returned as 2.76%.

10.3.4 Quote Response - Funds

Quote Response Fields for MUTUAL FUNDS

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
quote-list	Complex	Container for one or more <quote> objects
error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**." would be returned if the symbol was not found. The error container exists within each level and returns a message specific to the layer.
quote	Complex	Container for detailed quote fields
symbol	String	Symbol of the security being quoted. For example, **FFFEX**
description	String	Contains a description of the symbol. For example, "**FID ABERDEEN ST TR FREEDM 2030**"
nav	Double	
offer	Double	
change	double	The difference between the NAV and the previous day's NAV.
real-time	String	Indicates whether the quote information is real-time or delayed. **true** — Real-time quote if the account is subscribed to RT **false** — Delayed quote if the account not subscribed to RT
asset-type	String	Type of asset the symbol represents (Mutual Fund in this case). It is a one character code, such as: **E** - Equity or ETF **F** - Mutual Fund **I** - Index. **O** - Option **B** - Bond

10.4 Request/Response Examples

Request URL

https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**
>&symbol=$SPX.X,AMTD_20070922P12.5,DELL,FFFEX

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <quote-list>
        <error></error>
        <quote>
            <error></error>
            <symbol>DELL</symbol>
            <description>DELL INC COM Status Alert: Delinquent</description>
            <bid>27.27</bid>
            <ask>27.28</ask>
            <bid-ask-size>2800X4600</bid-ask-size>
            <last>27.28</last>
            <last-trade-size>200</last-trade-size>
            <last-trade-date>2007-06-12 10:03:11 EDT</last-trade-date>
            <open>27.35</open>
            <high>27.49</high>
            <low>27.25</low>
            <close>27.60</close>
            <volume>2028822</volume>
            <year-high>28.04</year-high>
            <year-low>18.95</year-low>
            <real-time>true</real-time>
            <exchange>NASDAQ</exchange>
            <asset-type>E</asset-type>
            <change>-0.32</change>
            <change-percent>-1.16%</change-percent>
        </quote>
        <quote>
            <error></error>
            <symbol>FFFEX</symbol>
            <description>FID ABERDEEN ST TR FREEDM 2030</description>
            <nav>16.75</nav>
            <change>0.02</change>
            <real-time>true</real-time>
            <asset-type>F</asset-type>
        </quote>
        <quote>
            <error></error>
```

```
            <symbol>AMTD_20070922P12.5</symbol>
            <description>AMTD Sep 22 2007 12.5 Put</description>
            <bid>3.50</bid>
            <ask>3.70</ask>
            <bid-ask-size>244X61</bid-ask-size>
            <last>3.70</last>
            <last-trade-size>1</last-trade-size>
            <last-trade-date>2007-06-12 10:45:27 EDT</last-trade-date>
            <open>4.00</open>
            <high>5.29</high>
            <low>3.40</low>
            <close>9.10</close>
            <volume>5017</volume>
            <strike-price>510.00</strike-price>
            <open-interest>14775</open-interest>
            <expiration-month>6</expiration-month>
            <expiration-year>2007</expiration-year>
            <real-time>true</real-time>
            <exchange>OPRA</exchange>
            <asset-type>O</asset-type>
            <underlying-symbol>GOOG</underlying-symbol>
            <delta>0.197</delta>
            <gamma>0.008</gamma>
            <theta>-1.193</theta>
            <vega>0.139</vega>
            <rho>0.01</rho>
            <implied-volatility>68.074</implied-volatility>
            <days-to-expiration>4</days-to-expiration>
            <time-value-index>3.70</time-value-index>
            <multiplier>100.00</multiplier>
        </quote>
        <quote>
            <error></error>
            <symbol>$SPX.X</symbol>
            <description>NEW S & P 500 INDEX</description>
            <high>1509.12</high>
            <low>1497.69</low>
            <last>1499.66</last>
            <close>1509.12</close>
            <year-high>1509.12</year-high>
            <year-low>1219.29</year-low>
            <real-time>true</real-time>
            <asset-type>I</asset-type>
        </quote>
    </quote-list>
</amtd>
```


10.4.1 Sample - Stock

Request URL

https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**>&symbol=DELL

Response:


```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <quote-list>
        <error></error>
        <quote>
            <error></error>
            <symbol>DELL</symbol>
            <description>DELL INC COM Status Alert: Delinquent</description>
            <bid>27.27</bid>
            <ask>27.28</ask>
            <bid-ask-size>2800X4600</bid-ask-size>
            <last>27.28</last>
            <last-trade-size>200</last-trade-size>
            <last-trade-date>2007-06-12 10:03:11 EDT</last-trade-date>
            <open>27.35</open>
            <high>27.49</high>
            <low>27.25</low>
            <close>27.60</close>
            <volume>2028822</volume>
            <year-high>28.04</year-high>
            <year-low>18.95</year-low>
            <real-time>true</real-time>
            <exchange>NASDAQ</exchange>
            <asset-type>E</asset-type>
            <change>-0.32</change>
            <change-percent>-1.16%</change-percent>
        </quote>
    </quote-list>
</amtd>
```


10.4.2 Sample - Option

Request URL

https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**>&symbol=GOOG_032010C510

Response:


```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <quote-list>
        <error></error>
        <quote>
            <error></error>
            <symbol>GOOG_032010C510</symbol>
            <description>GOOG Mar 20 2010 510.0 Call</description>
```

```xml
            <bid>3.50</bid>
            <ask>3.70</ask>
            <bid-ask-size>244X61</bid-ask-size>
            <last>3.70</last>
            <last-trade-size>1</last-trade-size>
            <last-trade-date>2007-06-12 10:45:27 EDT</last-trade-date>
            <open>4.00</open>
            <high>5.29</high>
            <low>3.40</low>
            <close>9.10</close>
            <volume>5017</volume>
            <strike-price>510.00</strike-price>
            <open-interest>14775</open-interest>
            <expiration-month>6</expiration-month>
            <expiration-day>15</expiration-day>
            <expiration-year>2007</expiration-year>
            <real-time>true</real-time>
            <exchange>OPRA</exchange>
            <asset-type>O</asset-type>
            <underlying-symbol>GOOG</underlying-symbol>
            <delta>0.197</delta>
            <gamma>0.008</gamma>
            <theta>-1.193</theta>
            <vega>0.139</vega>
            <rho>0.01</rho>
            <implied-volatility>68.074</implied-volatility>
            <days-to-expiration>4</days-to-expiration>
            <time-value-index>3.70</time-value-index>
            <multiplier>100.00</multiplier>
        </quote>
    </quote-list>
</amtd>
```


10.4.3 Sample - Index

Request URL

https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**>&symbol=$SPX.X

Response:


```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <quote-list>
        <error></error>
        <quote>
            <error></error>
            <symbol>$SPX.X</symbol>
            <description>NEW S & P 500 INDEX</description>
```

```
            <open>1508.82</open>
            <high>1509.12</high>
            <low>1497.69</low>
            <last>1499.66</last>
            <close>1509.12</close>
            <year-high>1509.12</year-high>
            <year-low>1219.29</year-low>
            <real-time>true</real-time>
            <asset-type>I</asset-type>
        </quote>
    </quote-list>
</amtd>
```


10.4.4 Sample - Fund

Request URL

https://apis.tdameritrade.com/apps/100/Quote?source=<**#sourceID#**>&symbol=FFFEX

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <quote-list>
        <error></error>
        <quote>
            <error></error>
            <symbol>FFFEX</symbol>
            <description>FID ABERDEEN ST TR FREEDM 2030</description>
            <nav>16.75</nav>
            <change>0.02</change>
            <real-time>true</real-time>
            <asset-type>F</asset-type>
        </quote>
    </quote-list>
</amtd>
```


10.5 Response Errors

Possible Errors

If the symbol is not valid you will get the following response:


```
<?xml version="1.0" ?>
```

```
<amtd>
    <result>OK</result>
    <quote-list>
        <error></error>
        <quote>
            <error>The Security Symbol is Invalid.</error>
            <symbol>XYZA</symbol>
            <description></description>
            <bid></bid>
            <ask></ask>
            <last></last>
            <open></open>
            <high></high>
            <low></low>
            <close></close>
            <volume>0</volume>
            <year-high></year-high>
            <year-low></year-low>
            <real-time>false</real-time>
            <asset-type>E</asset-type>
            <change></change>
            <change-percent></change-percent>
        </quote>
    </quote-list>
</amtd>
```


If the session is invalid, you will get the following response:


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```


11 SymbolLookup

Service provides the ability to lookup symbols for stocks and ETFs.

11.1 SymboLookup Request

Request URL

https://apis.tdameritrade.com/apps/100/SymbolLookup?source=<**#sourceID#**>&matchstring=**some search string**

Parameters

The matchstring parameter contains the text that you wish to search for.

11.2 SymboLookup Parameters

The quote parameter contains one or more symbols in a comma or space separated list. The TD AMERITRADE quote standards apply. The query string should be URL encoded

Quote Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
matchstring	List of symbols for which the quotes should be returned	YES	The string being searched for. Partial name of the company for example

11.3 SymboLookup Response

The response to the Quote request will be in XML format.

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <symbol-lookup-result>
        <error></error>
        <search-string></search-string>
        <symbol-result>
            <symbol></symbol>
            <description></description>
        </symbol-result>
        ......
        <symbol-result>
            <symbol></symbol>
            <description></description>
        </symbol-result>
    </symbol-lookup-result>
</amtd>
```

SymbolLookup Response Fields for STOCKS

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
symbol-lookup-result	Complex	Container for one or more <symbol-result> objects

error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**." would be returned if the symbol was not found. The error container exists within each level and returns a message specific to the layer.
search-string	String	The search string that was requested
symbol-result	Complex	Container for detailed fields for each result record
symbol	String	Symbol of the security being returned as a possible match. For example, DELL
description	String	Description of the symbol. For example, "**DELL INC COM**"

11.4 Request/Response Examples

Request URL

https://apis.tdameritrade.com/apps/100/SymbolLookup?source=<**#sourceID#**>&matchstring=bank%20of

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <symbol-lookup-result>
        <error></error>
        <search-string>bank of</search-string>
        <symbol-result>
            <symbol>BAC</symbol>
            <description>BANK OF AMERICA CORPORATION COM</description>
        </symbol-result>
        <symbol-result>
            <symbol>BAC-D</symbol>
            <description>BANK OF AMERICA CORPORATION 1/1000 6.204%D</
            description>
        </symbol-result>
        <symbol-result>
            <symbol>BAC-L</symbol>
            <description>BANK OF AMERICA CORPORATION 7.25%CNV PFD L</
            description>
        </symbol-result>
    </symbol-lookup-result>
</amtd>
```

11.5 Response Errors

Possible Errors

If search string was not specified, you will get following response:

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Error -- No Symbol entered.</error>
</amtd>
```

If source was not specified, you will get following response:

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Source parameter is required.</error>
</amtd>
```

12 PriceHistory

This service provides the ability to retrieve historical intraday and end of day quote data consolidated in OHLC format.

NOTE: in order to have the current session returned as part of the data, you must set the enddate parameter to today

12.1 PriceHistory Request

Request URL

https://apis.tdameritrade.com/apps/100/PriceHistory?source=<**##SourceID#**>&startdate=&enddate=&requestvalue=&intervaltype=&periodtype=&extended=&intervalduration=&period=&requestidentifiertype=

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as the various parameters defining the data being requested.

12.2 PriceHistory Parameters

Each PriceHistory request must contain the source id assigned by TD Ameritrade as well as the parameters indicating the information about the data that is being requested.

PriceHistory Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The SOURCE ID of the software client assigned by TD Ameritrade.	YES	String UPPER case IDs

requestidentifiertype	Indicates the type of data that will be specified in "*requestvalue*" parameter	YES	**SYMBOL**
requestvalue	Comma-separated list of symbols being requested.	YES	**NOTE:** for multiple symbol requests, use comma AND space to separate the symbols. For example: DELL, INTC. Just comma will not work
intervaltype	Indicates how the intervalduration (*frequency*) will be specified.	YES	If the *periodtype* is DAY - *intervaltype* can be **MINUTE** ONLY. MONTH - The *intervaltype* can be **DAILY**, **WEEKLY** YEAR - The *intervaltype* can be **DAILY**, **WEEKLY**, **MONTHLY** YTD - The *intervaltype* can be **DAILY**, **WEEKLY**.
intervalduration	The size of the interval (frequency) for the data being requested. For example if *intervaltype* is DAY, then 1 for 1 minute bars, 5 for 5 minute bars	YES	INTEGER. The valid values are dependent on the specified *intervaltype*: MINUTE - **1, 5, 10, 15, 30** DAILY - **1** WEEKLY - **1** MONTHLY - **1**
periodtype	Indicates the units in which the "period" parameter will be specified	NO	**DAY**,**MONTH**,**YEAR**,**YTD**
period	The number of periods for which the data is returned. For example, if periodtype=DAY and period=10, then the request is for 10 days of data	NO	INTEGER. The valid values are dependent on the specified *periodtype*: DAY - **1, 2, 3, 4, 5, 10** MONTH - **1, 2, 3, 6** YEAR - **1, 2, 3, 5 10, 15, 20** YTD - **1** If the *period* is NOT SPECIFIED and the *periodtype* is: DAY - Then default period is 10. c) MONTH - Then default period is 1. d) YEAR - Then default period is 1. e) YTD - Then default period is 1.
startdate	The start date of the data being requested (Inclusive)	NO	Date (YYYYMMDD) If the *startdate* is not specified, then it will be (*enddate* - *period*) excluding weekends and holidays.

enddate	The end date of the data being requested (Inclusive) **NOTE:** Requires *startdate* or *period*. If a *period* is given instead of *startdate*, then the request will return data for the specified period starting with specified end date.	NO	Date (YYYYMMDD) If NULL, then the default is the previous business day
extended	Indicates if extended hours data is to be included in the response. FALSE if not specified. **NOTE:** Only valid for intraday data requests.	NO	true false

12.3 PriceHistory Validation Rules

requestidentifiertype must be set to **SYMBOL**

requestvalue - if requesting multiple symbols, they must be separated by a **comma AND a space**

periodtype must be one of these values: **DAY**, **MONTH**,**YEAR**,**YTD**

period - The valid values are dependent on the *periodtype*:
If *periodtype*=DAY, then period can be **1, 2, 3, 4, 5** or **10**
If *periodtype*=MONTH, then period can be **1, 2, 3** or **6**
If *periodtype*=YEAR, then period can be **1, 2, 3, 5 10, 15,** or **20**
If *periodtype*=YTD, then period can only be **1**

intervaltype - The valid values are dependent on the *periodtype.*
- if *periodtype*=DAY, then *intervaltype* can be only be **MINUTE**.
- if *periodtype*=MONTH, then *intervaltype* can be **DAILY** or **WEEKLY**
- if *periodtype*=YEAR, then *intervaltype* can be **DAILY**, **WEEKLY**, or **MONTHLY**
- if *periodtype*=YTD, then *intervaltype* can be **DAILY** or **WEEKLY**

intervalduration - The valid values are dependent on the *intervaltype.*
- if *intervaltype*=MINUTE, then *intervalduration* can be **1, 5, 10, 15,** or **30**
- if *intervaltype*=DAILY, then *intervalduration* can only be **1**
- if *intervaltype*=WEEKLY, then *intervalduration* can only be **1**
- if *intervaltype*=MONTHLY, then *intervalduration* can only be **1**

startdate - If specified, then *period* and *periodtype* CANNOT be specified

12.4 PriceHistory Response

The PriceHistory response is in binary format represented by a symbol, number of chart bars, chart bar data (repeating series of fields) and terminator. Refer to the Streaming Data Type Definitions for type reference.

PriceHistory Response

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Symbol Count	Integer	4	Number of symbols for which data is being returned. The subsequent sections are repeated this many times
REPEATING SYMBOL DATA			
Symbol Length	Short	2	Length of the Symbol field
Symbol	String	Variable	The symbol for which the historical data is returned
Error Code	Byte	1	0=OK, 1=ERROR
Error Length	Short	2	Only returned if Error Code=1. Length of the Error string
Error Text	String	Variable	Only returned if Error Code=1. The string describing the error
Bar Count	Integer	4	# of chart bars; only if error code=0
REPEATING PRICE DATA			
close	Float	4	
high	Float	4	
low	Float	4	
open	Float	4	
volume	Float	4	in 100's
timestamp	Long	8	time in milliseconds from 00:00:00 UTC on January 1, 1970
END OF REPEATING PRICE DATA			
Terminator	Bytes	2	0xFF, 0XFF
END OF REPEATING SYMBOL DATA			

12.5 Request/Response Samples

Request:

Example requesting intraday data by specifying start and end date:
https://apis.tdameritrade.com/apps/100/PriceHistory?source=**<#sourceID#>**

&requestidentifiertype=SYMBOL&requestvalue=AMTD&intervaltype=MINUTE&intervalduration=1&extended=true&startdate=20071107&enddate=20071126

Example requesting 10 days of intraday data by specifying # of days:
https://apis.tdameritrade.com/apps/100/PriceHistory?source=**<#sourceID#>**
&requestidentifiertype=SYMBOL&requestvalue=AMTD&intervaltype=MINUTE&intervalduration=1&periodtype=DAY&period=10&extended=true

Example requesting historical (EOD) data:
https://apis.tdameritrade.com/apps/100/PriceHistory?source=**<#sourceID#>**
&requestidentifiertype=SYMBOL&requestvalue=AMTD&intervaltype=DAILY&intervalduration=1&startdate=20071122&enddate=20071126

Response (To the EOD request):

```
00 00 00 01 00 04 41 4D 54 44 00 00 00 00 02 41
97 33 33 41 99 5C 29 41 90 3D 71 41 91 D7 0A 47
0F C6 14 00 00 01 16 6A E0 68 80 41 93 B4 05 41
97 1E B8 41 90 7A E1 41 96 8F 57 46 E6 2E 80 00
00 01 16 7A 53 7C 80 FF FF
```

NOTE: November 22, 2007 is a holiday, which is why only 2 days are returned for the above request

DATA	TYPE	DESCRIPTION
00 00 00 01	4 bytes	Symbol Count =1
00 04	2 bytes	Symbol Length = 4
41 4D 54 44	4 bytes	Symbol = AMTD
00	1 byte	Error code = 0 (OK)
00 00 00 02	4 bytes	Bar Count = 2
FIRST BAR		
41 97 33 33	4 bytes	Close = 18.90
41 99 5C 29	4 bytes	High = 19.17
41 90 3D 71	4 bytes	Low = 18.03
41 91 D7 0A	4 bytes	Open = 18.23
47 0F C6 14	4 bytes	Volume = 3,680,608
00 00 01 16 6A E0 68 80	8 bytes	Timestamp = November 23,2007
SECOND BAR		
41 93 B4 05	4 bytes	Close = 18.4629
41 97 1E B8	4 bytes	High = 18.89
41 90 7A E1	4 bytes	Low = 18.06
41 96 8F 57	4 bytes	Open = 18.82
46 E6 2E 80	4 bytes	Volume = 2,946,325
00 00 01 16 7A 53 7C 80	8 bytes	Timestamp = November 26,2007
TERMINATOR		
FF FF	2 bytes	

12.6 PriceHistory Error Response

Responses which contain errors will include an HTTP response code in the header indicating the type of error (4xx, 5xx, etc.), as appropriate to the type of error. The body of the response will contain a single UTF-String with the error message.

Possible Errors

Validation failed for 1 of 1 requests.Your request is not valid.

Internal Server Error

13 VolatilityHistory

This service provides the ability to retrieve historical implied volatility data.

13.1 VolatilityHistory Request

Request URL

https://apis.tdameritr ade.com/apps/100/VolatilityHistory?source=<**##SourceID#**>&requestidentifiertype=&requestvalue=&volatilityhistorytype=&intervaltype=&intervalduration=&periodtype=&period=&startdate=&enddate=&daystoexpiration=&surfacetypeidentifier=&surfacetypevalue=

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as the various parameters defining the data being requested.

13.2 VolatilityHistory Parameters

Each VolatilityHistory request must contain the source id assigned by TD Ameritrade as well as the parameters indicating the information about the data that is being requested.

PriceHistory Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The SOURCE ID of the software client assigned by TD Ameritrade.	YES	String UPPER case IDs
requestidentifiertype	Indicates the type of data that will be specified in "*requestvalue*" parameter	YES	**SYMBOL**
requestvalue	The actual symbol being requested.	YES	

volatilityhistorytype	The type of volatility history being requested	YES	**I** or **H** - **I**=Implied (calculated) **H**= Historical (actual)
intervaltype	Indicates the granularity of the data being requested. Used to specify the consolidation of the data into 1-day, 1-week, etc	YES	If the *periodtype* is DAY - *intervaltype* can be **DAILY** ONLY. MONTH - The *intervaltype* can be **DAILY**, **WEEKLY, MONTHLY** YEAR - The *intervaltype* can be **WEEKLY**, **MONTHLY** YTD - The *intervaltype* can be **WEEKLY**, **MONTHLY**
intervalduration	The size of the interval (frequency) for the data being requested.	YES	Always set to **1**
periodtype	Indicates the units in which the "period" parameter will be specified	YES	**DAY, WEEK, MONTH,YEAR,YTD**
period	The number of periods for which the data is returned. For example, if periodtype=DAY and period=10, then the request is for 10 days of data	NO	INTEGER. The valid values are dependent on the specified *periodtype*: DAY - **1, 2, 3, 4, 5, 10** MONTH - **1, 2, 3, 6** YEAR - **1, 2, 3, 5 10, 15, 20** YTD - **1** If the *period* is NOT SPECIFIED and the *periodtype* is: DAY - Then default period is 10. c) MONTH - Then default period is 1. d) YEAR - Then default period is 1. e) YTD - Then default period is 1.
startdate	The start date of the data being requested (Inclusive)	NO	Date (YYYYMMDD) If the *startdate* is not specified, then it will be (*enddate* - *period*) excluding weekends and holidays.
enddate	The end date of the data being requested (Inclusive) **NOTE:** Requires *startdate* or *period*. If a *period* is given instead of *startdate*, then the request will return data for the specified period starting with specified end date.	NO	Date (YYYYMMDD) If NULL, then the default is the previous business day
daystoexpiration	Specifies the period for which the implied volatility is calculated.	NO	**30, 60, 90, 120, 150, 180** Default: 30

surfacetypeidentifier	Specifies the type of implied volatility data to be returned.	YES	**DELTA** - The ratio comparing the change in the price of the underlying asset to the corresponding change in the price of a derivative. The Contract Type will be "CALL" if the delta is positive. The Contract Type will be "PUT" if the delta is negative. **DELTA_WITH_COMPOSITE** - The mean of 2 delta values, one CALL and one PUT **SKEW** - Difference between the implied volatilities at 2 delta values. The Contract Type will be "CALL" if the delta is positive. The Contract Type will be "PUT" if the delta is negative.
surfacetypevalue	1 integer if DELTA, 2 integers for composite or skew	YES	list of integers (comma-separated). **NOTE:** Make sure to use URLEncoded comma (%2C)

13.3 VolatilityHistory Validation Rules

requestidentifiertype must be set to **SYMBOL**

volatilityhistorytype - Must be either **I** or **H**

periodtype - If specified, must be one of these values: **DAY, WEEK, MONTH,YEAR,YTD**

period - If specified, the valid values are dependent on the *periodtype*:
If *periodtype*=DAY, then period can be **1, 2, 3, 4, 5** or **10**
If *periodtype*=MONTH, then period can be **1, 2, 3** or **6**
If *periodtype*=YEAR, then period can be **1, 2, 3, 5 10, 15,** or **20**
If *periodtype*=YTD, then period can only be **1**

intervaltype - The valid values are dependent on the *periodtype.*
- if *periodtype*=DAY, then *intervaltype* can be only be **DAILY**
- if *periodtype*=MONTH, then *intervaltype* can be **DAILY, WEEKLY**, or **MONTHLY**
- if *periodtype*=YEAR, then *intervaltype* can be **WEEKLY** or **MONTHLY**
- if *periodtype*=YTD, then *intervaltype* can be **WEEKLY** or **MONTHLY**

intervalduration - Must be set to **1**

startdate - If specified, then *period* and *periodtype* CANNOT be specified

daystoexpiration - If specified, must be one of these values: **30, 60, 90, 120, 150, 180**

surfacetypeidentifier - Must be one of these values: **DELTA, DELTA_WITH_COMPOSITE, SKEW**

surfacetypevalue - Requirements are dependent on the *surfacetypeidentifier*:
If *surfacetypeidentifier*=DELTA, then must be one of these values: **30, 40, 50, 60, 70, -30, -40, -50, -60, -70**
If *surfacetypeidentifier*=DELTA_WITH_COMPOSITE, then must be TWO values: one +ve, one -ve
If *surfacetypeidentifier*=SKEW, then must be TWO values with the same sign: +ve, +ve or -ve, -ve

13.4 VolatilityHistory Response

The VolatilityHistory response is in binary format represented by a repeating series of fields and terminator. Refer to the Streaming Data Type Definitions for type reference.

VolatilityHistory Response

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Symbol Count	Integer	4	Number of symbols for which data is being returned. The subsequent sections are repeated this many times
REPEATING DATA			
Symbol Length	Short	2	Length of the Symbol field
Symbol	UTF-String	Variable	The symbol for which the historical data is returned
Error Code	Byte	1	0=OK, 1=ERROR
Error Length	Short	2	Length of the Error String
Error	UTF-String	Variable	Only returned if Error Code=1. The string describing the error
Value Count	Integer	4	# of values being returned; only if error code=0
REPEATING DATA			
Value	Float	4	
Timestamp	Long	8	time in milliseconds from 00:00:00 UTC on January 1, 1970
END OF REPEATING DATA			
Terminator	Bytes	2	0xFF, 0XFF

13.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/VolatilityHistory?source=**<#sourceID#>**&requestidentifiertype=SYMBOL&requestvalue=AMTD&intervaltype=DAILY&intervalduration=1&periodtype=DAY&period=3&enddate=20071126&daystoexpiration=30&volatilityhistorytype=I&surfacetypeidentifier=DELTA_WITH_COMPOSITE&surfacetypevalue=50%2C-50

Response (To the EOD request):

```
00 00 00 01 00 04 41 4D 54 44 00 00 00 00 03 3F
10 44 FA 00 00 01 16 60 93 B0 80 3F 08 2F 5A 00
00 01 16 6A E0 68 80 3F 15 00 04 00 00 01 16 7A
53 7C 80 FF FF
```

NOTE: November 22, 2007 is a holiday

DATA	TYPE	DESCRIPTION
00 00 00 01	4 bytes	Symbol Count =1
00 04	2 bytes	Symbol Length = 4
41 4D 54 44	4 bytes	Symbol = AMTD
00	1 byte	Error code = 0 (OK)
00 00 00 03	4 bytes	Value Count = 3
FIRST VALUE		
3F 10 44 FA	4 bytes	Value = 0.5635525
00 00 01 16 60 93 B0 80	8 bytes	Timestamp = November 21,2007
SECOND VALUE		
3F 08 2F 5A	4 bytes	Value = 0.5319725
00 00 01 16 6A E0 68 80	8 bytes	Timestamp = November 23,2007
THIRD VALUE		
3F 15 00 04	4 bytes	Value = 0.5820315
00 00 01 16 7A 53 7C 80	8 bytes	Timestamp = November 26,2007
TERMINATOR		
FF FF	2 bytes	

13.6 VolatilityHistory Error Response

Responses which contain errors will include an HTTP response code in the header indicating the type of error (4xx, 5xx, etc.), as appropriate to the type of error. The body of the response will contain a single UTF-String with the error message.

Possible Errors

Validation failed for 1 of 1 requests.Your request is not valid.

Internal Server Error

14 OptionChain

Service provides option chain information with or without quotes on a single underlying equity symbol.

14.1 OptionChain Request

Request URL

https://apis.tdameritrade.com/apps/200/OptionChain?source=<**#sourceID#**>&symbol=<**#symbol#**>

Parameters

The symbol parameter contains the underlying symbol for the option chain being requested.

14.2 OptionChain Parameters

Quote Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
symbol	The underlying symbol for the option chain	YES	Can be a stock, or index symbol. Standard TD AMERITRADE symbology applies.
type	The option chain type. If the type is not included, then both calls and puts will be returned.	NO	If type is not specified, calls and puts are returned. Otherwise: **C** - Only Calls **P** - Only Puts **VCS** - Vertical Call Spread **VPS** - Vertical Put Spread **CCS** - Calendar Call Spread **CPS** - Calendar Put Spread **BW** - Buy/Write **STDL** - Straddle **STGL** - Strangle
interval	The interval or Spread for Vertical Spreads and Strangles	NO	Only required for Vertical Spreads and Strangles requests

strike	The option chain strike price. If the strike price is not included, then all strike prices will be returned.	NO	a valid strike price, e.g. 15 or 17.5 A valid strike price will return those calls and puts for that strike price. An invalid strike price will return no data.
expire	The option expiration. If the expiration is not included or has a value of AL, then all standard months and leap months will be included.	NO *	**al, a, l, w** or a date matching the YYYYMM pattern AL = ALL+Leaps, including Weeklies A = All standard expirations, including Weeklies L = leap expirations W = weeklies only (this is only supported for a few index options). YYYYMM = returns only the option month requests. A date that doesn't have options will return an empty data set. * Required for Vertical Spreads
range	The Strike Price range of the options to be returned	NO	**N** - Near The Money I - In The Money **O** - Out of The Money **ALL** - All The following are for use with Straddes and Strangles: **SNK** - Strikes Near Market **SBK** - Strikes Below Market **SAK** - Strikes Above Market
neardate	for use with Calendar spreads	NO	YYYYMM Format date
fardate	for use with Calendar spreads	NO	YYYYMM Format date
quotes	A flag to request quote data. The default behavior is not to send quote data.	NO	**true/false**

14.3 OptionChain Response

The response to the OptionChain request will be in XML format. The Structure of the response will vary depending on whether or not quotes were requested with the chain and is documented at:

OptionChain - With Quotes
OptionChain - Without Quotes

14.3.1 OptionChain Response - Without Quotes

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <option-chain-results>
        <error></error>
        <symbol></symbol>
        <option-date>
            <date></date>
            <expiration-type></expiration-type>
            <days-to-expiration></days-to-expiration>
            <option-strike>
                <strike-price></strike-price>
                <standard-option></standard-option>
                <put>
                        <symbol></symbol>
                        <description></description>
                </put>
                <call>
                        <symbol></symbol>
                        <description></description>
                </call>
            </option-strike>
            .......
        </option-date>
    </option-chain-results>
</amtd>
```

OptionChain Response Fields for no-quote request

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
option-chain-results	Complex	Container for all the option chain info
error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**."
symbol	String	The Symbol of the security for which the option chain is being retrieved. For example, **AMTD**
option-date	Complex	Container for all the option chain data for a particular option expiration date
date	String	The options expiration date in YYYYMMDD format
expiration-type	String	**R or L** - Regular or LEAP

days-to-expiration	Integer	Number of days till the options expire
option-strike	Complex	Container for the options at a given strike price
strike-price	String	Option Strike Price
standard-option	String	true/false - indicates if the options in question are standard or non-standard
put	Complex	Container for the PUT option symbol fields at the given date and strike price
symbol	String	The PUT Option Symbol. For example: **SPY_112109P111**
description	String	The PUT Option Symbol Description. For example: **SPY Nov 21 2009 111.0 Put**
call	Complex	Container for the PUT option symbol fields at the given date and strike price
symbol	String	The CALL Option Symbol. For example: **SPY_112109C111**
description	String	The CALL Option Symbol Description. For example: **SPY Nov 21 2009 111.0 Call**

14.3.2 OptionChain Response - With Quotes

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <option-chain-results>
        <error></error>
        <symbol></symbol>
        <description></description>
        <bid></bid>
        <ask></ask>
        <bid-ask-size></bid-ask-size>
        <last></last>
        <open></open>
        <high></high>
        <low></low>
        <close></close>
        <volume></volume>
        <change></change>
        <quote-punctuality></quote-punctuality>
        <time></time>
        <option-date>
            <date></date>
            <expiration-type></expiration-type>
            <days-to-expiration></days-to-expiration>
            <option-strike>
```

```
            <strike-price></strike-price>
            <standard-option></standard-option>
            <put>
                <option-symbol></option-symbol>
                <description></description>
                <bid></bid>
                <ask></ask>
                <bid-ask-size></bid-ask-size>
                <last></last>
                <last-trade-date></last-trade-date>
                <volume></volume>
                <open-interest></open-interest>
                <real-time></real-time>
                <underlying-symbol></underlying-symbol>
                <delta></delta>
                <gamma></gamma>
                <theta></theta>
                <vega></vega>
                <rho></rho>
                <implied-volatility></implied-volatility>
                <time-value-index></time-value-index>
                <multiplier></multiplier>
                <change></change>
                <change-percent></change-percent>
                <in-the-money></in-the-money>
                <near-the-money></near-the-money>
                <theoretical-value></theoretical-value>
                <deliverable-list>
                    <notes-description></notes-description>
                    <cash-in-lieu-dollar-amount></cash-in-lieu-dollar-amount>
                    <cash-dollar-amount></cash-dollar-amount>
                    <index-option></index-option>
                    <row>
                        <symbol></symbol>
                        <shares></shares>
                    </row>
                    <row>
                        <symbol></symbol>
                        <shares></shares>
                    </row>
                    ...
                </deliverable-list>
            </put>
            <call>
                .... (Same fields as in the PUT record)
            </call>
        </option-strike>
        ...... (More option-strike records)
    </option-date>
    ...... (More option-date records)
</option-chain-results>
</amtd>
```

OptionChain Response Fields for with-quote request

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
option-chain-results	Complex	Container for all the option chain info
error	String	Contains an error message, if any. For example, "**The Security Symbol is Invalid**."
symbol	String	Symbol of the security being quoted. For example, **DELL**
description	String	Contains a description of the symbol. For example, "**TD AMERITRADE HLDG CORP COM**"
bid	Double	Underlying Symbol - BID
ask	Double	Underlying Symbol - ASK
bid-ask-size	String	Underlying Symbol - bid/ask Size The value is displayed as "bid qty X ask qty". For example, **390X41**
last	Double	Underlying Symbol - The price of the last trade
open	Double	Underlying Symbol - The price of the first trade at normal market open.
high	Double	Underlying Symbol - The highest price trade for the symbol during the normal trading session
low	Double	Underlying Symbol - The lowest price trade for the symbol during the normal trading session.
close	Double	Underlying Symbol - The price of the last trade for the symbol at the end of the previous trading session.
volume	Integer	Underlying Symbol - The number of shares traded for the symbol.
change	Double	Underlying Symbol - CHANGE
quote-punctuality	String	R or D - Real-time or Delayed
time	String	Last Trade Time for the quote
option-date	Complex	Container for all the option chain data for a particular option expiration date
date	String	The options expiration date in YYYYMMDD format
expiration-type	String	**R or L** - Regular or LEAP
days-to-expiration	Integer	Number of days till the options expire

option-strike	Complex	Container for the options at a given strike price
strike-price	Double	Option Strike Price
standard-option	String	**true/false** - indicates if the options in question are standard or non-standard
put	Complex	Container for the fields describing the PUT option symbol at the given date and strike price
call	Complex	Container for the fields describing the CALL option symbol at the given date and strike price
option-symbol	String	The option symbol
description	String	The option symbol description
bid	Double	
ask	Double	
bid-ask-size	String	
last	Double	
last-trade-date	String	
volume	Integer	
open-interest	Integer	
real-time	String	
underlying-symbol	String	
delta	Double	
gamma	Double	
theta	Double	
vega	Double	
rho	Double	
implied-volatility	Double	
time-value-index	Double	
multiplier	Integer	
change	Double	
change-percent	String	

in-the-money	String	**true/false**
near-the-money	String	**true/false**
theoretica-value	Double	
deliverable-list	Complex	Container for Deliverables
notes-description	String	For non-standard options, this describes what needs to be delivered. For example:"**$600.0 cash in lieu of shares, 100 shares of AMTD**"
cash-in-lieu-dollar-amount	Double	
cash-dollar-amount	Double	
index-option	String	**true/false**
row	Complex	Container
symbol	String	
shares	Integer	

14.4 Request/Response Examples

14.4.1 Sample - Without Quotes

Request URL

https://apis.tdameritrade.com/apps/**200**/OptionChain?source=<**#sourceID#**>&symbol=AMTD&expire=200709

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <option-chain-results>
        <error></error>
        <symbol>AMTD</symbol>
        <description>TD AMERITRADE HLDG CORP COM</description>
        <option-date>
            <date>20070922</date>
            <expiration-type>R</expiration-type>
            <days-to-expiration>4</days-to-expiration>
            <option-strike>
                <strike-price>12.50</strike-price>
```

```
        <standard-option>true</standard-option>
        <put>
        <symbol>AMTD_092207P12.5</symbol>
        <description>AMTD Sep 22 2007 12.5 Put</description>
        </put>
        <call>
        <symbol>AMTD_092207C12.5</symbol>
        <description>AMTD Sep 22 2007 12.5 Call</description>
        </call>
    </option-strike>
    <option-strike>
        <strike-price>15.00</strike-price>
        <standard-option>true</standard-option>
        <put>
        <symbol>AMTD_092207P15</symbol>
        <description>AMTD Sep 22 2007 15.0 Put</description>
        </put>
        <call>
        <symbol>AMTD_092207C15</symbol>
        <description>AMTD Sep 22 2007 15.0 Call</description>
        </call>
    </option-strike>
    <option-strike>
        <strike-price>20.00</strike-price>
        <standard-option>true</standard-option>
        <put>
        <symbol>AMTD_092207P20</symbol>
        <description>AMTD Sep 22 2007 20.0 Put</description>
        </put>
        <call>
        <symbol>AMTD_092207C20</symbol>
        <description>AMTD Sep 22 2007 20.0 Call</description>
        </call>
    </option-strike>
    <option-strike>
        <strike-price>22.50</strike-price>
        <standard-option>true</standard-option>
        <put>
        <symbol>AMTD_092207P22.5</symbol>
        <description>AMTD Sep 22 2007 22.5 Put</description>
        </put>
        <call>
        <symbol>AMTD_092207C22.5</symbol>
        <description>AMTD Sep 22 2007 22.5 Call</description>
        </call>
    </option-strike>
    <option-strike>
        <strike-price>25.00</strike-price>
        <standard-option>true</standard-option>
        <put>
        <symbol>AMTD_092207P25</symbol>
        <description>AMTD Sep 22 2007 25.0 Put</description>
        </put>
```

```
                <call>
                <symbol>AMTD_092207C25</symbol>
                <description>AMTD SEP 22 2007 25.0 Call</description>
                </call>
            </option-strike>
        </option-date>
    </option-chain-results>
</amtd>
```


14.4.2 Sample - With Quotes

Request URL

[https://apis.tdameritrade.com/apps/**200**/OptionChain?source=<**#sourceID#**>&symbol=AMTD&expire=200709&quotes=true](https://apis.tdameritrade.com/apps/200/OptionChain?source=<#sourceID#>&symbol=AMTD&expire=200709&quotes=true)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <option-chain-results>
        <error></error>
        <symbol>AMTD</symbol>
        <description>TD AMERITRADE HLDG CORP COM</description>
        <bid>17.94</bid>
        <ask>17.95</ask>
        <bid-ask-size>400X2600</bid-ask-size>
        <last>17.9401</last>
        <open>17.94</open>
        <high>18.12</high>
        <low>17.69</low>
        <close>17.78</close>
        <volume>2402786.0</volume>
        <change>0.1601</change>
        <quote-punctuality>R</quote-punctuality>
        <time>14:03:45</time>
        <option-date>
            <date>20071020</date>
            <expiration-type>R</expiration-type>
            <days-to-expiration>32</days-to-expiration>
            <option-strike>
                <strike-price>10.00</strike-price>
                <standard-option>true</standard-option>
                <put>
                    <option-symbol>AMTD_102007P10</option-symbol>
                    <description>AMTD Oct 20 2007 10.0 Put</description>
                    <bid></bid>
                    <ask>0.05</ask>
                    <bid-ask-size>0X619</bid-ask-size>
                    <last></last>
                    <volume>0</volume>
                    <open-interest>0</open-interest>
```

```
        <real-time>false</real-time>
        <underlying-symbol>AMTD</underlying-symbol>
        <delta>-0.013</delta>
        <gamma>0.007</gamma>
        <theta>-0.003</theta>
        <vega>0.002</vega>
        <rho>0.00</rho>
        <implied-volatility>95.18</implied-volatility>
        <time-value-index>0.03</time-value-index>
        <multiplier></multiplier>
        <change>0.00</change>
        <change-percent></change-percent>
        <in-the-money>false</in-the-money>
        <near-the-money>false</near-the-money>
        <theoretical-value>0.03</theoretical-value>
        <deliverable-list>
            <row>
                <cash-in-lieu-dollar-amount></cash-in-lieu-dollar-amount>
                <cash-dollar-amount></cash-dollar-amount>
                <index-option>false</index-option>
                <symbol>AMTD</symbol>
                <shares>100</shares>
            </row>
        </deliverable-list>
    </put>
    <call>
        <option-symbol>AMTD_102007C10</option-symbol>
        <description>AMTD Oct 20 2007 10.0 Call</description>
        <bid>7.90</bid>
        <ask>8.10</ask>
        <bid-ask-size>210X121</bid-ask-size>
        <last>7.50</last>
        <last-trade-date>2007-09-07 10:03:34 EDT</last-trade-date>
        <volume>0</volume>
        <open-interest>139</open-interest>
        <real-time>false</real-time>
        <underlying-symbol>AMTD</underlying-symbol>
        <delta>0.991</delta>
        <gamma>0.005</gamma>
        <theta>-0.003</theta>
        <vega>0.001</vega>
        <rho>0.009</rho>
        <implied-volatility>88.79</implied-volatility>
        <time-value-index>-0.44</time-value-index>
        <multiplier></multiplier>
        <change>0.00</change>
        <change-percent></change-percent>
        <in-the-money>true</in-the-money>
        <near-the-money>false</near-the-money>
        <theoretical-value>8.00</theoretical-value>
        <deliverable-list>
            <row>
                <cash-in-lieu-dollar-amount></cash-in-lieu-dollar-amount>
```

```
                        <cash-dollar-amount></cash-dollar-amount>
                        <index-option>false</index-option>
                        <symbol>AMTD</symbol>
                        <shares>100</shares>
                    </row>
                </deliverable-list>
            </call>
        </option-strike>
        ...... (More option-strike records)
    </option-date>
</option-chain-results>
</amtd>
```

14.5 OptionChain Error Response

Possible Errors

If the symbol is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Please enter a valid symbol.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

15 BinaryOptionChain

Service provides option chain information with or without quotes on a single underlying equity symbol in Binary format. Unlike the OptionChain request, BnaryOptionChain does not support complex option chains

15.1 BinaryOptionChain Request

Request URL

https://apis.tdameritrade.com/apps/200/BinaryOptionChain?source=<**#sourceID#**>&symbol=<**#symbol#**>

Parameters

The symbol parameter contains the underlying symbol for the option chain being requested.

15.2 BinaryOptionChain Parameters

Quote Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
symbol	The underlying symbol for the option chain	YES	Can be a stock, or index symbol. Standard TD AMERITRADE symbology applies.
type	The option chain type. If the type is not included, then both calls and puts will be returned.	NO	If type is not specified, calls and puts are returned. Otherwise: **C** - Only Calls **P** - Only Puts
strike	The option chain strike price. If the strike price is not included, then all strike prices will be returned.	NO	a valid strike price, e.g. 15 or 17.5 A valid strike price will return those calls and puts for that strike price. An invalid strike price will return no data.
expire	The option expiration. If the expiration is not included or has a value of AL, then all standard months and leap months will be included.	NO *	**al, a, l, w** or a date matching the YYYYMM pattern AL = ALL+Leaps, including Weeklies A = All standard expirations, including Weeklies L = leap expirations W = weeklies only (this is only supported for a few index options). YYYYMM = returns only the option month requests. A date that does not have options will return an empty data set.
range	The Strike Price range of the options to be returned	NO	**N** - Near The Money **I** - In The Money **O** - Out of The Money **ALL** - All The following are for use with Straddes and Strangles: **SNK** - Strikes Near Market **SBK** - Strikes Below Market **SAK** - Strikes Above Market

quotes	A flag to request quote data. The default behavior is not to send quote data.	NO	**true/false** - Default is false if not specified

15.3 BinaryOptionChain Response

The response to the BinaryOptionChain request will be in Binary format. The Structure of the response will vary depending on whether or not quotes were requested with the chain and is documented at:

BinaryOptionChain - With Quotes
BinaryOptionChain - Without Quotes

15.3.1 BinaryOptionChain Response - Without Quotes

The BinaryOptionChain response is in binary format. Refer to the Streaming Data Type Definitions for type reference

BinaryOptionChain Response - without Quotes

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Error Code	Byte	1	0=OK, 1=ERROR
Error Length	Short	2	Only returned if Error Code=1. Length of the Error string
Error Text	String	Variable	Only returned if Error Code=1. The string describing the error
Underlying Symbol Length	Short	2	Length of the Underlying Symbol field
Underlying Symbol	String	Variable	The Underlying symbol for which the option chain data is being returned
Underlying Symbol Description Length	Short	2	Length of the Underlying Symbol Description field
Underlying Symbol Description	String	Variable	The description of the Underlying Symbol (Company or Index Name)
Row Count	Integer	4	Number of option symbols returned
REPEATING Block for each Option Symbol			
Option Date Length	Short	2	
Option Date	String	Variable	YYYYMMDD

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Expiration Type Length	Short	2	
Expiration Type	String	Variable	**R or L** - Regular or LEAP
Strike Price	Double	8	
Standard Option Flag	Byte	1	1 = true, 0=false
Put/Call Indicator	Char	2	**P** or **C** (2 bytes because it is Unicode)
Option Symbol Length	Short	2	
Option Symbol	String	Variable	
Option Description Length	Short	2	
END OF REPEATING DATA			

15.3.2 BinaryOptionChain Response - With Quotes

The BinaryOptionChain response is in binary format. Refer to the Streaming Data Type Definitions for type reference

BinaryOptionChain Response - with Quotes

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Error Code	Byte	1	0=OK, 1=ERROR
Error Length	Short	2	Only returned if Error Code=1. Length of the Error string
Error Text	String	Variable	Only returned if Error Code=1. The string describing the error
Underlying Symbol Length	Short	2	Length of the Underlying Symbol field
Underlying Symbol	String	Variable	The Underlying symbol for which the option chain data is being returned
Underlying Symbol Description Length	Short	2	Length of the Underlying Symbol Description field

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Underlying Symbol Description	String	Variable	The description of the Underlying Symbol (Company or Index Name)
Bid	Double	8	
Ask	Double	8	
Bid/Ask Size Length	Short	2	
Bid/Ask Size	String	Variable	
Last	Double	8	
Open	Double	8	
High	Double	8	
Low	Double	8	
Close	Double	8	
Volume	Double	8	
Change	Double	8	
Real Time Quote Flag	Char	2	**R** or **D** (2 bytes because it is Unicode)
Quote Time Length	Short	2	
Quote Time	String	Variable	
Row Count	Integer	4	Number of option symbols returned
REPEATING Block for each Option Symbol			
Option Date Length	Short	2	
Option Date	String	Variable	YYYYMMDD
Expiration Type Length	Short	2	

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Expiration Type	String	Variable	**R or L** - Regular or LEAP
Strike Price	Double	8	
Standard Option Flag	Byte	1	1 = true, 0=false
Put/Call Indicator	Char	2	**P** or **C** (2 bytes because it is Unicode)
Option Symbol Length	Short	2	
Option Symbol	String	Variable	
Option Description Length	Short	2	
Option Description	String	Variable	
Bid	Double	8	
Ask	Double	8	
Bid/Ask Size Length	Short	2	
Bid/Ask Size	String	Variable	
Last	Double	8	
Last Trade Size Length	Short	2	
Last Trade Size	String	Variable	
Last Trade Date Length	Short	2	
Last Trade Date	String	Variable	
Volume	Long	8	
Open Interest	Integer	4	

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
Real Time Quote Flag	Byte	1	1 = true, 0=false
Underlying Symbol Length	Short	2	
Underlying Symbol	String	Variable	
Delta	Double	8	
Gamma	Double	8	
Theta	Double	8	
Vega	Double	8	
Rho	Double	8	
Implied Volatility	Double	8	
Time Value Index	Double	8	
Multiplier	Double	8	
Change	Double	8	
Change Percentage	Double	8	
In The Money Flag	Byte	1	1 = true, 0=false
Near The Money Flag	Byte	1	1 = true, 0=false
Theoretical Value	Double	8	
Deliverable Note Length	Short	2	
Deliverable Note	String	Variable	
CIL Dollar Amount	Double	8	

FIELD	TYPE	LENGTH (8 BIT BYTES)	DESCRIPTION
OP Cash Dollar Amount	Double	8	
Index Option Flag	Byte	1	1 = true, 0=false
Number of Deliverables	Integer	4	
REPEATING Block for each Deliverable			
Deliverable Symbol Length	Short	2	
Deliverable Symbol	String	Variable	
Deliverable Shares	Integer	4	
END OF REPEATING Deliverable Block			
END OF REPEATING Option Symbol Block			
END OF RESPONSE			

15.4 Request/Response Examples

15.4.1 Sample - Without Quotes

Request URL

https://apis.tdameritrade.com/apps/**200**/BinaryOptionChain?source=<**#sourceID#**>&symbol=AMTD&expire=201005&strike=17.5

Response:

```
00 00 04 41 4D 54 44 00 1B 54 44 20 41 4D 45 52 49 54 52 41 44 45 20 48 4C
44 47 20 43 4F 52 50 20 43 4F 4D 00 00 00 02 00 08 32 30 31 30 30 35 32 32
00 01 52 40 31 80 00 00 00 00 00 01 00 43 00 11 54 51 41 5F 32 30 31 30 30
35 32 32 43 31 37 2E 35 00 19 54 51 41 20 4D 61 79 20 32 32 20 32 30 31 30
20 31 37 2E 35 20 43 61 6C 6C 00 08 32 30 31 30 30 35 32 32 00 01 52 40 31
80 00 00 00 00 00 01 00 50 00 11 54 51 41 5F 32 30 31 30 30 35 32 32 50 31
37 2E 35 00 18 54 51 41 20 4D 61 79 20 32 32 20 32 30 31 30 20 31 37 2E 35
20 50 75 74
```

NOTE: November 22, 2007 is a holiday, which is why only 2 days are returned for the above request

DATA	TYPE	DESCRIPTION
00	1 byte	Error Code = 0 (OK)
00 04	2 bytes	Underlying Symbol Length = 4
41 4D 54 44	4 bytes	Underlying Symbol = AMTD
00 1B	2 bytes	Underlying Symbol Description Length = 27
54 44 20 41 4D 45 52 49 54 52 41 44 45 20 48 4C 44 47 20 43 4F 52 50 20 43 4F 4D	27 bytes	Underlying Symbol Description = TD AMERITRADE HLDG CORP COM

00 00 00 02	4 bytes	Row Count = 2
FIRST OPTION		
00 08	2 bytes	Option Date Length = 8
32 30 31 30 30 35 32 32	8 bytes	Option Date = 20100522
00 01	2 bytes	Expiration Type Length = 1
52	1 byte	Expiration Type= R
40 31 80 00 00 00 00 00	8 bytes	Strike Price = 17.50
01	4 bytes	Standard Option Flag = 1
00 43	8 bytes	Put/Call Indicator = C
00 0F	2 bytes	Option Symbol Length = 15
54 51 41 5F 30 35 32 32 31 30 43 31 37 2E 35	15 bytes	Option Symbol = TQA_052210C17.5
00 19	2 bytes	Option Symbol Description Length = 25
54 51 41 20 4D 61 79 20 32 32 20 32 30 31 30 20 31 37 2E 35 20 43 61 6C 6C	25 bytes	Option Symbol Description = TQA May 22 2010 17.5 Call
SECOND OPTION		
00 08	2 bytes	Option Date Length = 8
32 30 31 30 30 35 32 32	8 bytes	Option Date = 20100522
00 01	2 bytes	Expiration Type Length = 1
52	1 byte	Expiration Type= R
40 31 80 00 00 00 00 00	8 bytes	Strike Price = 17.50
01	4 bytes	Standard Option Flag = 1
00 50	8 bytes	Put/Call Indicator = P
00 0F	2 bytes	Option Symbol Length = 15
54 51 41 5F 32 30 31 30 30 35 32 32 50 31 37 2E 35	15 bytes	Option Symbol = TQA_052210P17.5
00 18	2 bytes	Option Symbol Description Length = 24
54 51 41 20 4D 61 79 20 32 32 20 32 30 31 30 20 31 37 2E 35 20 43 61 6C 6C	24 bytes	Option Symbol Description = TQA May 22 2010 17.5 Put
END OF REPEATING DATA		

15.4.2 Sample - With Quotes

Request URL

https://apis.tdameritrade.com/apps/**200**/BinaryOptionChain?source=<**#sourceID#**>&symbol=AMTD&expire=201005&strike=17.5&quotes=true

Response:

Sample not currently available

15.5 BinaryOptionChain Error Response

Possible Errors

If the symbol is not valid you will get

Response:

`01 00 0E 49 6E 76 61 6C 69 64 20 53 79 6D 62 6F 6C`

NOTE: November 22, 2007 is a holiday, which is why only 2 days are returned for the above request

DATA	TYPE	DESCRIPTION
01	1 byte	Error Code = 1 (FAILURE)
00 0E	2 bytes	Error Length = 14
49 6E 76 61 6C 69 64 20 53 79 6D 62 6F 6C	14 bytes	Error = Invalid Symbol

NOTE: Invalid Session response will show up as an Internal Server Error with error code 500 in the HTTP Headers

16 Balances and Positions

Service provides detailed information on positions and balances in the account

16.1 B & P Request

Request URL

https://apis.tdameritrade.com/apps/100/BalancesAndPositions?source=<**#sourceID#**>

Parameters

The only parameter required is the source assigned by TD Ameritrade. You can also specify the account id for which the data is to be returned and the type of data to be returned - only balances, or only positions. Both are returned if not specified. If account is not specified, then the data for the default "associated" account will be returned.

16.2 B & P Parameters

The only parameter required is the source assigned by TD Ameritrade. The type and account id parameters are optional and can be specified togather

BalancesAndPositions Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES

source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account
type	Specifies the type of data to be returned - balances or positions	NO	**b** or **p.** If not specified, then both balances and positions for the account will be returned
suppressquotes	Flag indicating whether or not quotes should be suppressed on the positions	NO	**true/false**. Defaults to false.

16.3 B & P Response

The response to the Balances And Positions request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <balance>
        <error></error>
        <account-id></account-id>
        <day-trader></day-trader>
        <round-trips></round-trips>
        <resticted-closing-transactions-only></resticted-closing-transactions-only>
        <cash-balance>
            <initial></initial>
            <current></current>
            <change></change>
        </cash-balance>
        <money-market-balance>
            <initial></initial>
            <current></current>
            <change></change>
        </money-market-balance>
        <long-stock-value>
            <initial></initial>
            <current></current>
```

```
            <change></change>
        </long-stock-value>
        <long-option-value>
            <initial></initial>
            <current></current>
            <change></change>
        </long-option-value>
        .......
        <in-call></in-call>
        <in-potential-call></in-potential-call>
    </balance>
    <positions>
        <error></error>
        <account-id></account-id>
        <stocks>
            <position>
                <error></error>
                <quantity></quantity>
                <security>
                    <symbol></symbol>
                    <symbol-with-type-prefix></symbol-with-type-prefix>
                    <description> </description>
                    <asset-type></asset-type>
                    <cusip></cusip>
                </security>
                <account-type></account-type>
                <close-price></close-price>
                <position-type></position-type>
                <average-price></average-price>
                <current-value></current-value>
                <underlying-symbol></underlying-symbol>
                <put-call-indicator></put-call-indicator>
            </position>
            .........
        </stocks>
        <options>
            .........
        </options>
        <funds>
            .........
        </funds>
        <bonds>
            .........
        </bonds>
        <money-market>
            .........
        </money-market>
        <savings>
            .........
        </savings>
    </positions>
</amtd>
```

Response Fields

The Positions response fields are the same for all asset types. The Balances however differ depending on whether or not the account is enabled for Margin and for Options trading. The fields are documented at:
Balances for CASH Account
Balances for MARGIN Account
Positions

16.3.1 Response - BALANCE - CASH Account

Many of the entries on Balances will be containers which will then contain tags for initial, current and change values. For example:

```
<cash-balance>
    <initial>45678.00</initial>
    <current>45792.45</current>
    <change>114.45</change>
</cash-balance>
```

For simplicity, the individual tags for initial, current and change will not be documented. The table below will just list the container tag as Complex/ICC type and for CALL types, Complex/IC (Initial, Current)

BALANCES Response Fields for CASH accounts

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
balance	Complex	Container for all balance information for the account
error	String	Contains error message, if any, related to retrieving the balances
account-id	String	
day-trader	String/Boolean	true/false
round-trips	Integer	
resticted-closing-transactions-only	String/Boolean	true/false
cash-balance	Complex/ICC	The amount of liquid funds in the account, including the monetary value of trades that may not have settled, excluding any money market funds. There are two digits to the right of the decimal point.
money-market-balance	Complex/ICC	

long-stock-value	Complex/ICC	The total value of individual long stock positions based on the last price for those securities.
short-stock-value	Complex/ICC	The total value of individual short stock positions based on the last price for those securities.
long-option-value	Complex/ICC	The total value of individual long options positions based on the last price for those securities.
short-option-value	Complex/ICC	
mutual-fund-value	Complex/ICC	The total current value of any mutual funds in the account. There may be up to six digits to the right of the decimal point.
bond-value	Complex/ICC	The total current value of any bonds in the account.
account-value	Complex/ICC	
pending-deposits	Complex/ICC	
cash-for-withdrawal	Double	
unsettled-cash	Double	
savings-balance	Complex/C	Savings Balance - CURRENT sub-tag only
cash-debit-call-value	Complex/IC	
available-funds-for-trading	Double	
non-marginable-funds	Double	
in-call	String/Boolean	true/false - Indicates if the account has any calls on it (CURRENT)
in-potential-call	String/Boolean	true/false - Indicates if the account has any Potential calls on it (POTENTIAL)

16.3.2 Response - BALANCE - MARGIN Account

Many of the entries on Balances will be containers which will then contain tags for initial, current and change values. For example:

```
<cash-balance>
    <initial>45678.00</initial>
    <current>45792.45</current>
    <change>114.45</change>
</cash-balance>
```

For simplicity, the individual tags for initial, current and change will not be documented. The table below will just list the container tag as Complex/ICC type and for CALL types, Complex/ICP (Initial, Current, Potential) or Complex/IP (Initial, Potential)

BALANCES Response Fields for MARGIN accounts

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
balance	Complex	Container for all balance information for the account
error	String	Contains error message, if any, related to retrieving the balances
account-id	String	
day-trader	String/Boolean	true/false
round-trips	Integer	
resticted-closing-transactions-only	String/Boolean	true/false
cash-balance	Complex/ICC	The amount of liquid funds in the account, including the monetary value of trades that may not have settled, excluding any money market funds. There are two digits to the right of the decimal point.
money-market-balance	Complex/ICC	
long-stock-value	Complex/ICC	The total value of individual long stock positions based on the last price for those securities.
long-option-value	Complex/ICC	The total value of individual long options positions based on the last price for those securities.
short-stock-value	Complex/ICC	The total value of individual short stock positions based on the last price for those securities.
short-option-value	Complex/ICC	
mutual-fund-value	Complex/ICC	The total current value of any mutual funds in the account. There may be up to six digits to the right of the decimal point.
bond-value	Complex/ICC	The total current value of any bonds in the account.
account-value	Complex/ICC	
pending-deposits	Complex/ICC	
savings-balance	Complex/C	Savings Balance - CURRENT sub-tag only
margin-balance	Complex/ICC	A negative number that represents a debit balance or the amount that is on loan. The debit balance is subject to margin interest charges.
short-balance	Complex/ICC	
long-marginable-value	Complex/ICC	The total value of the long marginable positions

short-marginable-value	Complex/ICC	The total value of the short marginable positions
margin-equity	Complex/ICC	The amount of marginable assets owned and paid for in the account
equity-percentage	Complex/ICC	
stock-buying-power	Double	Projected total stock buying power, including any pending deposits
option-buying-power	Double	
day-trading-buying-power	Double	
available-funds-for-trading	Double	
maintenance-requirement	Complex/ICC	The minimum amount of equity needed to hold the marginal positions in the account
maintenance-call-value	Complex/ICP	Amount of equity needed to bring Margin Equity equal to Maintenance Requirement
regulation-t-call-value	Complex/ICP	
day-trading-call-value	Complex/IP	
day-equity-call-value	Double	
in-call	String/Boolean	true/false - Indicates if the account has any calls on it (CURRENT)
in-potential-call	String/Boolean	true/false - Indicates if the account has any Potential calls on it (POTENTIAL)

16.3.3 Response - POSITIONS

NOTE: You can have multiple position entries for the same symbol (one long and one short for example). You will need to handle that accordingly.

POSITIONS Response Fields

XML Attribute Name	Type	Definitions
error	String	Contains error message, if any, related to retrieving all the positions.
positions	Complex	Container for all types of position objects
stocks	Complex	Container for all STOCK position objects
options	Complex	Container for all OPTIONS position objects
funds	Complex	Container for all FUND position objects

bonds	Complex	Container for all BOND position objects
money-market	Complex	Container for all MONEY-MARKET position objects
savings	Complex	Container for all SAVINGS position objects
position	Complex	Container for the individual position fields
error	String	Contains an error message, if any, related to retrieving an individual security.
quantity	Double	Number of units in the position. Field may have up to 3 digits to the right of the decimal point.
security	Complex	Container for detailed information about the security.
symbol	String	Contains symbol. For example, **AMTD**.
symbol-with-type-prefix	String	
description	String	Contains a description of the symbol. For example, "**TD AMERITRADE HLDG CORP COM**"
asset-type	String	Type of asset the symbol represents. It is a one character code, such as: **E** - Equity or ETF **F** - Mutual Fund **O** - Option **B** - Bond **M** - Money Market **V** - Savings
cusip	String	An identification number assigned to all stocks and registered bonds. The Committee on Uniform Securities Identification Procedures (CUSIP) oversees the entire CUSIP system.
account-type	Integer	Indicates the account type for this security. Possible values are: **1** - Cash **2** - Margin **3** - Short **4** - Dividend/Interest
close-price	Double	The price of the last trade for the symbol at the end of the previous trading session.
position-type	String	**LONG** or **SHORT**
average-price	Double	Contains the value calculated by dividing the total cost to acquire the position by the number of units in the position. The field may contain up to six digits to the right of the decimal point.
current-value	Double	
underlying-symbol	String	Underlying symbol if it is an Option position NOTE: Only returned if **suppressquotes=true**

put-call	String	**P** or **C** - For Option symbols, indicates if its a PUT or a CALL. Otherwise null
maintenance-requirement	Double	The minimum amount of equity needed to hold the marginal positions in the account
quote	Complex	Container for the quote fields for the symbol. Only returned if **suppressquotes=false** is requested or if suppressquotes is not specified
NOTE:The Quote response will be the same as what you would get from the Quote request for the same security, in the QUOTE container		

16.4 Request/Response Examples

Request:

https://apis.tdameritrade.com/apps/100/BalancesAndPositions?source=<**#sourceID#**>

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <balance>
        <error></error>
        <account-id>123456789</account-id>
        <day-trader>true</day-trader>
        <round-trips>20</round-trips>
        <resticted-closing-transactions-only>false</resticted-closing-transactions-only>

        <cash-balance>
            <initial>970.08</initial>
            <current>46329.83</current>
            <change>45359.75</change>
        </cash-balance>
        <money-market-balance>
            <initial>0</initial>
            <current>0</current>
            <change>0</change>
        </money-market-balance>
        <long-stock-value>
            <initial>45129</initial>
            <current>0</current>
            <change>-45129</change>
        </long-stock-value>
        <long-option-value>
            <initial>0</initial>
            <current>0</current>
            <change>0</change>
```

```
</long-option-value>
<short-option-value>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</short-option-value>
<mutual-fund-value>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</mutual-fund-value>
<bond-value>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</bond-value>
<account-value>
    <initial>46142.25</initial>
    <current>46373</current>
    <change>230.75</change>
</account-value>
<pending-deposits>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</pending-deposits>
<in-call>false</in-call>
<in-potential-call>false</in-potential-call>
<short-balance>
    <initial>43.17</initial>
    <current>43.17</current>
    <change>0</change>
</short-balance>
<margin-balance>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</margin-balance>
<short-stock-value>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</short-stock-value>
<long-marginable-value>
    <initial>45129</initial>
    <current>0</current>
    <change>-45129</change>
</long-marginable-value>
<short-marginable-value>
    <initial>0</initial>
    <current>0</current>
    <change>0</change>
</short-marginable-value>
```

```xml
    <margin-equity>
        <initial>46329.83</initial>
        <current>46373</current>
        <change>230.75</change>
    </margin-equity>
    <equity-percentage>
        <initial>100.00</initial>
        <current>100.00</current>
        <change>0.00</change>
    </equity-percentage>
    <option-buying-power>46373</option-buying-power>
    <stock-buying-power>132456.44</stock-buying-power>
    <day-trading-buying-power>139440</day-trading-buying-power>
    <available-funds-for-trading>46373</available-funds-for-trading>
    <maintenance-requirement>
        <initial>13538.7</initial>
        <current>0</current>
        <change>-13538.7</change>
    </maintenance-requirement>
    <maintenance-call-value>
        <initial>0</initial>
        <current>0</current>
        <potential>0</potential>
    </maintenance-call-value>
    <regulation-t-call-value>
        <initial>0</initial>
        <current>0</current>
        <potential>0</potential>
    </regulation-t-call-value>
    <day-trading-call-value>
        <potential>0</potential>
        <initial>0</initial>
    </day-trading-call-value>
    <day-equity-call-value>0</day-equity-call-value>
</balance>
<positions>
    <error></error>
    <account-id>123456789</account-id>
    <stocks>
        <position>
            <error></error>
            <quantity>100.0</quantity>
            <security>
                <symbol>CLN</symbol>
                <symbol-with-type-prefix>CLN</symbol-with-type-prefix>
                <description>CELSION CORPORATION COM</description>
                <asset-type>E</asset-type>
                <cusip>15117N305</cusip>
            </security>
            <account-type>2</account-type>
            <close-price>6.36</close-price>
            <position-type>LONG</position-type>
            <average-price>0.00</average-price>
```

```
            <current-value>629.88</current-value>
        </position>
        <position>
            <error></error>
            <quantity>200.0</quantity>
            <security>
                <symbol>MSFT</symbol>
                <symbol-with-type-prefix>MSFT</symbol-with-type-prefix>
                <description>MICROSOFT CORP COM</description>
                <asset-type>E</asset-type>
                <cusip>594918104</cusip>
            </security>
            <account-type>2</account-type>
            <close-price>30.02</close-price>
            <position-type>LONG</position-type>
            <average-price>0.00</average-price>
            <current-value>6022.00</current-value>
        </position>
        <position>
            <error></error>
            <quantity>100.0</quantity>
            <security>
                <symbol>OKE</symbol>
                <symbol-with-type-prefix>OKE</symbol-with-type-prefix>
                <description>ONEOK INC CM (NEW)</description>
                <asset-type>E</asset-type>
                <cusip>682680103</cusip>
            </security>
            <account-type>2</account-type>
            <close-price>50.07</close-price>
            <position-type>LONG</position-type>
            <average-price>0.00</average-price>
            <current-value>4990.00</current-value>
        </position>
    </stocks>
    <options>
        <position>
            <error></error>
            <quantity>5.0</quantity>
            <security>
                <symbol>FMD_20070930C8.70</symbol>
                <symbol-with-type-prefix>FMD_032010C8.70</
                symbol-with-type-prefix>
                <description>FMD Feb 20 2007 8.70 Call</description>
                <asset-type>O</asset-type>
                <cusip>999FMDIF6</cusip>
            </security>
            <account-type>2</account-type>
            <close-price>8.70</close-price>
            <position-type>LONG</position-type>
            <average-price>8.80</average-price>
            <current-value>4150.00</current-value>
    </position>
```

</options>
<funds>
<position>
<error></error>
<quantity>**616.196**</quantity>
<security>
<symbol>**DODIX**</symbol>
<symbol-with-type-prefix>**DODIX**</symbol-with-type-prefix>
<description>**DODGE & COX INCOME FD COM**</description>
<asset-type>**F**</asset-type>
<cusip>**256210105**</cusip>
</security>
<account-type>**2**</account-type>
<close-price>**12.52**</close-price>
<position-type>**LONG**</position-type>
<average-price>**0.00**</average-price>
<current-value>**7714.77**</current-value>
</position>
<position>
<error></error>
<quantity>**1693.48**</quantity>
<security>
<symbol>**VIPSX**</symbol>
<symbol-with-type-prefix>**VIPSX**</symbol-with-type-prefix>
<description>**VANGUARD FIXED INCOME SECS FD INFLATION PROTECTED SEC FD**</description>
<asset-type>**F**</asset-type>
<cusip>**922031869**</cusip>
</security>
<account-type>**2**</account-type>
<close-price>**11.77**</close-price>
<position-type>**LONG**</position-type>
<average-price>**0.00**</average-price>
<current-value>**19932.25**</current-value>
</position>
<position>
<error></error>
<quantity>**700.405**</quantity>
<security>
<symbol>**VUSTX**</symbol>
<symbol-with-type-prefix>**VUSTX**</symbol-with-type-prefix>
<description>**VANGUARD FIXED INCOME SECS FD LONG TERM US TREASURY PORT**</description>
<asset-type>**F**</asset-type>
<cusip>**922031505**</cusip>
</security>
<account-type>**2**</account-type>
<close-price>**10.57**</close-price>
<position-type>**LONG**</position-type>
<average-price>**0.00**</average-price>
<current-value>**7403.28**</current-value>
</position>
</funds>

```
            <bonds>
                <position>
                    <error></error>
                    <quantity>25.0</quantity>
                    <security>
                        <symbol></symbol>
                        <symbol-with-type-prefix></symbol-with-type-prefix>
                        <description>WASHINGTON CNTY PA MUN FAC LEASE REV BDS
                        </description>
                        <asset-type>B</asset-type>
                        <cusip>938589BP2</cusip>
                    </security>
                    <account-type>2</account-type>
                    <close-price>132.173996</close-price>
                    <position-type>LONG</position-type>
                    <average-price>0.00</average-price>
                    <current-value>33043.50</current-value>
                </position>
            </bonds>
        </positions>
</amtd>
```


16.5 B & P Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```


If the session is invalid, you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```


17 OrderStatus

This service provides detailed order status for an account, allowing filtering of orders by status, entry date, order id, and other criteria. OrderStatus is a synchronous request. If you need an update, you need to issue the request again.

NOTE: You would want to use this in conjunction with the streaming asyncronous Account

Activity request.

17.1 OrderStatus Request

Request URL

https://apis.tdameritrade.com/apps/100/OrderStatus?source=<**#sourceID#**>

Parameters

The only parameter required is the source assigned by TD Ameritrade. You can also specify the account id for which the data is to be returned as well as various other parameters to filter the results.

17.2 OrderStatus Parameters

The only parameter required is the source assigned by TD Ameritrade. The rest are optional and are used to filter the data being returned. You can combine multiple parameters to filter the data unless stated otherwise in the parameter description.

OrderStatus Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned Can be combined with any other parameter	NO	One of the account IDs returned from the LogIn service. The default is the associated account
time	This will return orders with activity since the time specified on the current day. The time request is exclusive. For example, if the time parameter is 13:05:00 only transactions occurring at 13:05:01 and later would be included. The time used is either the reported time or if it is not available, the received time **NOTE:** time overrides the type, days, fromdate, and todate parameters. The days value is set to 0	NO	The format is <hh:mm:ss> and is assumed to be Eastern Time for the current day

orderid	This will return the order or orders that match the passed orderids. Multiple orderids can be passed. E.g. orderid=1234&orderid=5678&orderid=9999 **NOTE:** The orderid overrides the type, days, fromdate, and todate parameters. The days value is set to the maximum days back. However, in the case where time is paired with orderid, the days value is set to 0	NO	any legitimate order ID for the account being queried. If the order is part of a conditional order or complex option order, the parent order will be returned along with all child orders, including the one being requested.
type	returns all orders that match the status type. If no date parameters are specified, the orders are filtered from all orders with activity in the current day and all open orders.	NO	**DEFAULT: all** all open filled canceled pending
fromdate / todate	This will return all orders that were received in the date range specified	NO	The date format is YYYYMMDD. Maximum 60 days back. Dates are Inclusive
days	This returns all orders that were received beginning with the number of days back to the current day. If zero, and type includes open orders, then all open orders will be returned as well, regardless of date. Refers to calander days, NOT market days **NOTE:** This parameter is mutually exclusive with the date range parameters and takes precedence over them.	NO	**DEFAULT: 0** Integer 0 to 60
numrec	This returns the last number of records specified	NO	`Integer 1 to 5000`
underlying	Indicates whether or not underlying symbol for options should be returned	NO	`TRUE (Default is FALSE)`

17.3 OrderStatus Response

The response to the OrderStatus request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <orderstatus-list>
        <account-id></account-id>
```

```
        <orderstatus>
            <order-number></order-number>
            <cancelable></cancelable>
            <editable></editable>
            <complex-option></complex-option>
            <enhanced-order></enhanced-order>
            <display-status></display-status>
            <order-routing-status></order-routing-status>
            <order-received-date-time></order-received-date-time>
            <remaining-quantity></remaining-quantity>
            <trailing-activation-price></trailing-activation-price>
            <underlying-symbol></underlying-symbol>
            <order>
                <security>
                  <symbol></symbol>
                  <symbol-with-type-prefix></symbol-with-type-prefix>
                  <description></description>
                  <asset-type></asset-type>
                </security>
                <quantity></quantity>
                <order-id></order-id>
                <action></action>
                <trade-type></trade-type>
                <requested-destination>
                    <routing-mode></routing-mode>
                    <option-exchange></option-exchange>
                    <response-description></response-description>
                </requested-destination>
                <actual-destination>
                    <routing-mode></routing-mode>
                    <option-exchange></option-exchange>
                    <response-description></response-description>
                </actual-destination>
                <routing-display-size></routing-display-size>
                <order-type></order-type>
                <limit-price></limit-price>
                <stop-price></stop-price>
                <trailing-stop-method></trailing-stop-method
                <special-conditions></special-conditions>
                <time-in-force>
                    <session></session>
                </time-in-force>
            </order>
        </orderstatus>
    </orderstatus-list>
</amtd>
```

Response Fields

The OrderStatus response fields have nested orders in them for complex option and conditional orders. The table below will indicate fields that are specific to complex options or conditional orders

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
orderstatus-list	Complex	Container for all OrderStatus entries being returned for the account
account-id	String	
orderstatus	Complex	Container for the order status for a single order.
order-number	String	Unique order number assigned by the Trading Platform
cancelable	String/Boolean	true/false - Indicates if the order can be canceled
editable	String/Boolean	true/false - Indicates if the order can be edited
complex-option	String/Boolean	true/false - Indicates if the order is part of a complex option order
enhanced-order	String/Boolean	true/false - Indicates if the order is part of an enhanced order (conditional order such as OTT, OCA, etc)
enhanced-type	String	The type of enhanced order: **OCA** **OTA** **OTT** **OTOCA** **OTOTA** **NOTE:** The list of possible enhanced order types can change
display-status	String	Contains the current status of the order. Values returned are: **Open** **Filled** **Expired** **Pending** **Pending Cancel** **Canceled** **Pending Replace** **Replaced** **Received** **Review/Release** **NOTE:** **"Pending"** means the order is waiting to be manually reviewed "**Review/Release"** means the order is being manually reviewed. Once reviewed it will either switch to Open or Canceled
order-routing-status	String	
order-received-date-time	String	
reported-time	String	

exp-date-time	String	Date/time when the order was expired
cancel-date-time	String	
remaining-quantity	Double	
trailing-activation-price	Double	The current price at which the Trailing Stop will trigger
underlying-symbol	String	Underlying symbol for a given option. Only on option orders and only if underlying=true parameter is specified in the request
order	Complex	Container for detailed information about the order
security	Complex	Container for detailed information about the security
symbol	String	Contains symbol. For example, **AMTD**.
symbol-with-type-prefix	String	
description	String	
asset-type	String	Type of asset the symbol represents. It is a one character code where: **E** — Equity or ETF **F** — Mutual Fund **O** — Option **B** — Bond
quantity	Double	Number of units ordered. Field may have up to three digits to the right of the decimal point.
order-id	String	Unique order number assigned by the Trading Platform
action	String	Code indicating the order action requested. Values returned are: **B** — Buy **S** — Sell **SS** — Short Sell **BC** — Buy to Cover **E** — Exchange **EX** — Exercise Option **NOTE:** For options, also check the open-close field.
trade-type	Integer	Valid values are: **1** — Normal Market **2** — External Hour Market **4** — German Market **8** — AM Session **16** — Seamless Session
requested-destination	Complex	
actual-destination	Complex	
routing-mode	String	SMART INET ECN **NOTE:** Can have other values

option-exchange	String	Auto **NOTE:** Can have other values
response-description	String	AutoRoute ISLD ARCA **NOTE:** Can have other values
routing-display-size	Integer	
order-type	String	Values returned are: **M** — Market **L** — Limit **S** — Stop **X** — Stop Limit **D** — Debit **C** — Credit **T** — Trailing Stop **EX** — Exercise Option
limit-price	Double	
stop-price	Double	
trailing-stop-method	String	**PERCENT,POINTS**
special-conditions	String	One or more of the following or "None": **A** - All Or None **D** - Do Not Reduce **F** - Fill Or Kill **AD** - Both AON and DNR
time-in-force	Complex	
session	String	**D** - Day **G** - Good Till Canceled (same as GTD) **GTD** - Good Till Date (same as G) **DX** - Day+Extended **GX** - GTC+Extended **XAM** - Extended hours trading - AM **XPM** - Estended hours trading - PM **MOC** - Market On Close
expiration	String	

strategy	String	The type of complex option that the order is part of. Possible values are: **Spread** **Straddle** **Buy Write** **Exercise** **Combo** **Exercise** **Swap** **Vertical** **Back Ratio** **Calendar** **Diagonal** **Butterfly** **Condor** **Iron Condor** **Vertical Roll** **Covered Stock** **Collar With Stock** **Collar Synthetic** **Double Diagonal** **Custom** (This field is only returned on complex option orders)
open-close	String	(This field is only returned on option orders) **O** or **C**
put-call	String	**P** or **C** - For Option symbols, indicates if its a PUT or a CALL. Otherwise null (This field is only returned on option orders)
fills	Complex	Container for information about Fills that occurred on this order. There will be one fills container for each fill for the given order, so if there were 5 partial fills, there will be 5 fills containers. (This field is only returned on orders that have fills)
fill-id	String	(This field is only returned on orders that have fills)
fill-quantity	Double	(This field is only returned on orders that have fills)
fill-price	Double	(This field is only returned on orders that have fills)
execution-reported-date-time	String	Example of the Date/Time format is: 2007-03-23 15:09:59 EDT NOTE: Timezone may be EDT or EST for now (This field is only returned on orders that have fills)
related-orders	Complex	Container for OrderStatus objects of related orders. (This field is only returned on complex options orders and conditional orders)
relationship-type	String	(This field is only returned on conditional orders)

17.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/OrderStatus?source=<**#sourceID#**>

Response:


```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <orderstatus-list>
        <account-id>123456789</account-id>
        <orderstatus>
            <order-number>2967109823</order-number>
            <cancelable>true</cancelable>
            <editable>true</editable>
            <complex-option>false</complex-option>
            <enhanced-order>false</enhanced-order>
            <display-status>Open</display-status>
            <order-routing-status>Ready</order-routing-status>
            <order-received-date-time>2007-06-13 20:18:03 EDT</
            order-received-date-time>
            <remaining-quantity>100.0</remaining-quantity>
            <order>
                <security>
                  <symbol>YHOO</symbol>
                  <symbol-with-type-prefix>YHOO</symbol-with-type-prefix>
                  <description></description>
                  <asset-type>E</asset-type>
                </security>
                <quantity>100</quantity>
                <order-id>2967109823</order-id>
                <action>B</action>
                <trade-type>1</trade-type>
                <requested-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
                </requested-destination>
                <actual-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
                </actual-destination>
                <routing-display-size>0</routing-display-size>
                <order-type>L</order-type>
                <limit-price>2.56</limit-price>
                <stop-price>0.00</stop-price>
                <special-conditions></special-conditions>
                <time-in-force>
                    <session>D</session>
                </time-in-force>
            </order>
```

```
</orderstatus>
<orderstatus>
    <order-number>2147591543</order-number>
    <cancelable>false</cancelable>
    <editable>false</editable>
    <complex-option>false</complex-option>
    <enhanced-order>false</enhanced-order>
    <display-status>Pending Cancel</display-status>
    <order-routing-status>Routed</order-routing-status>
    <order-received-date-time>2007-05-24 13:42:43 EDT</
    order-received-date-time>
    <reported-time>2007-05-24 13:42:43 EDT</reported-time>
    <cancel-date-time>2007-06-05 20:44:21 EDT</cancel-date-time>
    <remaining-quantity>2.0</remaining-quantity>
    <order>
        <security>
            <symbol>IBM</symbol>
            <symbol-with-type-prefix>IBM</symbol-with-type-prefix>
            <description></description>
            <asset-type>E</asset-type>
        </security>
        <quantity>2</quantity>
        <order-id>2147591543</order-id>
        <action>B</action>
        <trade-type>1</trade-type>
        <requested-destination>
            <routing-mode>SMART</routing-mode>
            <option-exchange>Auto</option-exchange>
            <response-description>AutoRoute</response-description>
        </requested-destination>
        <actual-destination>
            <routing-mode>SMART</routing-mode>
            <option-exchange>Auto</option-exchange>
            <response-description>AutoRoute</response-description>
        </actual-destination>
        <routing-display-size>0</routing-display-size>
        <order-type>L</order-type>
        <limit-price>3.00</limit-price>
        <stop-price>0.00</stop-price>
        <special-conditions></special-conditions>
        <time-in-force>
            <session>GTD</session>
            <expiration>2007-08-24</expiration>
        </time-in-force>
    </order>
</orderstatus>
<orderstatus>
    <order-number>2147591539</order-number>
    <cancelable>false</cancelable>
    <editable>false</editable>
    <complex-option>false</complex-option>
    <enhanced-order>false</enhanced-order>
    <display-status>Pending Replace</display-status>
```

```
        <order-routing-status>Routed</order-routing-status>
        <order-received-date-time>2007-05-24 13:42:34 EDT</
        order-received-date-time>
        <reported-time>2007-05-24 13:42:34 EDT</reported-time>
        <cancel-date-time>2007-05-24 13:42:43 EDT</cancel-date-time>
        <remaining-quantity>2.0</remaining-quantity>
        <order>
            <security>
                <symbol>IBM</symbol>
                <symbol-with-type-prefix>IBM</symbol-with-type-prefix>
                <description></description>
                <asset-type>E</asset-type>
            </security>
            <quantity>2</quantity>
            <order-id>2147591539</order-id>
            <action>B</action>
            <trade-type>1</trade-type>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <actual-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </actual-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</order-type>
            <limit-price>2.00</limit-price>
            <stop-price>0.00</stop-price>
            <special-conditions></special-conditions>
            <time-in-force>
                <session>GTD</session>
                <expiration>2007-08-24</expiration>
            </time-in-force>
        </order>
    </orderstatus>
    <orderstatus>
        <order-number>2147594455</order-number>
        <cancelable>false</cancelable>
        <editable>false</editable>
        <complex-option>false</complex-option>
        <enhanced-order>true</enhanced-order>
        <enhanced-type>OTT</enhanced-type>
        <display-status>Filled</display-status>
        <order-routing-status>Routed</order-routing-status>
        <order-received-date-time>2007-06-05 14:19:19 EDT</
        order-received-date-time>
        <reported-time>2007-06-05 14:19:22 EDT</reported-time>
        <remaining-quantity>0.0</remaining-quantity>
        <order>
            <security>
```

```
        <symbol>AMTD</symbol>
        <symbol-with-type-prefix>AMTD</symbol-with-type-prefix>
        <description></description>
        <asset-type>E</asset-type>
    </security>
    <quantity>1500</quantity>
    <order-id>2147594455</order-id>
    <action>B</action>
    <trade-type>1</trade-type>
    <requested-destination>
        <routing-mode>SMART</routing-mode>
        <option-exchange>Auto</option-exchange>
        <response-description>AutoRoute</response-description>
    </requested-destination>
    <actual-destination>
        <routing-mode>SMART</routing-mode>
        <option-exchange>Auto</option-exchange>
        <response-description>AutoRoute</response-description>
    </actual-destination>
    <routing-display-size>0</routing-display-size>
    <order-type>L</order-type>
    <limit-price>11.00</limit-price>
    <stop-price>0.00</stop-price>
    <special-conditions></special-conditions>
    <time-in-force>
        <session>GTD</session>
        <expiration>2007-07-31</expiration>
    </time-in-force>
</order>
<fills>
    <fill-id>214758730</fill-id>
    <fill-quantity>600.0</fill-quantity>
    <fill-price>11.00</fill-price>
    <execution-reported-date-time>2007-06-05 14:19:22 EDT</
    execution-reported-date-time>
</fills>
<fills>
    <fill-id>2147594461</fill-id>
    <fill-quantity>900.0</fill-quantity>
    <fill-price>11.00</fill-price>
    <execution-reported-date-time>2007-06-05 14:19:23 EDT</
    execution-reported-date-time>
</fills><related-orders>
    <orderstatus>
        <order-number>2147594456</order-number>
        <cancelable>false</cancelable>
        <editable>false</editable>
        <complex-option>false</complex-option>
        <enhanced-order>true</enhanced-order>
        <enhanced-type>OTT</enhanced-type>
        <display-status>Canceled</display-status>
        <order-routing-status>Received</order-routing-status>
        <order-received-date-time>2007-06-05 14:22:03 EDT</
```

```
        order-received-date-time>
        <remaining-quantity>200.0</remaining-quantity>
        <order>
            <security>
                <symbol>SCHW</symbol>
                <symbol-with-type-prefix>SCHW</symbol-with-type-prefix
                >
                <description></description>
                <asset-type>E</asset-type>
            </security>
            <quantity>1500</quantity>
            <order-id>2147594456</order-id>
            <action>S</action>
            <trade-type>1</trade-type>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <actual-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </actual-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</order-type>
            <limit-price>24.00</limit-price>
            <stop-price>0.00</stop-price>
            <special-conditions> </special-conditions>
            <time-in-force>
                <session>GTD</session>
                <expiration>2007-07-31</expiration>
            </time-in-force>
        </order>
        <relationship-type>Triggers</relationship-type>
    </orderstatus>
    <orderstatus>
        <order-number>2147594457</order-number>
        <cancelable>false</cancelable>
        <editable>false</editable>
        <complex-option>false</complex-option>
        <enhanced-order>true</enhanced-order>
        <enhanced-type>OTT</enhanced-type>
        <display-status>Canceled</display-status>
        <order-routing-status>Received</order-routing-status>
        <order-received-date-time>2007-06-05 14:22:03 EDT</
        order-received-date-time>
        <remaining-quantity>1500.0</remaining-quantity>
        <order>
            <security>
                <symbol>ETFC</symbol>
                <symbol-with-type-prefix>ETFC</symbol-with-type-prefix>
                <description></description>
```

<asset-type>**E**</asset-type>
</security>
<quantity>**200**</quantity>
<order-id>**2147594457**</order-id>
<action>**S**</action>
<trade-type>**1**</trade-type>
<requested-destination>
<routing-mode>**SMART**</routing-mode>
<option-exchange>**Auto**</option-exchange>
<response-description>**AutoRoute**</response-description>
</requested-destination>
<actual-destination>
<routing-mode>**SMART**</routing-mode>
<option-exchange>**Auto**</option-exchange>
<response-description>**AutoRoute**</response-description>
</actual-destination>
<routing-display-size>**0**</routing-display-size>
<order-type>**L**</order-type>
<limit-price>**25.00**</limit-price>
<stop-price>**0.00**</stop-price>
<special-conditions> </special-conditions>
<time-in-force>
<session>**GTD**</session>
<expiration>**2007-07-31**</expiration>
</time-in-force>
</order>
<relationship-type>**Triggers**</relationship-type>
</orderstatus>
</related-orders>
</orderstatus>
</orderstatus-list>
</amtd>

17.4.1 Response Sample - Simple, OPEN

<?xml version="1.0" ?>
<amtd>
<result>**OK**</result>
<orderstatus-list>
<account-id>**123456789**</account-id>
<orderstatus>
<order-number>**4196528959**</order-number>
<cancelable>**true**</cancelable>
<editable>**true**</editable>
<complex-option>**false**</complex-option>
<enhanced-order>**false**</enhanced-order>
<display-status>**Open**</display-status>
<order-routing-status>**Ready**</order-routing-status>
<order-received-date-time>**2009-07-27 10:26:08 EDT**</
order-received-date-time>
<remaining-quantity>**100.0**</remaining-quantity>

```
        <trailing-activation-price>0</trailing-activation-price>
        <order>
          <symbol>YHOO</symbol>
          <security>
            <symbol>YHOO</symbol>
            <symbol-with-type-prefix>YHOO</symbol-with-type-prefix>
            <description> </description>
            <asset-type>E</asset-type>
          </security>
          <quantity>100</quantity>
          <order-id>4196528959</order-id>
          <action>B</action>
          <trade-type>2</trade-type>
          <requested-destination>
          <routing-mode>SMART</routing-mode>
            <option-exchange>Auto</option-exchange>
            <response-description>AutoRoute</response-description>
          </requested-destination>
          <actual-destination>
             <routing-mode>SMART</routing-mode>
             <option-exchange>Auto</option-exchange>
             <response-description>AutoRoute</response-description>
          </actual-destination>
          <routing-display-size>0</routing-display-size>
          <order-type>L</order-type>
          <limit-price>1.00</limit-price>
          <stop-price> </stop-price>
          <special-conditions> </special-conditions>
          <time-in-force>
               <session>XPM</session>
               <expiration>2009-07-27</expiration>
          </time-in-force>
          </order>
       </orderstatus>
       </orderstatus-list>
</amtd>
```


17.4.2 Response Sample - Simple, PENDING


```
<?xml version="1.0" ?>
<amtd>
<result>OK</result>
<orderstatus-list>
  <account-id>123456789</account-id>
<orderstatus>
   <order-number>4196528804</order-number>
  <cancelable>false</cancelable>
  <editable>false</editable>
  <complex-option>true</complex-option>
  <enhanced-order>false</enhanced-order>
  <display-status>Pending Cancel</display-status>
  <order-routing-status>Routed</order-routing-status>
  <order-received-date-time>2009-07-23 13:48:11 EDT</order-received-date-time>
```

```
    <reported-time>2009-07-23 13:48:11 EDT</reported-time>
    <remaining-quantity>100.0</remaining-quantity>
    <trailing-activation-price>0</trailing-activation-price>
    <strategy>Buy Write</strategy>
    <order>
      <symbol>IBM</symbol>
      <security>
        <symbol>IBM</symbol>
        <symbol-with-type-prefix>IBM</symbol-with-type-prefix>
        <description> </description>
        <asset-type>E</asset-type>
      </security>
      <quantity>100</quantity>
      <order-id>4196528804</order-id>
      <action>B</action>
      <trade-type>1</trade-type>
      <requested-destination>
        <routing-mode>SMART</routing-mode>
        <option-exchange>Auto</option-exchange>
        <response-description>AutoRoute</response-description>
      </requested-destination>
      <actual-destination>
        <routing-mode>SMART</routing-mode>
        <option-exchange>Auto</option-exchange>
        <response-description>AutoRoute</response-description>
      </actual-destination>
      <routing-display-size>0</routing-display-size>
      <order-type>C</order-type>
      <limit-price>17.00</limit-price>
      <stop-price> </stop-price>
      <special-conditions> </special-conditions>
      <time-in-force>
      <session>D</session>
      <expiration>2009-07-27</expiration>
      </time-in-force>
    </order>
    </orderstatus>
    </orderstatus-list>
  </amtd>
```


17.4.3 Response Sample - Conditional, FILLED

Note the nested orders under the **related-orders** tag, as well as the **enhanced-type** and **relationship-type** tags.


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <orderstatus-list>
        <account-id>123456789</account-id>
        <orderstatus>
            <order-number>2147594455</order-number>
            <cancelable>false</cancelable>
```

```
<editable>false</editable>
<complex-option>false</complex-option>
<enhanced-order>true</enhanced-order>
<enhanced-type>OTT</enhanced-type>
<display-status>Filled</display-status>
<order-routing-status>Routed</order-routing-status>
<order-received-date-time>2007-06-05 14:19:19 EDT</
order-received-date-time>
<reported-time>2007-06-05 14:19:22 EDT</reported-time>
<remaining-quantity>0.0</remaining-quantity>
<order>
    <symbol>AMTD</symbol>
    <security>
        <symbol>AMTD</symbol>
        <symbol-with-type-prefix>AMTD</symbol-with-type-prefix>
        <description></description>
        <asset-type>E</asset-type>
    </security>
    <quantity>700</quantity>
    <order-id>2147594455</order-id>
    <action>B</action>
    <trade-type>1</trade-type>
    <requested-destination>
        <routing-mode>SMART</routing-mode>
        <option-exchange>Auto</option-exchange>
        <response-description>AutoRoute</response-description>
    </requested-destination>
    <actual-destination>
        <routing-mode>SMART</routing-mode>
        <option-exchange>Auto</option-exchange>
        <response-description>AutoRoute</response-description>
    </actual-destination>
    <routing-display-size>0</routing-display-size>
    <order-type>L</order-type>
    <limit-price>11.00</limit-price>
    <stop-price>0.00</stop-price>
    <special-conditions></special-conditions>
    <time-in-force>
        <session>GTD</session>
        <expiration>2007-07-31</expiration>
    </time-in-force>
</order>
<fills>
    <fill-id>2147594461</fill-id>
    <fill-quantity>400.0</fill-quantity>
    <fill-price>10.99</fill-price>
    <execution-reported-date-time>2007-06-05 14:19:21 EDT</
    execution-reported-date-time>
</fills>
<fills>
    <fill-id>2147594482</fill-id>
    <fill-quantity>100.0</fill-quantity>
    <fill-price>11.00</fill-price>
```

```
        <execution-reported-date-time>2007-06-05 14:19:22 EDT</
        execution-reported-date-time>
    </fills>
    <fills>
        <fill-id>2147595193</fill-id>
        <fill-quantity>200.0</fill-quantity>
        <fill-price>11.00</fill-price>
        <execution-reported-date-time>2007-06-05 14:19:22 EDT</
        execution-reported-date-time>
    </fills>
    <related-orders>
        <orderstatus>
            <order-number>2147594456</order-number>
            <cancelable>false</cancelable>
            <editable>false</editable>
            <complex-option>false</complex-option>
            <enhanced-order>true</enhanced-order>
            <enhanced-type>OTT</enhanced-type>
            <display-status>Canceled</display-status>
            <order-routing-status>Received</order-routing-status>
            <order-received-date-time>2007-06-05 14:22:03 EDT</
            order-received-date-time>
            <remaining-quantity>200.0</remaining-quantity>
            <order>
                <symbol>SCHW</symbol>
                <security>
                    <symbol>SCHW</symbol>
                    <symbol-with-type-prefix>SCHW</symbol-with-type-prefix
                    >
                    <description></description>
                    <asset-type>E</asset-type>
                </security>
                <quantity>200</quantity>
                <order-id>2147594456</order-id>
                <action>S</action>
                <trade-type>1</trade-type>
                <requested-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
                </requested-destination>
                <actual-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
                </actual-destination>
                <routing-display-size>0</routing-display-size>
                <order-type>L</order-type>
                <limit-price>24.00</limit-price>
                <stop-price>0.00</stop-price>
                <special-conditions></special-conditions>
                <time-in-force>
                    <session>GTD</session>
```

```
                <expiration>2007-07-31</expiration>
            </time-in-force>
        </order>
        <relationship-type>Triggers</relationship-type>
    </orderstatus>
    <orderstatus>
        <order-number>2147594457</order-number>
        <cancelable>false</cancelable>
        <editable>false</editable>
        <complex-option>false</complex-option>
        <enhanced-order>true</enhanced-order>
        <enhanced-type>OTT</enhanced-type>
        <display-status>Canceled</display-status>
        <order-routing-status>Received</order-routing-status>
        <order-received-date-time>2007-06-05 14:22:03 EDT</order-received-date-time>
        <remaining-quantity>200.0</remaining-quantity>
        <order>
            <symbol>ETFC</symbol>
            <security>
                <symbol>ETFC</symbol>
                <symbol-with-type-prefix>ETFC</symbol-with-type-prefix>
                <description></description>
                <asset-type>E</asset-type>
            </security>
            <quantity>200</quantity>
            <order-id>2147594457</order-id>
            <action>S</action>
            <trade-type>1</trade-type>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <actual-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </actual-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</order-type>
            <limit-price>25.00</limit-price>
            <stop-price>0.00</stop-price>
            <special-conditions></special-conditions>
            <time-in-force>
                <session>GTD</session>
                <expiration>2007-07-31</expiration>
            </time-in-force>
        </order>
        <relationship-type>Triggers</relationship-type>
    </orderstatus>
</related-orders>
</orderstatus>
```

```
    </orderstatus-list>
</amtd>
```


17.4.4 Response Sample - Complex Option

Note the nested orders under the **related-orders** tag, as well as the **complex-option** and **oe-strategy** tags.


```
<?xml version="1.0" ?>
<amtd>
   <result>OK</result>
   <orderstatus-list>
   <account-id>123456789</account-id>
   <orderstatus>
      <order-number>4196528728</order-number>
      <cancelable>true</cancelable>
      <editable>false</editable>
      <complex-option>true</complex-option>
      <enhanced-order>false</enhanced-order>
      <display-status>Open</display-status>
      <order-routing-status>Routed</order-routing-status>
      <order-received-date-time>2009-07-22 14:19:38 EDT</
order-received-date-time>
      <remaining-quantity>1.0</remaining-quantity>
      <trailing-activation-price>0</trailing-activation-price>
      <strategy>Iron Condor</strategy>
      <order>
        <symbol>IBM_20100822P110</symbol>
        <security>
          <symbol>IBM_20100822P110</symbol>
          <symbol-with-type-prefix>IBM_082210P110</symbol-with-type-prefix>
          <description>IBM Aug 22  2010 110.0 Put</description>
          <asset-type>O</asset-type>
        </security>
        <quantity>1</quantity>
        <order-id>4196528728</order-id>
        <action>B</action>
        <trade-type>1</trade-type>
        <requested-destination>
          <routing-mode>SMART</routing-mode>
          <option-exchange>Auto</option-exchange>
          <response-description>AutoRoute</response-description>
        </requested-destination>
        <actual-destination>
           <routing-mode>SMART</routing-mode>
           <option-exchange>Auto</option-exchange>
           <response-description>AutoRoute</response-description>
        </actual-destination>
        <routing-display-size>0</routing-display-size>
        <order-type>M</order-type>
        <limit-price></limit-price>
        <stop-price></stop-price>
```

```
          <special-conditions></special-conditions>
          <time-in-force>
          <session>D</session>
          <expiration>2009-07-23</expiration>
          </time-in-force>
          <put-call>P</put-call>
          <open-close>O</open-close>
        </order>
        <related-orders>
          <orderstatus>
            <order-number>4196528731</order-number>
            <cancelable>false</cancelable>
            <editable>false</editable>
            <complex-option>true</complex-option>
            <enhanced-order>false</enhanced-order>
            <display-status>Open</display-status>
            <order-routing-status>Routed</order-routing-status>
            <order-received-date-time>2009-07-22 14:19:38 EDT</
order-received-date-time>
            <remaining-quantity>1.0</remaining-quantity>
            <trailing-activation-price>0</trailing-activation-price>
            <strategy>Iron Condor</strategy>
            <order>
              <symbol>IBM_082210P110</symbol>
              <security>
                <symbol>IBM_082210P110</symbol>
                <symbol-with-type-prefix>IBM_082210P110</symbol-with-type-prefix
>
                <description>IBM Aug 22  2010 110.0 Put</description>
                  <asset-type>O</asset-type>
              </security>
              <quantity>1</quantity>
              <order-id>4196528731</order-id>
              <action>S</action>
              <trade-type>1</trade-type>
              <requested-destination>
                 <routing-mode>SMART</routing-mode>
                 <option-exchange>Auto</option-exchange>
                 <response-description>AutoRoute</response-description>
              </requested-destination>
              <actual-destination>
                 <routing-mode>SMART</routing-mode>
                 <option-exchange>Auto</option-exchange>
                 <response-description>AutoRoute</response-description>
              </actual-destination>
              <routing-display-size>0</routing-display-size>
              <order-type>M</order-type>
              <limit-price></limit-price>
              <stop-price></stop-price>
              <special-conditions></special-conditions>
              <time-in-force>
                  <session>D</session>
                  <expiration>2009-07-23</expiration>
```

```
                </time-in-force>
                <put-call>P</put-call>
                <open-close>O</open-close>
              </order>
          </orderstatus>
          <orderstatus>
              <order-number>4196528730</order-number>
              <cancelable>false</cancelable>
              <editable>false</editable>
              <complex-option>true</complex-option>
              <enhanced-order>false</enhanced-order>
              <display-status>Open</display-status>
              <order-routing-status>Routed</order-routing-status>
              <order-received-date-time>2009-07-22 14:19:38 EDT</
order-received-date-time>
              <remaining-quantity>1.0</remaining-quantity>
              <trailing-activation-price>0</trailing-activation-price>
              <strategy>Iron Condor</strategy>
              <order>
                <symbol>IBM_082210C110</symbol>
                  <security>
                        <symbol>IBM_082210C110</symbol>
                        <symbol-with-type-prefix>IBM_082210C110</
symbol-with-type-prefix>
                        <description>IBM Aug 22  2010 110.0 Call</description>
                  <asset-type>O</asset-type>
                </security>
                <quantity>1</quantity>
                <order-id>4196528730</order-id>
                <action>S</action>
                <trade-type>1</trade-type>
                <requested-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
               </requested-destination>
               <actual-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
              </actual-destination>
              <routing-display-size>0</routing-display-size>
              <order-type>M</order-type>
              <limit-price></limit-price>
              <stop-price></stop-price>
              <special-conditions>   </special-conditions>
              <time-in-force>
                 <session>D</session>
                 <expiration>2009-07-23</expiration>
              </time-in-force>
              <put-call>C</put-call>
              <open-close>O</open-close>
           </order>
```

```
      </orderstatus>
      <orderstatus>
         <order-number>4196528729</order-number>
         <cancelable>false</cancelable>
         <editable>false</editable>
         <complex-option>true</complex-option>
         <enhanced-order>false</enhanced-order>
         <display-status>Open</display-status>
         <order-routing-status>Routed</order-routing-status>
         <order-received-date-time>2009-07-22 14:19:38 EDT</
order-received-date-time>
         <remaining-quantity>1.0</remaining-quantity>
         <trailing-activation-price>0</trailing-activation-price>
         <strategy>Iron Condor</strategy>
         <order>
           <symbol>IBM_20100822C110</symbol>
           <security>
                  <symbol>IBM_082210C110</symbol>
                  <symbol-with-type-prefix>IBM_082210C110</
symbol-with-type-prefix>
                  <description>IBM Aug 22  2010 110.0 Call</description>
             <asset-type>O</asset-type>
           </security>
           <quantity>1</quantity>
           <order-id>4196528729</order-id>
           <action>B</action>
           <trade-type>1</trade-type>
           <requested-destination>
             <routing-mode>SMART</routing-mode>
             <option-exchange>Auto</option-exchange>
             <response-description>AutoRoute</response-description>
           </requested-destination>
           <actual-destination>
             <routing-mode>SMART</routing-mode>
             <option-exchange>Auto</option-exchange>
             <response-description>AutoRoute</response-description>
           </actual-destination>
           <routing-display-size>0</routing-display-size>
           <order-type>M</order-type>
           <limit-price></limit-price>
           <stop-price></stop-price>
           <special-conditions>  </special-conditions>
           <time-in-force>
             <session>D</session>
             <expiration>2009-07-23</expiration>
           </time-in-force>
           <put-call>C</put-call>
           <open-close>O</open-close>
           </order>
          </orderstatus>
        </related-orders>
      </orderstatus>
    </orderstatus-list>
```

```
</amtd>
```

17.5 OrderStatus Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

18 LastOrderStatus

This call returns the date and time of the last order status activity for the primary account associated with the login userID, or the explicitly specified account.

Request URL

https://apis.tdameritrade.com/apps/100/LastOrderStatus?source=#sourceID#&accountid=123456789

Parameters

No parameters are needed other then the source. AccountID is optional. If not specified, then the result will be for the default associated account

General Structure of XML Response

The response to the LogOut request will be in XML format

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <last-order-status>
      <account-id></account-id>
      <last></last>
    </last-order-status>
</amtd>
```

Login Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
last-order-status	Complex	Container for the result data
account-id	String	Account ID being checked
last	String	The Date/Time of the last order status activity or "**No Orders**" if there are no open orders and no transactions for the current day

Response:

```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <last-order-status>
      <account-id>123456789</account-id>
      <last>2007-06-14 09:37:05 EDT</last>
    </last-order-status>
</amtd>
```

or

```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <last-order-status>
      <account-id>123456789</account-id>
      <last>No Orders</last>
    </last-order-status>
</amtd>
```

19 History

This service provides detailed order status for an account, allowing filtering of orders by status, entry date, order id, and other criteria.

19.1 History Request

Request URL

https://apis.tdameritrade.com/apps/100/History?source=<**#sourceID#**>&type=**<#type#>**

Parameters

The required parameters are the source assigned by TD Ameritrade and the type. You can also specify the account id for which the data is to be returned as well as various other parameters to filter the results.

19.2 History Parameters

The only parameters required are the source assigned by TD Ameritrade and the transaction type. The rest are optional and are used to filter the data being returned. You can combine multiple parameters to filter the data.

History Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned Can be combined with any other parameter	NO	One of the account IDs returned from the LogIn service. The default is the associated account
type	Identifies the type of transactions that should be returned	YES	**0** - All Trans types **1** - Trades **2** - Buy **3** - Sell **4** - Deposits & Withdrawals **5** - Checking **6** - Dividends **7** - Interest **8** - Other
startdate	The date of the oldest transaction that should be returned	NO	The date format is YYYYMMDD
enddate	The date of the newest transaction that should be returned	NO	The date format is YYYYMMDD

19.3 History Response

The response to the History request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
```

```xml
    <result></result>
    <history>
        <account-id></account-id>
        <startDate></startDate>
        <endDate></endDate>
        <searchTransactionType></searchTransactionType>
        <requestType></requestType>
        <transaction-list>
            <error></error>
            <orderDateTime></orderDateTime>
            <type></type>
            <subType></subType>
            <buySellCode></buySellCode>
            <assetType></assetType>
            <symbol></symbol>
            <description></description>
            <cusip></cusip>
            <price></price>
            <quantity></quantity>
            <transactionId></transactionId>
            <value></value>
            <commission></commission>
            <orderNumber></orderNumber>
            <fees></fees>
            <additionalFees></additionalFees>
            <cashBalanceEffect></cashBalanceEffect>
            <openClose></openClose>
            <optionType></optionType>
            <optionStrike></optionStrike>
            <accruedInterest></accruedInterest>
            <parentChildIndicator></parentChildIndicator>
            <sharesBefore></sharesBefore>
            <sharesAfter></sharesAfter>
            <otherCharges></otherCharges>
            <redemptionFee></redemptionFee>
            <cdscFee></cdscFee>
            <bondInterestRate></bondInterestRate>
        </transaction-list>
    </history>
</amtd>
```

History Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
history	Complex	Container for all History entries being returned for the request
account-id	String	Account ID of the account for which history is being returned
startDate	String	Search Criteria - Start Date

endDate	String	Search Criteria - End Date
searchTransactionType	String	Search Criteria - Transaction Type
requestType	String	Search Criteria - request Type
searchSymbols	Sting	Search Criteria - Symbols
transaction-list	Complex	Container each individual transaction entry being returned
error	String	If an error occurred, the error text. Otherwise null.
orderDateTime	String	Date/Time of the transaction. For example: **2008-03-25 10:39:47 EDT**
executedDate	String	Date/Time that the order was executed. For example: **2008-03-25 10:39:47 EDT**
type		**AD** – ACH Disbursements **CD** – Cash Disbursements **DV** – Dividend / Interest **JN** – Journal Entry **MI** – Margin Interest **RD** – Receive & Deliver **TR** – Trade **WI** – Wire IN **AR** – ACH Receipt **CR** – Cash Receipts **EF** – Electronic Funds **MC** – Margin Call **MM** – Money Market **SM** – SMA Adjustment **WO** – Wire OUT
subType		Transaction Sub-Type - Refer to Sub-Type documentation page
buySellCode		Indication if the transaction was a BUY or a SELL. Values are **B** or **S**
assetType		**S** – Stocks **U** – Units **O** – Option **W** – Rights, Warrants **F** – Funds **B** - Bonds
symbol		Symbol of the security involved in the transaction
description		
cusip		
price		Price per share or per contract. (6 digits maximum to the right of the decimal point)
quantity		Number of shares or contracts
transactionId		Unique identifier for each transaction.
value		Net amount including commission or fees charged. It is the net value of the transaction.
commission		Cost of Commission charged for the transaction
orderNumber		Order Key associated with the transaction if it is a trade. (same as the order ID used in OrderStatus and other trading commands)
fees		Fees that SEC charges the customer for the transaction
additionalFees		Contains the total fees that the customer was charged for the trade.
cashBalanceEffect		To indicate if the transaction effects balance calculations. **N** – Balance is not affected **Y** – Balance is affected.
bankingStatus		

openClose		Option Open/Close Identifier **C** – Close **O** – Open
OptionExpireDate		Expiration Date. YYYYMMDD format
optionUnderlyingSymbol		This is the underlying symbol of the option or Base Symbol
optionType		**P** – Put **C** – Call
optionStrike		Strike Price of an Option
accruedInterest		
parentChildIndicator		Indicates if an order is a normal order, parent order or child order. (NOT for Conditional Orders) **P** – Parent **C** – Child **Null** - Normal
splitRatio		
sharesBefore		Shares Before Trade
sharesAfter		Shares After Trade
otherCharges		Other Charges that can be imposed to trades
redemptionFee		
cdscFee		
bondInterestRate		
bondMaturityDate		

19.4 History Response Sub-Types

The response to the History request will be in XML format

History Response Fields

Type	Sub-Type	Type Description	Sub-Type Description
AD	AD	ACH Disbursement	ACH Out
AD	NULL	ACH Disbursement	NULL
AR	AR	ACH Receipt	ACH In
AR	DD	ACH Receipt	Direct Deposit
AR	NULL	ACH Receipt	NULL
CD	AB	Cash Disbursement	Disbursement (Automatic Process)
CD	AC	Cash Disbursement	Disbursement (ACD Request)
CD	AM	Cash Disbursement	Cash Disbursement Alt Payee or Address
CD	AP	Cash Disbursement	Disbursement to Alternate Payee or Address
CD	CD	Cash Disbursement	Check (Manual Request)

CD	CK	Cash Disbursement	Check (Written against Brokerage Account)
CD	DC	Cash Disbursement	Debit Cards (against Brokerage Account)
CD	TO	Cash Disbursement	Account Transfer Out (via Check)
CD	TP	Cash Disbursement	Third Party
CD	NULL	Cash Disbursement	NULL
CR	BD	Cash Receipt	Bank Draft Received
CR	BP	Cash Receipt	Cash Receipt Bill Payments
CR	CH	Cash Receipt	Credit Card Check Received
CR	CR	Cash Receipt	Personal Check Receipt
CR	CU	Cash Receipt	Currency (Cash) Received
CR	FF	Cash Receipt	Foreign Funds Instrument
CR	FX	Cash Receipt	Cash Receipt Via Fax
CR	GC	Cash Receipt	Government Check Received
CR	LB	Cash Receipt	Lock Box Receipt
CR	MO	Cash Receipt	Money Order Received
CR	SC	Cash Receipt	Cash Receipts Starter Check
CR	TC	Cash Receipt	Travelers Check Received
CR	TI	Cash Receipt	Account Transfer Incoming (via Check)
CR	TP	Cash Receipt	Cash Receipts Third Party
CR	NULL	Cash Receipt	NULL
DV	AF	Dividend Receipt	ADR Fees
DV	CA	Dividend Receipt	Free Balance Interest Adjustment
DV	CF	Dividend Receipt	Contract Fee
DV	CL	Dividend Receipt	Cash in Lieu of Fractional Shares
DV	DC	Dividend Receipt	Return of Capital
DV	DD	Dividend Receipt	Cash Liquidation
DV	DE	Dividend Receipt	Non-Taxable Dividends
DV	DG	Dividend Receipt	Long Term Gain Distribution
DV	DN	Dividend Receipt	Non-Cash Liquidation Value
DV	DQ	Dividend Receipt	Qualified 5 Year Gain
DV	DS	Dividend Receipt	Short Term Capital Gains
DV	DX	Dividend Receipt	Investment Expense
DV	GN	Dividend Receipt	Section 1202 Gain
DV	IG	Dividend Receipt	US Treasury Interest
DV	IN	Dividend Receipt	Non-Taxable Interest
DV	IS	Dividend Receipt	Interest Income - Securities
DV	IX	Dividend Receipt	Investment Expense -- Interest
DV	MA	Dividend Receipt	Margin Interest Adjustment
DV	MF	Dividend Receipt	Management Fees
DV	NC	Dividend Receipt	Partnership Distribution
DV	ND	Dividend Receipt	Foreign Tax Withheld
DV	NI	Dividend Receipt	Foreign Interest Withholding
DV	NP	Dividend Receipt	Principal Credited

DV	OD	Dividend Receipt	Ordinary Dividend
DV	OI	Dividend Receipt	Other Income
DV	OX	Dividend Receipt	Investment Expense -- OID
DV	PI	Dividend Receipt	Price Improvement
DV	QF	Dividend Receipt	Qualified Dividend
DV	QI	Dividend Receipt	Qualified Interest Income Dividend
DV	RB	Dividend Receipt	Commission or Other Rebate
DV	RE	Dividend Receipt	Dividend Reinvestment
DV	RG	Dividend Receipt	28% Gain Distribution
DV	RI	Dividend Receipt	Industrial Royalties
DV	RM	Dividend Receipt	Motion Picture Royalties
DV	RO	Dividend Receipt	Other Royalties
DV	RP	Dividend Receipt	Short-term Real Property Gain
DV	RY	Dividend Receipt	Natural Resource Royalties
DV	SH	Dividend Receipt	Dividend Short Sale
DV	SI	Dividend Receipt	Short Credit Interest Adjustment
DV	SP	Dividend Receipt	Substitute Payment
DV	SR	Dividend Receipt	Sale of Rights
DV	SS	Dividend Receipt	Sale of Stock
DV	SW	Dividend Receipt	Sale of Warrants
DV	UR	Dividend Receipt	Unrecaptured 1250 Gain
DV	NULL	Dividend Receipt	NULL
EF	CI	Electronic Funds In	Incoming ACH (Customer initiated at the Bank - push)
EF	DD	Electronic Funds In	Direct Deposit
EF	FI	Electronic Funds In	Client Requested Electronic Funding Receipt (Funds NOW)
EF	FO	Electronic Funds Out	Client Requested Electronic Funding Disbursement (Funds NOW)
EF	NA	Electronic Funds In	Electronic New Account Funding
EF	RF	Electronic Funds In	Recurring Electronic Transfer Funding
EF	NULL	Electronic Funds In	NULL
JN	AD	Miscellaneous Journal Action	Accrued Dividend or Interest Payment
JN	AF	Miscellaneous Journal Action	Alternate Address or Payee Fee
JN	AP	Miscellaneous Journal Action	Appraisal or Dissenters Fee
JN	AR	Miscellaneous Journal Action	ACH Return Fee
JN	AV	Miscellaneous Journal Action	Amerivest Maintenance Fee
JN	BC	Miscellaneous Journal Action	Debit Client for Bounced Check
JN	BD	Miscellaneous Journal Action	Beneficiary Distribution Fee

JN	BI	Miscellaneous Journal Action	Bond Interest Payment
JN	BN	Miscellaneous Journal Action	B-Notice Withholding Fee
JN	BP	Miscellaneous Journal Action	Bankruptcy Distribution Payment
JN	BW	Miscellaneous Journal Action	Backup Withholding (W-9)
JN	CA	Miscellaneous Journal Action	Commission Adjustment
JN	CB	Miscellaneous Journal Action	Credit Back Pre-Paid Trades
JN	CD	Miscellaneous Journal Action	Option Cash Distribution
JN	CF	Miscellaneous Journal Action	Roth Conversion Fee
JN	CL	Miscellaneous Journal Action	Cash in Lieu Payment
JN	CR	Miscellaneous Journal Action	Charge Refunding Fee for Pre-Paid Trades
JN	DC	Miscellaneous Journal Action	Craft Copy Request Fee
JN	DD	Miscellaneous Journal Action	ACH Direct Deposits
JN	DF	Miscellaneous Journal Action	Roth Distribution Fee
JN	DP	Miscellaneous Journal Action	Distribution Payment
JN	DR	Miscellaneous Journal Action	Reorganization Deposit Fee
JN	DS	Miscellaneous Journal Action	Short Term Capital Gains
JN	DT	Miscellaneous Journal Action	DWAC Fee
JN	DV	Miscellaneous Journal Action	Duplicate 1099 Fee
JN	DW	Miscellaneous Journal Action	Cash Movement of ESPP transaction
JN	EF	Miscellaneous Journal Action	Early Settlement Fee
JN	EP	Miscellaneous Journal Action	Early Payment Fee
JN	ER	Miscellaneous Journal Action	EF Return Fee
JN	ES	Miscellaneous Journal Action	Monetary Escheatment

JN	FD	Miscellaneous Journal Action	Returned Draft Fee
JN	FE	Miscellaneous Journal Action	Account Transfer Out Fee
JN	FN	Miscellaneous Journal Action	Failed Statement Notification Fee
JN	FS	Miscellaneous Journal Action	Foreign Security Fee
JN	FT	Miscellaneous Journal Action	Free Trade
JN	FW	Miscellaneous Journal Action	Federal Withholding on IRA Distributions
JN	GL	Miscellaneous Journal Action	Transaction between two GL Accounts
JN	IA	Miscellaneous Journal Action	Intra-account Transfer
JN	IF	Miscellaneous Journal Action	Inactivity Fee
JN	IP	Miscellaneous Journal Action	Interest Payment
JN	IR	Miscellaneous Journal Action	Recharacterization Fee
JN	IT	Miscellaneous Journal Action	Internal Transfer of Cash
JN	IW	Miscellaneous Journal Action	International Wire Fee
JN	JN	Miscellaneous Journal Action	Miscellaneous Journal Entry
JN	LA	Miscellaneous Journal Action	LP Annual Fee
JN	LF	Miscellaneous Journal Action	Legal Fee
JN	LP	Miscellaneous Journal Action	Late Payment Fee
JN	LQ	Miscellaneous Journal Action	Liquidation Payment
JN	LR	Miscellaneous Journal Action	LP Review Fee
JN	LT	Miscellaneous Journal Action	LP Transfer In Fee
JN	MF	Miscellaneous Journal Action	Mandatory Reorganization Fee
JN	MI	Miscellaneous Journal Action	Mortgage Interest
JN	MK	Miscellaneous Journal Action	Mark to the Market

JN	MM	Miscellaneous Journal Action	Money Market Movement
JN	MO	Miscellaneous Journal Action	Mortgage Principal
JN	MP	Miscellaneous Journal Action	Mandatory Reorg Payment
JN	NF	Miscellaneous Journal Action	Management Fee
JN	ON	Miscellaneous Journal Action	Overnight Fee
JN	OR	Miscellaneous Journal Action	Wire Out Returns
JN	PC	Miscellaneous Journal Action	Paper Confirmation Fee
JN	PD	Miscellaneous Journal Action	Premature Distribution Fee
JN	PF	Miscellaneous Journal Action	Reorganization Processing Fee
JN	PI	Miscellaneous Journal Action	Preferred Stock – Interest Payment
JN	PM	Miscellaneous Journal Action	Money Market Purchase
JN	PP	Miscellaneous Journal Action	Purchase Pre-Paid Trades
JN	PR	Miscellaneous Journal Action	Promotions
JN	PS	Miscellaneous Journal Action	Paper Statement Fee
JN	QM	Miscellaneous Journal Action	Quarterly Maintenance Fee
JN	QW	Miscellaneous Journal Action	Quarterly Maintenance Fee Write Off
JN	RA	Miscellaneous Journal Action	ACH Returns
JN	RB	Miscellaneous Journal Action	Rebate
JN	RC	Miscellaneous Journal Action	Return Check Fee
JN	RD	Miscellaneous Journal Action	Redemption
JN	RE	Miscellaneous Journal Action	Removal of Excess Fee
JN	RF	Miscellaneous Journal Action	IRA Termination Fee
JN	RI	Miscellaneous Journal Action	Wire in Returns

JN	RJ	Miscellaneous Journal Action	DTC Reject Fee
JN	RM	Miscellaneous Journal Action	Money Market Redemption
JN	RO	Miscellaneous Journal Action	Check Reorder Fee
JN	RP	Miscellaneous Journal Action	Realize Pre-Paid Trades as Revenue
JN	RQ	Miscellaneous Journal Action	Real Time Quote Fee
JN	RR	Miscellaneous Journal Action	Refund from Reorganization
JN	RS	Miscellaneous Journal Action	Restricted Stock Fee
JN	RT	Miscellaneous Journal Action	Reg T Extension Fee
JN	RW	Miscellaneous Journal Action	Reorg Withdrawal Fee
JN	SF	Miscellaneous Journal Action	Software Fee
JN	SL	Miscellaneous Journal Action	Sellout Fee
JN	SO	Miscellaneous Journal Action	ESOP Fee
JN	SP	Miscellaneous Journal Action	Stop Payment on Distribution
JN	SR	Miscellaneous Journal Action	Voluntary - Rights Subscription Fee
JN	SS	Miscellaneous Journal Action	Subscription Service Charge
JN	ST	Miscellaneous Journal Action	Stop Payment Fee
JN	SW	Miscellaneous Journal Action	State Withholding on IRA Distributions
JN	TC	Miscellaneous Journal Action	Security Transfer Fee
JN	TD	Miscellaneous Journal Action	TOD Fee
JN	TI	Miscellaneous Journal Action	Cash Movement of Incoming Account Transfer
JN	TO	Miscellaneous Journal Action	Cash Movement of Outgoing Account Transfer
JN	TP	Miscellaneous Journal Action	Third Party
JN	TR	Miscellaneous Journal Action	{none identified}

JN	TT	Miscellaneous Journal Action	Cash Movement of DVP transaction
JN	TW	Miscellaneous Journal Action	12B-1 Fees
JN	UI	Miscellaneous Journal Action	UIT - Interest Payment
JN	VC	Miscellaneous Journal Action	Void Check
JN	VD	Miscellaneous Journal Action	Verification of Deposit Fee
JN	VF	Miscellaneous Journal Action	Voluntary Fee
JN	VP	Miscellaneous Journal Action	Voluntary - Reorganization Payment
JN	VR	Miscellaneous Journal Action	Voluntary - Reorganization Withdrawal Fee
JN	WB	Miscellaneous Journal Action	Account Write Off - Bankruptcy
JN	WC	Miscellaneous Journal Action	Wire Charge (Fee)
JN	WD	Miscellaneous Journal Action	Wells Fargo ACH Direct Deposits
JN	WE	Miscellaneous Journal Action	Voluntary - Warrant Exercise Fee
JN	WF	Miscellaneous Journal Action	W-8 Withholding
JN	WI	Miscellaneous Journal Action	Wired From Suspense Account to Client
JN	WO	Miscellaneous Journal Action	Account Write Off
JN	WT	Miscellaneous Journal Action	Outgoing wires to 401K customers
JN	XC	Miscellaneous Journal Action	Tax Correction Fee
JN	ZA	Miscellaneous Journal Action	NRA Withholding - Ordinary Income
JN	ZB	Miscellaneous Journal Action	NRA Withholding - Partnership Distribution
JN	ZC	Miscellaneous Journal Action	NRA Withholding - REIT Capital Gains
JN	ZD	Miscellaneous Journal Action	NRA Withholding - Substitute Payment
JN	ZE	Miscellaneous Journal Action	NRA Withholding - Resource Royalties
JN	ZF	Miscellaneous Journal Action	NRA Withholding - Industrial Royalties

JN	ZG	Miscellaneous Journal Action	NRA Withholding - Copyright Royalties
JN	ZH	Miscellaneous Journal Action	NRA Withholding - Other Royalties
JN	NULL	Miscellaneous Journal Action	NULL
MI	{Any}	MARGIN INTEREST	MARGIN INTEREST
MC	{Any}	CLOSE MARGIN CALLS/ MAINTENANCE CALL	NULL
RD	AT	Receive and Deliver	Transfer Request Entries
RD	BD	Receive and Deliver	Bankruptcy Distribution
RD	CA	Receive and Deliver	Recapitalization
RD	CC	Receive and Deliver	Mandatory - Cusip Change
RD	CL	Receive and Deliver	Reclassification
RD	CV	Receive and Deliver	Voluntary - Conversion Offer
RD	DA	Receive and Deliver	Voluntary - Dutch Auction
RD	DM	Receive and Deliver	Default Mandatory
RD	DR	Receive and Deliver	Move Position due to Reorganization
RD	DW	Receive and Deliver	Security movement for ESPP transaction
RD	EM	Receive and Deliver	Voluntary - Election Merger
RD	EO	Receive and Deliver	Voluntary - Exchange Offer
RD	ER	Receive and Deliver	Removal of Expired Rights
RD	ES	Receive and Deliver	Security Escheatment
RD	EW	Receive and Deliver	Removal of Expired Warrants
RD	EX	Receive and Deliver	Mandatory - Exchange
RD	FS	Receive and Deliver	Mandatory - Forward Split

RD	HS	Receive and Deliver	Movement involving Hard Seg location code 80
RD	IA	Receive and Deliver	Internal Transfer between location codes
RD	IT	Receive and Deliver	Internal Transfer between accounts or account types
RD	LC	Receive and Deliver	Leap Conversion
RD	LG	Receive and Deliver	Deposit of Non-Negotiable Securities
RD	MA	Receive and Deliver	Option Position Change
RD	MC	Receive and Deliver	Share Class Conversion
RD	MM	Receive and Deliver	Money Market Movement
RD	MR	Receive and Deliver	Mandatory - Rounding
RD	NA	Receive and Deliver	Option Name Change
RD	NC	Receive and Deliver	Mandatory - Name Change
RD	NP	Receive and Deliver	Non Pre-refunding
RD	NS	Receive and Deliver	Non-Taxable Spin off/Liquidation Distribution
RD	OA	Receive and Deliver	Removal of Option due to assignment
RD	OE	Receive and Deliver	Removal of Option due to exercise
RD	OL	Receive and Deliver	Voluntary - Odd Lot Offer
RD	OX	Receive and Deliver	Removal of Option due to expiration
RD	PC	Receive and Deliver	Move Position Due to Partial Call
RD	PD	Receive and Deliver	Physical Deposit of Negotiable Securities
RD	PF	Receive and Deliver	Pre-refunding
RD	PL	Receive and Deliver	Partial Call Lottery
RD	PM	Receive and Deliver	Money Market Purchase
RD	PO	Receive and Deliver	Remove Paid Out Securities

RD	PR	Receive and Deliver	Pre-refunding
RD	RC	Receive and Deliver	Option Reorganization/Corporate Action
RD	RD	Receive and Deliver	Redemption
RD	RE	Receive and Deliver	Option Right Expiration
RD	RM	Receive and Deliver	Money Market Redemption
RD	RN	Receive and Deliver	Option Right Distribution
RD	RO	Receive and Deliver	Voluntary - Rights Oversubscription
RD	RP	Receive and Deliver	Remove Position Per Client Request
RD	RS	Receive and Deliver	Mandatory Reverse Split
RD	RT	Receive and Deliver	Voluntary - Rights Subscription
RD	SA	Receive and Deliver	Scheme of Arrangement
RD	SD	Receive and Deliver	Stock Dividend
RD	SE	Receive and Deliver	Option Even Split Reorganization
RD	SN	Receive and Deliver	Option Spin Off
RD	SO	Receive and Deliver	Option Odd Split Reorganization
RD	SP	Receive and Deliver	Stock Split
RD	SU	Receive and Deliver	Rights Subscription
RD	TD	Receive and Deliver	Voluntary - Tender Offer
RD	TI	Receive and Deliver	Transfer of Security or Option In
RD	TO	Receive and Deliver	Transfer of Security or Option Out
RD	TP	Receive and Deliver	{none identified}
RD	TR	Receive and Deliver	Dividend Reinvestment (Shares)
RD	TS	Receive and Deliver	Taxable Spin off/Stock Dividend

RD	TT	Receive and Deliver	Security movement for DVP transaction
RD	US	Receive and Deliver	Mandatory - Unit Separation
RD	WE	Receive and Deliver	Voluntary - Warrant Exercise
RD	WS	Receive and Deliver	Removal of Worthless Securities
RD	ZI	Receive and Deliver	Transfer of Security in Safekeeping
RD	ZO	Receive and Deliver	Transfer of Security Out Safekeeping
RD	NULL	Receive and Deliver	NULL
SM	{Any}	SMA Adjustment	SMA Adjustment
TR	AT	Trade	Mandatory - ADR Termination
TR	BY	Trade	Buy Trade
TR	CM	Trade	Mandatory - Cash Merger
TR	CS	Trade	Close Short Position
TR	DM	Trade	{none identified}
TR	DP	Trade	Voluntary - Dutch Auction Payment
TR	DR	Trade	Dividend Reinvest
TR	DT	Trade	DWAC Fee
TR	FC	Trade	Bonds - Full Call
TR	LP	Trade	Mandatory - Liquidation Payment
TR	MA	Trade	Bonds - Maturity
TR	OA	Trade	Option Assignment
TR	OE	Trade	Option Exercise
TR	OP	Trade	Voluntary - Odd Lot Tender Payment
TR	PC	Trade	Bonds - Partial Call
TR	PO	Trade	Voluntary - Odd Lot Purchase Payment
TR	RE	Trade	Bonds - Redemption
TR	RR	Trade	Redemption of Rights
TR	RW	Trade	Redemption of Warrants
TR	SL	Trade	Sell Trade
TR	SS	Trade	Short Sale
TR	TA	Trade	{none identified}
TR	TC	Trade	Trade Correction
TR	TO	Trade	Voluntary - Tender Payment
TR	UB	Trade	UIT Rollover - Buy
TR	US	Trade	UIT Rollover - Sell
TR	UT	Trade	Voluntary - UIT Termination
TR	VO	Trade	Voluntary - Odd Lot Tender Payment
TR	WI	Trade	When Issued Security

TR	NULL	Trade	NULL
WI	WI	Wire In	Wire Incoming
WI	NULL	Wire In	NULL
WO	AC	Wire Out	Wire Outgoing (ACD Wire Disbursements)
WO	IW	Wire Out	International Wire
WO	TP	Wire Out	{none identified}
WO	WO	Wire Out	Wire Outgoing (Non ACD Wire Disbursements)
WO	NULL	Wire Out	NULL

19.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/History?source=<**#sourceID#**>&type=1

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <history>
        <account-id>123456789</account-id>
        <startDate>Jan 01 2008</startDate>
        <endDate>Apr 08 2008</endDate>
        <searchTransactionType>1</searchTransactionType>
        <requestType>T</requestType>
        <transaction-list>
            <error></error>
            <orderDateTime>2008-03-26 15:04:14 EDT</orderDateTime>
            <type>TR</type>
            <subType>BY</subType>
            <buySellCode>B</buySellCode>
            <assetType>S</assetType>
            <symbol>SSO</symbol>
            <description>BUY TRADE</description>
            <cusip>74347R107</cusip>
            <price>68.70</price>
            <quantity>800</quantity>
            <transactionId>3759123456</transactionId>
            <value>-54,965.00</value>
            <commission>5.00</commission>
            <orderNumber>3741234567</orderNumber>
            <fees></fees>
            <additionalFees></additionalFees>
            <cashBalanceEffect>Y</cashBalanceEffect>
            <openClose></openClose>
            <optionType></optionType>
            <optionStrike></optionStrike>
            <accruedInterest></accruedInterest>
            <parentChildIndicator></parentChildIndicator>
            <sharesBefore></sharesBefore>
            <sharesAfter></sharesAfter>
```

```
            <otherCharges>0.00</otherCharges>
            <redemptionFee></redemptionFee>
            <cdscFee></cdscFee>
            <bondInterestRate></bondInterestRate>
            </transaction-list>
         <transaction-list>
            <error></error>
            <orderDateTime>2008-03-25 15:36:33 EDT</orderDateTime>
            <type>TR</type>
            <subType>SL</subType>
            <buySellCode>S</buySellCode>
            <assetType>S</assetType>
            <symbol>SDS</symbol>
            <description>SELL TRADE</description>
            <cusip>74347R883</cusip>
            <price>61.1301</price>
            <quantity>800</quantity>
            <transactionId>3756123456</transactionId>
            <value>48,898.54</value>
            <commission>5.00</commission>
            <orderNumber>3738123456</orderNumber>
            <fees>0.54</fees>
            <additionalFees></additionalFees>
            <cashBalanceEffect>Y</cashBalanceEffect>
            <openClose></openClose>
            <optionType></optionType>
            <optionStrike></optionStrike>
            <accruedInterest></accruedInterest>
            <parentChildIndicator></parentChildIndicator>
            <sharesBefore></sharesBefore>
            <sharesAfter></sharesAfter>
            <otherCharges>0.00</otherCharges>
            <redemptionFee></redemptionFee>
            <cdscFee></cdscFee>
            <bondInterestRate></bondInterestRate>
         </transaction-list>
      </history>
   </amtd>
```


19.6 History Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

20 QuoteDelayed

This service provides clients the ability to retrieve delayed quotes; used by the wireless site to provide quotes to clients that are not logged in.

20.1 QuoteDelayed Request

Request URL

https://apis.tdameritrade.com/apps/100/QuoteDelayed?source=<**#sourceID#**>&symbol=<**#symbol#**>

For information on QuoteDelayed parameters, response, samples and errors, refer Snapshot (XML) Quotes

21 MarketOverview

This service provides clients a snapshot of the five indices.

21.1 MarketOverview Request

Request URL

https://apis.tdameritrade.com/apps/100/MarketOverview?source=AMTDTest&format=compact&submit=Submit

Request Parameters

No parameters required.

21.2 MarketOverview Parameters

No parameters required.

21.3 MarketOverview Response

The response to the MarketOverview request will be in XML format, general structure of the response will be as follows:

General Structure of XML Response


```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <market-snapshot>
        <quote-timestamp></quote-timestamp>
        <markets>
            <index>
                <error/>
                <symbol></symbol>
                <description></description>
                <open></open>
                <high></high>
                <low></low>
                <last></last>
                <last-trade-date></last-trade-date>
                <close></close>
                <year-high></year-high>
                <year-low></year-low>
                <real-time></real-time>
                <asset-type></asset-type>
                <change></change>
                <change-percent></change-percent>
                <volume></volume>
            </index>
            <index>
                <error/>
                <symbol></symbol>
                <description></description>
                <open></open>
                <high></high>
                <low></low>
                <last></last>
                <last-trade-date></last-trade-date>
                <close></close>
                <year-high></year-high>
                <year-low></year-low>
                <real-time></real-time>
                <asset-type></asset-type>
                <change></change>
                <change-percent></change-percent>
                <volume></volume>
            </index>
            ......

        </markets>
```

```xml
    </market-snapshot>
</amtd>
```


21.4 MarketOverview Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/MarketOverview?source=AMTDTest&format=compact&submit=Submit

Response:


```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <market-snapshot>
        <quote-timestamp>2009-07-20 13:35:25 CDT</quote-timestamp>
        <markets>
            <index>
                <error/>
                <symbol>$COMPX</symbol>
                <description>NASDAQ COMPOSITE</description>
                <open>1896.99</open>
                <high>1906.24</high>
                <low>1890.00</low>
                <last>1898.64</last>
                <last-trade-date>2009-07-20 14:35:24 EDT</last-trade-date>
                <close>1886.61</close>
                <year-high>2485.00</year-high>
                <year-low>1265.52</year-low>
                <real-time>true</real-time>
                <asset-type>I</asset-type>
                <change>12.03</change>
                <change-percent>0.64%</change-percent>
                <volume>1336260384</volume>
            </index>
            <index>
                <error />
                <symbol>$DJI</symbol>
                <description>DOW JONES INDUSTRIAL AVERAGE</description>
                <open>8746.05</open>
                <high>8827.07</high>
                <low>8745.90</low>
                <last>8809.46</last>
                <last-trade-date>2009-07-20 14:35:23 EDT</last-trade-date>
                <close>8743.94</close>
                <year-high>12369.23</year-high>
                <year-low>6469.95</year-low>
                <real-time>true</real-time>
                <asset-type>I</asset-type>
                <change>65.52</change>
                <change-percent>0.75%</change-percent>
                <volume>129705882</volume>
            </index>
```

```
<index>
    <error />
    <symbol>$NYA.X</symbol>
    <description>NYSE COMPOSITE INDEX NEW</description>
    <open>6038.11</open>
    <high>6128.91</high>
    <low>6038.11</low>
    <last>6117.68</last>
    <last-trade-date>2009-07-20 14:35:23 EDT</last-trade-date>
    <close>6038.11</close>
    <year-high>9408.30</year-high>
    <year-low>4181.75</year-low>
    <real-time>true</real-time>
    <asset-type>I</asset-type>
    <change>79.57</change>
    <change-percent>1.32%</change-percent>
    <volume>0</volume>
</index>
<index>
    <error />
    <symbol>$SPX.X</symbol>
    <description>S&P 500 Index</description>
    <open>942.07</open>
    <high>948.90</high>
    <low>940.99</low>
    <last>946.70</last>
    <last-trade-date>2009-07-20 14:35:23 EDT</last-trade-date>
    <close>940.38</close>
    <year-high>1370.63</year-high>
    <year-low>666.79</year-low>
    <real-time>true</real-time>
    <asset-type>I</asset-type>
    <change>6.32</change>
    <change-percent>0.67%</change-percent>
    <volume>0</volume>
</index>
<index>
    <error />
    <symbol>$XAX.X</symbol>
    <description>AMEX COMPOSITE INDEX-XAX</description>
    <open>1624.55</open>
    <high>1638.02</high>
    <low>1622.48</low>
    <last>1636.18</last>
    <last-trade-date>2009-07-20 14:35:19 EDT</last-trade-date>
    <close>1624.55</close>
    <year-high>2374.66</year-high>
    <year-low>354.60</year-low>
    <real-time>true</real-time>
    <asset-type>I</asset-type>
    <change>11.63</change>
    <change-percent>0.72%</change-percent>
    <volume>0</volume>
```

```
            </index>
        </markets>
    </market-snapshot>
</amtd>
```

22 News

This service provides clients a list of the top news headlines. This command is provided "As Is" from CBS MarketWatch with minimal documentation.

22.1 News Request

Request URL

https://apis.tdameritrade.com/apps/100/NewsManager?source=AMTDTest&type=A&format=compact&submit=Submit

Request Parameters

There is a "type" parameter associated with the news headline retrieval. Type = A or empty means both the top ten and market headlines are returned. Type=M returns just the market news, and type= T returns just the top ten news headlines.

22.2 News Parameters

News Request Parameters

There is a "type" parameter associated with the news headline retrieval. Type = A or empty means both the top ten and market headlines are returned. Type=M returns just the market news, and type= T returns just the top ten news headlines.

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
Source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
Type	Type is a parameter that indicates what information you want.	NO	**Type = A** or empty means both the top ten and market headlines are returned. **Type=M** returns just the market news. **Type= T** returns just the top ten news headlines.

22.3 News Response

The response to the News request will be in XML format, the response is passed through as is from CBS MarketWatch.

22.4 News Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/NewsManager?source=AMTDTest&type=A&format=compact&submit=Submit

Response:


```
<?xml version="1.0" ?>
   <amtd>
      <result>OK</result>
         <LIST NEXTKEY="40016.68125-965868853">
         <ITEM idDoc="965868853" idTrack="102" idTrackType="102">
         <NEWSITEM>
         <DOCTYPE ID="103">
            <NAME>MarketWatch Databased News</NAME>
            <PROVIDER>MarketWatch.com</PROVIDER>
         </DOCTYPE>
         <HEADLINE>U.S. stocks up on earnings, less so on revenue</HEADLINE>

         <COLUMN>Market Snapshot</COLUMN>
         <TARGETING PRIORITY="DAILY">
            <CHANNEL NAME="News & Commentary" ID="1">
                  <SUBCHANNEL NAME="News & Commentary|Markets" ID="
            1025" />
            </CHANNEL>
         </TARGETING>
         <CODE-LIST NSURI="urn:marketwatch-com:Channel" CATEGORY="
         Target">
            <CODE>Markets</CODE>
         </CODE-LIST>
         <CODE-LIST NSURI="urn:marketwatch-com:Subchannel" CATEGORY="
         Target">
            <CODE>Markets|U.S. &Canada</CODE>
         </CODE-LIST>
         <PACKAGES> </PACKAGES>
         <PLACEMENT-LIST>
            <PLACEMENT>cbs</PLACEMENT>
         </PLACEMENT-LIST>
         <INDUSTRY-LIST>  </INDUSTRY-LIST>
         <ISSUE-LIST>
            <ISSUE>Markets/Exchanges</ISSUE>
            <ISSUE>Market News</ISSUE>
         </ISSUE-LIST>
         <FUNDTYPE-LIST> </FUNDTYPE-LIST>
         <REGION-LIST> </REGION-LIST>
         <PERSONALFINANCE-LIST>  </PERSONALFINANCE-LIST>
         <RELATED-LIST>  </RELATED-LIST>
         <ABSTRACT>
            <PARAGRAPH TYPE="NORMAL">With quarterly financial results now
```

```
reported by more than a fifth of S&P 500 companies, actual
operating earnings are 10% ahead of estimates but rank nearly
30% behind year-ago figures, according to Standard &Poor's
calculations.</PARAGRAPH>
</ABSTRACT>
<SUMMARY ID="">
<SUMMARY-BODY>
<HEADLINE>U.S. stocks up on earnings, less so on revenue</
HEADLINE>
<PARAGRAPH TYPE="NORMAL">With quarterly financial results now
reported by more than a fifth of S&P 500 companies, actual
operating earnings are 10% ahead of estimates but rank nearly
30% behind year-ago figures, according to Standard &Poor's
calculations.</PARAGRAPH>
</SUMMARY-BODY>
</SUMMARY>
<CREATED TIMESTAMP="40016.68125" ET-FORMAT="4:21 PM ET Jul 22,
2009" CTIME-FORMAT="1248301260">7/22/2009 4:21:00 PM
</CREATED>
<LASTUPDATED TIMESTAMP="40016.68125" ET-FORMAT="4:21 PM ET Jul
22, 2009" CTIME-FORMAT="1248301260">7/22/2009 4:21:00 PM
</LASTUPDATED>
<AUTHOR GUID="{E2958AE6-BB1D-0A4C-A3E9-DBAFBCE64306}"
ORGANIZATION="MarketWatch" ADDRESS="
mailto:kgibson@marketwatch.com" BIO="Kate Gibson is a reporter for
MarketWatch, based in New York.">Kate Gibson
</AUTHOR>
<GUID>{AC9AF409-EF15-4AA4-BAC4-D84F773D819A}</GUID>
<LOCATION>
<STORAGE GUID="{AC9AF409-EF15-4AA4-BAC4-D84F773D819A}" SLUG
="snapshot" />
</LOCATION>
<SYMBOLS>
<SYMBOL COUNTRY="US" TICKER="$INDU">$INDU</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="COMP">COMP</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="$SPX">$SPX</SYMBOL>
</SYMBOLS>
</NEWSITEM>
</ITEM>
</LIST>
</amtd>
```


22.5 News Errors

Possible Errors

If the CBS feed is down, user would get following error message:

"System Unavailable"

23 FullStoryNews

This service provides clients the full story of the headline they chose to view. This command is provided "As Is" from CBS MarketWatch with minimal documentation.

23.1 FullStoryNews Request

Request URL

https://apis.tdameritrade.com/apps/100/FullStoryNews?source=AMTDTest&GUID={AB904891-9A58-48E1-8B35-2C35EEA45913}&format=compact&submit=Submit

Request Parameters

The GUID is required to retrieve a specific story. The GUID is in the response from CBS market Watch from the headline news.

23.2 FullStoryNews Parameters

FullStoryNews Parameters

The GUID is required to retrieve a specific story. The GUID is in the response from CBS market Watch from the headline news.

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
Source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
GUID	GUID is a news story identification ID provided by CBS Marketwatch. That ID is used to retrieve a particular story. The format of the GUID is: **{B4C52924-044D-41A3-9F22-629E3199E4F8}**	YES	This is a unique ID provided by CBS. For example, **{B4C52924-044D-41A3-9F22-629E3199E4F8}**

23.3 FullStoryNews Response

The response to the FullStoryNews request will be in XML format, the response is passed through as is from CBS MarketWatch.

23.4 FullStoryNews Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/FullStoryNews?source=AMTDTest&GUID={AB904891-9A58-48E1-8B35-2C35EEA45913}&format=compact&submit=Submit

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
        <LIST>
        <ITEM idDoc="957828744" idTrack="103" idTrackType="103">
            <NEWSITEM>
            <DOCTYPE ID="103">
                <NAME>MarketWatch Databased News</NAME>
                <PROVIDER>MarketWatch.com</PROVIDER>
            </DOCTYPE>
            <HEADLINE>Investors warm to Geithner comments; banks stage
            turnaround</HEADLINE>
            <COLUMN>Financial Stocks</COLUMN>
            <TARGETING PRIORITY="DAILY">
                <CHANNEL NAME="News & Commentary" ID="1">
                    <SUBCHANNEL NAME="News & Commentary|Markets" ID="1025" />
                </CHANNEL>
            </TARGETING>
            <PACKAGES> </PACKAGES>
            <PLACEMENT-LIST>
                <PLACEMENT>cbs</PLACEMENT>
            </PLACEMENT-LIST>
            <INDUSTRY-LIST>
                <INDUSTRY>Banks</INDUSTRY>
                <INDUSTRY>Financial Services</INDUSTRY>
            </INDUSTRY-LIST>
            <ISSUE-LIST>
                <ISSUE>Markets/Exchanges</ISSUE>
                <ISSUE>Market News</ISSUE>
            </ISSUE-LIST>
            <FUNDTYPE-LIST> </FUNDTYPE-LIST>
            <REGION-LIST>  </REGION-LIST>
            <PERSONALFINANCE-LIST> </PERSONALFINANCE-LIST>
            <RELATED-LIST>  </RELATED-LIST>
            <ABSTRACT>
                <PARAGRAPH TYPE="NORMAL">The financial sector stages a dramatic
                turnaround on soothing words from Treasury Secretary Timothy
                Geithner, after banking stocks open lower with several key names
                reporting disappointing quarterly results.
                </PARAGRAPH>
            </ABSTRACT>
            <CREATED TIMESTAMP="39924.6902777778" ET-FORMAT="4:34 PM ET Apr
            21, 2009" CTIME-FORMAT="1240353240">4/21/2009 4:34:00 PM
            </CREATED>
            <LASTUPDATED TIMESTAMP="39924.6902777778" ET-FORMAT="4:34 PM ET
```

Apr 21, 2009" CTIME-FORMAT="1240353240">**4/21/2009 4:34:00 PM**
</LASTUPDATED>
<AUTHOR GUID="{3FE480F5-108F-11D4-8BE7-005004D02F49}">
MarketWatch</AUTHOR>
<GUID>**{AB904891-9A58-48E1-8B35-2C35EEA45913}**</GUID>
<LOCATION>
<STORAGE GUID="{AB904891-9A58-48E1-8B35-2C35EEA45913}"
SLUG="sector_financial" />
</LOCATION>
<SYMBOLS>
<SYMBOL COUNTRY="US" TICKER="ZION">**ZION**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="HBAN">**HBAN**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="KEY">**KEY**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="FITB">**FITB**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="RF">**RF**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="NTRS">**NTRS**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="BK">**BK**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="KBE">**KBE**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="BAC">**BAC**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="USB">**USB**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="CMA">**CMA**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="BLK">**BLK**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="AMTD">**AMTD**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="C">**C**</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="MS">**MS**</SYMBOL>
</SYMBOL>
<BODY>
<SECTION TYPE="HEADER">
<PARAGRAPH TYPE="HEADLINE">**Investors like what they hear from Geithner**
</PARAGRAPH>
<PARAGRAPH TYPE="HEADLINE2">**Financial sector bounces back from early losses; earnings in full swing**
</PARAGRAPH>
</SECTION>
<SECTION TYPE="INFORMATION">
<PARAGRAPH TYPE="BYLINE" BIO="">
By
<AUTHOR GUID="{3FE480F5-108F-11D4-8BE7-005004D02F49}">
MarketWatch</AUTHOR>
</PARAGRAPH>
<PARAGRAPH TYPE="LASTUPDATE">
Last Update:
<LASTUPDATED TIMESTAMP="39924.6902777778">**4:34 PM ET Apr 21, 2009**</LASTUPDATED>
</PARAGRAPH>
</SECTION>
<SECTION TYPE="CONTENT">
<PARAGRAPH TYPE="NORMAL">**BOSTON (MarketWatch) -- The financial sector staged a dramatic turnaround Tuesday on soothing words from Treasury Secretary Timothy Geithner, after banking stocks opened lower with several key names reporting disappointing quarterly results.**

```
    </PARAGRAPH>
    <PARAGRAPH TYPE="NORMAL">Geithner, speaking before the
    financial bailout package's congressional oversight panel, said most
    U.S. banks have adequate capital and that there are signals that
    credit markets are on the mend.
    <CROSSREF GUID="{4F9086D2-7DAA-4037-9F80-7400B2C0D4FB}"
    LASTUPDATE="39924.5847222222"                ARCHIVE="20090421"
    SLUG="bailout" HEADLINE="TARP funds sufficient to do the job, Treasury's
    Geithner says">
    Read more on Geithner's testimony.
    </CROSSREF>
    </PARAGRAPH>
    <PARAGRAPH TYPE="NORMAL">Investors were also mulling quarterly
    results from a bevy of regional and custody banks, including Zions
    Bancorp
    <SYMBOL TICKER="ZION" COUNTRY="US" />
    , Huntington Bancshares Inc.
    <SYMBOL TICKER="HBAN" COUNTRY="US" />
    , KeyCorp
    <SYMBOL TICKER="KEY" COUNTRY="US" />
    , Fifth Third Bancorp
    <SYMBOL TICKER="FITB" COUNTRY="US" />
    , Regions Financial Corp.
    <SYMBOL TICKER="RF" COUNTRY="US" />
    , Northern Trust Corp.
    <SYMBOL TICKER="NTRS" COUNTRY="US" />
    and Bank of New York Mellon Corp.
    <SYMBOL TICKER="BK" COUNTRY="US" />
    </PARAGRAPH>
    <PARAGRAPH TYPE="NORMAL">
    The SPDR KBW Bank ETF
    <SYMBOL TICKER="KBE" COUNTRY="US" />
    , an exchange-traded fund tied to banking stocks, closed up nearly
    7% after declining earlier in the session. The bank ETF lost about
    15% on Monday when Bank of America Corp.'s
    <SYMBOL TICKER="BAC" COUNTRY="US" />
    warning about lingering credit losses spooked investors.
    </PARAGRAPH>
    <PARAGRAPH TYPE="NORMAL">After opening lower Tuesday, the
    financial sector shot higher in morning         trading around the
    time Geithner's prepared remarks were released.
    </PARAGRAPH>
<SIDEBAR PADDING="7">
    <PARAGRAPH TYPE="GRAPHIC">
    <CHART APPEARANCE="S1" TITLE="KBE" TICKER="KBE"
    ARGS="symb=KBE&time=4&freq=1&compidx=aaaaa:0&comp=&uf=0&lf=
    1&lf2=0&lf3=0&state=0&sid=2814487&startdate=&enddate=39924&noset
    tings=1">
    <IMAGE WIDTH="125" HEIGHT="99" TRACK="201" />
    </CHART>
    </PARAGRAPH>
</SIDEBAR>
    <PARAGRAPH TYPE="NORMAL">In earnings news, custody banks
```

Northern Trust and Bank of New York Mellon, which slashed its dividend, saw their profits dinged by falling stock values.
<CROSSREF GUID="{E0F373CB-D359-4380-B6E5-58E149B3D3D9}" LASTUPDATE="39924.3618055556" ARCHIVE="20090421" SLUG="bk" HEADLINE="Custody banks' profits hit by falling stock values, results show">
See earnings coverage.
</CROSSREF>
</PARAGRAPH>
<PARAGRAPH TYPE="NORMAL">**Several regional banks also released quarterly results.**</PARAGRAPH>
<PARAGRAPH TYPE="NORMAL">**Zions Bancorp's shares fell after the bank swung to a first-quarter net loss on a write-down of goodwill and other charges, prompting Moody's Investors Service to slash its ratings.**
<CROSSREF GUID="{67495ACC-0E65-4B18-9A15-CA8B9FCD8513}" LASTUPDATE="0" ARCHIVE="18991230" SLUG="" HEADLINE="Zions Bancorp Swings To 1Q Loss; Moody's Cuts Ratings">
Read more on Moody's downgrade of Zions.
</CROSSREF>
</PARAGRAPH>
<PARAGRAPH TYPE="NORMAL">**KeyCorp of Cleveland reported a quarterly loss of nearly $500 million and set plans to cut its dividend to beef up its balance sheet.**
</PARAGRAPH>
<PARAGRAPH TYPE="NORMAL">**Regions Financial's first-quarter net income slid 77% on still-rising credit costs, while the company reported deposit growth.**
<CROSSREF GUID="{4F823F24-9CA4-4EAF-BEB3-7FDC1D592C70}" LASTUPDATE="0" ARCHIVE="18991230" SLUG="" HEADLINE="Regions Financial 1Q Net Down 77% On Credit Woes">**See complete article on Regions Financial.**
</CROSSREF>
</PARAGRAPH>
<PARAGRAPH TYPE="NORMAL">**Also in regional banks, Ohio's Huntington Bancshares said Tuesday that it swung to a $2.49 billion loss in the first quarter after it took a goodwill-impairment charge and boosted losses for credit provisions.**
<CROSSREF GUID="{E3070B25-94A2-4BE2-88D1-3A4C59A9CDC5}" LASTUPDATE="39924.3381944444" ARCHIVE="20090421" SLUG="hban" HEADLINE="Huntington Bancshares swings to $2.5 billion loss">**Read full story on Huntington Bancshares.**
</CROSSREF>
</PARAGRAPH>
<PARAGRAPH TYPE="NORMAL">**Other financial-services firms reporting quarterly results included U.S. Bancorp**
<SYMBOL TICKER="USB" COUNTRY="US" />
, Comerica Inc.
<SYMBOL TICKER="CMA" COUNTRY="US" />
, BlackRock Inc
<SYMBOL TICKER="BLK" COUNTRY="US" />
and TD Ameritrade Holding Corp.
<SYMBOL TICKER="AMTD" COUNTRY="US" />

```
            </PARAGRAPH>
            <PARAGRAPH TYPE="NORMAL">Aside from the crush of earnings
            reports, traders were looking ahead to results from the
            government's "stress tests" for banks, and also to a key Bank of
            America shareholder meeting with Chief Executive Ken Lewis, who
            is facing criticism over a string of high-profile acquisitions.
            </PARAGRAPH>
            <PARAGRAPH TYPE="NORMAL">
            Another bank CEO on the hot seat, Citigroup Inc.'s
            <SYMBOL TICKER="C" COUNTRY="US" />
            Vikram Pandit, made headlines Tuesday. Senior officials at the
            Federal Deposit Insurance Corp. have privately discussed who
            might replace Pandit as Citigroup's CEO if the bank needed more
            government aid, the Financial Times reported, citing a person
            familiar with the matter. Citi was holding its shareholder meeting
            Tuesday.
            </PARAGRAPH>
            <PARAGRAPH TYPE="NORMAL">Separately, a report by the TARP
            watchdog said the Treasury Department should be more vigilant in
            managing its financial bailout so taxpayer dollars are protected and
            programs are more fraud-resistant, accountable and transparent.
            <INTERNET
            URL="http://online.wsj.com/article/SB124028315955338071.html"
            LOCATION="EXTERNAL"TYPE="UNKNOWN">
            See more coverage at WSJ.com
            </INTERNET>
            </PARAGRAPH>
            <PARAGRAPH TYPE="NORMAL">Also, Morgan Stanley
            <SYMBOL TICKER="MS" COUNTRY="US" />
            is considering the acquisition of U.S. regional banks in a move to
            boost the company's retail banking operations, according to a
            published report.
            <CROSSREF GUID="{5B05A6B5-3D01-4915-989B-9847571CA9AA}"
            LASTUPDATE="39923.9388888889"
            ARCHIVE="20090420" SLUG="ms" HEADLINE="Morgan Stanley mulling
            buy of U.S. regional banks: report">      Read more on Morgan Stanley.
            </CROSSREF>
            </PARAGRAPH>
         </SECTION>
         <SECTION TYPE="FOOTER">  </SECTION TYPE="FOOTER">
         </BODY>
         </NEWSITEM>
      </ITEM>
   </LIST>
</amtd>
```

23.5 FullStoryNews Errors

Possible Errors

If user has not entered the symbol, they would get the following message:

```
<?xml version="1.0" ?>
```

```
<amtd>
   <result>FAIL</result>
   <error>Error -- No Symbol entered.</error>
</amtd>
```


If MarketWatch feed is down, user would get following error message:

"System Unavailable"

24 QuoteNews

This service displays to clients the top stories based on the symbol the client entered to retrieve news on. This command is provided "As Is" from CBS MarketWatch with minimal documentation.

24.1 QuoteNews Request

Request URL

https://apis.tdameritrade.com/apps/100/QuoteNews?source=AMTDTest&symbol=AMTD

Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
Source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
Symbol	List of symbols for which the quote news should be returned.	YES	Any valid symbol. Symbols should be comma separated.

24.2 QuoteNews Parameters

QuoteNews Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES

Source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
Symbol	List of symbols for which the quote news should be returned.	YES	Comma separated list of symbols.

24.3 QuoteNews Response

The response to the FullStoryNews request will be in XML format, the response is passed through as is from CBS MarketWatch.

24.4 QuoteNews Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/QuoteNews?source=AMTDTest&symbol=AMTD&format=compact&submit=Submit

Response:


```
<?xml version="1.0" ?>
    <amtd>
        <result>OK</result>
            <XML_MULTISYMBOL_NEWS>
            <LIST SYMBOL="AMTD" SID="2192348" NEXTKEY="
            39924.7201388889-957833622">
            <ITEM idDoc="965369676" idTrack="102" idTrackType="102">
            <NEWSITEM>
            <DOCTYPE ID="103">
                <NAME>MarketWatch Databased News</NAME>
                <PROVIDER>MarketWatch.com</PROVIDER>
            </DOCTYPE>
            <HEADLINE>Financials remain higher on CIT rescue</HEADLINE>
            <COLUMN>Financial Stocks</COLUMN>
            <TARGETING PRIORITY="DAILY">
                <CHANNEL NAME="News & Commentary" ID="1">
                    <SUBCHANNEL NAME="News & Commentary|Markets" ID="
                    1025" />
                </CHANNEL>
            </TARGETING>
            <CODE-LIST NSURI="urn:marketwatch-com:Channel" CATEGORY="
            Target">
                <CODE>Markets</CODE>
            </CODE-LIST>
            <CODE-LIST NSURI="urn:marketwatch-com:Subchannel" CATEGORY="
            Target">
                <CODE>Markets|U.S. &Canada</CODE>
            </CODE-LIST>
```

<PACKAGES> </PACKAGES>
<PLACEMENT-LIST>
<PLACEMENT>**cbs**</PLACEMENT>
</PLACEMENT-LIST>
<INDUSTRY-LIST>
<INDUSTRY>**Banks**</INDUSTRY>
<INDUSTRY>**Financial Services**</INDUSTRY>
</INDUSTRY-LIST>
<ISSUE-LIST>
<ISSUE>**Newsmakers**</ISSUE>
<ISSUE>**SEC**</ISSUE>
<ISSUE>**Earnings**</ISSUE>
<ISSUE>**Markets/Exchanges**</ISSUE>
<ISSUE>**Analyst**</ISSUE>
<ISSUE>**Company Announcements**</ISSUE>
<ISSUE>**Market News**</ISSUE>
</ISSUE-LIST>
<FUNDTYPE-LIST> </FUNDTYPE-LIST>
<REGION-LIST>
<REGION>**Washington DC**</REGION>
<REGION>**US**</REGION>
</REGION-LIST>
<PERSONALFINANCE-LIST> </PERSONALFINANCE-LIST>
<RELATED-LIST> </RELATED-LIST>
<ABSTRACT>
<PARAGRAPH TYPE="**NORMAL**">**Investors in financial firms cheered news that CIT Group has secured financing, easing fears for the time being that the business lender will file for bankruptcy protection.**
</PARAGRAPH>
</ABSTRACT>
<SUMMARY ID="">
<SUMMARY-BODY>
<HEADLINE>**Financials remain higher on CIT rescue**</HEADLINE>
<PARAGRAPH TYPE="**NORMAL**">**Investors in financial firms cheered news that CIT Group has secured financing, easing fears for the time being that the business lender will file for bankruptcy protection.**
</PARAGRAPH>
</SUMMARY-BODY>
</SUMMARY>
<CREATED TIMESTAMP="**40014.6666666667**" ET-FORMAT="**4:00 PM ET Jul 20, 2009**" CTIME-FORMAT="**1248127200**">**7/20/2009 4:00:00 PM**
</CREATED>
<LASTUPDATED TIMESTAMP="**40014.6666666667**" ET-FORMAT="**4:00 PM ET Jul 20, 2009**" CTIME-FORMAT=" **1248127200**">
7/20/2009 4:00:00 PM</LASTUPDATED>
<AUTHOR GUID="**{B1EC6A08-BFC2-0642-93E2-6DB9DCB28E7A}**" ORGANIZATION="**MarketWatch**" ADDRESS="
mailto:rwilliams@marketwatch.com" BIO="**Ryan Williams is a reporter for MarketWatch, based in New York.**">**Ryan Williams**
</AUTHOR>
<GUID>**{7A460343-08C5-4B1E-B870-57C3807C6BCF}**</GUID>

```
<LOCATION>
    <STORAGE GUID="{7A460343-08C5-4B1E-B870-57C3807C6BCF}"
    SLUG="sector_financial" />
</LOCATION>
<SYMBOLS>
    <SYMBOL COUNTRY="US" TICKER="XLF">XLF</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="CIT">CIT</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="STT">STT</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="WFC">WFC</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="AXP">AXP</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="MTB">MTB</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="AMTD">AMTD</SYMBOL>
    <SYMBOL COUNTRY="US" TICKER="SCHW">SCHW</SYMBOL>
</SYMBOLS>
</NEWSITEM>
</ITEM>
<ITEM idDoc="965356483" idTrack="102" idTrackType="102">
<NEWSITEM>
<DOCTYPE ID="103">
    <NAME>MarketWatch Databased News</NAME>
    <PROVIDER>MarketWatch.com</PROVIDER>
</DOCTYPE>
<HEADLINE>Financials get lift from CIT emergency rescue</HEADLINE
>
<COLUMN>Financial Stocks</COLUMN>
<TARGETING PRIORITY="DAILY">
    <CHANNEL NAME="News & Commentary" ID="1">
        <SUBCHANNEL NAME="News & Commentary|Markets" ID="
        1025" />
    </CHANNEL>
</TARGETING>
<CODE-LIST NSURI="urn:marketwatch-com:Channel" CATEGORY="
Target">
    <CODE>Markets</CODE>
</CODE-LIST>
<CODE-LIST NSURI="urn:marketwatch-com:Subchannel" CATEGORY="
Target">
    <CODE>Markets|U.S. &Canada</CODE>
</CODE-LIST>
<PACKAGES>  </PACKAGES>
<PLACEMENT-LIST>
    <PLACEMENT>cbs</PLACEMENT>
</PLACEMENT-LIST>
<INDUSTRY-LIST>
    <INDUSTRY>Banks</INDUSTRY>
    <INDUSTRY>Financial Services</INDUSTRY>
</INDUSTRY-LIST>
<ISSUE-LIST>
    <ISSUE>Newsmakers</ISSUE>
    <ISSUE>Earnings</ISSUE>
    <ISSUE>Markets/Exchanges</ISSUE>
    <ISSUE>Analyst</ISSUE>
    <ISSUE>Company Announcements</ISSUE>
```

<ISSUE>Market News</ISSUE>
</ISSUE-LIST>
<FUNDTYPE-LIST> </FUNDTYPE-LIST>
<REGION-LIST>
<REGION>Washington DC</REGION>
<REGION>US</REGION>
</REGION-LIST>
<PERSONALFINANCE-LIST> </PERSONALFINANCE-LIST>
<RELATED-LIST> </RELATED-LIST>
<ABSTRACT>
<PARAGRAPH TYPE="NORMAL">The lenders efforts to get financing are alleviating fears for the time being that the business lender will file for bankruptcy protection.</PARAGRAPH>
</ABSTRACT>
<SUMMARY ID="">
<SUMMARY-BODY>
<HEADLINE>Financials get lift from CIT emergency rescue</HEADLINE>
<PARAGRAPH TYPE="NORMAL">The lenders efforts to get financing are alleviating fears for the time being that the business lender will file for bankruptcy protection.</PARAGRAPH>
</SUMMARY-BODY>
</SUMMARY>
<CREATED TIMESTAMP="40014.5194444444" ET-FORMAT="12:28 PM ET Jul 20, 2009" CTIME-FORMAT="1248114480">7/20/2009 12:28:00 PM
</CREATED>
<LASTUPDATED TIMESTAMP="40014.5194444444" ET-FORMAT="12:28 PM ET Jul 20, 2009" CTIME-FORMAT=" 1248114480">
7/20/2009 12:28:00 PM
</LASTUPDATED>
<AUTHOR GUID="{B1EC6A08-BFC2-0642-93E2-6DB9DCB28E7A}" ORGANIZATION="MarketWatch" ADDRESS="
mailto:rwilliams@marketwatch.com" BIO="Ryan Williams is a reporter for MarketWatch, based in New York.">Ryan Williams</AUTHOR>
<GUID>{D897B551-84C4-49A1-BEEB-058D1BDA2DBC}</GUID>
<LOCATION>
<STORAGE GUID="{D897B551-84C4-49A1-BEEB-058D1BDA2DBC}
" SLUG="sector_financial" />
</LOCATION>
<SYMBOLS>
<SYMBOL COUNTRY="US" TICKER="XLF">XLF</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="CIT">CIT</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="STT">STT</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="WFC">WFC</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="AXP">AXP</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="MTB">MTB</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="AMTD">AMTD</SYMBOL>
<SYMBOL COUNTRY="US" TICKER="SCHW">SCHW</SYMBOL>
</SYMBOLS>
</NEWSITEM>
</ITEM>

```
        </LIST>
        </XML_MULTISYMBOL_NEWS>
    </amtd>
```

24.5 QuoteNews Errors

Possible Errors

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <XML_MULTISYMBOL_NEWS>
        <NOTFOUND>
            <SYMBOL>111</SYMBOL>
        </NOTFOUND>
    </XML_MULTISYMBOL_NEWS>
</amtd>
```

If user has not entered the symbol, they would get the following message:

```
<?xml version="1.0" ?>
<amtd>
   <result>OK</result>
   <XML_MULTISYMBOL_NEWS>
       <ERROR>Please pass a symbol(s).</ERROR>
   </XML_MULTISYMBOL_NEWS>
</amtd>
```

If MarketWatch feed is down, user would get following error message:

"System Unavailable"

25 EquityTrade

Service provides the ability to trade equities. EquityTrade is used for simple, single leg orders. Conditional orders are handled by the ConditionalEquityTrade service

25.1 EquityTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/EquityTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order being sent.

25.2 EquityTrade Parameters

Each EquityTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

EquityTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

NOTE: The **accountid** is specified inside the orderstring, not as a separate parameter

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
action	Order action	YES	**sell, buy, sellshort, buytocover**
actprice	The stop price for stop market or stop limit orders	NO	
displaysize	Level II display size if routing is INET otherwise null.	NO	Integer value 100 or higher, in increments of 100
expire	Time in force	YES	**day, moc, day_ext, gtc, gtc_ext, am, pm**
exday	Two digit expiration day, only specified if expire is set to gtc otherwise null	NO	
exmonth	Two digit expiration month, only specified if expire is set to gtc otherwise null.	NO	
exyear	Two digit expiration year, only specified if expire is set to gtc otherwise null.	NO	
ordtype	Case sensitive value for order type	YES	**market, limit, stop_market, stop_limit, tstoppercent, tstopdollar**

price	Limit price for limit or stop limit orders otherwise null	NO	
quantity	The number of shares being bought or sold in the transaction	YES	
routing	User specified order routing destination. Default is auto	NO	**auto, inet, ecn_arca**
spinstructions	Special Instructions for the order execution. Set to none if not specified	NO	**none, fok, aon, dnr, aon_dnr**
symbol	Symbol being bought or sold. UPPER CASE	YES	
tsparam	Trailing Stop parameter in dollars or percent depending on order type otherwise null.	NO	In case of percent, 5 means 5 percent

25.3 EquityTrade Validation Rules

action, symbol, ordtype, quantity, accountid, and **expire** are required parameters

if ordtype=market
- expire must be day or moc
- spinstructions must none
- price must be empty or 0
- actprice must be empty or 0
- routing must be auto

if ordtype=limit
- price must be a valid value
- actprice must be empty or 0
- expire must be day, day_ext, gtc, gtc_ext, am, or pm
  - o if expire=day, spinstructions must be non, aon, fok
  - o if expire=day_ext, gtc_ext, am, or pm, spinstructions must be none
  - o if expire=gtc, spinstructions must be none, aon, dnr, aon_dnr

if ordtype=stop_limit
- price must be a valid value
- actprice must be a valid value
- expire must be day or gtc
  - o if expire=day, spinstructions must be none

if ordtype=stop_market
- price must be empty or 0
- actprice must be a valid value
- expire must be day or gtc
-

if ordtype=tstoppercent
- price must be empty or 0
- actprice must be empty or 0
- tsparam must be a whole number between 1 and 99

- expire must be day or gtc

if ordtype=tstopdollar
- price must be empty or 0
- actprice must be empty or 0
- tsparam must be greater than 0 and a decimal value (#.##)
- expire must be day or gtc

if exday, exmonth, and exyear are not empty
- all fields must contain valid date values
- expire must be gtc or gtc_ext
- the date cannot be before the current date
- the date cannot be after six months from the current date

if expire=gtc or gtc_ext
- a valid expiration date (exday, exmont, and exyear) is required

if account does not have margin approval
- action must not be buytocover or sellshort

if displaysize is not empty (display size is not a required field and is only related to INET orders)
- display size must be a valid quantity
- the minimum display size is 100
- the maximum display size is the quantity

if spinstructions is aon or fok
- , then expire must be day or gtc

if spinstructions is dnr
- , then expire must be gtc

if routing is not empty (auto is the default value)
- routing must be auto, inet, or ecn_arca
- if routing is not auto, then the account must be enabled for direct routing

if routing=inet or ecn_arca
- the maximum quantity is 250000
- the ordtype must be limit
- the expire must be day or day_ext
- the spinstructions must be none

if the equity being traded is an OTCBB
- action must be buy, buytocover, or sell
- if action is buy or buytocover
    - ordertype must be limit or stop_limit
- if action is sell
    - ordertype must be market, limit, stop_market, or stop_limit

25.4 EquityTrade Response

The response to the EquityTrade request will be in XML format

General Structure of XML Response

```xml
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <trade-type></trade-type>
            <requested-destination>
                <routing-mode></routing-mode>
                <market-maker-id></market-maker-id>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <limit-price></limit-price>
            <stop-price></stop-price>
            <time-in-force>
                <session></session>
            </time-in-force>
        </order>
    </order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.

order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the security
symbol	String	Contains symbol. For example, **AMTD**.
symbol-with-type-prefix	String	
description	String	
asset-type	String	Type of asset the symbol represents. It is a one character code: **E** — Equity
exchange	String	
quantity	Double	Number of units ordered. Field may have up to three digits to the right of the decimal point.
order-id	String	A unique number identifying the order. Generated by the servers
action	String	Code indicating the order action requested. Values returned are: **B** — Buy **S** — Sell **SS** — Short Sell **BC** — Buy to Cover **E** — Exchange **EX** — Exercise Option
trade-type	Integer	Valid values are: **1** — Normal Market **2** — External Hour Market **4** — German Market **8** — AM Session **16** — Seamless Session
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	

order-type	String	Values returned are: **M** — Market **L** — Limit **S** — Stop **X** — Stop Limit **T** — Trailing Stop **EX** — Exercise Option
limit-price	Double	
stop-price	Double	
special-conditions	String	
time-in-force	Complex	
session	String	
expiration	String	

25.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/EquityTrade?source=<**#sourceID#**>&orderstring=action%3Dbuy%7Equantity%3D400%7Esymbol%3DDELL%7Eordtype%3DLimit%7Eprice%3D27.49%7Eactprice%3D%7Etsparam%3D%7Eexpire%3Dday%7Espinstructions%3Dnone%7Erouting%3Dauto%7Edisplaysize%3D%7Eexmonth%3D%7Eexday%3D%7Eexyear%3D%7Eaccountid%3D123456789

(Order String is URL Encoded version of action=buy~quantity=400~symbol=DELL~ordtype=Limit~price=27.49~actprice=~tsparam=~expire=day~spinstructions=none~routing=auto~displaysize=~exmonth=~exday=~exyear=~accountid=123456789)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        action=buy~quantity=400~symbol=DELL~ordtype=Limit~price=27.49~actprice=~tsparam=~expire=day~spinstructions=none~routing=auto~displaysize=~exmonth=~exday=~exyear=~accountid=123456789

        </orderstring>
        <error></error>
        <order>
            <account-id>123456789</account-id>
            <security>
                <symbol>DELL</symbol>
                <symbol-with-type-prefix>DELL</symbol-with-type-prefix>
```

```xml
                <description>DELL INC COM Status Alert: Delinquent</description>
                <asset-type>E</asset-type>
                <exchange>NASDAQ</exchange>
            </security>
            <quantity>400</quantity>
            <order-id>2148878993</order-id>
            <action>B</action>
            <trade-type>1</trade-type>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <market-maker-id></market-maker-id>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</order-type>
            <limit-price>27.49</limit-price>
            <stop-price></stop-price>
            <time-in-force>
                <session>D</session>
            </time-in-force>
        </order>
    </order-wrapper>
</amtd>
```

25.6 EquityTrade Errors

Possible Errors

If the Account ID specified is not valid you will get

```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        action=sell~quantity=100~symbol=DELL~ordtype=Limit~price=27.4~ac
        tprice=~tsparam=~expire=day~spinstructions=none~routing=auto~dis
        playsize=~exmonth=~exday=~exyear=~accountid=123456789</
        orderstring>
```

<error>**Please check your portfolio and/or open orders. Your sell order is greater than the quantity held in your account.**</error>
</order-wrapper>
</amtd>

26 OptionTrade

Service provides the ability to trade options. OptionTrade is used for simple, single leg orders. Complex option orders and Conditional orders will be implemented separately

26.1 OptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/OptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order being sent.

26.2 OptionTrade Parameters

Each OptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

OptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application

orderstring	consists of variable =value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

NOTE: The **accountid** is specified inside the orderstring, not as a separate parameter

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
action	Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
actprice	The stop price for stop market or stop limit orders	NO	
expire	Time in force	YES	**day, gtc**

exday	Two digit expiration day, only specified if expire is set to gtc otherwise null	NO	
exmonth	Two digit expiration month, only specified if expire is set to gtc otherwise null.	NO	
exyear	Two digit expiration year, only specified if expire is set to gtc otherwise null.	NO	
ordtype	Case sensitive value for order type	YES	**market, limit, stop_market, stop_limit**

price	Limit price for limit or stop limit orders otherwise null	NO	
quantity	The number of shares being bought or sold in the transaction	YES	
routing	User specified order routing destination. Default is auto	NO	**auto, isex, cboe, amex, phlx, pacx, bosx**
spinstructions	Special Instructions for the order execution. Set to none if not specified	NO	**none, fok, aon**
symbol	Symbol being bought or sold. UPPER CASE	YES	

26.3 OptionTrade Validation Rules

action, symbol, ordtype, quantity, accountid, and **expire** are required parameters

if ordtype=market
- expire must be day

- spinstructions must be none
- price must be empty or 0
- actprice must be empty or 0

if ordtype=limit
- price must be valid
- actprice must be empty or 0
- expire must be day or gtc
  - if expire=gtc, spinstructions must be none or aon
  - if expire=day, spinstructions must be none, aon, fok

if ordertype=stop_market
- price must be empty or 0
- actprice must be valid
- expire must be day or gtc
- spinstructions must be none or aon

if ordtype=stop_limit
- price must be valid
- actprice must be valid
- expire must be day or gtc
- spinstructions must be none or aon

if routing is not empty (auto is the default value)
- routing must be auto, isex, cboe, amex, phlx, pacx, or bosx
- if routing is not auto, then the account must be enabled for option direct routing

if exday, exmonth, and exyear are not empty
- all fields must contain valid date values
- expire must be gtc or gtc_ext
- the date cannot be before the current date
- the date cannot be after the last day of the following month

if expire=gtc or gtc_ext
- a valid expiration date (exday, exmont, and exyear) is required

26.4 OptionTrade Response

The response to the OptionTrade request will be in XML format. You will notice that its almost identical to the EquityTrade Response. In fact, the only 2 fields that are added are the <put-call> and <open-close>. Some of the values returned may be slightly different.

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
```

```
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <trade-type></trade-type>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <limit-price></limit-price>
            <stop-price></stop-price>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
        </order>
    </order-wrapper>
</amtd>
```


Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order

accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the security
symbol	String	Contains symbol. For example, **KQM_082210P6.30**
symbol-with-type-prefix	String	
description	String	
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	
quantity	Double	Number of units ordered. Field may have up to three digits to the right of the decimal point.
order-id	String	Unique order number assigned by the Trading Platform
action	String	Code indicating the order action requested. Values returned are: **B** — Buy **S** — Sell **E** — Exchange **EX** — Exercise Option (NOTE: Look at the **OPEN-CLOSE** tag too)
trade-type	Integer	Valid values are: **1** — Normal Market **2** — External Hour Market **4** — German Market **8** — AM Session **16** — Seamless Session
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **M** — Market **L** — Limit **S** — Stop **X** — Stop Limit **T** — Trailing Stop **EX** — Exercise Option
limit-price	Double	
stop-price	Double	

special-conditions	String	
time-in-force	Complex	
session	String	
expiration	String	
put-call	String	
open-close	String	

26.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/OptionTrade?source=<**#sourceID#**>&orderstring=action%3Dselltoclose%7Equantity%3D10%7Esymbol%3DKQM_082210P6.30%7Eordtype%3DLimit%7Eprice%3D6.3%7Eactprice%3D%7Eexpire%3Dday%7Espinstructions%3Dnone%7Erouting%3Dauto%7Eexmonth%3D%7Eexday%3D%7Eexyear%3D%7Eaccountid%3D123456789

(Order String is URL Encoded version of action=selltoclose~quantity=10~symbol=KQM_082210P6.30 ~ordtype=Limit~price=6.3~actprice=~expire=day~spinstructions=none~routing=auto ~exmonth=~exday=~exyear~accountid=123456789)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        action=selltoclose~quantity=10~symbol=KQM_082210P6.30~ordtype=Limit~price=6.3~actprice=~expire=day~spinstructions=none~routing=auto~exmonth=~exday=~exyear=~tabid=0~accountid=123456789
        </orderstring>
        <error></error>
        <order>
            <account-id>123456789</account-id>
            <security>
                <symbol>KQM_082210P6.30</symbol>
                <symbol-with-type-prefix>KQM_082210P6.30</
                symbol-with-type-prefix>
                <description>KQM Feb 20 2010 6.30 Put</description>
                <asset-type>O</asset-type>
                <exchange>OPRA</exchange>
            </security>
            <quantity>10</quantity>
            <order-id>2148878993</order-id>
            <action>S</action>
```

```
            <trade-type>1</trade-type>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <market-maker-id>Auto</market-maker-id>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</order-type>
            <limit-price>6.30</limit-price>
            <stop-price></stop-price>
            <time-in-force>
                <session>D</session>
            </time-in-force>
            <put-call>P</put-call>
            <open-close>C</open-close>
        </order>
    </order-wrapper>
</amtd>
```

26.6 OptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        action=sell~quantity=5~symbol=KQM_082210P6.30~ordtype=Limit~pri
        ce=3.4~actprice=~tsparam=~expire=day~spinstructions=none~routing
        =auto~exmonth=~exday=~exyear=~accountid=123456789</orderstring
        >
        <error>Please check your portfolio and/or open orders. Your sell order is
        greater than the quantity held in your account.</error>
    </order-wrapper>
</amtd>
```

27 EditOrder

Service provides the ability to modify unfilled or partially filled orders. Only certain fields may be modified for an existing order. If non-modifiable fields require modification the order must be cancelled and replaced.

NOTE: If editing an order that was just edited, make sure that the original order's status is "Canceled" before submitting a new EditOrder command

27.1 EditOrder Request

Request URL

https://apis.tdameritrade.com/apps/100/EditOrder?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order being sent.

27.2 EditOrder Parameters

Each EditOrder request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent.

Note: This operation can now be used on Equity, Option and Complex Option orders.

Edit Order Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES

clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
orderid	The Order ID of an existing open order	YES	One of the existing open orders that is in one of the logged in accounts. The orderid would be either returned from EquityTrade or OptionTrade, or retrievied from the OrderStatus Response When editing a complex option order use the orderid of the first leg of the order.
actprice	The stop price for stop market or stop limit orders	NO	
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
expire	Time in force	YES	**day, day_ext, gtc, gtc_ext, am, pm**
exday	Two digit expiration day, only specified if expire is set to gtc otherwise null	NO	
exmonth	Two digit expiration month, only specified if expire is set to gtc otherwise null.	NO	
exyear	Two digit expiration year, only specified if expire is set to gtc otherwise null.	NO	
ordtype	Case sensitive value for order type	YES	**market, limit, stop_market, stop_limit, tstoppercent, tstopdollar, net_credit, net_debit**
price	Limit price for limit, stop limit, net_credit, or net_debit orders otherwise null	NO	
quantity	The number of shares being bought or sold in the transaction	YES	In cases where the original order was partially filled, the specified quantity needs to be the REMAINING quantity

27.3 EditOrder Validation Rules

orderId, accountid, quantity, ordtype, and **expire** are required parameters

For the order to be eligible for editing the following conditions must be met:
- originalOrder assetType = Option or Equity
- originalOrder status=Open, Pending, or Partially Filled

if ordtype=limit or stop_limit or stop_market

- price must be a valid value
- actprice must be empty or 0

if ordtype=stop_limit
- price must be a valid value
- actprice must be a valid value

if ordtype=stop_market
- price must be empty or 0
- actprice must be a valid value

if ordtype=tstoppercent or ordtype=tstopdollar
- not allowed to be edited

if exday, exmonth, and exyear are not empty
- all fields must contain valid date values
- expire must be gtc or gtc_ext
- the date cannot be before the current date
- the date cannot be after the last day of the following month

if expire=gtc or gtc_ext
- a valid expiration date (exday, exmont, and exyear) is required

if expire value cannot switch session.
- If the original order is day or gtc, edit can be day or gtc.
- If the original order is day+ext or gtc+ext, edit can be day+ext or gtc+ext.
- If the original order is AM or PM, the expiration on the EditOrder must be the same as the original order.

27.4 EditOrder Response

The response to the EditOrder request will be in XML format, and will be exactly the same as the response to EquityTrade

Note: You can not edit Complex Option Order. You just need to cancel the order.

27.5 EditOrder Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
```

```
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        orderid=200635201~ordtype=limit~quantity=100~price=21~actprice=
        ~expire=day~exmonth=~exday=~exyear=~accountid=123456789</
        orderstring>
        <error>When your account equity is less than $2,000, only your cash
        balance is available for purchases.</error>
    </order-wrapper>
</amtd>
```

If the order was already filled, you will get the following response

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        action=sell~quantity=100~symbol=DELL~ordtype=Limit~price=27.4~ac
        tprice=~tsparam=~expire=day~spinstructions=none~routing=auto~dis
        playsize=~exmonth=~exday=~exyear=~accountid=123456789</
        orderstring>
        <error>Order cannot be replaced.</error>
    </order-wrapper>
</amtd>
```

28 OrderCancel

Service provides the ability to cancel one or more open orders or the balance of partially filled orders.

NOTE: If canceling an order that was just edited, make sure that the original order's status is "Canceled" before submitting the OrderCancel command

28.1 OrderCancel Request

Request URL

https://apis.tdameritrade.com/apps/100/OrderCancel?source=<**#SourceID#**>&orderid=<**#order-id#**>&orderid=<**#order-id#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderid parameters for existing open orders

28.2 OrderCancel Parameters

The source assigned by TD Ameritrade is a required parameter, as well as one or more orderid paramethers

OrderCancel Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which the data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account
orderid	The Order ID of an existing open order	YES	One of the existing open orders that is in one of the logged in accounts. The orderid would be either returned from EquityTrade or OptionTrade, or retrievied from the OrderStatus Response If the orderid is not in the associated account, then the accountid must be specified

28.3 OrderCancel Response

The response to the OrderCancel request will be in XML format.

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <cancel-order-messages>
    <account-id></account-id>
    <order>
        <order-id></order-id>
        <message></message>
```

```
        <error><error>
    </order>
    </cancel-order-messages>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
cancel-order-messages	Complex	Container for the canceled order information that was submitted
account-id	String	The account ID in which the order was placed
order	String	Container for the order information
order-id	String	Unique ID identifying the order being canceled
message	String	Status message to be displayed to the user if the OrderCancel was successfully submitted: **NOTE:** This does NOT mean the order was successfully canceled. Just successfully submitted
error	String	Error message if an error occurred when submitting the OrderCancel request. Null otherwise

28.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/OrderCancel?source=<**#sourceID#**>&orderid=2148881399&orderid=2147949999&orderid=2147949948

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <cancel-order-messages>
    <account-id>123456789</account-id>
    <order>
        <order-id>2148881399</order-id>
        <message>The order was submitted for cancellation. Please check its
        status.</message>
        <error></error>
    </order>
    <order>
        <order-id>2147949999</order-id>
        <message></message>
        <error>The order could not found. Please check its status.</error>
```

```
    </order>
    </cancel-order-messages>
</amtd>
```

Note: If you want to cancel Complex Option Order, you need to pass Primary Legged Order ID.

28.5 OrderCancel Errors

Possible Errors

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

if the order was closed or could not be found, you will get the following errors:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <cancel-order-messages>
    <account-id>123456789</account-id>
    <order>
        <order-id>2147949999</order-id>
        <message></message>
        <error>The order could not found. Please check its status.</error>
    </order>
    </cancel-order-messages>
</amtd>
```

If no orders were specified, then the following error will be returned:

```
<?xml version="1.0" ?>

<amtd>
    <result>FAIL</result>
    <error>No Orders have been selected to be canceled.</error>
</amtd>
```

29 ConditionalEquityTrade

Service provides the ability to trade equities. Unlike EquityTrade, this one is used to enter multi-leg conditional orders - where the execution of one order will trigger one or more other orders or cancellation of one or more orders

29.1 ConditionalEquityTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/ConditionalEquityTrade?source=<**#SourceID#**

>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order group being sent.

29.2 ConditionalEquityTrade Parameters

Each ConditionalEquityTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

EquityTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client. NOTE: only one clientorderid is specified for the entire conditional order, not one for each leg
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
ordticket	The type of conditional order	YES	**oca,ota,ott,otoca,otota** **NOTE:** More conditional order types may be added in the future
totlegs	The number of legs of the conditional order	YES	2 or 3.
The rest of the parameters are the same as for EquityTrade, just followed by a Leg #. For example, instead of actionprice, it would be actionprice1, actionprice2			

29.3 ConditionalEquityTrade Validation Rules

All existing EquityTrade Validation Rules apply. Additional rules specific to Conditional Equity Trade orders are:

if ordticket=oca
- expire must be the same for both legs
- ordtype cannot be Market for either leg
- symbol + action + ordertype cannot be the exact same for both legs

- if symbol is the same for both legs and either leg has an ordtype=limit
  - o other leg ordertype=stop_limit, stop_market, tstoppercent, tstopdollar

- if symbol and action on both legs are the same and action is buy and if one leg is limit and other leg is stop limit or stop market
  - o stop price must be higher than the limit price

- if symbol and action on both legs are the same and action is sell and if one leg is limit and other leg is stop limit or stop market
  - o stop price must be lower than the limit price

29.4 ConditionalEquityTrade Response

The response to the ConditionalEquityTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from EquityTrade Response

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <enhancled-order>
            <eot-type></eot-type>
            <order>
                <account-id></account-id>
                <security>
                    <symbol></symbol>
                    <symbol-with-type-prefix></symbol-with-type-prefix>
                    <description></description>
                    <asset-type></asset-type>
                    <exchange></exchange>
                </security>
                <quantity></quantity>
                <order-id></order-id>
                <action></action>
```

```
                <trade-type></trade-type>
                <requested-destination>
                    <routing-mode></routing-mode>
                    <market-maker-id></market-maker-id>
                    <response-description></response-description>
                </requested-destination>
                <routing-display-size></routing-display-size>
                <order-type></order-type>
                <limit-price></limit-price>
                <stop-price></stop-price>
                <time-in-force>
                    <session></session>
                </time-in-force>
            </order>
            <order>
                ....
            </order>
            ....
        </enhancled-order>
    </order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
enhanced-order	Complex	Container for detailed information about the enhanced order
eot-type	String	The type of conditional order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the security
symbol	String	Contains symbol. For example, **AMTD**.
symbol-with-type-prefix	String	
description	String	

asset-type	String	Type of asset the symbol represents. It is a one character code: **E** — Equity
exchange	String	
quantity	Double	Number of units ordered. Field may have up to three digits to the right of the decimal point.
order-id	String	Unique order number assigned by the Trading Platform
action	String	Code indicating the order action requested. Values returned are: **B** — Buy **S** — Sell **SS** — Short Sell **BC** — Buy to Cover **E** — Exchange **EX** — Exercise Option
trade-type	Integer	Valid values are: **1** — Normal Market **2** — External Hour Market **4** — German Market **8** — AM Session **16** — Seamless Session
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **M** — Market **L** — Limit **S** — Stop **X** — Stop Limit **T** — Trailing Stop **EX** — Exercise Option
limit-price	Double	
stop-price	Double	
special-conditions	String	
time-in-force	Complex	
session	String	
expiration	String	

29.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/ConditionalEquityTrade?orderstring=ordticket%3Dota%7Etotlegs%3D2%7Eaction1%3Dbuy%7Equantity1%3D2%7Esymbol1%3Dibm%7Eordtype1%3Dlimit%7Eprice1%3D2%7Eactprice1%3D%7Etsparam1%3D%7Eexpire1%3Dday%7Espinstructions1%3Dnone%7Erouting1%3Dauto%7Edisplaysize1%3D%7Eexmonth1%3D%7Eexday1%3D%7Eexyear1%3D%7Eaction2%3Dbuy%7Equantity2%3D4%7Esymbol2%3Ddis%7Eordtype2%3Dlimit%7Eprice2%3D25%7Eactprice2%3D%7Etsparam2%3D%7Eexpire2%3Dday%7Espinstructions2%3Dnone%7Erouting2%3Dauto%7Eexmonth2%3D%7Eexday2%3D%7Eexyear2%3D%7Eaccountid%3D123456789%7Eclientorderid%3D1

(Order String is URL Encoded version of ordticket=ota~totlegs=2~action1=buy~quantity1=2~symbol1=ibm~ordtype1=limit~price1=2~actprice1=~tsparam1=~expire1=day~spinstructions1=none~routing1=auto~displaysize1=~exmonth1=~exday1=~exyear1=~action2=buy~quantity2=4~symbol2=dis~ordtype2=limit~price2=25~actprice2=~tsparam2=~expire2=day~spinstructions2=none~routing2=auto~exmonth2=~exday2=~exyear2=~accountid=123456789~clientorderid=1)

Response:


```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <client-order-id>1</client-order-id>
        <orderstring>
        ordticket=ota~totlegs=2~action1=buy~quantity1=2~symbol1=ibm~ordtype1=limit~price1=2~actprice1=~tsparam1=~expire1=day~spinstructions1=none~routing1=auto~displaysize1=~exmonth1=~exday1=~exyear1=~action2=buy~quantity2=4~symbol2=dis~ordtype2=limit~price2=25~actprice2=~tsparam2=~expire2=day~spinstructions2=none~routing2=auto~exmonth2=~exday2=~exyear2=~accountid=123456789~clientorderid=1</orderstring>
        <error></error>
        <enhanced-order>
            <eot-type>OTA</eot-type>
            <order>
                <security>
                    <symbol>IBM</symbol>
                    <symbol-with-type-prefix>IBM</symbol-with-type-prefix>
                    <description>INTERNATIONAL BUSINESS MACHS COM</description>
                    <asset-type>E</asset-type>
                    <exchange>NYSE</exchange>
                </security>
                <quantity>2</quantity>
                <order-id>2148123456</order-id>
                <action>B</action>
                <trade-type>1</trade-type>
                <requested-destination>
                    <routing-mode>SMART</routing-mode>
```

```
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</ order-type >
            <limit-price>2.00</limit-price>
            <stop-price>  </stop-price>
            <stop-limit-price>2.00</stop-limit-price>
            <time-in-force>
                <session>D</session>
            </time-in-force>
        </order>
        <order>
            <security>
            <symbol>DIS</symbol>
                <symbol-with-type-prefix>DIS</symbol-with-type-prefix>
                <description>DISNEY WALT CO COM DISNEY</description>
                <asset-type>E</asset-type>
                <exchange>NYSE</exchange>
            </security>
            <quantity>4</quantity>
            <order-id>2148123457</order-id>
            <action>B</action>
            <trade-type>1</trade-type>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>L</ order-type >
            <limit-price>25.00</limit-price>
            <stop-price>  </stop-price>
            <stop-limit-price>25.00</stop-limit-price>
            <time-in-force>
                <session>D</session>
            </time-in-force>
        </order>
    </enhanced-order>
  </order-wrapper>
</amtd>
```


29.6 ConditionalEquityTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
```

```
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        ordticket=ota~totlegs=2~action1=buy~quantity1=2~symbol1=ibm~ord
        type1=limit~price1=2~actprice1=~tsparam1=~expire1=day~spinstructi
        ons1=none~routing1=auto~displaysize1=~exmonth1=~exday1=~exyea
        r1=~action2=buy~quantity2=4~symbol2=dis~ordtype2=limit~price2=2
        5~actprice2=~tsparam2=~expire2=day~spinstructions2=none~routing
        2=auto~exmonth2=~exday2=~exyear2=~accountid=
        123456789~clientorderid=1</orderstring>
        <error>Please check your portfolio and/or open orders. Your sell order is
        greater than the quantity held in your account.</error>
    </order-wrapper>
</amtd>
```

30 ConditionalOptionTrade

Service provides the ability to trade options. Unlike OptionTrade, this one is used to enter multi-leg conditional orders - where the execution of one order will trigger one or more other orders or cancellation of one or more orders.

NOTE: This should not be confused with complex option orders (Vertical Spreads, Buy/Write, Butterflies, etc). Those are implemented separately

30.1 ConditionalOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/ConditionalOptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each defining one order group being sent.

30.2 ConditionalOptionTrade Parameters

The Parameters for the ConditionalOptionTrade service are the same as for ConditionalEquityTrade

30.3 ConditionalOptionTrade Validation Rules

All existing OptionTrade Validation Rules apply. Additional rules specific to Conditional Option Trade orders are:

if ordticket=oca

- expire must be the same for both legs
- ordtype cannot be Market for either leg
- symbol + action + ordertype cannot be the exact same for both legs

- if symbol is the same for both legs and either leg has an ordtype=limit
  - o other leg ordertype=stop_limit, stop_market

- if symbol and action on both legs are the same and action is buy and if one leg is limit and other leg is stop limit or stop market
  - o stop price must be higher than the limit price

- if symbol and action on both legs are the same and action is sell and if one leg is limit and other leg is stop limit or stop market
  - o stop price must be lower than the limit price

30.4 ConditionalOptionTrade Response

The response to the ConditionalOptionTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from OptionTrade Response

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <enhancled-order>
            <eot-type></eot-type>
            <order>
                <account-id></account-id>
                <security>
                    <symbol></symbol>
                    <symbol-with-type-prefix></symbol-with-type-prefix>
                    <description></description>
                    <asset-type></asset-type>
```

```
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <trade-type></trade-type>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <limit-price></limit-price>
            <stop-price></stop-price>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
        </order>
        <order>
            ....
        </order>
        ....
    </enhancled-order>
  </order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
enhanced-order	Complex	Container for detailed information about the enhanced order
eot-type	String	The type of conditional order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed

security	Complex	Container for detailed information about the security
symbol	String	Contains symbol. For example, **GOOG_071010C520**.
symbol-with-type-prefix	String	Contains symbol. For example, **GOOG_071010C520**.
description	String	Contains the option symbol description. For example: **GOOG Jul 10 2010 520.0 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** - Option
exchange	String	
quantity	Double	Number of units ordered. Field may have up to three digits to the right of the decimal point.
order-id	String	Unique order number assigned by the Trading Platform
action	String	Code indicating the order action requested. Values returned are: **B** — Buy **S** — Sell **SS** — Short Sell **BC** — Buy to Cover **E** — Exchange **EX** — Exercise Option
trade-type	Integer	Valid values are: **1** — Normal Market **2** — External Hour Market **4** — German Market **8** — AM Session **16** — Seamless Session
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **M** — Market **L** — Limit **S** — Stop **X** — Stop Limit **T** — Trailing Stop **EX** — Exercise Option
limit-price	Double	
stop-price	Double	

special-conditions	String	
time-in-force	Complex	
session	String	
expiration	String	

30.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/ConditionalOptionTrade?orderstring=ordticket%3Dota%7Etotlegs%3D2%7Eaction1%3Dbuytoopen%7Equantity1%3D2%7Esymbol1%3DGOOG_071010C520%7Eordtype1%3Dlimit%7Eprice1%3D10%7Eactprice1%7Eexpire1%3Dday%7Espinstructions1%3Dnone%7Erouting1%3Dauto%7Eexmonth1%3D%7Eexday1%3D%7Eexyear1%3D%7Eaction2%3Dbuytoopen%7Equantity2%3D4%7Esymbol2%3DGOOG_071010C520%7Eordtype2%3Dlimit%7Eprice2%3D8%7Eactprice2%3D%7Eexpire2%3Dday%7Espinstructions2%3Dnone%7Erouting2%3Dauto%7Eexmonth2%3D%7Eexday2%3D%7Eexyear2%3D%7Eaccountid%3D123456789%7Eclientorderid%3D1

(Order String is URL Encoded version of ordticket=ota~totlegs=2~action1=buytoopen~quantity1=2~symbol1=GOOG_071010C520~ordtype1=limit~price1=10~actprice1~expire1=day~spinstructions1=none~routing1=auto~exmonth1=~exday1=~exyear1=~action2=buytoopen~quantity2=4~symbol2=GOOG_071010C520~ordtype2=limit~price2=8~actprice2=~expire2=day~spinstructions2=none~routing2=auto~exmonth2=~exday2=~exyear2=~accountid=123456789~clientorderid=1)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <client-order-id>1</client-order-id>
        <orderstring>
        ordticket=ota~totlegs=2~action1=buytoopen~quantity1=2~symbol1=GOOG_071010C520~ordtype1=limit~price1=10~actprice1~expire1=day~spinstructions1=none~routing1=auto~exmonth1=~exday1=~exyear1=~action2=buytoopen~quantity2=4~symbol2=GOOG_071010C520~ordtype2=limit~price2=8~actprice2=~expire2=day~spinstructions2=none~routing2=auto~exmonth2=~exday2=~exyear2=~accountid=123456789~clientorderid=1</orderstring>
        <error></error>
        <enhanced-order>
            <eot-type>OTA</eot-type>
            <order>
                <security>
                    <symbol>GOOG_071010C520</symbol>
                    <symbol-with-type-prefix>GOOG_071010C520</
                    symbol-with-type-prefix>
                    <description>GOOG Jul 10 2010 520.0 Call</description>
                    <asset-type>O</asset-type>
                    <exchange>OPRA</exchange>
```

```xml
                </security>
                <quantity>2</quantity>
                <order-id>2148883013</order-id>
                <action>B</action>
                <trade-type>1</trade-type>
                <requested-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
                </requested-destination>
                <routing-display-size>0</routing-display-size>
                <order-type>L</ order-type >
                <limit-price>10.00</limit-price>
                <stop-price>  </stop-price>
                <stop-limit-price>10.00</stop-limit-price>
                <time-in-force>
                    <session>D</session>
                </time-in-force>
                <put-call>C</put-call>
                <open-close>O</open-close>
            </order>
            <order>
                <security>
                    <symbol>GOOG_071010C520</symbol>
                    <symbol-with-type-prefix>GOOG_071010C520</
                    symbol-with-type-prefix>
                    <description>GOOG Jul 10 2010 520.0 Call</description>
                    <asset-type>O</asset-type>
                    <exchange>OPRA</exchange>
                </security>
                <quantity>4</quantity>
                <order-id>2148883014</order-id>
                <action>B</action>
                <trade-type>1</trade-type>
                <requested-destination>
                    <routing-mode>SMART</routing-mode>
                    <option-exchange>Auto</option-exchange>
                    <response-description>AutoRoute</response-description>
                </requested-destination>
                <routing-display-size>0</routing-display-size>
                <order-type>L</ order-type >
                <limit-price>8.00</limit-price>
                <stop-price>  </stop-price>
                <stop-limit-price>8.00</stop-limit-price>
                <time-in-force>
                    <session>D</session>
                </time-in-force>
                <put-call>C</put-call>
                <open-close>O</open-close>
            </order>
        </enhanced-order>
    </order-wrapper>
</amtd>
```

30.6 ConditionalOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        ordticket=ota~totlegs=2~action1=buytoopen~quantity1=2~symbol1=G
        OOG_071010C520~ordtype1=limit~price1=10~actprice1~expire1=day~
        spinstructions1=none~routing1=auto~exmonth1=~exday1=~exyear1=
        ~action2=buytoopen~quantity2=4~symbol2=GOOG_071010C520~ordty
        pe2=limit~price2=8~actprice2=~expire2=day~spinstructions2=none~r
        outing2=auto~exmonth2=~exday2=~exyear2=~accountid=123456789
        ~clientorderid=1</orderstring>
        <error>Please check your portfolio and/or open orders. Your sell order is
        greater than the quantity held in your account.</error>
    </order-wrapper>
</amtd>
```

31 BuyWriteOptionTrade

Service provides the ability to trade complex orders. Unlike OptionTrade, this one is used to enter a buy on the underlying symbol and a sell of the corresponding option in one transaction.

NOTE: This should not be confused with conditional option orders (where execution of one order can trigger one or more other orders). Those are implemented separately

31.1 BuyWriteOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/BuyWriteOptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order group being sent.

31.2 BuyWriteOptionTrade Parameters

Each BuyWriteOptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

OptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES

clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
action	Equity Order action	YES	**buy, sell**
action2	Option Order action	YES	**selltoopen, buytoclose**
ordtype	Case sensitive value for order type	YES	**net_debit,net_credit,market**
price	Limit price for net_debit or net_credit orders. Otherwise null	NO	

quantity	The number of **equity shares** being bought or sold in the transaction.	YES	Integer in multiples of 100
quantity2	The number of option contracts to be bought or sold.	NO	If empty, the default number of contracts will be 1/100 of the number of shares (rounded down to the nearest integer). E.g. 100 shares = 1 contract 150 shares = 1 contract 199 shares = 1 contract 200 shares = 2 contracts
symbol	The symbol for the **Equity** security to be traded	YES	
symbol2	The symbol for the **option** security to be traded	YES	
expire	Time-in-force	NO	**D or GTC**, if not supplied defaults to day expiration (D)
exday	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

exmonth	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	
exyear	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

31.3 BuyWriteOptionTrade Validation Rules

The validation rules specific to BuyWriteOptionTrade are:

- **action, action2, symbol, symbol2, ordtype, quantity**, and **accountid** are required parameters

- account must be approved for option trading.

- **if action=buy**
  action2 must be selltoopen
  ordtype must be net_debit or market

- **if action=sell**
  action2 must be buytoclose
  ordtype must be net_credit or market

- **if ordtype=net_debit or netcredit**, a valid price is required

- The underlying symbol of the option leg must be the same as the equity leg.

- The option must be a call.

31.4 BuyWriteOptionTrade Response

The response to the BuyWriteOptionTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from OptionTrade Response

General Structure of XML Response

```xml
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
            <strategy></strategy>
            <premium></premium>
            <action2></action2>
            <open-close2></open-close2>
            <security2>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security2>
        </order>
```

```
    </order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the EQUITY security
symbol	String	Contains the EQUITY symbol. For example, **DELL**
symbol-with-type-prefix	String	
description	String	
asset-type	String	Type of asset the symbol represents. It is a one character code: **E** — Equity
exchange	String	
quantity	Double	Number of shares ordered of the Equity leg of the order
order-id	String	Unique order number assigned to this order
action	String	Code indicating the order action requested. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE** tag too)
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	

order-type	String	Values returned are: **D** — Net Debit **C** — Net Credit **M** — Market
limit-price	Double	
stop-price	Double	
time-in-force	Complex	
session	String	
put-call	String	
strategy	String	**Buy Write**
premium	String	
action2	String	
open-close2	String	O or C depending if the OPTION order is to Open or Close
security2	Complex	Container for detailed information about the OPTION security
symbol	String	Contains OPTION symbol. For example, **AMTD_20090922P12.5**.
symbol-with-type-prefix	String	Contains OPTION symbol. For example, **AMTD_20090922P12.5**.
description	String	Description of the option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	
exchange	String	

31.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/BuyWriteOptionTrade?orderstring=accountid%3D123456789%7Esymbol%3DIBM%7Eordtype%3Dnet_debit%7Eaction%3Dbuy%7Equantity%3D100%7Esymbol2IBM_111010C113.5%7Eaction2%3Dselltoopen%7Eprice%3D113.54%7Eclientorderid%3D1

(Order String is URL Encoded version of accountid=123456789~symbol=IBM~ordtype=net_debit~action=buy~quantity=100~symbol2=IBM_111010C113.5~action2=selltoopen~price=113.54~clientorderid=1)

Response:


```
<?xml version="1.0" ?>
<amtd>
  <result>OK</result>
  <order-wrapper>
    <client-order-id>1</client-order-id>
    <orderstring>
    accountid=123456789~symbol=IBM~ordtype=net_debit~action=buy~quantity=100~symbol2=IBM_111010C113.5~action2=selltoopen~price=113.54~clientorderid=1</orderstring>
```

```
        <error></error>
        <order>
            <account-id>123456789</account-id>
            <security>
                <symbol>IBM</symbol>
                <symbol-with-type-prefix>IBM</symbol-with-type-prefix>
                <description>INTERNATIONAL BUSINESS MACHS COM</description>
                <asset-type>E</asset-type>
            </security>
            <quantity>100</quantity>
            <order-id>2147607354</order-id>
            <action>B</action>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>D</order-type>
            <time-in-force>
                <session>D</session>
            </time-in-force>
            <put-call>C</put-call>
            <open-close></open-close>
            <strategy>Buy Write</strategy>
            <premium>113.54</premium>
            <action2>S</action2>
            <open-close2>O</open-close2>
            <security2>
                <symbol>IBM_111010C113.5</symbol>
                <symbol-with-type-prefix>IBM_111010C113.5</
                symbol-with-type-prefix>
                <description>IBM Nov 10 2010 113.5 Call</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


31.6 BuyWriteOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

<?xml version="1.0" ?>
<amtd>
<result>**FAIL**</result>
<error>**Invalid Session**</error>
</amtd>

Order not allowed because of account positions/balances:

<?xml version="1.0" ?>
<amtd>
<result>**OK**</result>
<order-wrapper>
<orderstring>
accountid=123456789~symbol=IBM~ordtype=net_debit~action=buy~quantity=100~symbol2=IBM_111010C113.5~action2=selltoopen~price=113.54~clientorderid=1</orderstring>
<error>**Your order was not accepted. Funds are not available for this transaction of a non-marginable security.** </error>
</order-wrapper>
</amtd>

32 SpreadOptionTrade

Service provides the ability to trade complex orders. Unlike OptionTrade, this one is used to enter a buy on one option symbol and a sell of another option in one transaction.

NOTE: This should not be confused with conditional option orders (where execution of one order can trigger one or more other orders). Those are implemented separately

32.1 SpreadOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/SpreadOptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order group being sent.

32.2 SpreadOptionTrade Parameters

Each SpreadOptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

OptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account

action	Leg 1 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
action2	Leg 2 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
ordtype	Case sensitive value for order type	YES	**net_debit,net_credit,market**
price	Limit price for net_debit or net_credit orders. Otherwise null	NO	
quantity	The number of option contracts to bought or sold in the first leg of the transaction.	YES	
quantity 2	The number of contracts for the second leg.	NO	If not provided, the number of contracts in quantity is used.

symbol	The symbol for the option security to be traded - Leg 1	YES	
symbol2	The symbol for the option security to be traded - Leg 2	YES	
expire	Time-in-force	NO	**D or GTC**, if not supplied defaults to day expiration (D)
exday	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	
exmonth	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

exyear	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

32.3 SpreadOptionTrade Validation Rules

The validation rules specific to SpreadOptionTrade are:

- **action, action2, symbol, symbol2, ordtype, quantity**, and **accountid** are required parameters
- account must be approved for option trading.
- **if action=buytoopen or buytoclose**
  action2 must be selltoopen or selltoclose
- **if action=selltoopen or selltoclose**
  action2 must be buytoopen or buytoclose
- **if ordtype=net_debit or netcredit**, a valid price is required
- **if the first leg is a put,** the second leg must be a put
- **if the first leg is a call**, the second leg must be a call

32.4 SpreadOptionTrade Response

The response to the SpreadOptionTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from OptionTrade Response

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
```

```
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
            <strategy></strategy>
            <premium></premium>
            <action2></action2>
            <open-close2></open-close2>
            <security2>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted

client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the OPTION security for LEG 1
symbol	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**
description	String	Description of the Leg 1 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	
quantity	Double	Number of contracts ordered of the first leg of the order
order-id	String	Unique order number assigned to this order
action	String	Code indicating the order action requested for Leg 1 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE** tag too)
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **D** — Net Debit **C** — Net Credit **M** — Market
limit-price	Double	
stop-price	Double	
time-in-force	Complex	
session	String	
put-call	String	**P** or **C** depending if the Leg 1 OPTION order is a Put or a Call

open-close	String	**O** or **C** depending if the Leg 1 OPTION order is to Open or Close
strategy	String	**Spread**
premium	String	
action2	String	Code indicating the order action requested for Leg 2 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE2** tag too)
open-close2	String	O or C depending if the Leg 2 OPTION order is to Open or Close
security2	Complex	Container for detailed information about the OPTION security for Leg 2
symbol	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**
description	String	Description of the Leg 2 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	
exchange	String	

32.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/StraddleOptionTrade?orderstring=accountid%3D123456789%7EsymbolIBM_061910C110%7Eordtype%3Dnet_debit%7Eaction%3Dbuytoopen%7Equantity%3D2%7Esymbol2IBM_061910C120%7Eaction2%3Dselltoopen%7Eprice%3D3.9%7Eclientorderid%3D1

accountid=123456789~symbol=IBM_061910C110~ordtype=net_debit~action=buytoopen~quantity=2~symbol2=IBM_061910C120~action2=selltoopen~price=3.9~clientorderid=1)

Response:

```
&lt;?xml version="1.0" ?&gt;
&lt;amtd&gt;
    &lt;result&gt;OK&lt;/result&gt;
    &lt;order-wrapper&gt;
        &lt;client-order-id&gt;1&lt;/client-order-id&gt;
        &lt;orderstring&gt;
        accountid=123456789~symbol=IBM_061910C110~ordtype=net_debit~action=buytoopen~quantity=2~symbol2=IBM_061910C120~action2=selltoopen~price=3.9~clientorderid=1&lt;/orderstring&gt;
        &lt;error&gt;&lt;/error&gt;
        &lt;order&gt;
            &lt;account-id&gt;123456789&lt;/account-id&gt;
            &lt;security&gt;
                &lt;symbol&gt;IBM_061910C110&lt;/symbol&gt;
                &lt;symbol-with-type-prefix&gt;IBM_061910C110</symbol-with-type-prefix
```

```
                >
                <description>IBM Jun 19 2010 110.0 Call</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security>
            <quantity>2</quantity>
            <order-id>2147607354</order-id>
            <action>B</action>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>D</order-type>
            <time-in-force>
                <session>D</session>
            </time-in-force>
            <put-call>C</put-call>
            <open-close>O</open-close>
            <strategy>Spread</strategy>
            <premium>3.90</premium>
            <action2>S</action2>
            <open-close2>O</open-close2>
            <security2>
                <symbol>IBM_061910C120</symbol>
                <symbol-with-type-prefix>IBM_061910C120</symbol-with-type-prefix
                >
                <description>IBM Jun 19 2010 120.0 Call</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


32.6 SpreadOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```


If the session is invalid, you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
```

```
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        accountid=123456789~symbol=IBM_061910C110~ordtype=net_debit~
        action=buytoopen~quantity=2~symbol2=IBM_061910C120~action2=se
        lltoopen~price=3.9~clientorderid=1</orderstring>
        <error>Your order was not accepted. Funds are not available for this
        transaction of a non-marginable security. </error>
    </order-wrapper>
</amtd>
```

33 StraddleOptionTrade

Service provides the ability to trade complex orders. Unlike OptionTrade, this one is used to enter a buy or a sell on two options on the same underlying symbol in one transaction.

NOTE: This should not be confused with conditional option orders (where execution of one order can trigger one or more other orders). Those are implemented separately

33.1 StraddleOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/StraddleOptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order group being sent.

33.2 StraddleOptionTrade Parameters

Each StraddleOptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

OptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES

source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
action	Leg 1 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**

action2	Leg 2 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
ordtype	Case sensitive value for order type	YES	**net_debit,net_credit,market**
price	Limit price for net_debit or net_credit orders. Otherwise null	NO	
quantity	The number of option contracts to bought or sold in the transaction. The number of contracts for the 2nd leg is automatically calculated at 1:1 ratio	YES	If not provided, the number of contracts in quantity is used.
quantity 2	The number of contracts for the second leg.	NO	If not provided, the number of contracts in quantity is used.

symbol	The symbol for the option security to be traded - Leg 1	YES	
symbol2	The symbol for the option security to be traded - Leg 2	YES	
expire	Time-in-force	NO	**D or GTC**, if not supplied defaults to day expiration (D)
exday	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	
exmonth	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

exyear	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

33.3 StraddleOptionTrade Validation Rules

The validation rules specific to StraddleOptionTrade are:

- **action, action2, symbol, symbol2, ordtype, quantity**, and **accountid** are required parameters
- account must be approved for option trading.
- **if action=buytoopen or buytoclose**
  action2 must be buytoopen or buytoclose
- **if action=selltoopen or selltoclose**
  action2 must be selltoopen or selltoclose
- **if ordtype=net_debit or netcredit**, a valid price is required
- **if the first leg is a put**, the second leg must be a call
- **if the first leg is a call,** the second leg must be a put

33.4 StraddleOptionTrade Response

The response to the StraddleOptionTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from OptionTrade Response

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
```

```
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
            <strategy></strategy>
            <premium></premium>
            <action2></action2>
            <open-close2></open-close2>
            <security2>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted

client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the OPTION security for LEG 1
symbol	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**
description	String	Description of the Leg 1 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	
quantity	Double	Number of contracts ordered of the first leg of the order
order-id	String	Unique order number assigned to this order
action	String	Code indicating the order action requested for Leg 1 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE** tag too)
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **D** — Net Debit **C** — Net Credit **M** — Market
limit-price	Double	
stop-price	Double	
time-in-force	Complex	
session	String	
put-call	String	**P** or **C** depending if the Leg 1 OPTION order is a Put or a Call

open-close	String	**O** or **C** depending if the Leg 1 OPTION order is to Open or Close
strategy	String	**Straddle**
premium	String	
action2	String	Code indicating the order action requested for Leg 2 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE2** tag too)
open-close2	String	**O** or **C** depending if the Leg 2 OPTION order is to Open or Close
security2	Complex	Container for detailed information about the OPTION security for Leg 2
symbol	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**
description	String	Description of the Leg 1 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	

33.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/StraddleOptionTrade?orderstring=accountid%3D123456789%7Esymbol%3DIBM_071710C110%7Eordtype%3Dnet_debit%7Eaction%3Dbuytoopen%7Equantity%3D2%7Esymbol2%3DIBM_071710P110%7Eaction2%3Dbuytoopen%7Eprice%3D4.15%7Eclientorderid%3D1

(Order String is URL Encoded version of accountid=123456789~symbol=IBM_071710C110~ordtype=net_debit~action=buytoopen~quantity=2~symbol2=IBM_071710P110~action2=buytoopen~price=4.15~clientorderid=1)

Response:


```
<?xml version="1.0" ?>
<amtd>
  <result>OK</result>
  <order-wrapper>
    <client-order-id>1</client-order-id>
    <orderstring>
    accountid=123456789~symbol=IBM_071710C110~ordtype=net_debit~action=buytoopen~quantity=2~symbol2=IBM_071710P110~action2=buytoopen~price=4.15~clientorderid=1</orderstring>
    <error></error>
    <order>
      <account-id>123456789</account-id>
```

```
            <security>
                <symbol>IBM_071710C110</symbol>
                <symbol-with-type-prefix>IBM_071710C110</symbol-with-type-prefix
                >
                <description>IBM Jul 17 2010 110 Call</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security>
            <quantity>100</quantity>
            <order-id>2147607354</order-id>
            <action>B</action>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>D</order-type>
            <time-in-force>
                <session>D</session>
            </time-in-force>
            <put-call>C</put-call>
            <open-close>O</open-close>
            <strategy>Straddle</strategy>
            <premium>4.15</premium>
            <action2>B</action2>
            <open-close2>O</open-close2>
            <security2>
                <symbol>IBM_071710P110</symbol>
                <symbol-with-type-prefix>IBM_071710P110</symbol-with-type-prefix
                >
                <description>IBM Jul 17 2010 110 Put</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


33.6 StraddleOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```


If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```


Order not allowed because of account positions/balances:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        accountid=123456789~symbol=IBM_071710C110~ordtype=net_debit~
        action=buytoopen~quantity=100~symbol2=IBM_071710P110~action2=
        buytoopen~price=5.1~clientorderid=1</orderstring>
        <error>Your order was not accepted. Funds are not available for this
        transaction of a non-marginable security.  </error>
    </order-wrapper>
</amtd>
```


34 StrangleOptionTrade

Service provides the ability to trade complex orders. Unlike OptionTrade, this one is used to enter a buy or a sell on two options on the same underlying symbol in one transaction.

NOTE: This should not be confused with conditional option orders (where execution of one order can trigger one or more other orders). Those are implemented separately

34.1 StrangleOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/StrangleOptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order group being sent.

34.2 StrangleOptionTrade Parameters

Each StrangleOptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

OptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account

action	Leg 1 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
action2	Leg 2 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
ordtype	Case sensitive value for order type	YES	**net_debit,net_credit,market**
price	Limit price for net_debit or net_credit orders. Otherwise null	NO	
quantity	The number of option contracts to bought or sold in the transaction. The number of contracts for the 2nd leg is automatically calculated at 1:1 ratio	YES	

quantity2	The number of contracts for the second leg.	NO	If not provided, the number of contracts in quantity is used.
symbol	The symbol for the option security to be traded - Leg 1	YES	
symbol2	The symbol for the option security to be traded - Leg 2	YES	
expire	Time-in-force	NO	**D or GTC**, if not supplied defaults to day expiration (D)
exday	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	
exmonth	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

exyear	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

34.3 StrangleOptionTrade Validation Rules

The validation rules specific to StrangleOptionTrade are:

- **action, action2, symbol, symbol2, ordtype, quantity**, and **accountid** are required parameters

- account must be approved for option trading.

- **if action=buytoopen or buytoclose**
action2 must be buytoopen or buytoclose

- **if action=selltoopen or selltoclose**
action2 must be selltoopen or selltoclose

- **if ordtype=net_debit or netcredit**, a valid price is required

- **if the first leg is a put**, the second leg must be a call

- **if the first leg is a call**, the second leg must be a put

- **the order cannot be a straddle order**, i.e. the first and second legs cannot have the same strike price in the same expiration month

34.4 StrangleOptionTrade Response

The response to the StrangleOptionTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from OptionTrade Response

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
```

```
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
            <strategy></strategy>
            <premium></premium>
            <action2></action2>
            <open-close2></open-close2>
            <security2>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.

order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the OPTION security for LEG 1
symbol	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**.
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**.
description	String	Description of the Leg 1 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	
quantity	Double	Number of contracts ordered of the first leg of the order
order-id	String	Unique order number assigned to this order
action	String	Code indicating the order action requested for Leg 1 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE** tag too)
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **D** — Net Debit **C** — Net Credit **M** — Market
limit-price	Double	
stop-price	Double	
time-in-force	Complex	
session	String	

put-call	String	**P** or **C** depending if the Leg 1 OPTION order is a Put or a Call
open-close	String	**O** or **C** depending if the Leg 1 OPTION order is to Open or Close
strategy	String	**Strangle**
premium	String	
action2	String	Code indicating the order action requested for Leg 2 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE2** tag too)
open-close2	String	**O** or **C** depending if the Leg 2 OPTION order is to Open or Close
security2	Complex	Container for detailed information about the OPTION security for Leg 2
symbol	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**.
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**.
description	String	Description of the Leg 2 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	
exchange	String	

34.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/StrangleOptionTrade?orderstring=accountid%3D123456789%7EsymbolIBM_061910C110%7Eordtype%3Dnet_credit%7Eaction%3Dselltoopen%7Equantity%3D100%7Esymbol2IBM_061910P100%7Eaction2%3Dselltoopen%7Eprice%3D3.7%7Eclientorderid%3D1

(Order String is URL Encoded version of accountid=123456789~symbol=IBM_061910C110~ordtype=net_credit~action=selltoopen~quantity=100~symbol2=IBM_061910P100~action2=selltoopen~price=3.7~clientorderid=1)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <client-order-id>1</client-order-id>
        <orderstring>
        accountid=123456789~symbol=IBM_061910C110~ordtype=net_credit~
        action=selltoopen~quantity=100~symbol2=IBM_061910P100~action2=s
        elltoopen~price=3.7~clientorderid=1</orderstring>
        <error></error>
        <order>
            <account-id>123456789</account-id>
            <security>
```

```xml
                <symbol>IBM_061910C110</symbol>
                <symbol-with-type-prefix>IBM_061910C110</symbol-with-type-prefix
                >
                <description>IBM Jun 19 2010 110.0 Call</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security>
            <quantity>100</quantity>
            <order-id>2147694954</order-id>
            <action>S</action>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
                <response-description>AutoRoute</response-description>
            </requested-destination>
            <routing-display-size>0</routing-display-size>
            <order-type>C</order-type>
            <time-in-force>
                <session>D</session>
            </time-in-force>
            <put-call>C</put-call>
            <open-close>O</open-close>
            <strategy>Strangle</strategy>
            <premium>3.70</premium>
            <action2>S</action2>
            <open-close2>O</open-close2>
            <security2>
                <symbol>IBM_061910P100</symbol>
                <symbol-with-type-prefix>IBM_061910P100</symbol-with-type-prefix
                >
                <description>IBM Jun 19 2010 100.0 Put</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


34.6 StrangleOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```


If the session is invalid, you will get


```xml
<?xml version="1.0" ?>
```

```
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```


Order not allowed because of account positions/balances:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        accountid=123456789~symbol=IBM_061910C110~ordtype=net_credit~
        action=selltoopen~quantity=100~symbol2=IBM_061910P100~action2=
        selltoopen~price=3.7~clientorderid=1</orderstring>
        <error>Your order was not accepted. Funds are not available for this
        transaction of a non-marginable security.  </error>
    </order-wrapper>
</amtd>
```


35 ComboOptionTrade

Service provides the ability to trade complex orders. Unlike the other complex option order commands (BuyWrite, Straddle, etc), This one will accept any two option legs without specifying the strategy

NOTE: This should not be confused with conditional option orders (where execution of one order can trigger one or more other orders). Those are implemented separately

35.1 ComboOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/ComboOptionTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order group being sent.

35.2 ComboOptionTrade Parameters

Each ComboOptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, which define the order being sent

OptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the orderstring MUST be URL Encoded prior to being added to the URL.

ORDERSTRING Trading Field Definitions

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account

action	Leg 1 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
action2	Leg 2 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
ordtype	Case sensitive value for order type	YES	**net_debit,net_credit,market**
price	Limit price for net_debit or net_credit orders. Otherwise null	NO	
quantity	The number of option contracts to bought or sold in the transaction. The number of contracts for the 2nd leg is automatically calculated at 1:1 ratio	YES	

quantity2	The number of contracts for the second leg.	NO	If not provided, the number of contracts in quantity is used.
symbol	The symbol for the option security to be traded - Leg 1	YES	
symbol2	The symbol for the option security to be traded - Leg 2	YES	
expire	Time-in-force	NO	**D or GTC**, if not supplied defaults to day expiration (D)
exday	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	
exmonth	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

exyear	Two digit expiration day, only specified if expire is set to GTC otherwise null	NO	

35.3 ComboOptionTrade Validation Rules

The validation rules specific to ComboOptionTrade are:

- **action, action2, symbol, symbol2, ordtype, quantity,** and **accountid** are required parameters
- account must be approved for option trading.
- The options must have the **same underlying equity**
- The first and second leg symbols must be different

35.4 ComboOptionTrade Response

The response to the ComboOptionTrade request will be in XML format.

NOTE: the structure of the ORDER section is the same as what you get from OptionTrade Response

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
        <client-order-id></client-order-id>
        <orderstring></orderstring>
        <error></error>
        <order>
            <account-id></account-id>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security>
```

```
            <quantity></quantity>
            <order-id></order-id>
            <action></action>
            <requested-destination>
                <routing-mode></routing-mode>
                <option-exchange></option-exchange>
                <response-description></response-description>
            </requested-destination>
            <routing-display-size></routing-display-size>
            <order-type></order-type>
            <time-in-force>
                <session></session>
            </time-in-force>
            <put-call></put-call>
            <open-close></open-close>
            <premium></premium>
            <action2></action2>
            <open-close2></open-close2>
            <security2>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description></description>
                <asset-type></asset-type>
                <exchange></exchange>
            </security2>
        </order>
    </order-wrapper>
</amtd>
```


Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
security	Complex	Container for detailed information about the OPTION security for LEG 1
symbol	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**.

symbol-with-type-prefix	String	Contains OPTION symbol for Leg 1. For example, **AMTD_092209P12.5**.
description	String	Description of the Leg 1 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	
quantity	Double	Number of contracts ordered of the first leg of the order
order-id	String	Unique order number assigned to this order
action	String	Code indicating the order action requested for Leg 1 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE** tag too)
requested-destination	Complex	
routing-mode	String	
market-maker-id	String	
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **D** — Net Debit **C** — Net Credit **M** — Market
limit-price	Double	
stop-price	Double	
time-in-force	Complex	
session	String	
put-call	String	**P** or **C** depending if the Leg 1 OPTION order is a Put or a Call
open-close	String	**O** or **C** depending if the Leg 1 OPTION order is to Open or Close
premium	String	
action2	String	Code indicating the order action requested for Leg 2 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE2** tag too)
open-close2	String	**O** or **C** depending if the Leg 2 OPTION order is to Open or Close

security2	Complex	Container for detailed information about the OPTION security for Leg 2
symbol	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**.
symbol-with-type-prefix	String	Contains OPTION symbol for Leg 2. For example, **AMTD_092209P12.5**.
description	String	Description of the Leg 1 option symbol. For example: **AMTD Sep 22 2009 12.5 Call**
asset-type	String	
exchange	String	

35.5 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/ComboOptionTrade?orderstring=accountid%3D123456789%7Esymbol%3DIBM_091010C120%7Eordtype%3Dnet_debit%7Eaction%3Dbuytoopen%7Equantity%3D10%7Esymbol2%3DIBM_081010P110%7Eaction2%3Dbuytoopen%7Eprice%3D3.5%7Eclientorderid%3D1

(Order String is URL Encoded version of accountid=123456789~symbol=IBM_091010C120~ordtype=net_debit~action=buytoopen~quantity=10~symbol2=IBM_081010P110~action2=buytoopen~price=3.5~clientorderid=1)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <client-order-id>1</client-order-id>
        <orderstring>
        accountid=123456789~symbol=IBM_091010C120~ordtype=net_debit~a
        ction=buytoopen~quantity=10~symbol2=IBM_20100810P110~action2=
        buytoopen~price=3.5~clientorderid=1</orderstring>
        <error></error>
        <order>
            <account-id>123456789</account-id>
            <security>
                <symbol>IBM_091010C120</symbol>
                <symbol-with-type-prefix>IBM_091010C120</symbol-with-type-prefix
                >
                <description>IBM Sep 10 2010 120.0 Call</description>
                <asset-type>O</asset-type>
                <exchange>OPR</exchange>
            </security>
            <quantity>10</quantity>
            <order-id>1234567890</order-id>
            <action>B</action>
            <requested-destination>
                <routing-mode>SMART</routing-mode>
                <option-exchange>Auto</option-exchange>
```

```
            <response-description>AutoRoute</response-description>
        </requested-destination>
        <routing-display-size>0</routing-display-size>
        <order-type>D</order-type>
        <time-in-force>
            <session>D</session>
        </time-in-force>
        <put-call>C</put-call>
        <open-close>O</open-close>
        <premium>3.50</premium>
        <action2>B</action2>
        <open-close2>O</open-close2>
        <security2>
            <symbol>IBM_081010P110</symbol>
            <symbol-with-type-prefix>IBM_081010P110</symbol-with-type-prefix
            >
            <description>IBM Aug 10 2010 110.0 Put</description>
            <asset-type>O</asset-type>
            <exchange>OPR</exchange>
        </security2>
    </order>
  </order-wrapper>
</amtd>
```

35.6 ComboOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
        <orderstring>
        accountid=123456789~symbol=IBM_091010C120~ordtype=net_debit~
        action=buytoopen~quantity=10~symbol2=IBM_081010P110~action2=b
        uytoopen~price=3.5~clientorderid=1</orderstring>
```

<error>**Your order was not accepted. Funds are not available for this transaction of a non-marginable security.** </error>
</order-wrapper>
</amtd>

36 MultiLegOptionTrade

Service provides the ability to trade complex orders. Unlike other complex option order commands (BuyWrite, Straddle, etc), this will accept multiple option legs.

NOTE: This should not be confused with conditional option orders (where execution of one order can trigger one or more other orders). Those are implemented separately.

36.1 MultiLegOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/MultiLegOptionTrade?source=<**#SourceID#**>&message=<**#messagexml#**>

where messagexml contains parameters as it is shown in the Request Sample

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more message parameters, each one defining one order group being sent.

36.2 MultiLegOptionTrade Parameters

Each MultiLegOptionTrade request must contain the source id assigned by TD Ameritrade as well as one or more messagexml parameters, which define the order being sent

MultiLegOptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
message	consists of variable=value pairs, separated by ~ (tilde).	YES	Values defined in the table below. **NOTE:** The contents of the message MUST be URL Encoded prior to being added to the URL.

MESSAGEXML Trading Field Definitions

The following parameters have to be POSTed:

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
clientorderid	Generated by the client software. It gets returned in the response for order matching.	NO	Any integer value. Should be unique for that client
accountid	The Account ID of the account in which the trade is being made	YES	One of the account IDs returned from the LogIn service. The default is the associated account
order-type	Case sensitive value for order type	YES	**net_debit,net_credit,market**
price	Limit price for net_debit or net_credit orders. Otherwise null	NO	Price per share or per contract. (6 digits maximum to the right of the decimal point)
expiration	Time-in-force	YES	**D or GTC**
exday	Two digit expiration day, only specified if expiration is set to GTC otherwise null	NO	
exmonth	Two digit expiration month, only specified if expiration is set to GTC otherwise null.	NO	
exyear	Two digit expiration year, only specified if expiration is set to GTC otherwise null.	NO	
legs			
orderleg			

symbol	The symbol for the option security to be traded - Leg 1	YES	
action	Leg 1 Option Order action	YES	**buytoopen, buytoclose, selltoopen, selltoclose**
quantity	The number of option contracts to bought or sold in the transaction. The number of contracts for the 2nd leg is automatically calculated at 1:1 ratio	YES	any positive integer value
assettype	The type of asset that is being traded		Equities or Options
strategy		YES	Any of the following values: **Spread** **Straddle** **Buy Write** **Combo** **Vertical** **Back Ratio** **Calendar** **Butterfly** **Condor** **Iron Condor** **Vertical Roll** **Covered Stock** **Collar With Stock** **Diagonal** **Collar Synthetic** **Double Diagonal** **Custom**

36.3 MultiLegOptionTrade Validation Rules

The validation rules specific to MultiLegOptionTrade are:

- The options must have the **same underlying equity**
- The first and second leg symbols must be different

36.4 MultiLegOptionTrade Response

The response to the MultiLegOptionTrade request will be in XML format.

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <order-wrapper>
    <client-order-id></client-order-id>
    <request>
        <complex-option-order-request>
        <account-id></account-id>
        <strategy></strategy>
        <order-type></order-type>
        <price></price>
        <client-order-id></client-order-id>
        <expiration></expiration>
        <legs>
            <order-leg>
                <symbol></symbol>
                <action></action>
                <quantity></quantity>
                <asset-type></asset-type>
            </order-leg>
            <order-leg>
                <symbol></symbol>
                <action></action>
                <quantity></quantity>
                <asset-type></asset-type>
            </order-leg>
            <order-leg>
                <symbol></symbol>
                <action></action>
                <quantity></quantity>
                <asset-type></asset-type>
            </order-leg>
        </legs>
        </complex-option-order-request>
    </request>
    <order>
        <account-id></account-id>
        <symbol></symbol>
        <security>
            <symbol></symbol>
            <symbol-with-type-prefix></symbol-with-type-prefix>
            <description></description>
            <asset-type></asset-type>
            <exchange></exchange>
        </security>
        <quantity></quantity>
        <order-id></order-id>
```

```
        <action></action>
        <requested-destination>
            <routing-mode></routing-mode>
            <option-exchange></option-exchange>
            <response-description></response-description>
        </requested-destination>
        <routing-display-size></routing-display-size>
        <order-type></order-type>
        <limit-price> </limit-price>
        <stop-price> </stop-price>
        <time-in-force>
            <session></session>
        </time-in-force>
        <put-call></put-call>
        <open-close></open-close>
        <premium></premium>
        <legs>
        <order-leg>
            <action></action>
            <quantity></quantity>
            <open-close></open-close>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description/>
                <asset-type></asset-type>
            </security>
        </order-leg>
        <order-leg>
            <action></action>
            <quantity></quantity>
            <open-close></open-close>
            <security>
                <symbol></symbol>
                <symbol-with-type-prefix></symbol-with-type-prefix>
                <description/>
                <asset-type>O</asset-type>
            </security>
        </order-leg>
        </legs>
        </order>
    </order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.

order-wrapper	Complex	Container for the order information that is being submitted
client-order-id	String	The ID Generated by the client software and submitted with the order. Used for order matching.
order	Complex	Container for detailed information about the order
accountid	String	The account id in which the order was placed
symbol	String	Contains OPTION symbol. For example, **AMTD_092210P12.5**.
security	Complex	Container for detailed information about the OPTION security for LEG 1
symbol-with-type-prefix	String	Contains OPTION symbol. For example, **AMTD_092210P12.5**.
description	String	Description of the OPTION symbol. For example: **AMTD Sep 22 2010 12.5 Call**
asset-type	String	Type of asset the symbol represents. It is a one character code: **O** — Option
exchange	String	
quantity	Double	Number of contracts ordered of the first leg of the order
order-id	String	Unique order number assigned to this order
action	String	Code indicating the order action requested for Leg 1 of the order. Values returned are: **B** — Buy **S** — Sell (NOTE: Look at the **OPEN-CLOSE** tag too)
requested-destination	Complex	
routing-mode	String	
optionexchange		
response-description	String	
routing-display-size	Integer	
order-type	String	Values returned are: **D** — Net Debit **C** — Net Credit **M** — Market
limit-price	Double	
stop-price	Double	
time-in-force	Complex	
session	String	
put-call	String	**P** or **C** depending if the Leg 1 OPTION order is a Put or a Call

open-close	String	**O** or **C** depending if the Leg 1 OPTION order is to Open or Close
premium	String	

36.5 MultiLegOptionTrade Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/MultiLegOptionTrade

POSTED DATA:

source=<##SourceID#>&message=<#messagexml#>

where message contains parameters as it is shown below:

```
<complex-option-order-request>
    <account-id>123456789</account-id>
    <strategy>Butterfly</strategy>
    <order-type>net_debit</order-type>
    <price>3.00</price>
    <client-order-id>1</client-order-id>
    <expiration>day</expiration>
    <legs>
        <order-leg>
            <symbol>IBM_061910C85</symbol>
            <action>buytoopen</action>
            <quantity>1</quantity>
            <asset-type>O</asset-type>
        </order-leg>
        <order-leg>
            <symbol>IBM_061910C115</symbol>
            <action>selltoopen</action>
            <quantity>2</quantity>
            <asset-type>O</asset-type>
        </order-leg>
        <order-leg>
            <symbol>IBM_061910C140</symbol>
            <action>buytoopen</action>
            <quantity>1</quantity>
            <asset-type>O</asset-type>
        </order-leg>
    </legs>
</complex-option-order-request>
```

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <order-wrapper>
    <client-order-id>4196530660</client-order-id>
    <request>
```

```
        <complex-option-order-request>
        <account-id>123456789</account-id>
        <strategy>Butterfly</strategy>
        <order-type>net_debit</order-type>
        <price>3.00</price>
        <client-order-id>1</client-order-id>
        <expiration>D</expiration>
        <legs>
            <order-leg>
                <symbol>IBM_061910C85</symbol>
                <action>buytoopen</action>
                <quantity>1</quantity>
                <asset-type>O</asset-type>
            </order-leg>
            <order-leg>
                <symbol>IBM_061910C115</symbol>
                <action>selltoopen</action>
                <quantity>2</quantity>
                <asset-type>O</asset-type>
            </order-leg>
            <order-leg>
                <symbol>IBM_061910C140</symbol>
                <action>buytoopen</action>
                <quantity>1</quantity>
                <asset-type>O</asset-type>
            </order-leg>
        </legs>
        </complex-option-order-request>
    </request>
    <order>
        <account-id>123456789</account-id>
        <symbol>IBM_061910C85</symbol>
        <security>
            <symbol>IBM_061910C85</symbol>
            <symbol-with-type-prefix>IBM_061910C85</symbol-with-type-prefix>
            <description>IBM Jun 19 2010 85.0 Call</description>
            <asset-type>O</asset-type>
            <exchange>OPRA</exchange>
        </security>
        <quantity>1</quantity>
        <order-id>4196530660</order-id>
        <action>B</action>
        <requested-destination>
            <routing-mode>SMART</routing-mode>
            <option-exchange>Auto</option-exchange>
            <response-description>AutoRoute</response-description>
        </requested-destination>
        <routing-display-size>0</routing-display-size>
        <order-type>D</order-type>
        <limit-price> </limit-price>
        <stop-price> </stop-price>
        <time-in-force>
            <session>Day</session>
```

```
        </time-in-force>
        <put-call>C</put-call>
        <open-close>O</open-close>
        <premium>3.00</premium>
        <legs>
        <order-leg>
            <action>S</action>
            <quantity>2.0</quantity>
            <open-close>O</open-close>
            <security>
                <symbol>IBM_061910C115</symbol>
                <symbol-with-type-prefix>IBM_061910C115</symbol-with-type-prefix
                >
                <description>IBM Jun 19 2010 115.0 Call</description>
                <asset-type>O</asset-type>
            </security>
        </order-leg>
        <order-leg>
            <action>B</action>
            <quantity>1.0</quantity>
            <open-close>O</open-close>
            <security>
                <symbol>IBM_061910C140</symbol>
                <symbol-with-type-prefix>IBM_061910C140</symbol-with-type-prefix
                >
                <description>IBM Jun 19 2010 140.0 Call </description>
                <asset-type>O</asset-type>
            </security>
        </order-leg>
        </legs>
        </order>
    </order-wrapper>
</amtd>
```


36.6 MultiLegOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```


If the session is invalid, you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

Order not allowed because of account positions/balances:


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <order-wrapper>
        <error>Your order was not accepted. Funds are not available for this
        transaction of a non-marginable security.
        </error>
    </order-wrapper>
</amtd>
```


If the account is not enabled for n-legged option trading:


```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <order-wrapper>
        <error>Account not enabled for n-legged option trading.
        </error>
    </order-wrapper>
</amtd>
```


37 SavedOrders (list of saved orders)

This service provides a way to view the previously saved Equity and Options orders

37.1 SavedOrders Request

Request URL

https://apis.tdameritrade.com/apps/100/SavedOrders?source=<**#sourceID#**>

Parameters

The only parameter required is the source assigned by TD Ameritrade. You can also specify the account id for which the data is to be returned

37.2 SavedOrders Parameters

The only parameter required is the source assigned by TD Ameritrade.

OrderStatus Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account

37.3 SavedOrders Response

The response to the SavedOrders request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <savedorder-list>
        <account-id></account-id>
        <savedorder>
            <action></action>
            <activation-price></activation-price>
            <asset-type></asset-type>
            <date-saved></date-saved>
            <display-size></display-size>
            <expiration-day></expiration-day>
            <expiration-month></expiration-month>
            <expiration-year></expiration-year>
            <id></id>
            <order-type></order-type>
            <price></price>
            <quantity></quantity>
            <routing></routing>
            <stop-param></stop-param>
            <symbol></symbol>
            <description></description>
            <special-conditions></special-conditions>
            <time-in-force></time-in-force>
            <underlying-symbol></underlying-symbol>
            <dividend-reinvest></dividend-reinvest>
            <capital-gains-reinvest></capital-gains-reinvest>
        </savedorder>
    </savedorder-list>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
savedorder-list	Complex	Container for all saved order entries being returned for the account
account-id	String	
savedorder	Complex	Contains all the fields for one saved order
action	String	Order action

activation-price	Float	The stop price for stop market or stop limit orders
asset-type	String	Type of asset the symbol represents. It is a one character code where: **E** — Equity or ETF **F** — Mutual Fund **O** — Option **B** — Bond
date-saved	String	The date this order was saved
display-size	Integer	Integer value 100 or higher, in increments of 100
expiration-day	Integer	Two digit expiration day, only specified if expire is set to gtc otherwise
expiration-month	Integer	Two digit expiration month, only specified if expire is set to gtc otherwise null.
expiration-year	Integer	Two digit expiration year, only specified if expire is set to gtc otherwise null.
id	Integer	the unique ID of the saved order
order--type	String	Order Type string
price	Float	Limit price for limit or stop limit orders otherwise null
quantity	Float	The number of shares being bought or sold in the transaction
routing		User specified order routing destination. Default is auto
stop-param	Float	Trailing Stop parameter in dollars or percent depending on order type otherwise null.
symbol	String	Symbol being bought or sold. UPPER CASE
description	String	
special-conditions	String	Special Instructions for the order execution. Set to none if not specified
time-in-force	String	The time in force for the order
dividend-reinvest	Boolean	**true/false**
capital-gains-reinvest	Boolean	**true/false**

37.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/SavedOrders?source=<**#sourceID#**>

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <savedorder-list>
        <account-id>123456789</account-id>
        <saved-order>
            <action>BUY</action>
            <activation-price>  </activation-price>
```

```
        <asset-type>E</asset-type>
        <date-saved>2008-05-19 22:40:29 EST</date-saved>
        <display-size>0</display-size>
        <id>23299</id>
        <order-type>LIMIT</order-type>
        <price>43.01</price>
        <quantity>70</quantity>
        <routing>AUTO</routing>
        <stop-param></stop-param>
        <symbol>CVS</symbol>
        <description></description>
        <time-in-force>DAY</time-in-force>
        <dividend-reinvest></dividend-reinvest>
        <capital-gains-reinvest></capital-gains-reinvest>
    </saved-order>
    <saved-order>
        <action>BUY</action>
        <activation-price>  </activation-price>
        <asset-type>E</asset-type>
        <date-saved>2008-05-19 22:51:31 EST</date-saved>
        <display-size>0</display-size>
        <expiration-day>1</expiration-day>
        <expiration-month>11</expiration-month>
        <expiration-year>2008</expiration-year>
        <id>23300</id>
        <order-type>LIMIT</order-type>
        <price>43.01</price>
        <quantity>70</quantity>
        <routing>AUTO</routing>
        <stop-param></stop-param>
        <symbol>CVS</symbol>
        <description></description>
        <special-conditions>CVS</special-conditions>
        <time-in-force>GTC_EXT</time-in-force>
        <underlying-symbol>CVS</underlying-symbol>
        <dividend-reinvest></dividend-reinvest>
        <capital-gains-reinvest></capital-gains-reinvest>
    </saved-order>
    <saved-order>
        <action>SELL</action>
        <activation-price>2.00</activation-price>
        <asset-type>E</asset-type>
        <date-saved>2008-05-19 22:53:27 EST</date-saved>
        <display-size>0</display-size>
        <id>23301</id>
        <order-type>TSTOPDOLLAR</order-type>
        <price></price>
        <quantity>70</quantity>
        <routing>AUTO</routing>
        <stop-param>2.00</stop-param>
        <symbol>CVS</symbol>
        <description></description>
        <time-in-force>DAY</time-in-force>
```

```
            <dividend-reinvest></dividend-reinvest>
            <capital-gains-reinvest></capital-gains-reinvest>
        </saved-order>
    </savedorder-list>
</amtd>
```

37.5 SavedOrders Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

38 SaveEquityTrade

Service provides the ability to Save an Equity Trade order on the servers for use in the future. Refer to SavedOrders command for retrieving saved orders.

38.1 SaveEquityTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/SaveEquityTrade?source=<**#SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order being saved. The orderstring parameter is exactly the same as what is used for the EquityTrade command.

38.2 SaveEquityTrade Parameters

Each SaveEquityTrade request must contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining an order being saved

SaveEquityTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	**orderstring definitions and validation rules are exactly the same as for the EquityTrade command**

NOTE: The **accountid** is specified inside the orderstring, not as a separate parameter

38.3 SaveEquityTrade Response

The response to the SaveEquityTrade request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <saved-order-wrapper>
        <orderstring></orderstring>
        <error></error>
        <order-saved></order-saved>
    </saved-order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
saved-order-wrapper	Complex	Container for the order information that is being submitted
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order-saved	String	true if successfully saved, false otherwise

38.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/SaveEquityTrade?source=<**#sourceID#**>&orderstring=action%3Dbuy%7Equantity%3D400%7Esymbol%3DDELL%7Eordtype%3DLimit%7Eprice%3D27.49%7Eactprice%3D%7Etsparam%3D%7Eexpire%3Dday%7Espinstructions%3Dnone%7Erouting%3Dauto%7Edisplaysize%3D%7Eexmonth%3D%7Eexday%3D%7Eexyear%3D%7Eaccountid%3D123456789

(Order String is URL Encoded version of
action=buy~quantity=400~symbol=DELL~ordtype=Limit~price=27.49~actprice=~tsparam=~expire=day~spinstructions=none~routing=auto~displaysize=~exmonth=~exday=~exyear=~accountid=123456789)

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <saved-order-wrapper>
        <orderstring>
        action=buy~quantity=400~symbol=DELL~ordtype=Limit~price=27.49~
        actprice=~tsparam=~expire=day~spinstructions=none~routing=auto~d
        isplaysize=~exmonth=~exday=~exyear=~accountid=123456789
        </orderstring>
        <error></error>
        <saved-order>true</saved-order>
    </saved-order-wrapper>
</amtd>
```


38.5 SaveEquityTrade Errors

Possible Errors

If the Account ID specified is not valid you will get


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <saved-order-wrapper>
        <orderstring>
        action=buy~quantity=400~symbol=DELL~ordtype=Limit~price=27.49~
        actprice=~tsparam=~expire=day~spinstructions=none~routing=auto~d
        isplaysize=~exmonth=~exday=~exyear=~accountid=123456789
        </orderstring>
        <error>Not a valid account for this user</error>
        <saved-order>false</saved-order>
    </saved-order-wrapper>
</amtd>
```


If the session is invalid, you will get


```
<?xml version="1.0" ?>
<amtd>
```

<result>**FAIL**</result>
<error>**Invalid Session**</error>
</amtd>

39 SaveOptionTrade

Service provides the ability to Save an Option Trade order on the servers for use in the future. Refer to SavedOrders command for retrieving saved orders.

39.1 SaveOptionTrade Request

Request URL

https://apis.tdameritrade.com/apps/100/SaveOptionTrade?source=<**##SourceID#**>&orderstring=<**#ORDERSTRING#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining one order being saved. The orderstring parameter is exactly the same as what is used for the OptionTrade command.

39.2 SaveOptionTrade Parameters

Each SaveOptionTrade request much contain the source id assigned by TD Ameritrade as well as one or more orderstring parameters, each one defining an order being saved

SaveOptionTrade Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
orderstring	consists of variable=value pairs, separated by ~ (tilde).	YES	**orderstring definitions and validation rules are exactly the same as for the OptionTrade command**

NOTE: The **accountid** is specified inside the orderstring, not as a separate parameter

39.3 SaveOptionTrade Response

The response to the SaveOptionTrade request will be in XML format

General Structure of XML Response

```xml
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <saved-order-wrapper>
        <orderstring></orderstring>
        <error></error>
        <order-saved></order-saved>
    </saved-order-wrapper>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
saved-order-wrapper	Complex	Container for the order information that is being submitted
orderstring	String	The orderstring used to submit this order
error	String	Error message if an error occurred when placing the order
order-saved	String	true if successfully saved, false otherwise

39.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/SaveOptionOrder?source=<**#sourceID#**>&orderstring=action%3Dselltoclose%7Equantity%3D10%7Esymbol%3D%2BIBM_061910C85%7Eordtype%3DLimit%7Eprice%3D6.3%7Eactprice%3D%7Eexpire%3Dday%7Espinstructions%3Dnone%7Erouting%3Dauto%7Eexmonth%3D%7Eexday%3D%7Eexyear%3D%7Etabid%3D0%7Eaccountid%3D123456789

(Order String is URL Encoded version of action=selltoclose~quantity=10~symbol=IBM_061910C85 ~ordtype=Limit~price=6.3~actprice=~expire=day~spinstructions=none~routing=auto ~exmonth=~exday=~exyear=~tabid=0~accountid=123456789)

Response:

```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <saved-order-wrapper>
        <orderstring>action=selltoclose~quantity=10~symbol=IBM_061910C85
        ~ordtype=Limit~price=6.3~actprice=~expire=day~spinstructions=none~routin
        g=auto~exmonth=~exday=~exyear=~tabid=0~accountid=123456789
        </orderstring>
        <error></error>
        <saved-order>true</saved-order>
```

```
    </saved-order-wrapper>
</amtd>
```

39.5 SaveOptionTrade Errors

Possible Errors

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <saved-order-wrapper>
        <orderstring>action=selltoclose~quantity=10~symbol=IBM_20071010C85
        ~ordtype=Limit~price=6.3~actprice=~expire=day~spinstructions=none
        ~routing=auto~exmonth=~exday=~exyear=~tabid=0~accountid=1234
        56789
        </orderstring>
        <error>Not a valid account for this user</error>
        <saved-order>false</saved-order>
    </saved-order-wrapper>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

40 DeleteSavedOrders

Service provides the ability to delete one or more previously saved orders.

40.1 DeleteSavedOrders Request

Request URL

https://apis.tdameritrade.com/apps/100/DeleteSavedOrders?source=<**#SourceID#**>&orderid=<**#order-id#**>&orderid=<**#order-id#**>

Request Parameters

The request must contain the source assigned by TD Ameritrade as well as one or more orderid parameters for previously saved orders

40.2 DeleteSavedOrders Parameters

The source assigned by TD Ameritrade is a required parameter, as well as one or more orderid paramethers

DeleteSavedOrders Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which the data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account
orderid	The Order ID of an existing saved order	YES	One of previously saved orders that is to be deleted. The orderid would be either returned from SaveEquityTrade or SaveOptionTrade, or retrievied from the SavedOrders response

40.3 DeleteSavedOrders Response

The response to the DeleteSavedOrders request will be in XML format.

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.

40.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/DeleteSavedOrders?source=<**#sourceID#**>&orderid=21481399&orderid=21479999&orderid=21479448

Response:

```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <deleted-savedorder-list>
       <account-id>123456789</account-id>
       <deleted-saved-order>
              <id>2148881399</id>
      </deleted-saved-order>
       <deleted-saved-order>
              <id>2147949999</id>
      </deleted-saved-order>
       <deleted-saved-order>
              <id>2147949948</id>
      </deleted-saved-order>
    </deleted-savedorder-list>
</amtd>
```

40.5 DeleteSavedOrders Errors

Possible Errors

If the session is invalid, you will get

```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

if the order was closed or could not be found, or a cancel for a conditional order was submitted, you will get the following errors:

```xml
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <cancel-order-messages>
    <account-id>123456789</account-id>
    <order>
        <order-id>2147949999</order-id>
        <message></message>
        <error>The order could not found. Please check its status.</error>
    </order>
    <order>
        <order-id>2147949948</order-id>
        <message></message>
        <error>Conditional orders cannot be cancelled through this interface.</
        error>
    </order>
    </cancel-order-messages>
```

```
</amtd>
```

If no orders were specified, then the following error will be returned:

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>No Orders have been selected to be canceled.</error>
</amtd>
```

41 GetWatchlists

This service provides a way to view the previously saved WatchLists

41.1 GetWatchlists Request

Request URL

https://apis.tdameritrade.com/apps/100/GetWatchlists?source=<**#sourceID#**>

Parameters

The only parameter required is the source assigned by TD Ameritrade. Other parameters are available

41.2 GetWatchlists Parameters

The only parameter required is the source assigned by TD Ameritrade.

OrderStatus Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account
listid	The ID of the previously saved watchlist		Any valid ID of a Watchlist saved for the particular login

41.3 GetWatchlists Response

The response to the GetWatchlists request will be in XML format

General Structure of XML Response


```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <watchlist-result>
        <error></error>
        <account-id></account-id>
        <watchlist>
            <name></name>
            <id></id>
            <symbol-list>
            <watched-symbol>
                <quantity></quantity>
                <security>
                    <symbol></symbol>
                    <symbol-with-type-prefix></symbol-with-type-prefix>
                    <description></description>
                    <asset-type></asset-type>
                </security>
                <position-type></position-type>
                <average-price></average-price>
                <commission></commission>
                <open-date></open-date>
            </watched-symbol>
            .....
            </symbol-list>
        </watchlist>
    </watchlist-result>
</amtd>
```


Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
watchlist-list	Complex	Container for all WatchList entries being returned
error	String	Error string in case of an error
account-id	String	
watchlist	Complex	Contains all the fields for one watchlist
name	String	The name assigned to the given watchlist
id	Integer	Numeric ID that identifies the watchlist
symbol-list	Complex	Container for all the symbols in the watchlist
watched-symbol	Complex	Container for one symbol in the watchlist

quantity	Double	Number of units in the position. Field may have up to 3 digits to the right of the decimal point.
security	Complex	Container for detailed information about the security.
symbol	String	Contains symbol. For example, **AMTD**.
symbol-with-type-prefix	String	
description	String	Contains a description of the symbol. For example "**TD AMERITRADE HLDG CORP COM**"
asset-type	String	Type of asset the symbol represents. It is a one character code where: **E** — Equity or ETF **F** — Mutual Fund **O** — Option **B** — Bond
position-type	String	**LONG** or **SHORT**
average-price	Double	Contains the value calculated by dividing the total cost to acquire the position by the number of units in the position. The field may contain up to six digits to the right of the decimal point.
commission	Double	
open-date	String	Date the "position" was opened

41.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/GetWatchlists?source=<**#sourceID#**>

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <watchlist-result>
        <error></error>
        <account-id>123456789</account-id>
        <watchlist>
            <name>Sample WatchList 1</name>
            <id>1234561</id>
            <symbol-list>
            <watched-symbol>
                <quantity>0</quantity>
                <security>
                    <symbol>BBBY</symbol>
                    <symbol-with-type-prefix>BBBY</symbol-with-type-prefix>
                    <description>BED BATH & BEYOND INC COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
```

```
                <average-price>0</average-price>
                <commission>0</commission>
            </watched-symbol>
            </symbol-list>
        </watchlist>
        <watchlist>
            <name>Some Other Test</name>
            <id>1234562</id>
            <symbol-list>
            <watched-symbol>
                <quantity>100</quantity>
                <security>
                    <symbol>DELL</symbol>
                    <symbol-with-type-prefix>DELL</symbol-with-type-prefix>
                    <description>DELL INC COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
                <average-price>11.11</average-price>
                <commission>10</commission>
                <open-date>01-11-2008</open-date>
            </watched-symbol>
            <watched-symbol>
                <quantity>2</quantity>
                <security>
                <symbol>INTC</symbol>
                    <symbol-with-type-prefix>INTC</symbol-with-type-prefix>
                    <description>INTEL CORP COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
                <average-price>12.12</average-price>
                <commission>20</commission>
                <open-date>02-22-2006</open-date>
            </watched-symbol>
            <watched-symbol>
                <quantity>3</quantity>
                <security>
                    <symbol>T</symbol>
                    <symbol-with-type-prefix>T</symbol-with-type-prefix>
                    <description>AT&T INC COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
                <average-price>13.13</average-price>
                <commission>30</commission>
                <open-date>03-30-2007</open-date>
            </watched-symbol>
            </symbol-list>
        </watchlist>
    </watchlist-result>
</amtd>
```

41.5 GetWatchlists Errors

Possible Errors

If an invalid listid is specified

```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Watchlist id not found.</error>
</amtd>
```

If the Account ID specified is not valid you will get

```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

42 CreateWatchlist

This service provides a way to create new Watchlists containing one or more entries. Each entry consists of a symbol, quantity, price, Purchase Month, Day, Year, and commission. All entries other than symbol can be null.

42.1 CreateWatchlist Request

Request URL

https://apis.tdameritrade.com/apps/100/CreateWatchlist?source=<**#sourceID#**>&watchlistname=<**#PortfolioName#**>&symbollist=<**#SymbolList#**>

Parameters

You must specify the **watchlistname** and **symbollist** parameters, with at least one symbol, as well as the **source** assigned by TD Ameritrade. Other parameters are available

42.2 CreateWatchlist Parameters

The only parameters that are required are the **watchlistname**, **symbollist** and the **source** assigned by TD Ameritrade.

The Watchlist can contain one or more entries. Each entry consists of a symbol, quantity, price, Purchase Month, Day, Year, and commission. All entries other than symbol can be null. When creating the Watchlist, you need to specify a list values for all entries for each of the fields. Unspecified entries will be added as null values. So if you want to create a Watchlist with the following entries:

Symbol: **DELL** Quantity: **100** Purchase Date **02/21/2008** Price Paid: **20.23** Commission: **9.99**
Symbol: **ABCD** Quantity: **200** Purchase Date **02/22/2008** Price Paid: **23.68**
Symbol: **EFGH** Quantity: **300** Purchase Date **02/23/2008** Price Paid: **10.82** Commission: **9.99**
Symbol: **IBM** Quantity: **400**
Symbol: **XYZ**

You would specify:

watchlistname=Some+Portfolio+Name+Example
symbollist=DELL,ABCD,EFGH,IBM,ZYZ
quantitylist=100,200,300,400
monthlist=2,2,2
daylist=21,22,23
yearlist=2008,2008,2008
pricelist=20.23,23.68,10.82
commissionlist=9.99,,9.99

NOTE: All the list parameters are comma delimited **with no spaces**

NOTE: that a null entry is added in the commissionlist. A zero could have been used as well

OrderStatus Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by TD Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account
watchlistname	The name to assign the watchlist as it will be displayed to the end-user	YES	Text string for the watchlist name that is not already in use by another watchlist. **NOTE: names are not case sensitive**
symbollist	The list of symbols that will comprise the WatchList	YES	list of symbols
quantitylist	The list of Quantity values for each of the entries in the WatchList.	NO	list of Quantity values. Leave each value as null if not specified.

pricelist	The list of Purchase Price values for each of the entries in the WatchList.	NO	list of Price Values. Leave each value as null if not specified.
monthlist	The list of Purchase Month values for each of the entries in the WatchList.	NO	list of Month Values. Leave each value as null if not specified.
daylist	The list of Purchase Day values for each of the entries in the WatchList.	NO	list of Day Values. Leave each value as null if not specified.
yearlist	The list of Purchase Year values for each of the entries in the WatchList.	NO	list of Year Values. Leave each value as null if not specified.
commissionlist	The list of Commission values for each of the entries in the WatchList.	NO	list of Commission Values. Leave each value as null if not specified.

42.3 CreateWatchlist Response

The response to the CreateWatchlist request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <create-watchlist-result>
        <error></error>
        <account-id></account-id>
        <created-watchlist>
            <name></name>
            <id></id>
            <symbol-list>
                <watched-symbol>
                    <quantity></quantity>
                    <security>
                        <symbol></symbol>
                        <symbol-with-type-prefix></symbol-with-type-prefix>
                        <description></description>
                        <asset-type></asset-type>
                    </security>
                    <position-type></position-type>
                    <average-price></average-price>
                    <commission></commission>
                    <open-date></open-date>
                </watched-symbol>
            </symbol-list>
        </created-watchlist>
    </create-watchlist-result>
```

</amtd>

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
create-watchlist-result	Complex	Container for the Result of the CreateWatchlist command being returned
error	String	Error string in case of an error
account-id	String	
created-watchlist	Complex	Contains all the fields for the watchlist that was created
NOTE: The rest of the fields are exactly the same as contained inside the watchlist tag in the result from the GetWatchLists command		
name	String	The name assigned to the given watchlist
id	Integer	Numeric ID that identifies the watchlist
symbol-list	Complex	Container for all the symbols in the watchlist
watched-symbol	Complex	Container for one symbol in the watchlist
quantity	Double	Number of units in the position. Field may have up to 3 digits to the right of the decimal point.
security	Complex	Container for detailed information about the security.
symbol		Contains symbol. For example, **AMTD**.
symbol-with-type-prefix		
description		Contains a description of the symbol. For example "**TD AMERITRADE HLDG CORP COM**"
asset-type	String	Type of asset the symbol represents. It is a one character code where: **E** — Equity or ETF **F** — Mutual Fund **O** — Option **B** — Bond
position-type	String	**LONG** or **SHORT**
average-price	Double	Contains the value calculated by dividing the total cost to acquire the position by the number of units in the position. The field may contain up to six digits to the right of the decimal point.
commission	Double	
open-date	String	Date the "position" was opened

42.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/CreateWatchlist?source=<**#sourceID#**>&watchlistname=<**#PortfolioName#**>&symbollist=<**#SymbolList#**>

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <create-watchlist-result>
        <error></error>
        <account-id>123456789</account-id>
        <created-watchlist>
            <name>Some Portfolio Name Example</name>
            <id>1234567</id>
            <symbol-list>
                <watched-symbol>
                    <quantity>100</quantity>
                    <security>
                        <symbol>DELL</symbol>
                        <symbol-with-type-prefix>DELL</symbol-with-type-prefix>
                        <description>DELL INC COM</description>
                        <asset-type>E</asset-type>
                    </security>
                    <position-type>LONG</position-type>
                    <average-price>20.23</average-price>
                    <commission>9.99</commission>
                    <open-date>2-21-2008</open-date>
                </watched-symbol>
                <watched-symbol>
                    <quantity>200</quantity>
                    <security>
                        <symbol>ABCD</symbol>
                        <symbol-with-type-prefix>ABCD</symbol-with-type-prefix>
                        <description> </description>
                        <asset-type>E</asset-type>
                    </security>
                    <position-type>LONG</position-type>
                    <average-price>23.68</average-price>
                    <commission>0</commission>
                    <open-date>2-22-2008</open-date>
                </watched-symbol>
                <watched-symbol>
                    <quantity>300</quantity>
                    <security>
                        <symbol>EFGH</symbol>
                        <symbol-with-type-prefix>EFGH</symbol-with-type-prefix>
                        <description> </description>
                        <asset-type>E</asset-type>
                    </security>
```

```
                    <position-type>LONG</position-type>
                    <average-price>10.82</average-price>
                    <commission>9.99</commission>
                    <open-date>2-23-2008</open-date>
                </watched-symbol>
                <watched-symbol>
                    <quantity>400</quantity>
                    <security>
                        <symbol>IBM</symbol>
                        <symbol-with-type-prefix>IBM</symbol-with-type-prefix>
                        <description>INTERNATIONAL BUSINESS MACHS COM</description>
                        <asset-type>E</asset-type>
                    </security>
                    <position-type>LONG</position-type>
                    <average-price>0</average-price>
                    <commission>0</commission>
                </watched-symbol>
                <watched-symbol>
                    <quantity>0</quantity>
                    <security>
                        <symbol>ZYZ</symbol>
                        <symbol-with-type-prefix>ZYZ</symbol-with-type-prefix>
                        <description> </description>
                        <asset-type>E</asset-type>
                    </security>
                    <position-type>LONG</position-type>
                    <average-price>0</average-price>
                    <commission>0</commission>
                </watched-symbol>
            </symbol-list>
        </created-watchlist>
    </create-watchlist-result>
</amtd>
```

42.5 CreateWatchlist Errors

Possible Errors

If one of the numeric parameter lists has a non-numeric value, you will get get the following error, with the given parameter filled in

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid value passed in parameter_name request parameter.</error>
</amtd>
```

if there was an error while trying to perform the save of a new watchlist, provided an more explicit error is not returned.

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
```

```
    <error>Error creating watchlist.</error>
</amtd>
```

if the watchlist being created has the same name as an existing watchlist.

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Error requested watchlist name already in use.</error>
</amtd>
```

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

43 EditWatchlist

This service provides a way to Edit a previously saved Watchlist

43.1 EditWatchlist Request

Request URL

https://apis.tdameritrade.com/apps/100/EditWatchlist?source=<**#sourceID#**>&listid=<**#ListID#**>&symbollist=<**#SymbolList#**>

Parameters

You must specify the **listid** and **symbollist** parameters, with at least one symbol, as well as the **source** assigned by TD Ameritrade. Other parameters are available

43.2 EditWatchlist Parameters

The only parameters that are required are the **listid**, **symbollist** and the **source** assigned by TD Ameritrade.

When editing the Watchlist, you need to specify the new values for all the entries in the watchlist. You identify the watchlist being edited with the listid parameter that you get from the GetWatchlists command. The rest of the parameters are the same as in the CreateWatchlist command

OrderStatus Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
[source](#)	The Application ID of the software client assigned by TD Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned	NO	One of the account IDs returned from the [LogIn](#) service. The default is the associated account
watchlistname	The name to assign the watchlist as it will be displayed to the end-user	YES *	Text string for the watchlist name that is not already in use by another watchlist. **CAN BE NULL, but must have at least "watchlistname=" in the request** **NOTE: names are not case sensitive**
listid	Numeric ID that identifies the watchlist	YES	A valid ID as returned by the [GetWatchlists](#) command
symbollist	The list of symbols that will comprise the WatchList	YES	list of symbols
quantitylist	The list of Quantity values for each of the entries in the WatchList.	NO	list of Quantity values. Leave each value as null if not specified.
pricelist	The list of Purchase Price values for each of the entries in the WatchList.	NO	list of Price Values. Leave each value as null if not specified.
monthlist	The list of Purchase Month values for each of the entries in the WatchList.	NO	list of Month Values. Leave each value as null if not specified.
daylist	The list of Purchase Day values for each of the entries in the WatchList.	NO	list of Day Values. Leave each value as null if not specified.
yearlist	The list of Purchase Year values for each of the entries in the WatchList.	NO	list of Year Values. Leave each value as null if not specified.
commissionlist	The list of Commission values for each of the entries in the WatchList.	NO	list of Commission Values. Leave each value as null if not specified.

43.3 EditWatchlist Response

The response to the EditWatchlist request will be in XML format

General Structure of XML Response

```
<?xml version="1.0" ?>
<amtd>
   <result></result>
```

```
    <edit-watchlist-result>
        <account-id></account-id>
        <edited-atchlist>
            <name></name>
            <id></id>
            <symbol-list>
            <watched-symbol>
                <quantity></quantity>
                <security>
                    <symbol></symbol>
                    <symbol-with-type-prefix></symbol-with-type-prefix>
                    <description></description>
                    <asset-type></asset-type>
                </security>
                <position-type></position-type>
                <average-price></average-price>
                <commission></commission>
                <open-date></open-date>
            </watched-symbol>
            .....
            </symbol-list>
        </edited-atchlist>
    </edit-watchlist-result>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.
edit-watchlist-list	Complex	Container for the Result of the EditWatchlist command being returned
account-id	String	
edited-watchlist	Complex	Contains all the fields for the edited watchlist
NOTE: The rest of the fields are exactly the same as contained inside the watchlist tag in the result from the GetWatchLists command		
name	String	The name assigned to the given watchlist
id	Integer	Numeric ID that identifies the watchlist
symbol-list	Complex	Container for all the symbols in the watchlist
watched-symbol	Complex	Container for one symbol in the watchlist
quantity	Double	Number of units in the position. Field may have up to 3 digits to the right of the decimal point.
security	Complex	Container for detailed information about the security.

symbol		Contains symbol. For example, **AMTD**.
symbol-with-type-prefix		
description		Contains a description of the symbol. For example "**TD AMERITRADE HLDG CORP COM**"
asset-type	String	Type of asset the symbol represents. It is a one character code where: **E** — Equity or ETF **F** — Mutual Fund **O** — Option **B** — Bond
position-type	String	**LONG** or **SHORT**
average-price	Double	Contains the value calculated by dividing the total cost to acquire the position by the number of units in the position. The field may contain up to six digits to the right of the decimal point.
commission	Double	
open-date	String	Date the "position" was opened

43.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/EditWatchlist?source=<**#sourceID#**>&listid=<**#ListID#**>&symbollist=<**#SymbolList#**>

Response:


```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <edit-watchlist-result>
        <error></error>
        <account-id>123456789</account-id>
        <edited-watchlist>
            <name>Some Other Test</name>
            <id>1234567</id>
            <symbol-list>
            <watched-symbol>
                <quantity>100</quantity>
                <security>
                    <symbol>DELL</symbol>
                    <symbol-with-type-prefix>DELL</symbol-with-type-prefix>
                    <description>DELL INC COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
                <average-price>11.11</average-price>
                <commission>10</commission>
                <open-date>01-11-2008</open-date>
```

```
            </watched-symbol>
            <watched-symbol>
                <quantity>2</quantity>
                <security>
                <symbol>INTC</symbol>
                    <symbol-with-type-prefix>INTC</symbol-with-type-prefix>
                    <description>INTEL CORP COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
                <average-price>12.12</average-price>
                <commission>20</commission>
                <open-date>02-22-2006</open-date>
            </watched-symbol>
            <watched-symbol>
                <quantity>3</quantity>
                <security>
                    <symbol>T</symbol>
                    <symbol-with-type-prefix>T</symbol-with-type-prefix>
                    <description>AT&T INC COM</description>
                    <asset-type>E</asset-type>
                </security>
                <position-type>LONG</position-type>
                <average-price>13.13</average-price>
                <commission>30</commission>
                <open-date>03-30-2007</open-date>
            </watched-symbol>
            </symbol-list>
        </edited-watchlist>
    </edit-watchlist-result>
</amtd>
```

43.5 EditWatchlist Errors

Possible Errors

If one of the numeric parameter lists has a non-numeric value, you will get get the following error, with the given parameter filled in

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid value passed in parameter_name request parameter.</error>
</amtd>
```

if there was an error while trying to perform the save of the watchlist, provided an more explicit error is not returned.

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Error creating watchlist.</error>
```

```
</amtd>
```

if the new watchlist name is the same as an existing watchlist.

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Error requested watchlist name already in use.</error>
</amtd>
```

If an invalid listid is specified

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Watchlist id not found.</error>
</amtd>
```

if the listid is not passed

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Request did not contain watchlist id.</error>
</amtd>
```

If the Account ID specified is not valid you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```

If the session is invalid, you will get

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```

44 DeleteWatchlist

This service provides a way to delete a previously saved WatchList

44.1 DeleteWatchlist Request

Request URL

https://apis.tdameritrade.com/apps/100/DeleteWatchlist?source=<**#sourceID#**>&listid=<**#ListID#**>

Parameters

You must specify the **listid** parameter, as well as the **source** assigned by TD Ameritrade.

Other parameters are available

44.2 DeleteWatchlist Parameters

The parameters required are the **listid** and the is the **source** assigned by TD Ameritrade.

OrderStatus Request Parameters

PARAMETER	DESCRIPTION	REQUIRED	POSSIBLE VALUES
source	The Application ID of the software client assigned by Ameritrade.	YES	The value is assigned by TD AMERITRADE to the application developer for the specific application
accountid	The Account ID of the account for which data is to be returned	NO	One of the account IDs returned from the LogIn service. The default is the associated account
listid	The ID of the previously saved watchlist that is to be deleted	YES	Any valid ID of a Watchlist saved for the particular login

44.3 DeleteWatchlist Response

The response to the DeleteWatchlist request will be in XML format

General Structure of XML Response

```xml
<?xml version="1.0" ?>
<amtd>
    <result></result>
    <delete-watchlist-result>
        <error></error>
        <account-id></account-id>
        <deleted-watchlist>
            <name></name>
            <id></id>
        </deleted-watchlist>
    </delete-watchlist-result>
</amtd>
```

Response Fields

XML Attribute Name	Type	Definitions
result	String	Contains the overall result for the request. **OK** - indicates the request was successful **FAIL** - indicates the request was unsuccessful.

delete-watchlist-list	Complex	Container for the Result of the DeleteWatchlist command being returned
error	String	Error string in case of an error
account-id	String	
deleted-watchlist	Complex	Containder for the fields identifying the deleted watchlist
name	String	The name assigned to the given watchlist
id	Integer	Numeric ID that identifies the watchlist

44.4 Request/Response Samples

Request:

https://apis.tdameritrade.com/apps/100/DeleteWatchlist?source=<**#sourceID#**>&listid=<**#ListID#**>

Response:

```
<?xml version="1.0" ?>
<amtd>
    <result>OK</result>
    <delete-watchlist-result>
        <error></error>
        <account-id>123456789</account-id>
        <deleted-watchlist>
            <name>Sample WatchList 1</name>
            <id>1234567</id>
        </deleted-watchlist>
    </delete-watchlist-result>
</amtd>
```

44.5 DeleteWatchlist Errors

Possible Errors

if there was an error while trying to perform the command, provided an more explicit error is not returned.

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Error creating watchlist.</error>
</amtd>
```

If an invalid listid is specified

```
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Watchlist id not found.</error>
</amtd>
```

if the listid is not passed


```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Request did not contain watchlist id.</error>
</amtd>
```


If the Account ID specified is not valid you will get


```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Not a valid account for this user.</error>
</amtd>
```


If the session is invalid, you will get


```xml
<?xml version="1.0" ?>
<amtd>
    <result>FAIL</result>
    <error>Invalid Session</error>
</amtd>
```


45 Streaming Data

This section provides the API and protocol needed to communicate with the Streamer Server.

Structural Specification

The data requests is made in the context of a valid session where the user has successfully logged into the TD Ameritrade Trading Platform.

The streaming server responds with continuous data in binary format. When the end of each segment is encountered it is followed by the next segment

Some data may be compressed. It will be indicated as such in the documentation. When the data is compressed, standard **GZIP deflate/inflate** is used. The server performs DEFLATE and the client conducts INFLATE. This is the same method used by IE, ASP, Java and other applications and other programming languages. Refer to http://www.gzip.org/zlib/zlib_faq.html for additional information.

Refer to Data Type Definitions for all datatypes used in the Streaming Protocol

Request URL

All commands to streamer can use either HTTP protocol (port 80) or HTTPS protocol (port 443), and begin with the domain from streamer-url (retrieved from StreamerInfo command). All REQUEST URLs **must** be URL Encoded. For example:

GET **http://ameritrade.streamer.com/***command.....*

or

POST **http://ameritrade.streamer.com/**
DATA *command...../n/n*

NOTE: If using a GET, you have a restriction on the URL length
If using a POST, you MUST have two newline characters (ascii 10) on the end of the posted data

If ACCT_ACTIVITY is one of the streaming requests, then the request MUST BE on a SSL secure connection (HTTPS)

Authentication

All commands to the Streamer Server start with the authentication information, such as username, encoded password, and other parameters. Data is in key=value pair, delimited by &

PARAMETER	DESCRIPTION
!U	account number - corresponds to "**account-id**" returned from LogIn command
W	encoded password - corresponds to "**token**" returned from StreamerInfo command
A=userid	account number - corresponds to "**account-id**" returned from LogIn command (same as the !U parameter)
token	corresponds to "**token**" returned from StreamerInfo command
company	corresponds to "**company**" returned from LogIn command
segment	corresponds to "**segment**" returned from LogIn command
cddomain	corresponds to "**cd-domain-id**" returned from StreamerInfo command
usergroup	corresponds to "**usergroup**" returned from StreamerInfo command
accesslevel	corresponds to "**access-level**" returned from StreamerInfo command
authorized	corresponds to "**authorized**" returned from StreamerInfo command
acl	Customer's entitlement information - corresponds to "**acl**" returned from StreamerInfo command
timestamp	Application server generated timestamp - corresponds to "**timestamp**" returned from StreamerInfo command
appid	Application Server ID - corresponds to "**app-id**" returned from StreamerInfo command

Request Sections

Requests for multiple types of data have to be submitted in one URL. The different data requests are separated by the "|" (bar) character. For the detailed description, refer to the Request Format section

FIRST REQUEST

The first streaming data request for an application needs to have MONOPOLIZE request in it.

Request Sample

The following is a sample for constructing generic request that just has the authentication info, but no actual data requests:

"!U=" + account_number + "&W=" + StreamingToken + "&A=userid=" + account_number + "&token=" + StreamingToken + "&company=" + company + "&segment=" + segment + "&cddomain=" + cddomain + "&usergroup=" + usergroup + "&accesslevel=" + accesslevel + "&authorized=" + authorized + "&acl=" + acl + "&timestamp=" + timestamp + "&appid=" + appId + "|"

URL Result (NOT URL Encoded)

http://ameritrade02.streamer.com/!U=123456789&**W**=b3cb690339ba220bde92c9f42e4e75bbaf5615db&**A=userid**=123456789&**token**=b3cb690339ba220bde92c9f42e4e75bbaf5615db&**company**=AMER&**segment**=UAMER&**cddomain**=A000001111111111&**usergroup**=ACCT&**accesslevel**=ACCT&**authorized**=Y&**acl**=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&**timestamp**=1143132528988&**appid**=testapp1|

URL Result (URL Encoded)

http://ameritrade02.streamer.com/!**U**%3D123456789%26**W**%3Db3cb690339ba220bde92c9f42e4e75bbaf5615db%26**A**%3D**userid**%3D123456789%26**token**%3Db3cb690339ba220bde92c9f42e4e75bbaf5615db%2**6company**%3DAMER%26**segment**%3DUAMER%26**cddomain**%3DA000001111111111%26**usergroup**%3DACCT%26**accesslevel**%3DACCT%26**authorized**%3DY%26**acl**%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26**timestamp**%3D1143132528988%26**appid**%3Dtestapp1|

45.1 Debugging

A useful practice during the development cycle is to compare the URL generated by your application software to one generated by a program that is already known to work. The simplest way to do that is to look at the log files generated by the TESTDLL sample app that is included with the DLL/ActiveX components.

45.2 Programming Tips

Java Developers - Use the DataInputStream, and the GetUTF, GetByte, GetInt, GetShort, etc. functions to decode data.
Strings are captured automatically: the length, and then the string as UTF characters.

.NET Developers - All numeric values must be loaded and reversed as they are read from the response stream. ("big-endian" to "little-endian"). For example, GetBytes(4) to get values: 00 00 00 04, save the bytes into an array, reverse the array

(Array.Reverse(myArray)), and then use BitConverter class to convert the array into an Int32 value -- in this case, 4.

Strings are handled like above: bytes (a byte array) = binaryStream.GetBytes(2) -> Array.Reverse(bytes) -> int len = BitConverter.GetInt16 (bytes, 0). To load the string, convert the characters using System.Text.Encoding.ASCII.GetString(bytes, len).

Float values: grab the 4 bytes, reverse them, and use BitConverter's GetSingle function

45.3 Data Type Definitions

KEYWORD	DESCRIPTION	SIZE/FORMAT
Integers		
Byte	Byte-length integer	8-bit two's complement
Short	Short integer	16-bit two's complement
Int	Integer	32-bit two's complement
Long	Long integer	64-bit two's complement
Real numbers		
Float	Single-precision floating point	32-bit IEEE 754
Double	Double-precision floating point	64-bit IEEE 754
Other types		
Char	A single character	16-bit Unicode character
Boolean	A Boolean value (true or false)	true or false, appears in the datastream as an 8-bit byte, where 0 is FALSE and any other value is TRUE.

45.4 Service IDs

The following table lists the services available from Streamer Server. Service Name is sent in the request to Streamer Server, while SID (Streaming ID) is used to identify response data. There are two types of connections to the Streamer Server, Snapshot and Streaming. Some of the services support snapshot, some streaming, and others support both.

Streaming Services

SID = Streaming ID
S = Streaming
N = Snapshot

SERVICE NAME	SID	COMMAND	S/N	DESCRIPTION
QUOTE	1	QUOTE	S, N	Level I quotes (**EQUITIES**)
TIMESALE	5	TIMESALE	S	Last sale quotes
RESPONSE	10	n/a (response)	N	General response
AUTHORIZER	16	AUTHORIZER	N	Entitlement
OPTION	18	OPTION	S	level 1 quotes (**OPTIONS**)
ACTIVES_ISLAND	14	ACTIVES_ISLAND	S	INET actives (deprecated)
ACTIVES_NYSE	23	ACTIVES_NYSE	S	NYSE actives
ACTIVES_NASDAQ	25	ACTIVES_NASDAQ	S	NASDAQ actives
ACTIVES_OTCBB	26	ACTIVES_OTCBB	S	OTCBB actives
NEWS	27	NEWS	S	News
NEWS_HISTORY	28	n/a	N	Historical news for the day
OPTION_LOOKUP	33	n/a	N	Option symbol lookup
ACTIVES_OPTIONS	35	ACTIVES_OPTIONS	S	Options actives
ADAP_NASDAQ	62	ADAP_NASDAQ	S	NASDAQ aggregated data
NYSE_BOOK	81	NYSE_BOOK	S, N	NYSE level 2 book
NYSE_CHART	82	NYSE_CHART	S, N	NASDAQ chart
NASDAQ_CHART	83	NASDAQ_CHART	S, N	NYSE chart
INDEX_CHART	85	INDEX_CHART	S, N	Indices chart
OPRA_BOOK	84	OPRA	S	Options level 2 book
TOTAL_VIEW	87	TOTAL_VIEW	S	NASDAQ TotalView and Level II data

SERVICE NAME	SID	COMMAND	S/N	DESCRIPTION
ACCT_ACTIVITY	90	ACCT_ACTIVITY	S	Account Activity Notifications
STREAMER_SERVER	100	n/a (response)	S, N	Streamer response

45.5 Request Format

Request Format

S = Service Name (QUOTE, LEVELII, OPTION etc)
C = GET (snapshot) or **SUBS** (streaming)
P = Parameters (optional)
T = Field #'s (optional)

Multiple commands are delimited by "|".

Example: **S=***QUOTE*&**C=***SUBS*&**P=***DELL+AMTD*&**T=** *0+1+2+3+8+10+11+12+13+15+16+19+20*

IMPORTANT: There can only be ONE streaming URL request at a time. Requests for multiple types of data, and thus multiple sections, have to be submitted in one URL.

For example if you want to request quotes on DELL and some Indices as well as NASDAQ Level II for DELL, you would request **S=QUOTE&C=SUBS&P=DELL+$SPX.X+$INX+$INDU&T=0+1+2+3+8+10+11+12+13+15+16+19+20|S=TOTAL_VIEW&C=SUBS&P=DELL>L2&T=0+1+2+3|**

If you have one streaming request and submit make another request, the first request will be terminated.

NOTE: symbol formats for some symbols are different for streaming data vs snapshot and what is entered by the user on the web site. For example, streaming servers would expect **BRK/A,** while the snapsnot quote servers would expect **BRK.A**. The application should always accept the symbol in the "Snapshot" format and convert for streaming as needed, since that is what the end-user would enter on the TD AMERITRADE web site. Here are some sample conversions:

User Entered Symbol => Actual Symbol submitted to stream server

BRK.A => BRK/A (class A - can be other than A)
MET-B => METpB (preferred with lower case p. Can be other than B)
AXG+ => AXG/WS (warrants - always /WS suffix instead of the + on the end)
QGP+A=>QGP/WS/A (warrants with a Class)
PFL^=>PFLr (Rights)

NOTE: OPTIONS SYMBOL FORMAT:

PRIOR TO FEB 05, 2010
+ABCDE => ABC.DE (option) - last 2 characters of the option symbol go after the period.
So, +CIQ => C.IQ

AFTER FEB 12, 2010

AMTD_092209P12.5 gets submitted unchanged, but end user should NEVER see this symbol. Option Description should always be shown to the user.

45.6 Response Format

Streamer response returns in bytes. All streaming responses begin with the following fields:

Common Streaming Response Header

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message length	Short		2	Streaming ID Length
SID	Short		2	ID from Server IDs list identifying the response content

Snapshot responses begin with the following fields:

Snapshot Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	N	1	Snapshot
Snapshot ID length	Short		2	next string length
Snapshot ID	String			Snapshot ID - usually ASCI version of the Streamable ID (SID)
Payload Length	Int		4	Payload length
Payload	Byte[]			The actual data (Starting with SID)
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

Heartbeat/Timestamp responses begin with the following fields:

Heartbeat Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	H	1	Heartbeat
Sub Type	Byte	T or H	1	T indicates next field is time; H indicates no time value is given
Timestamp	Long		8	time in milliseconds from 00:00:00 UTC on January 1, 1970

45.7 MONOPOLIZE

The MONOPOLIZE server ID is used to identify a new streaming connection client. Other client applications connected with the same account will be notified to disconnect. This allows for proper notification to disconnect for the other applications, instead of them just getting the connections terminated.

The monopolize request should be attached to the first streaming connection made by an application. It should NOT be sent with every streaming request. Just the first one.

45.7.1 Request

Parameters

S = QUOTE
C = MONOPOLIZE

Request Example

!U=123456789&W=104256ff2f0aa06195fdc225908c05b34450703b&A=userid=123456789&token=104256ff2f0aa06195fdc225908c05b34450703b&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143663111397&appid=testapp1|**S=QUOTE&C=MONOPOLIZE**

Request Example (URL Encoded)

!U%3D123456789%26W%3D104256ff2f0aa06195fdc225908c05b34450703b%26A%3Duserid%3D123456789%26token%3D104256ff2f0aa06195fdc225908c05b34450703b%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143663111397%26appid%3Dtestapp1%7C**S**%3D**QUOTE**%26**C%**3D**MONOPOLIZE**

Request Example with a Level I request

!U=123456789&W=104256ff2f0aa06195fdc225908c05b34450703b&A=userid=123456789&token=104256ff2f0aa06195fdc225908c05b34450703b&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143663111397&appid=testapp1**|S=QUOTE&C=MONOPOLIZE|**
S=QUOTE&C=SUBS&P=DELL+$SPX.X+$INX+$INDU&T=0+1+2+3+8+10+11+12+13+15+16+19+20

45.7.2 Response

MONOPOLIZE Response will be a STREAMER SERVER message with MONOPOLIZE_ACK return code

45.8 Account Activity

The ACCT_ACTIVITY server ID is used to request streaming updates for one or more accounts associated with the logged in User ID. Common usage would involve issuing the OrderStatus request to get all transactions for an account, and subscribing to ACCT_ACTIVITY to get any updates.The messages returned for the ACCT_ACTIVITY subscription may not contain all the fields that you may be interested in for a transaction. Upon receipt of such a message, you may want to issue another OrderStatus request for the specific orderid referenced in the message in order to get the full update.

In order to subscribe to ACCT_ACTIVITY, you must first request a MessageKey for the accounts that you wish to get the updates on.

NOTE: If ACCT_ACTIVITY is one of the streaming requests, then the request MUST BE on a SSL secure connection (HTTPS)

45.8.1 Request

Parameters

S = ACCT_ACTIVITY
C = SUBS
P = Token returned from the MessageKey command
T = 0+1+2+3

Request Example

!U=#chen5582&W=acd04d27e1ebe685fc3a3d4286103fcdebefa589&A=userid=123456789&token=acd04d27e1ebe685fc3a3d4286103fcdebefa589&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143136132906&appid=testapp1|
S=ACCT_ACTIVITY&C=SUBS&P=adfl457kg2342s9045U&T=0+1+2+3

Request Example (URL Encoded)

!U%3D%23chen5582%26W%3Dacd04d27e1ebe685fc3a3d4286103fcdebefa589%26A%3D

userid%3D123456789%26token%3Dacd04d27e1ebe685fc3a3d4286103fcdebefa589%26c
ompany%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26use
rgroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRES
GKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143136132906%26appid
%3Dtestapp1%7C**S**%3D**ACCT_ACTIVITY&C**%26**C**%3D**SUBS**%26**P**%3D
adfl457kg2342s9045U%26**T**%3D**0**%2B**1**%2B**2**%2B**3**

NOTE: If ACCT_ACTIVITY is one of the streaming requests, then the request MUST BE on a SSL secure connection (HTTPS)

All 4 field numbers should be requested

ACCT_ACTIVITY Request Fields

FIELD NAME	COLUMN NUMBER	DATA TYPE	DESCRIPTION
SUBSCRIPTION KEY	0	String	
ACCOUNT #	1	String	Account # that the message is for
MESSAGE TYPE	2	String	XML Document Tag name describing the message
MESSAGE DATA	3	String	XML Message data

45.8.2 Response

Account Activity Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message length
SID	Short	0x005A	2	SID for Account Activity
Column #	Byte	0x00	1	Field ID for Subscription Key
Subscription Key length	Short		2	Length of Subscription Key string
Subscription Key	String			Subscription Key value
Column #	Byte	0x01	1	Field ID for Account #
Account # Length	Short		2	Length of the Account #

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Account #	String			Account # Value
Column #	Byte	0x02	1	Field ID for Message Type
Message Type Length	Short		2	Length of the Message Type String
Message Type	String			Message Type Value **(Refer to table below)**
Column #	Byte	0x03	1	Field ID for XML Message Data
Message Data Length	Short		2	Length of the XML Message Data
Core Message Data	String			The core data for the message. Either XML Message data describing update, NULL in some cases, or plain text in case of ERROR
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

Message Type - Possible values

MESSAGE TYPE	DESCRIPTION
SUBSCRIBED	First message returned when a subscription for ACCT_ACTIVITY is returned. **NOTE:** NO XML DATA will be returned with this message type
ERROR	Indicates some sort of error. **NOTE:** For this message type, the core data would contain error type. It will NOT be XML. Currently, possible values are: **INVALID_KEY** - If the Message Key used in the subscription is invalid **EXPIRED_KEY** - If the key is more than 24 hours old at the time of subscription (can also happen overnight). You need to get a new Key from the MessageKey Command **SYSTEM_ERROR** - Some internal server error
BrokenTrade	After an order was filled, the trade is reversed or "Broken" and the order is changed to Canceled.
ManualExecution	The order is manually entered (and filled) by the broker. Usually due to some system issue.
OrderActivation	A Stop order has been Activated

MESSAGE TYPE	DESCRIPTION
OrderCancelReplaceRequest	A request to modify an order (Cancel/Replace) has been received (You will also get a UROUT for the original order)
OrderCancelRequest	A request to cancel an order has been received
OrderEntryRequest	A new order has been submitted
OrderFill	An order has been completely filled
OrderPartialFill	An order has been partial filled
OrderRejection	An order was rejected
TooLateToCancel	A request to cancel an order has been received but the order cannot be canceled either because it was already canceled, filled, or for some other reason
UROUT	Indicates "You Are Out" - that the order has been canceled

45.8.3 XML Messages

NOTE: The response messages are in XML format and contain some fields that may not be relevant. We are only documenting the fields that are relevant. Any other fields should be ignored during processing.

Majority portion of the XML message will be the same regardless of the message type. The extra fields that will differ for each message type are documented explicitly at the end of the XML structure

General Structure of XML Response

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<OrderCancelReplaceRequestMessage xmlns="urn:xmlns:api.tdameritrade.com">
    <OrderGroupID>
        <Firm></Firm>
        <Branch></Branch>
        <ClientKey></ClientKey>
        <AccountKey></AccountKey>
        <SubAccountType></SubAccountType>
        <CDDomainID></CDDomainID>
    </OrderGroupID>
    <ActivityTimestamp></ActivityTimestamp>
    <Order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
    ="....">
        <OrderKey></OrderKey>
        <Security>
            <CUSIP></CUSIP>
            <Symbol></Symbol>
            <SecurityType></SecurityType>
            <SecurityCategory></SecurityCategory>
```

```
            <ShortDescription></ShortDescription>
            <SymbolUnderlying></SymbolUnderlying>
        </Security>
        <OrderPricing xsi:type="....">
            <Last></Last>
            <Ask></Ask>
            <Bid></Bid>
            <Limit></Limit>
        </OrderPricing>
        <OrderType></OrderType>
        <OrderDuration></OrderDuration>
        <OrderEnteredDateTime></OrderEnteredDateTime>
        <OrderInstructions></OrderInstructions>
        <OriginalQuantity></OriginalQuantity>
        <SpecialInstructions></SpecialInstructions>
        <Discretionary></Discretionary>
        <OrderSource></OrderSource>
        <Solicited></Solicited>
        <MarketCode></MarketCode>
        <Capacity></Capacity>
        <GoodTilDate></GoodTilDate>
    </Order>
    <LastUpdated></LastUpdated>
    <PendingCancelQuantity></PendingCancelQuantity>
    <OriginalOrderId></OriginalOrderId>
</OrderCancelReplaceRequestMessage>
```

General Account Activity XML Message Response Fields

XML Attribute Name	Type	Definitions
*Message	Complex	The main container will be named the same as the Message Type, followed by "Message". For example OrderCancelReplaceRequestMessage
OrderGroupID	Complex	
Firm	Integer	Internal
Branch	Integer	Internal
ClientKey	String	Internal
AccountKey	String	Account # of the account in which the order is in
SubAccountType	String	One of the following values: **Cash** **Margin** **Income** **Short** **DVP/RVP** **Dividend**
ActivityTimestamp	String	
Order	Complex	
OrderKey	String	The Order Number of the order for which the activity message is sent

Security	Complex	
CUSIP	String	
Symbol	String	
SecurityType	String	One of the following values: **Common Stock** **Preferred Stock** **Convertible Preferred Stock** **Rights** **Warrant** **Mutual Fund** **Call Option** **Put Option** **Bank Call Option** (Other values are also possible)
SecurityCategory	String	One of the following values: **Equity** **Option** **Hybrid** **Rights** **Warrant** **Mutual Fund** **Fixed Income** **Commercial Paper** **Other**
ShortDescription	String	Description of the Symbol. Particularly important for option symbols
SymbolUnderlying	String	If the symbol is an option, then this field is the underlying symbol for that option
OrderPricing	Complex	
Last	Float	
Ask	Float	
Bid	Float	
Limit	Float	Only returned for orders where LIMIT price is relevant
Method	String	The Method for Trailing Stop orders: **Points** or **Percent**
Amount	Float	The amount used for Trailing Stop. If trails 2%, it would be 2.
OrderType	String	One of the following values: **MarketLimit** **Stop** **Stop Limit** **On Close** **Trailing Stop** **Excercised**
OrderDuration	String	**Day** **Good Till Date** **Good Till Cancel** **Good Till Night** **Fill Or Kill** **Immediate Or Cancel** **End Of Month** **End Of Week** **Market At Open** **Next End Of Month** **NOTE:** Refer to MarketCode as well. OrderDuration and MarketCode together translate to the usually entered expiration values
OrderEnteredDateTime	String	
OrderInstructions	String	**Buy** **Sell** **Short** **Short ExemptBuy To Cover** **Exchange** **Exercise**
OriginalQuantity	Float	

SpecialInstructions	Complex	Container for various special instructions flags
AllOrNone	Integer	Value will either be 1, or the tag will not be returned
DoNotReduceIncreaseFlag	String	Value will either be "**Do Not Reduce**", or tag will not be returned
NotHeld	Integer	Value will either be 1, or the tag will not be returned
TryToStop	Integer	Value will either be 1, or the tag will not be returned
Discretionary	Integer	Internal
OrderSource	String	Internal
Solicited	Boolean	Internal
MarketCode	String	**Normal** **Seamless** **Extended** **PreSession** **German** **NOTE:** Seamless refers to DAY+Ext and GTC+Ext **AM** orders will be OrderDuration=**DAY**, MarketCode= **PreSession** **PM** orders will be OrderDuration=**GTC**, MarketCode= **Extended**
Capacity	String	Internal
GoodTilDate	String	Expiration date of the order
ActivationPrice	Float	The price at which the order was activated Only returned for Message Type: **OrderActivation**
LastUpdated	String	The date/time when the order was last updated Only returned for Message Type: **OrderCancelReplaceRequest, OrderEntryRequest**
OriginalOrderId	Integer	The Order ID of the original order being replaced. Only returned for Message Type: **OrderCancelReplaceRequest**
PendingCancelQuantity	Float	The Quantity of the order that being canceled. Only returned for Message Type: **OrderCancelReplaceRequest, OrderCancelRequest**
CancelledQuantity	Float	The Quantity of the order that was canceled. Only returned for Message Type: **UROUT**
RejectCode	Integer	The code indicating why the order was rejected Only returned for Message Type: **OrderRejection**
RejectReason	String	Plain text reason why the order was rejected Only returned for Message Type: **OrderRejection**
ReportedBy	String	The ID of the person reporting the order rejection Only returned for Message Type: **OrderRejection**
RemainingQuantity	Float	The size of the order that remains to be filled Only returned for Message Type: **OrderPartialFill**
OrderCompletionCode	String	Description of completion. For example: Normal Completion Only returned for Message Type: **OrderFill** and **TooLateToCancel**

Charges	Complex	Container for multiple "Charge" entries
Charge	Complex	Container for a single Charge entry
Type	String	The description of the type of charge
Amount	Float	The amount of the chart
OrderAssociation	Complex	Container for fields describing relationship of multiple legs of a Complex Option order
Type		**Complex** - for Complex Options
AssociatedOrders	Complex	Container for Associated order description fields
OrderKey		Order ID of the associated Order (child order)
Relationship		**Parent, Child, Other Leg, Contingent, Enhanced, Cancel Other**
ComplexOrderType		**Spread** **Straddle** **Strangle** **Buy-Writes** **Combo** **Exercise** **Swap** **OCA** **Vertical** **Back Ratio** **Calendar** **Diagonal** **Butterfly** **Condor** **Iron Condor** **Vertical Roll** **Covered Stock** **Collar With Stock** **Collar Synthetic** **Double Diagonal** **Custom**
CreditOrDebit		**Credit** or **Debit**
All subsequent tags are inside the ExecutionInformation container		
ExecutionInformation	Complex	Container for the Execution information **All subsequent tags are inside this container** only returned for Message Types: **OrderPartialFill** and **OrderFill**
Type	String	**Bought** **Sold** **Sold Short** **Sold Short Exempt** **Buy To Cover** **Exchange** **Exercised**
Timestamp	String	
Quantity	Float	
ExecutionPrice	Float	
AveragePriceIndicator	Boolean	**true/false -** Indicates if the price represents an average price of multiple fills
LeavesQuantity	Float	The size of the order that remains to be filled
ID	String	Execution ID

45.8.4 Sample Response

In the following sample, data is shown in mixed format. Bytes represented in alpha are shown as alphabetic characters. The {XX} represents non-alpha data where XX is the hex value of the byte.

DATA	TYPE	DESCRIPTION
S	1 byte	Identifies a streaming record
{05}{CD}	2 bytes	Length of message through delimiter
{00}{5A}	2 bytes	{00}{5A} indicates ACCT_ACTIVITY
{00}	1 byte	FieldID: Message Key
{00}{41}	2 bytes	Message Key length (65)
a19351dd6c64a4e786376b435 842a06fb889ae091e1c7863428 b66385ced04e3f	variable	Message Key Value
{01}	1 byte	FieldID: Account #
{00}{09}	1 bytes	Account # length (9)
150340610	variable	Account # value (150340610)
{02}	1 byte	FieldID: Message Type
{00}{11}	2 bytes	Message Type Length (17(
OrderEntryRequest	variable	Message Type (OrderEntryRequest)
{03}	1 byte	FieldID: XML Message
{05}{64}	2 bytes	XML Message Length (1380)
XML Data String (not shown)	variable	XML Message - 1380 bytes. Not shown for clarity
{FF}	1 byte	FieldID: final field
{0A}	1 byte	End of record

The following is the XML Message Returned


```
<?xml version="1.0" encoding="iso-8859-1" ?>
<OrderEntryRequestMessage xmlns="urn:xmlns:api.tdameritrade.com">
    <OrderGroupID>
        <Firm>111</Firm>
        <Branch>111</Branch>
        <ClientKey>123456789</ClientKey>
        <AccountKey>150340610</AccountKey>
        <SubAccountType>Margin</SubAccountType>
        <CDDomainID>A000000001234567</CDDomainID>
    </OrderGroupID>
    <ActivityTimestamp>2008-03-28T14:12:06Z</ActivityTimestamp>
    <Order xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type
```

```
    ="EquityOrderT">
        <OrderKey>2148111147</OrderKey>
        <Security>
            <CUSIP>458140100</CUSIP>
            <Symbol>INTC</Symbol>
            <SecurityType>Common Stock</SecurityType>
            <SecurityCategory>Equity</SecurityCategory>
        </Security>
        <OrderPricing xsi:type="LimitT">
            <Last>21.33000</Last>
            <Ask>21.34000</Ask>
            <Bid>21.33000</Bid>
            <Limit>21.09000</Limit>
        </OrderPricing>
        <OrderType>Limit</OrderType>
        <OrderDuration>Day</OrderDuration>
        <OrderEnteredDateTime>2008-03-28T14:11:42Z</OrderEnteredDateTime>
        <OrderInstructions>Buy</OrderInstructions>
        <OriginalQuantity>150.000</OriginalQuantity>
        <SpecialInstructions></SpecialInstructions>
        <Discretionary>0</Discretionary>
        <OrderSource>Web</OrderSource>
        <Solicited>false</Solicited>
        <MarketCode>Seamless</MarketCode>
        <Capacity>Agency</Capacity>
        <Charges>
            <Charge>
                <Type>Commission</Type>
                <Amount>8.000</Amount>
            </Charge>
            <Charge>
                <Type>SEC Fee</Type>
                <Amount>0.000</Amount>
            </Charge>
        </Charges>
    </Order>
    <LastUpdated>2008-03-28T14:12:06Z</LastUpdated>
</OrderEntryRequestMessage>
```


45.9 Level I Quote

The QUOTE server ID is used to request Level I quotes on stocks, indices and mutual funds. For options, you would use the OPTION Service ID

45.9.1 Request

Parameters

S = QUOTE
C = SUBS or GET
P = symbol+symbol+symbol ..
T = fieldNum+fieldNum+ .

Request Example

!U=#chen5582&W=acd04d27e1ebe685fc3a3d4286103fcdebefa589&A=userid=123456789&token=acd04d27e1ebe685fc3a3d4286103fcdebefa589&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143136132906&appid=testapp1|
S=QUOTE&C=SUBS&P=DELL+$SPX.X+$INX+$INDU&T=0+1+2+3+8+10+11+12+13+15+16+20

Request Example (URL Encoded)

!U%3D%23chen5582%26W%3Dacd04d27e1ebe685fc3a3d4286103fcdebefa589%26A%3Duserid%3D123456789%26token%3Dacd04d27e1ebe685fc3a3d4286103fcdebefa589%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143136132906%26appid%3Dtestapp1%7C**S**%3D**QUOTE**%26**C**%3D**SUBS**%26**P**%3D**DELL**%2B**%24SPX.X**%2B**%24INX**%2B**%24INDU**%26**T**%3D**0**%2B**1**%2B**2**%2B**3**%2B**8**%2B**10**%2B**11**%2B**12**%2B**13**%2B**15**%2B**16**%2B**20**

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

IMPORTANT: There can only be ONE streaming URL request at a time. Requests for multiple types of data, and thus multiple sections, have to be submitted in one URL.

Level I Quote Request Fields

FIELD NAME	COLUMN NUMBER	DATA TYPE	DESCRIPTION
SYMBOL	0	String	
BID	1	Float	
ASK	2	Float	
LAST	3	Float	
BIDSIZE	4	Int	
ASKSIZE	5	Int	
BIDID	6	Char	
ASKID	7	Char	
VOLUME	8	Long	
LASTSIZE	9	Int	
TRADETIME	10	Int	# of seconds since midnight, eastern time

FIELD NAME	COLUMN NUMBER	DATA TYPE	DESCRIPTION
QUOTETIME	11	Int	# of seconds since midnight, eastern time
HIGH	12	Float	
LOW	13	Float	
TICK	14	Char	
CLOSE	15	Float	
EXCHANGE	16	Char	
MARGINABLE	17	Boolean	
SHORTABLE	18	Boolean	
QUOTEDATE	22	Int	# of days sine 1/1/1970
TRADEDATE	23	Int	# of days sine 1/1/1970
VOLATILITY	24	Float	
DESCRIPTION	25	String	
TRADE_ID	26	Char	
DIGITS	27	Int	max # of digits to display after the decimal
OPEN	28	Float	
CHANGE	29	Float	
WEEK_HIGH_52	30	Float	
WEEK_LOW_52	31	Float	
P_E_RATIO	32	Float	
DIVIDEND_AMT	33	Float	
DIVIDEND_YIELD	34	Float	
NAV	37	Float	
FUND	38	Float	
EXCHANGE_NAME	39	String	
DIVIDEND_DATE	40	String	
The following will be available soon. Currently only in the GET request			
LAST_MARKET_HOURS	41	Float	
LASTSIZE_MARKET_HOURS	42	Int	

FIELD NAME	COLUMN NUMBER	DATA TYPE	DESCRIPTION
TRADEDATE_MARKET_HOURS	43	Int	
TRADETIME_MARKET_HOURS	44	Int	
CHANGE_MARKET_HOURS	45	Float	
IS_REGULAR_MARKET_QUOTE	46	Boolean	
IS_REGULAR_MARKET_TRADE	47	Boolean	

45.9.2 Response

Level I STREAMING Quote Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message length
SID	Short	0x0012	2	SID for QUOTE
Column #	Byte	0x00	1	Symbol Column always returned
Symbol length	Short		2	Length of symbol string
Symbol	String	Symbol		Symbol value
REPEATING DATA				
Column #	Byte	0x01 – 0x28	1	Next field as defined in the Level I Quote Request
Column Value	Depends	depends		Depends on column #, string response fields will be preceded with a 'short' length field (2 bytes) indicating the length of the string that follows
REPEATS UNTIL DELIMITER IS ENCOUNTERED				
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

Level I SNAPSHOT Quote Response

Refer to SNAPSHOT RESPONSE FORMAT for the core format for the response. The PAYLOAD is documented in the table below

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
REPEATING BLOCK OF DATA FOR EACH SYMBOL - Blocks separated by semicolon delimiter				
SID	Short	0x0012	2	SID for QUOTE
Column #	Byte	0x00	1	Symbol Column always returned
Symbol length	Short		2	Length of symbol string
Symbol	String	Symbol		Symbol value
REPEATING DATA				
Column #	Byte	0x01 – 0x28	1	Next field as defined in the Level I Quote Request
Column Value	Depends	depends		Depends on column #, string response fields will be preceded with a 'short' length field (2 bytes) indicating the length of the string that follows
REPEATS FIELD DATA UNTIL DELIMITER IS ENCOUNTERED				
Delimiter	Byte	0x3B	1	Semicolon - separator between symbols

45.9.3 Sample Response

In the following sample, data is shown in mixed format. Bytes represented in alpha are shown as alphabetic characters. The {XX} represents non-alpha data where XX is the hex value of the byte.

DATA	TYPE	DESCRIPTION
{'H' 'T'}	2 bytes	HEARTBEAT
{00 00 01 0c c5 5b 00 9a}	8 bytes	timestamp
'N'	1 byte	SNAPSHOT
{00 03}	2 bytes	next field length
'100'	variable	next field value (STREAMER_SERVER_ID)
{00 00 00 0e}	4 bytes	message length
{00 0a}	2 bytes	SID (RESPONSE)

{00}	1 byte	field #1 (Service ID)
{00 64}	2 bytes	value
{01}	1 byte	field #2 (Return code)
{00 12}	2 bytes	value (STREAMER_SERVER_ID)
{02}	1 byte	field #3 (Description)
{00 03}	2 bytes	length of following field
'118'	variable	value
S	1 byte	Identifies a streaming record
{00}{62}	2 bytes	Length of message through delimiter
{00}{01}	2 bytes	{00}{01} indicates L1 quote
{00}	1 byte	FieldID: symbol
{00}{04}	2 bytes	symbol length
MSFT	variable	symbol value
{01}	1 byte	FieldID: Bid
{41}{BB}{EB}{85}	4 bytes	float (23.49)
{02}	1 byte	FieldID: Ask
{41}{BC}{00}{00}	4 bytes	float (23.5)
{03}	1 byte	FieldID: Last
{41}{BC}{00}{00}	4 bytes	float (23.5)
{04}	1 byte	FieldID: Bid Size
{00}{00}{00}{48}	4 bytes	integer (72)
{05}	1 byte	FieldID: Ask Size
{00}{00}{02}{49}	4 bytes	integer (585)
{06}	1 byte	FieldID: Bid exchange
{00}{51}	2 bytes	the exchange
{07}	1 byte	FieldID: Ask exchange
{00}{51}	2 bytes	the exchange
{08}	1 byte	FieldID: Volume
{00}{00}{00}{00}{02}{0C}{4C}{09}	8 bytes	long integer (34360329)
{09}	1 byte	FieldID: Last Volume
{00}{00}{00}{03}	4 bytes	integer (3)
{0B}	1 byte	FieldID: Timestamp
{00}{00}{C2}{75}	4 bytes	# of seconds since midnight (49781 or 1:49:41 pm)
{0C}	1 byte	FieldID: High
{41}{BD}{33}{33}	4 byte	float (23.65)
{0D}	1 byte	FieldID: Low
{41}{BB}{0A}{A6}	4 byte	float (23.3802)

{0E}	1 byte	FieldID: Bid Tick
{00}{44}	2 bytes	D - down
{0F}	1 byte	FieldID: prev close
{41}{BB}{C2}{F8}	4 bytes	float (23.4702)
{19}	1 byte	FieldID: Company Name
{00}{12}	2 bytes	length of string (18)
MICROSOFT{20}CORP{20}COM	variable	the company name string
{FF}	1 byte	FieldID: final field
{0A}	1 byte	End of record

45.10 OPTION Quote

The QUOTE server ID is used to request Level I quotes on Options

45.10.1 Request

Parameters

S = OPTION
C = SUBS
P = symbol+symbol+symbol ..
T = fieldNum+fieldNum+ .

Request Example

NOTE: Option symbols change based on time. Do not use the symbols in this example for testing application software. Use current symbols for testing.

!U=123456789&W=104256ff2f0aa06195fdc225908c05b34450703b&A=userid=123456789&token=104256ff2f0aa06195fdc225908c05b34450703b&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143663111397&appid=testapp1|**S=OPTION&C=SUBS&P=AMTD_092209P12.5+QQQQ_092210P30+SPY_032110P80&T=0+1+2+3+4+5+6+7+8+11+12+16+17+28**

Request Example (URL Encoded)

!U%3D123456789%26W%3D104256ff2f0aa06195fdc225908c05b34450703b%26A%3Duserid%3D123456789%26token%3D104256ff2f0aa06195fdc225908c05b34450703b%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143663111397%26appid%3Dtestapp1%7C**S**%3D**OPTION**%26**C%**3D**SUBS**%26**P**%3D**AMTD_092209P12.5**%2B**QQQQ_092210P30**%2B**SPY_032110P80**%26**T**%3D**0**%2B**1**%2B**2**%2B**3**%2B**4**%2B**5**%2B**6**%2B**7**%2B**8**%2B**11**%2B**12**%2B**16**%2B**17**%2B**28**

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

Level I OPTION Request Fields

FIELD	VALUE	TYPE	DESCRIPTION
SYMBOL	0	String	
CONTRACT	1	String	Description of the option
BID	2	Float	
ASK	3	Float	
LAST	4	Float	
HIGH	5	Float	
LOW	6	Float	
CLOSE	7	Float	
VOLUME	8	Long	
OPEN_INTEREST	9	Int	
VOLATILITY	10	Float	
QUOTETIME	11	Int	
TRADETIME	12	Int	
IN_THE_MONEY	13	Float	
QUOTEDATE	14	Int	
TRADEDATE	15	Int	
YEAR	16	Int	
MULTIPLIER	17	Float	
OPEN	19	Float	
BIDSIZE	20	Int	
ASKSIZE	21	Int	
LASTSIZE	22	Int	
CHANGE	23	Float	
STRIKE	24	Float	
CONTRACT_TYPE	25	Char	**C** or **P** (**NOTE:** 2 bytes - Unicode) - **NOT USED ANYMORE**
UNDERLYING	26	String	Underlying Symbol. **NOTE:** be careful with Index options, you'll get SPX instead of $SPX.X, for example

FIELD	VALUE	TYPE	DESCRIPTION
MONTH	27	Int	
NOTE	28	String	Contains description of the DELIVERABLES for non-standard options. Standard options will not have this field returned
TIMEVALUE	29	Float	
DAYS_TO_EXP	31	Int	Days To Expiration
DELTA_INDEX	32	Float	
GAMMA_INDEX	33	Float	
THETA_INDEX	34	Float	
VEGA_INDEX	35	Float	
RHO_INDEX	36	Float	

45.10.2 Response

Level I OPTION Response format is exactly the same Level I (QUOTE) Response for stocks.

45.10.3 Sample Response

Level I OPTION Sample Response format is exactly the same Level I (QUOTE) Sample Response for stocks, except different field IDs will be returned

45.11 Time & Sales

The TIMESALE server ID is used to request Time & Sales data for all supported symbols

45.11.1 Request

Parameters

S = TIMESALE
C = SUBS
P = symbol+symbol+symbol ..
T = fieldNum+fieldNum+ .

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

Time & Sale Request Fields

FIELD	VALUE	TYPE

SYMBOL	0	String
TIME	1	Int
PRICE	2	Float
VOLUME	3	Long
SEQUENCE	4	Int
UPDATED	5	Int

Request Example

!U=#chen5582&W=3b2fcefd18b9913d2987d19e811c20251bf87eb4&A=userid=123456789&token=3b2fcefd18b9913d2987d19e811c20251bf87eb4&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143564221288&appid=testapp1|**S=TIMESALE&C=SUBS&P=DELL+AMTD&T=0+1+2+3+4**

Request Example (URL Encoded)

!U%3D%23chen5582%26W%3D3b2fcefd18b9913d2987d19e811c20251bf87eb4%26A%3Duserid%3D123456789%26token%3D3b2fcefd18b9913d2987d19e811c20251bf87eb4%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143564221288%26appid%3Dtestapp1%7CS%3D**TIMESALE**%26C%3D**SUBS**%26**P**%3D**DELL**%2B**AMTD**%26**T**%3D**0**%2B**1**%2B**2**%2B**3**%2B**4**

45.11.2 Response

Time & Sales Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message Length
SID	Short	0x0005	2	SID for TIMESALE
Column #	Byte	0x00	1	Symbol Column always returned
Symbol length	Short		2	Length of symbol string
Symbol	String	Symbol		Symbol value
Column #	Byte	0x01 – 0x05	1	Next field as defined in the Time & Sales Request
Column Value	Depends	depends		Depending on column #, string response fields will be preceded with a 'short' length field (two bytes) indicating the length of the string that follows

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Repeat ..				Repeat Column # and Column Value
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

Repeat column # and column value of the fields subscribed.

45.12 Level II - OPRA, NYSE_BOOK

For OPTIONS market depth, use OPRA Service ID. For NYSE and AMEX symbols, use NYSE_BOOK.

45.12.1 Request

Parameters

S = OPRA, NYSE_BOOK
C = SUBS
P = symbol+symbol+symbol ..
T = fieldNum+fieldNum+ .

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

Level II Quote Request Fields

FIELD	VALUE	TYPE
SYMBOL	0	String
TIME	1	Int
Data	2	Short, Byte[]

Request Example - NYSE_BOOK

!U=123456789&W=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&A=userid=123456789&token=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143580609994&appid=testapp1|**S=NYSE_BOOK&C=SUBS&P=IBM&T=0+1**

Request Example (URL Encoded) - NYSE_BOOK

!U%3D123456789%26W%3Decdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9%26A%3Duserid%3D123456789%26token%3Decdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9%26compa

ny%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergrou
p%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIP
MAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143580609994%26appid%3Dt
estapp1%7CS%3D**NYSE_BOOK**%26**C**%3D**SUBS**%26P%3D**IBM**%26**T**%3D**0**%2B**1**

Request Example - OPTIONS

!U=123456789&W=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&A=userid=123456789&
token=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&company=AMER&segment=UAMER&
cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=A
DAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143580609994&appi
d=testapp1|**S=OPRA&C=SUBS&P=AMTD_092209P12.5&T=0+1**

Request Example (URL Encoded) - OPTIONS

!U%3D123456789%26W%3Decdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9%26A%3Duser
id%3D123456789%26token%3Decdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9%26compa
ny%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergrou
p%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIP
MAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143580609994%26appid%3Dt
estapp1%7C**S**%3D**OPRA**%26**C**%3D**SUBS**%26**P**%3D**AMTD_092209P12.5**%26**T**%3D**0**
%2B**1**

45.12.2 Response

Level II Response (with compressed data)

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message length
SID	Short	0x0051 0x0054	2	NYSE Level II Book (NYSE_BOOK) Options Level II Book (OPRA)
Column #	Byte	0x00	1	SYMBOL Field ID - Symbol Column always returned
Symbol length	Short		2	Length of symbol string
Symbol	String	Symbol		Symbol value
Column #	Byte	0x01	1	TIME Field ID
Column Value	Int		4	Time for level 2 book in seconds from midnight
Column #	Byte	0x02	1	DATA Field ID
Book Length	Short		2	Length of level 2 book compressed data
Book Data	Binary			Compressed data; length is Book Length

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Delimiter	Byte	0xFF	1	Final field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream – '\n'

Level II Response Compressed Data Format After Uncompress

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Bid Count	Int		varies	Counts how many bid rows in book
Delimiter	Byte	;	1	Field delimiter for book data
Bid Total	Int		varies	Counts how many total bids in book
Delimiter	Byte	;	1	Field delimiter for book data
BEGINNING OF REPEAT BID DATA – repeat Bid Count **times**				
Bid MMID	String		4	Market Maker ID
Delimiter	Byte	;	1	Field delimiter for book data
Bid Price	Double		varies	One bid price level
Delimiter	Byte	;	1	Field delimiter for book data
Bid Size	Int		varies	One bid size
Delimiter	Byte	;	1	Field delimiter for book data
END OF REPEAT BID DATA				
Ask Count	Int		varies	Counts how many ask rows in book
Delimiter	Byte	;	1	Field delimiter for book data
Ask Total	Int		varies	Counts how many total asks in book
Delimiter	Byte	;	1	Field delimiter for book data
BEGINNING OF REPEAT ASK DATA – repeat Ask Count **times**				
Ask MMID	String		4	Market Maker ID
Delimiter	Byte	;	1	Field delimiter for book data
Ask Price	String		varies	One ask price level
Delimiter	Byte	;	1	Field delimiter for book data
Ask Size	String		varies	One ask size
Delimiter	Byte	;	1	Field delimiter for book data

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
END OF REPEAT ASK DATA				

45.12.3 Sample Response

In the following sample, data is shown in mixed format. Bytes represented in alpha are shown as alphabetic characters. The {XX} represents non-alpha data

DATA	TYPE	DESCRIPTION
S	1 byte	Shows it's a streaming record
{00}{66}	2 bytes	Message Length
{00}{51}	2 bytes	{00}{51} indicates it is a NYSE_BOOK quote
{00}	1 byte	FieldID: symbol
{00}{03}	2 bytes	symbol length
IBM	variable	the symbol
{01}	1 byte	FieldID: timestamp
{00}{00}{7C}{70}	4-byte	integer (31856)
{02}	1 byte	FieldID: the L2 compressed data record
{00}{56}	2 bytes	integer - length of compressed data (86)
...207 bytes	variable	the compressed data

When uncompressed, it is:

4;4;NASDAQ;90.7;1;PACX;90.52;16;ISE;89.48;30;NASD;0.01;1;4;4;NASDAQ;90.89;16;PACX;90.9;1;ISE;92.05;30;NASD;301000.0;1;

{FF}	1 byte	FieldID: final field
{0A}	1 byte	End of record

45.13 NASDAQ LEVEL II

The TOTAL_VIEW subscription is used to request NASDAQ Level II and Aggregated Depth data on NASDAQ Stocks. For OPTIONS market depth, use OPRA Service ID. For NYSE and AMEX symbols, use NYSE_BOOK.

45.13.1 Request

Parameters

S = TOTAL_VIEW

C = SUBS
P = symbol+symbol+symbol ..
T = fieldNum+fieldNum+ .

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

Since data can return in 3 formats: Total View Aggregate, Level 2, and Aggregated Depth at Price (aka Depth Chart), we will need to overload the symbol field (P) of the command. The following suffixes have to be attached to each symbol to indicate which type of data is to be returned

Symbol Attachment

FIELD	VALUE
TotalView Aggregate (NOT AVAILABLE)	>TV
Level II	>L2
Aggregated Depth	>AD

Level II Quote Request Fields

FIELD	VALUE	TYPE
SYMBOL	0	String
TIME	1	Int
Bid Book	2	Multi-field Data
Ask Book	3	Multi-field Data
NOII Data	4	Multi-field Data (NOT AVAILABLE)

Request Example - Level II data for AMTD

!U=123456789&W=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&A=userid=123456789&token=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143580609994&appid=testapp1|**S=TOTAL_VIEW&C=SUBS&P=AMTD>L2&T=0+1+2+3**

Request Example - Aggregated Depth data for AMTD

!U=123456789&W=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&A=userid=123456789&token=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143580609994&appid=testapp1|**S=TOTAL_VIEW&C=SUBS&P=AMTD>AD&T=0+1+2+3**

Request Example - Combined

!U=123456789&W=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&A=userid=123456789&token=ecdcdfc7c79fc9dbdbadf0af2e9697d19c0bead9&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143580609994&appid=testapp1|**S=TOTAL_VIEW&C=SUBS&P=DELL>AD+AMTD>L2&T=0+1+2+3**

45.13.2 Response

TotalView Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message length
SID	Short	0x0057	2	TOTAL_VIEW
Column #	Byte	0x00	1	SYMBOL Field ID
Symbol length	Short		2	Length of symbol string
Symbol	String			Symbol value
Column #	Byte	0x01	1	TIME Field ID
Timestamp	Int		4	Time for TV in milliseconds from midnight
Column #	Byte	0x02	1	BID BOOK Field ID
BID Book Length	Short		2	Length of the BID Side Book
BID Book Data	String	Multi-field		**BID** Side Book data. Please see Bid/Ask Book Table below
Column #	Byte	0x03	1	DATA Field ID
ASK Book Length	Short		2	Length of the ASK Side Book
ASK Book Data	String	Multi-field		**ASK** Side Book data. Please see Bid/Ask Book Table below
Column #	Byte	0x04	1	NOII Field ID
NOII Data Length	Short		2	Length of the NOII Data
NOII Data	Binary	Multi-field		**NOII** data. Please see HNOII Table below
Delimiter	Byte	0xFF	1	Final field

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream – '\n'

BID/ASK Response Data Format

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Total Row Count	Int		varies	Total number of rows for this side of the book. Can be more than the total # returned in subsequent sections, in which case, just display "X more rows" and use this field to compute that
Delimiter	Byte	;	1	Field delimiter for book data
Level Count	Int		varies	Counts how many levels in the book
Delimiter	Byte	;	1	Field delimiter for book data
BEGINNING OF REPEAT LEVEL DATA – repeat **Level Count** times				
Price	Double		Varies	Price for this level
Delimiter	Byte	;	1	Field delimiter for book data
Aggregate Size	Int		varies	Aggregate size of this level
Delimiter	Byte	;	1	Field delimiter for book data
MPID Count	Int		varies	Number of MPID's in this level
Delimiter	Byte	;	1	Field delimiter for book data
BEGINNING OF REPEAT Quote for each MPID – repeat **MPID Count** times				
MPID	String		varies	Market Participant ID, "?" if empty
Delimiter	Byte	;	1	Field delimiter for book data
Size	Int		varies	Size of this quote
Delimiter	Byte	;	1	Field delimiter for book data
Time	Long		varies	Time in milliseconds since midnight
Delimiter	Byte	;	1	Field delimiter for book data
END OF REPEAT BID/ASK DATA				

NOII Response Data Format

FIELD	TYPE	DESCRIPTION
Cross Type	Char	Cross Type
Shares Paired	Int	Shares Paired
Shares Imbalance	Int	Shares Imbalance
Imbalance Side	Char	Imbalance Side
Reference Price	Double	Reference Price
Far Indicative Price	Double	Far Indicative Price
Near Indicative Price	Double	Near Indicative Price
Price Variation Indicator	Char	Price Variation Indicator

45.13.3 Sample Response

In the following sample, data is shown in mixed format. Bytes represented in alpha are shown as alphabetic characters. The {XX} represents non-alpha data

DATA	TYPE	DESCRIPTION
S	1 byte	Shows it's a streaming record
{00}{4F}{00}{08}	4 bytes	the 4th byte {08} indicates it is a NASDAQ L2 quote
{00}	1 byte	FieldID: symbol
{00}{04}	2 bytes	symbol length
MSFT	variable	the symbol
{01}	1 byte	FieldID: timestamp
{00}{00}{C7}{EA}	4-byte	integer (51178)

{02}	1 byte	FieldID: the L2 data record
{00}{CF}	2 bytes	integer - length of data (207)
...207 bytes	variable	The multi-field data

Where the multi-field data is:

15;91;arcx;23.51;41;BRUT;23.51;40;cinn;23.51;31;SIZE;23.51;10;EDGX;23.51;10;TRAC;23.5;9;UBSS;23.5;5;LEHM;23.5;1;WCHV;23.5;1;PERT;23.49;22;FBCO;23.49;1;NFSC;23.49;1;amex;23.48;500;MOKE;23.48;1;BOFA;23.47;1;15;91;cinn;23.52;847;arcx;23.52;700;BRUT;23.52;213;SIZE;23.52;142;TRAC;23.52;43;BTRD;23.52;8;SBSH;23.53;11;MOKE;23.53;1;WCHV;23.53;1;NFSC;23.54;100;TMBR;23.54;50;PERT;23.54;5;BOFA;23.54;1;LEHM;23.54;1;COWN;23.54;1;

{FF} 1 byte FieldID: final field
{0A} 1 byte End of record

45.14 Option Lookup

THIS COMMAND IS NO LONGER SUPPORTED

45.15 ACTIVES

The ACTIVES_* server IDs are used to request the "Most Actives" lists of symbols based on various criteria

45.15.1 Request

Parameters

S = ACTIVES_NYSE, ACTIVES_NASDAQ, ACTIVES_OTCBB, ACTIVES_OPTIONS
C = SUBS
P = Venue-Duration+Venue-Duration....
T = 0+1

Field numbers should always be 0+1 for this request. The Service ID and Venue has to correspond. For example, if the select ACTIVES_NYSE, then the Venue must be NYSE as well.

You can specify multiple Venue-Duration combinations for each exchange. For example, NASDAQ-ALL+NASDAQ-3600. They have to be separated by a + sign

ACTIVES Request Fields

FIELDS	VALUE	TYPE
Actives Type	0	String
Data	1	String

ACTIVES Venue Duration

ACTIVES TYPE	VALUE	TYPE	DESCRIPTION
Venue	NASDAQ NYSE OTCBB CALLS-DESC* PUTS-DESC* OPTS-DESC*	String	Parameter to determine which venue/exchange
Duration	ALL 3600 1800 600 300 60	String	ALL = all day 3600 = 60 min 1800 = 30 min 600 = 10 min 300 = 5 min 60 = 1 min

*** Options**

Request Example NASDAQ Actives All day

!U=123456789&W=4416e81d1ac9b5e9203d1bcbd5e05844aa208894&A=userid=123456789&token=4416e81d1ac9b5e9203d1bcbd5e05844aa208894&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143736789195&appid=testapp1|**S=ACTIVES_NASDAQ&C=SUBS&P=NASDAQ-ALL&T=0+1**

Request Example NASDAQ Actives All day (URL Encoded)

!U%3D123456789%26W%3D4416e81d1ac9b5e9203d1bcbd5e05844aa208894%26A%3Duserid%3D123456789%26token%3D4416e81d1ac9b5e9203d1bcbd5e05844aa208894%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143736789195%26appid%3Dtestapp1%7C**S**%3D**ACTIVES_NASDAQ**%26**C**%3D**SUBS**%26**P**%3D**NASDAQ-ALL**%26**T**%3D**0**%2B**1**

Request Example NYSE 10 minutes

!U=123456789&W=054ba633eb878596346ab7acd825454f7ae131f5&A=userid=123456789&token=054ba633eb878596346ab7acd825454f7ae131f5&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143740871398&appid=testapp1|**S=ACTIVES_NYSE&C=SUBS&P=NYSE-600&T=0+1**

Request Example NYSE 10 minutes (URL Encoded)

!U%3D123456789%26W%3D054ba633eb878596346ab7acd825454f7ae131f5%26A%3Duserid%3D123456789%26token%3D054ba633eb878596346ab7acd825454f7ae131f5%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143740871398%26appid%3Dtestapp1%7C**S**%3D**ACTIVES_NYSE**%26**C**%3D**SUBS**%26**P**%3D**NYSE-600**%26**T**%3D**0**%2B**1**

Request Example OTCBB 1 minute

!U=123456789&W=054ba633eb878596346ab7acd825454f7ae131f5&A=userid=123456789&token=054ba633eb878596346ab7acd825454f7ae131f5&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143740871398&appid=testapp1|**S=ACTIVES_OTCBB&C=SUBS&P=OTCBB-60&T=0+1**

Request Example OTCBB 1 minute (URL Encoded)

!U%3D123456789%26W%3D054ba633eb878596346ab7acd825454f7ae131f5%26A%3Duserid%3D123456789%26token%3D054ba633eb878596346ab7acd825454f7ae131f5%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143740871398%26appid%3Dtestapp1%7C**S**%3D**ACTIVES_OTCBB**%26**C**%3D**SUBS**%26**P**%3D**OTCBB-60**%26**T**%3D**0**%2B**1**

Request Example Options Calls 60 min

!U=123456789&W=054ba633eb878596346ab7acd825454f7ae131f5&A=userid=123456789&token=054ba633eb878596346ab7acd825454f7ae131f5&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143740871398&appid=testapp1|**S=ACTIVES_OPTIONS&C=SUBS&P=CALLS-DESC-3600&T=0+1**

Request Example Options Calls 60 min (URL Encoded)

!U%3D123456789%26W%3D054ba633eb878596346ab7acd825454f7ae131f5%26A%3Duserid%3D123456789%26token%3D054ba633eb878596346ab7acd825454f7ae131f5%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143740871398%26appid%3Dtestapp1%7C**S**%3D**ACTIVES_OPTIONS**%26**C**%3D**SUBS**%26**P**%3D**CALLS-DESC-3600**%26**T**%3D**0**%2B**1**

Request Example Options Puts All day

!U=123456789&W=054ba633eb878596346ab7acd825454f7ae131f5&A=userid=123456789&token=054ba633eb878596346ab7acd825454f7ae131f5&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143740871398&appid=testapp1|**S=ACTIVES_OPTIONS&C=SUBS&P=OPTS-DESC-ALL&T=0+1**

Request Example Options Puts All day (URL Encoded)

!U%3D123456789%26W%3D054ba633eb878596346ab7acd825454f7ae131f5%26A%3Duserid%3D123456789%26token%3D054ba633eb878596346ab7acd825454f7ae131f5%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143740871398%26appid%3Dtestapp1%7C**S**%3D**ACTIVES_OPTIONS**%26**C**%3D**SUBS**%26**P**%3D**OPTS-DESC-ALL**%26**T**%3D**0**%2B**1**

45.15.2 Response

ACTIVES Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	SID Length
SID	Short	0x0017 0x0018 0x0019 0x001A 0x0023	2	NYSE Actives AMEX Actives NASDAQ Actives OTCBB Actives Options Actives
Column #	Byte	0x00	1	ACTIVES Field ID
Actives Type	String			String response fields will be preceded with a 'short' length field (2 bytes) indicating the length of the string that follows
Column #	Byte	0x01	1	DATA Field ID
Data	String			Actives Data delimited by ';' and ':'
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

Actives Data Parsing

The data is returned in groups. Each group is delimited by a semicolon '**;**'. The returned groups are the following (in the order that they are returned:

INITIAL GROUP: Descriptor Group. It contains information about the rest of the groups being returned

If number of groups (not counting the first one) is 2, then:

GROUP 0: Most Active Stocks based on "# of trades"
GROUP 1: Most Active Stocks based on "# of Shares" traded

If number of groups (not counting the first one) is 6, then:

GROUP 0: Most Active Stocks based on "Buy # of Trades"
GROUP 1: Most Active Stocks based on "Sell # of Trades"
GROUP 2: Most Active Stocks based on "Buy # of Shares"
GROUP 3: Most Active Stocks based on "Sell # of Shares"
GROUP 4: Most Active Stocks based on "# of trades"
GROUP 5: Most Active Stocks based on "# of Shares" traded

NOTE: Groups 0, 1, 2, 3 are not returned now. Deprecated

If the DATA parameter contains the following:

57510;0;00:00:00;15:58:30;6;0:0:0;1:0:0;2:0:0;3:0:0;4:10:5234081:AAPL:230692:4.41:SYMC:87008:1.66:LVLT:86452:1.65:GOOG:79025:1.51:INTC:75896:1.45:QQQQ:73049:1.4:ATYT:71554:1.37:BRCM:68573:1.31:CSCO:59819:1.14:VPHM:58947:1.13;5:10:2096769681:QQQQ:112137833:5.35:LVLT:101695066:4.85:SUNW:68187431:3.25:INTC:65462262:3.12:JDSU:55355659:2.64:CSCO:50843090:2.42:MSFT:49805550:2.38:AAPL:48921314:2.33:ORCL:47951340:2.29:SIRI:35386791:1.69;

Then it should be interpreted as:

ID: 57510
sampleDuration: 0
Start Time: 00:00:00
Display Time: 15:58:30
Num Groups: 6

followed by the strings for each group
0:0:0
1:0:0
2:0:0
3:0:0
4:
10:5234081:AAPL:230692:4.41:SYMC:87008:1.66:LVLT:86452:1.65:GOOG:79025:1.51:INTC:75896:1.45:QQQQ:73049:1.4:ATYT:71554:1.37:BRCM:68573:1.31:CSCO:59819:1.14:VPHM:58947:1.13
5:
10:2096769681:QQQQ:112137833:5.35:LVLT:101695066:4.85:SUNW:68187431:3.25:INTC:65462262:3.12:JDSU:55355659:2.64:CSCO:50843090:2.42:MSFT:49805550:2.38:AAPL:48921314:2.33:ORCL:47951340:2.29:SIRI:35386791:1.69

Data within each Group is separated by colons '**:**'

FOR STOCKS:

The first 3 entries are:

Group #: (0-based, 0 to Num) - the Group Number - refer to the list above.
NumEntries: # of actives entries (triplets) that follow
TotVol: Total volume (or total # of trades) for the exchange

followed by NumEntries of triplets, each of which is:

Symbol
Volume
Percent - same as Volume/TotVol

Example:

4:10:5234081:AAPL:230692:4.41:SYMC:87008:1.66:LVLT:86452:1.65:GOOG:79025:1.51:INTC:75896:1.45:QQQQ:73049:1.4:ATYT:71554:1.37:BRCM:68573:1.31:CSCO:59819:1.14:VPHM:58947:1.13

would be interpreted as:
GROUP 4: Most Active Stocks based on "# of trades"
10 Entries
Total Volume is 5,234,081

AAPL 230,692 trades 4.41%
SYMC 87,008 trades 1.66%
LVLT 86,452 trades 1.65%
GOOG 79,025 trades 1.51%
INTC 75,896 trades 1.45%
QQQQ 73,049 trades 1.4%
ATYT 71,554 trades 1.37%
BRCM 68,573 trades 1.31%
CSCO 59,819 trades 1.14%
VPHM 58,947 trades 1.1%

FOR OPTIONS:

Actives for OPTIONS are handled almost the same way as for stocks. The difference is that the Symbol Description is returned as well as the symbol

The first 3 entries are:

Group #: (0-based, 0 to Num) - the Group Number - refer to the list above.
NumEntries: # of actives entries (triplets) that follow
TotVol: Total volume (or total # of trades) for the exchange

followed by NumEntries of 4 value groups, each of which is:

Symbol
Symbol Description
Volume
Percent - same as Volume/TotVol

Example:

0:10:347642:APV_112109C200:APV Nov 21 2009 200.0 Call:3233:0.93:APV_112109P200:APV Nov 21 2009 200.0 Put:2669:0.77:SPY_112109C110:SPY Nov 21 2009 110.0 Call:1973:0.57:SPY_112109P110:SPY Nov 21 2009 110.0 Put:1846:0.53:AJL_121909C210:AJL Dec 19 2009 210.0 Call:1708:0.49:SWG_112109P109:SWG Nov 21 2009 109.0 Put:1651:0.47:SWG_112109C109:SWG Nov 21 2009 109.0 Call:1359:0.39:APV_121909C200:APV Dec 19 2009 200.0 Call:1276:0.37:SPY_112109C111:SPY Nov 21 2009 111.0 Call:1227:0.35:SWG_121909P109:SWG Dec 19 2009 109.0 Put:1214:0.35;

would be interpreted as:
GROUP 0: Most Active Stocks based on "# of trades"
10 Entries
Total Volume is 347,642

APV_112109C200	"APV Nov 21 2009 200.0 Call"	3233 trades 0.93%
APV_112109P200	"APV Nov 21 2009 200.0 Put"	2669 trades 0.77%

```
SPY_112109C110      "SPY Nov 21 2009 110.0 Call"   1973 trades 0.57%
SPY_112109P110      "SPY Nov 21 2009 110.0 Put"    1846 trades 0.53%
AJL_121909C210      "AJL Dec 19 2009 210.0 Call"   1708 trades 0.49%
SWG_112109P109      "SWG Nov 21 2009 109.0 Put"    1651 trades 0.47%
SWG_112109C109      "SWG Nov 21 2009 109.0 Call"   1359 trades 0.39%
APV_121909C200      "APV Dec 19 2009 200.0 Call"   1276 trades 0.37%
SPY_112109C111      "SPY Nov 21 2009 111.0 Call"   1227 trades 0.35%
SWG_121909P109      "SWG Dec 19 2009 109.0 Put"    1214 trades 0.35;%
```

45.16 News

The NEWS server ID is used to request streaming news headlines for various stocks and indices

45.16.1 Request

Parameters

S = NEWS
C = SUBS
P = symbol+symbol+symbol....*ALL*+*HOT*
T = fieldNum+fieldNum+ .

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

News Request Fields

FIELD	VALUE	TYPE
SYMBOL	0	String
TIME	1	Int
DATA	2	String
FIELD SIZE	3	Int

Request Example

!U=123456789&W=36a7b62ce5f82d8bd07443f62d02277c4da41f03&A=userid=123456789&token=36a7b62ce5f82d8bd07443f62d02277c4da41f03&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1143830624229&appid=testapp1|**S=NEWS&C=SUBS&P=DELL+AMTD&T=0+1+2+3**

Request Example (URL Encoded)

!U%3D123456789%26W%3D36a7b62ce5f82d8bd07443f62d02277c4da41f03%26A%3Duserid%3D123456789%26token%3D36a7b62ce5f82d8bd07443f62d02277c4da41f03%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1143830624229%26appid%3Dtestapp1%7C**S**%3D**NEWS**%26**C**%3D**SUBS**%26**P**%3D**DELL**%2B**AMTD**%26**T**%3D**0**%2B

1%2B**2**%2B**3**

45.16.2 Response

NEWS Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message Length
SID	Short	0x001B	2	SID for News
Column #	Byte	0x00	1	Symbol Field ID - always returned
Symbol length	Short		2	Length of symbol string.
Symbol	String	Symbol	variable	Symbol value
Column #	Byte	0x01	1	Timestamp Field ID - always returned
Timestamp Length	Short		2	Length of the timestamp string
Timestamp String	String		variable	Timestamp – an integer - # of milliseconds since 1/1/1970 UTC
REPEATING DATA				
Data Column #	Byte	0x02	1	Data Column #
Data Length	Short		2	Length of the timestamp string
Column Value	Byte Array			Compressed Data (see News Data Fields table below)
END OF REPEATING DATA				
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

News Data Fields delimited by 0x02

FIELDS	VALUES/DESCRIPTION
Modifier	A, R (for resetting news from the previous day)
Delimiter	0x02

FIELDS	VALUES/DESCRIPTION
Source	News source: DJN, BW, PR
Delimiter	0x02
Headline	News Story Headline
Delimiter	0x02
News URL	News story URL
Delimiter	0x02
Vendor ID	Vendor ID
Delimiter	0x02
Count	News Count – accumulated number of headlines per symbol generated by News Publisher (from Midnight). For display with symbol in Level 1.
Delimiter	0x02
Keywords	A list of keywords delimited by 0x03
Delimiter	0x02

45.17 News History

The NEWS_HISTORY server ID is used to return a list of headlines that have already occurred for the current day for a specific symbol.

45.17.1 Request

Parameters

S = NEWS
C = GET
P = symbol;FromTime (Number of milliseconds since 1/1/1970)

Request Example

!U=123456789&W=ba59869373ede93dec6486e284cfc3a8e805853a&A=userid=123456789&token=ba59869373ede93dec6486e284cfc3a8e805853a&company=AMER&segment=DATEK&cddomain=A000000009663628&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=AQC1DRESGKMAOCPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1144073528716&appid=prd_lapp3|**S=NEWS**&**C=GET**&**P=DELL;1144036800306**

Request Example (URL Encoded)

!U%3D123456789%26W%3Dba59869373ede93dec6486e284cfc3a8e805853a%26A%3Duserid%3D123456789%26token%3Dba59869373ede93dec6486e284cfc3a8e805853a%26company%3DAMER%26segment%3DDATEK%26cddomain%3DA000000009663628%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DAQC1DRESGKMAOCPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1144073528716%26appid%3Dprd_lapp3%7C**S**%3D**NEWS**%26**C**%3D**GET**%26**P**%3D**DELL**%3B**1144036800306**

45.17.2 Response

NOTE: The HTTP request for NEWS HISTORY may return multiple snapshot records. It may return one or more STREAMER SERVER records followed by the NEWS HISTORY record. Please make sure that you properly handle any records returned and do NOT just assume that it will be only News History response

News History Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	N	1	Snapshot
SID Length	Short	0x02	2	Length of the SID string that follows
SID String	String	28	2	The Service ID of News history in ASCI format
Payload Length	Int		4	The length of the payload - all data following this field, not including the terminating delimiters
SID	Short	0x001C	2	SID for News History - binary version of the one received before
Symbol Length	Short		2	Length of the symbol string
Symbol	String	Symbol	variable	Symbol value
Message Type	Short	0x00	2	0 = successful retrieval
Length	Short		2	Length of the DATA sting
Data	Byte Array		variable	Compressed News History (see Data Fields table below)
Delimiter	Byte	0xFF	1	
Ending delimiter	Byte	0x0A	1	

News History Data Fields

FIELDS	VALUES/DESCRIPTION
Headline ID	Headline ID
Delimiter	0x02
Headline	News headline
Delimiter	0x02
Timestamp	News headline timestamp

FIELDS	VALUES/DESCRIPTION
Delimiter	0x02
Source	News source
Delimiter	0x02
News URL	News story URL
Delimiter	0x01

45.18 CHART Data (Backfill)

The CHART data (also known as Backfill Data and Historical Intraday data) for a specific symbol is retrieved using NYSE_CHART, NASDAQ_CHART and INDEX_CHART server IDs

The specific Service ID that needs to be used depends on the symbol. NASDAQ_CHART would be NASDAQ symbols. NYSE_CHART for NYSE and AMEX symbols and INDEX_CHART for all Indices (symbols starting with $)

This data can be requested as a snapshot history, or as streaming bars.

45.18.1 STREAMING Bars

45.18.1.1 Request

Parameters

S = NYSE_CHART, NASDAQ_CHART, INDEX_CHART
C = SUBS
P = symbol+symbol+symbol....
T = fieldNum+fieldNum+ .

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

Charts Field Requests

FIELD	VALUE	TYPE
SYMBOL	0	String
SEQUENCE	1	Int (0 - 629) (minutes since 8:00 AM US Eastern time)
OPEN	2	Float
HIGH	3	Float
LOW	4	Float
CLOSE	5	Float
VOLUME	6	Int

FIELD	VALUE	TYPE
TIME	7	Int (seconds since midnight)
DATE	8	Int (days since 1/1/1970)

Request Example

!U=123456789&W=c11fa0e798aba3b9d65866befce078537dc3ed18&A=userid=123456789&token=c11fa0e798aba3b9d65866befce078537dc3ed18&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1144091133399&appid=testapp1|**S=NASDAQ_CHART&C=SUBS&P=DELL&T=0+1+2+3+4+5+6+7**

Request Example (URL Encoded)

!U%3D123456789%26W%3Dc11fa0e798aba3b9d65866befce078537dc3ed18%26A%3Duserid%3D123456789%26token%3Dc11fa0e798aba3b9d65866befce078537dc3ed18%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1144091133399%26appid%3Dtestapp1%7C**S**%3D**NASDAQ_CHART**%26**C**%3D**SUBS**%26**P**%3D**DELL**%26**T**%3D**0**%2B**1**%2B**2**%2B**3**%2B**4**%2B**5**%2B**6**%2B**7**

45.18.1.2 Response

NOTE: There are a few notes to consider when processing the responses:

1. Use the bar sequence for the time on each bar instead of the time
2. Only complete bars will be returned. If you get historical bars, the last bar returned may be newer than the last returned Streaming CHART bar since historical request can return partial data for the most current bar
3. You can get out of sequence or duplicate bars. This occurs if there is a correction

Streaming Chart Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	S	1	Streaming
Message Length	Short		2	Message Length
SID	Short	0x0052 0x0053 0x0055	2	NASDAQ Chart NYSE Chart Indices Chart
Column #	Byte	0x00	1	SYMBOL Field ID - Always returned
Symbol length	Short		2	Length of symbol string

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Symbol	String			Symbol
Column #	Byte	0x01	1	SEQUENCE Field ID - Always returned
Sequence	Int			0 – 629 (minutes since 8:00 AM US Eastern time)
Column #	Byte	0x02	1	OPEN Field ID - Always returned
Open	Float			OPEN price for the chart bar
Column #	Byte	0x03	1	HIGH Field ID - Always returned
High	Float			Highest trade price for the chart bar
Column #	Byte	0x04	1	LOW Field ID - Always returned
Low	Float			Lowest trade price for the chart bar
Column #	Byte	0x05	1	CLOSE Field ID - Always returned
Close	Float			closing price for the chart bar
Column #	Byte	0x06	1	VOLUME Field ID - Always returned
Volume	Int			Number of shares traded during the chart bar
Column #	Byte	0x07	1	TIME Field ID - Always returned
Time	Int			Seconds since midnight
Column #	Byte	0x08	1	DATE Field ID - Always returned
Date	Int			Days since 1/1/1970
REPEATS UNTIL DELIMITER IS ENCOUNTERED				
Delimiter	Int	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

45.18.2 SNAPSHOT History

45.18.2.1 Request

NOTE: snapshot History via the stream server should no longer be used. Please use PriceHistory instead

Parameters

S = NYSE_CHART, NASDAQ_CHART, INDEX_CHART
C = GET
P = *symbol, first bar sequence,last bar sequence*,5d,1m (parameters – "last chart period" is number of bars to return per day, 610 returns all; next two parameters are interpreted as 5 days of data, 1 minute frequency)
T = N/A.

Field numbers must be listed in ascending order. A symptom that there is a lower number following a higher number – the fields from the out of order number to the end are not returned.

NOTE: Here is some background in order to understand the data being requested/returned: The servers return 1 minute bar data for each day, starting at 8:00AM. Each bar has a sequence number, starting with Zero for each day. The Nth bar is always for data for N minutes from 8:00 AM. The request allows you to specify the start and end sequence of the bars that you want returned for each day.

FREQUENCY should always be 1m. that parameter is not used at this time

NOTE: we strongly suggest you serialize this request. If you need to get data on multiple symbols at the same time you should do this request one at a time, NOT concurrently

Request Example

!U=123456789&W=0f4907562c6fdcf69b31e2e63cf149eccee4608d&A=userid=123456789&token=0f4907562c6fdcf69b31e2e63cf149eccee4608d&company=AMER&segment=UAMER&cddomain=A000001111111111&usergroup=ACCT&accesslevel=ACCT&authorized=Y&acl=ADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS&timestamp=1144091069407&appid=testapp1|**S=NASDAQ_CHART&C=GET&P=DELL,0,720,5d,1m**

Request Example (URL Encoded)

!U%3D123456789%26W%3D0f4907562c6fdcf69b31e2e63cf149eccee4608d%26A%3Duserid%3D123456789%26token%3D0f4907562c6fdcf69b31e2e63cf149eccee4608d%26company%3DAMER%26segment%3DUAMER%26cddomain%3DA000001111111111%26usergroup%3DACCT%26accesslevel%3DACCT%26authorized%3DY%26acl%3DADAQC1DRESGKIPMAOLPNQSRFSPTETFTOTTUAWSQ2NS%26timestamp%3D1144091069407%26appid%3Dtestapp1%7C**S**%3D**NASDAQ_CHART**%26**C**%3D**GET**%26**P**%3D**DELL**%2C**0**%2C**720**%2C**5d**%2C**1m**

45.18.2.2 Response

NOTE: The HTTP request for CHART data may return multiple snapshot records. It may return one or more STREAMER SERVER records followed by the Snapshot CHART record. Please make sure that you properly handle any records returned and do NOT just assume that it will be only Snapshot CHART response

Chart History Snapshot Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	N	1	Snapshot
Snapshot ID length	Short	0x02	2	Skip these 4 bytes
Snapshot ID	String	82 83 85	2	ASCI version of the Streamable ID (SID) being returned

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Message Length	Int		4	Message Length
SID	Short	0x0052 0x0053 0x0055	2	NYSE Chart NASDAQ Chart Indices Chart
Symbol length	Short		2	Length of symbol string
Symbol	String	Symbol		Symbol value
Status	Short	0x0000	2	0 = success
Length	Int		4	Length of compressed message
Data	Byte Array			Compressed Chart Bar Data (Refer to table below)
Delimiter	Byte	0xFF	1	
Ending delimiter	Byte	0x0A	1	

NOTE: each bar is terminated with a semicolon. You may get NULL values for a bar, with just the semicolon terminator. For example, ;;;;;;;;. This happens because there were no trades for a particular minute, yet the server has to return a "Bar" for that minute. In that situation, just skip the null bar.

Bar Data Format

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Symbol	String		variable	
Delimiter	Char	,	1	Delimiter for bar data
Sequence	Int		variable	0 – 629 (minutes since 8:00 AM US Eastern time)
Delimiter	Char	,	1	Delimiter for bar data
Open	Float		variable	Opening price for symbol
Delimiter	Char	,	1	Delimiter for bar data

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
High	Float		variable	Highest trade price for symbol at this time
Delimiter	Char	,	1	Delimiter for bar data
Low	Float		variable	Lowest trade price for symbol at this time
Delimiter	Char	,	1	Delimiter for bar data
Close	Float		variable	Previous trading session closing price
Delimiter	Char	,	1	Delimiter for bar data
Volume	Int		variable	Share volume at this time
Delimiter	Char	,	1	Delimiter for bar data
Time	Int		variable	Seconds since midnight. **NOTE: DO NOT USE THIS FIELD.** Use the Sequence instead
Delimiter	Char	,	1	Delimiter for bar data
Date	Int		variable	Days since 1/1/1970
Delimiter	Char	;	1	Delimiter for chart bar

45.19 STREAMER SERVER

STREAMER SERVER Response comes in Snapshot RESPONSE format

This is a general response from the Streamer Server that may include messages from service providers, entitlement information or issues encountered (such as the service provider is not returning data etc).

These messages are NOT necessarily errors. Please review the Return Code Values table for the intent of the response

NOTE: When receiving this N, string length, 100 value, consider it a Streamer Server Response message

Streamer Server Response

FIELD	TYPE	VALUE	LENGTH (8 BIT BYTES)	DESCRIPTION
Header ID	Byte	N	1	Snapshot
Next string length	Short		2	
Snapshot ID	String	100		100 ascii in this case
Message Length	Int		4	Message Length
SID	Short	0x0A	2	N/A
Column #	Byte	0x00	1	Column ID for "Service ID"
Service ID	Short		2	The Server SID of the request for which to which this Server reply applies
Column #	Byte	0x01	1	Column ID for Return code
Return code	Short	0x0000 – 0x0012	2	Return code value (see Table below)
Column #	Byte	0x02	1	Description
Description Length	Short		2	Length of description string
Description	String			Return Code description string (see Table below)
Delimiter	Byte	0xFF	1	Last field
Ending Delim	Byte	0x0A	1	Ending delimiter for this data stream

Return Code Values

RETURN CODE DESCRIPTION	CODE VALUE
SUCCESS	0
SERVICE_DOWN	1
SERVICE_TIMEOUT	2
LOGIN_DENIED	3
AUTHORIZER_BUSY	4
AUTHORIZER_DOWN	5
USER_NOT_FOUND	6
ACCOUNT_ON_HOLD	7
ACCOUNT_FROZEN	8
UNKNOWN_FAILURE	9
FAILURE	10

RETURN CODE DESCRIPTION	CODE VALUE
SERVICE_NOT_AVAILABLE	11
CLOSE_APPLET	12
USER_STATUS	13
ACCOUNT_EMPTY	14
MONOPOLIZE_ACK	15
NOT_AUTHORIZED_FOR_SERVICE	16
NOT_AUTHORIZED_FOR_QUOTE	17
STREAMER_SERVER_ID	18
REACHED_SYMBOL_LIMIT	19
STREAM CONNECTION NOT FOUND	20
BAD_COMMAND_FORMAT	21
FAILED_COMMAND_SUBS	22
FAILED_COMMAND_UNSUBS	23
FAILED_COMMAND_ADD	24
FAILED_COMMAND_VIEW	25
SUCCEEDED_COMMAND_SUBS	26
SUCCEEDED_COMMAND_UNSUBS	27
SUCCEEDED _COMMAND_ADD	28
SUCCEEDED _COMMAND_VIEW	29

Index

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- X -