E\*TRADE FINANCIAL

***PHP-SDK for ETRADE APIs***

|  |  |  |
| --- | --- | --- |
| Language | : | PHP |
| Version | : | Developer Beta |

### PHP-SDK Prerequisite

1. PHP-SDK requires PHP language version 5.2 or newer. You can obtain new version of PHP from www.php.net. You can have it on your choice of Operating System.
2. In addition to that, PHP must be compiled with curl lib support.
3. PHP-SDK requires Google’s OAuth library, download it from : <http://oauth.googlecode.com/svn/code/php/OAuth.php> (Subversion revision : 1263) and copy under SDK’s OAuth folder.

### Installation

1. Download/checkout PHP-SDK from ETRADE website to your choice of folder/subfolder

1. Create config.php for your application. Sample config.php can be copied from Samples/ directory.
2. Include your local config.php on top of your application.

Include <SDK PATH>/Common/common.php afterwards.

1. Start writing your application.

1. Start using SDK functions. Comprehensive list of functions can be found in SDK detail document.
2. For additional details on data type, options and other details of each parameter, please check “Etrade API reference guide”.
3. Using sample scripts

* Sample scripts are located under Samples/directory. These are command line scripts and can be invoked on linux console or windows command line by using 'php <scriptname>'. e.g.

$php test\_etOAuth.php

* Edit given sample script with your configuration.
* You will need consumer application key and secret provided by Etrade.
* As first step, you must obtain access token and secret by using test\_etOAuth.php
* Configure accounts, market and orders sample script with access token and secret generated by test\_etOAuth.php

### Understanding config file:

PHP-SDK provides default configuration file under **Common**/ folder as **config.php** . This file provides basic configurations required by SDK. You can override any or all configuration values by creating your own config.php file under your application directory and including it on top of your application.

Important configurations:

**ET\_SDK\_PATH**  : Location of your PHP-SDK directory.

**DEBUG\_MODE** : Provide debug messages, good to enable for development purpose.

**RESPONSE\_FORMAT** : Format of response returned by server. Options are ‘xml’ or ‘json’.

Rest of config values provides URLs for using APIs. Please read config.php for details.

### ETRADE OAuth Library

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

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

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

### PHP-SDK contains etOAUTH, which library provides all needful function to complete authentication process.

**Pre-requisites:** You must accept the agreement to use Etrade API and must have API key and secret provided by Etrade.

**Getting Access Token:**

In order to final access token you need to perform following steps :

1. Get request token from Etrade OAuth server.
2. Get Authorize URL using request token.
3. Access Authorize URL using browser and login with your credentials.
4. After login, you will get verifier code.
5. Using verifier code, get final access token from Etrade OAuth Server.

### etOAuthConsumer

etOAuthConsumer class provide a consumer object to use for OAuth operations.

**\_\_construct()**

Params : API key, API secret, callback method (optional, as Etrade is using only ‘oob’ at present)

Return : etOAuthConsumer object.

### etOAuth

etOAuth class provide following methods to complete OAuth operations :

**\_\_construct()**

This is class constructor and constructs a request object using consumer object. e.g.

$consumer = new etOAuthConsumer(‘KEY’,’SECRET’);

$request = new etOAuth($consumer);

**GetRequestToken**()

Params : none

Return : Array of request token values.

**GetAuthorizeURL()**

Params : none

Return : String, authorize URL.

**GetAccessToken()**

Params : verifier code.

Return : Array, access token.

**RenewAccessToken()**

Params : none.

Returns : xml message returned from API.

**RevokeAccessToken** ()

Params: none.

Returns: xml message returned from API.

Example work flow :

**Construct an object of etOAuthConsumer:**

$consumer = new etOAuthConsumer(‘KEY’,’SECRET’);

**Construct request object using consumer object:**

$request = new etOAuth($consumer);

**Get request token:**

$request->GetRequestToken();

**Get authorize URL:**

$auth\_url = $request->GetAuthorizeURL();

**Open above URL in browser and follow the instructions. You will get verifier code.**

**Get access token:**

$access\_token = $request->GetAccessToken(‘VERIFIER CODE’);

Once you get the access token, you are good to go for accessing other resources.

### ETRADE Accounts Library

Accounts library “etAccounts” provides method to access account related resources.

**Pre-requisites** : You should have your access token ready to call accounts methods.

**\_\_construct()**

Param : consumer object.

Return : etAccounts object.

**GetAccountList()**

Provides list of accounts belong to the user.

Param : none

Return : list of accounts as xml or json string

**GetAccountBalance()**

Provides account balance.

Param : account ID.

Return : account balance as xml or json string

**GetAccountPositions()**

Provides account positions.

Param : account ID, AccountPositionsRequest object.

AccountPositionsRequest object properties are :

count

marker

symbol

typeCode

callPut

expYear

expMonth

expDay

strikePrice

*(For data type, options and other details of each parameter, please check “****Etrade API reference guide”*** *)*

Return : account positions as xml or json string

**GetAlerts()**

Provides list of alerts

Param : none

Return : List of alerts as xml or json string

**GetAlertDetails()**

Provides alerts details.

Param : alert Id.

Return : Details of an alert as xml or json string

**DeleteAlert()**

Delete an alert

Param : alert Id.

Return : string, success or failure message from server.

**Usage Example:**

**Create a consumer object:**

$consumer = **new** etOAuthConsumer(‘KEY’,’SECRET’);

$consumer->oauth\_token = OAUTH\_ACCESS\_TOKEN;

$consumer->oauth\_token\_secret = OAUTH\_ACCESS\_TOKEN\_SECRET;

**Create an etAccounts object:**

$ac\_obj = **new** etAccounts($consumer);

**Get account list:**

$ac\_list = $ac\_obj->GetAccountList();

**Get account balance:**

$ac\_balance = $ac\_obj->GetAccountBalance(‘AccountID’);

**Get account positions:**

//Construct params object

$request\_params = new AccountPositionsRequest();

$request\_params->\_\_set(count,3);

$request\_params->\_\_set(expDay,16);

//Call GetAccountPositions

$ac\_positions = $ac\_obj->GetAccountPositions($ac\_id,$request\_params);

**Get account alerts:**

$ac\_alerts = $ac\_obj->GetAlerts();

**Get Alert details:**

$ac\_alert\_dtl = $ac\_obj->GetAlertDetails(‘alert\_id’);

**Delete alert:**

$ac\_alert\_del = $ac\_obj->DeleteAlert(‘alert\_id’);

**Get Transaction History:**

$request\_params = new TransactionHistoryRequest();

$request\_params->\_\_set(count,3);

$RESTParams = 'TRADES/ASSIGNMENT,EXPIRATION';

$ac\_xml = $ac\_obj->GetTransactionHistory($ac\_id,$request\_params, $RESTParams);

$ac\_responce\_obj = etHttpUtils::GetResponseObject($ac\_xml);

**Get Transaction Details**

$ac\_ xml= $ac\_obj->GetTransactionHistoryDetails($ac\_id,$activity\_id);

$ac\_responce\_obj = etHttpUtils::GetResponseObject($ac\_xml);

**Market Library**

Market library “MarketClient” provides interface to ETRADE Market API.

**\_\_construct()**

Constructor creates an object of MarketClient class, using consumer object.

Param : consumer object.

Return : Market client object.

**getOptionChain()**

Provides option chain response

Param : getOptionChainsParams object

Valid getOptionChainsParams properties.

underlier

chainType

skipAdjusted

expirationMonth

expirationYear

Return: Option chains as xml or json string

**productLookup()**

Provides product details.

Param : productLookupParams object

Valid productLookupParams object properties

company;

type

Return: product detail as xml or json string.

**getExpiryDates()**

Provides expiry dates of a product.

Param : getExpiryDateParams object.

Valid getExpiryDateParams properties.

underlier;

expiryType;

Return: expiry date list as xml or json string.

**getQuote()**

Provides quote for a product.

Param : getExpiryDateParams object

Valid getExpiryDateParams properties:

symbolList

afterhourFlag

detailFlag

Return: quote detail as xml or json string.

**Usage Example:**

**Create a consumer object :**

$consumer = **new** etOAuthConsumer(‘KEY’,’SECRET’);

$consumer->oauth\_token = OAUTH\_ACCESS\_TOKEN;

$consumer->oauth\_token\_secret = OAUTH\_ACCESS\_TOKEN\_SECRET;

**Create a MarketClient object**

$ac\_obj = **new** MarketClient($consumer);

**Get Option Chains**

$request\_params = **new** getOptionChainsParams();

$request\_params->\_\_set('expirationMonth', '1');

$request\_params->\_\_set('expirationYear', '2012');

$request\_params->\_\_set('chainType', 'PUT');

$request\_params->\_\_set('skipAdjusted', 'true');

$request\_params->\_\_set('underlier', 'GOOG');

$out = $ac\_obj->getOptionChain($request\_params);

**Product lookup**

$request\_params = **new** productLookupParams();

$request\_params->\_\_set('company', 'cisco');

$request\_params->\_\_set('type', 'eq');

$out = $ac\_obj->productLookup($request\_params);

**Get expiry dates**

$request\_params = **new** getExpiryDateParams();

$request\_params->\_\_set('underlier', 'GOOG');

$out = $ac\_obj->getExpiryDates($request\_params);

**Get quote**

$request\_params = **new** getQuoteParams();

$request\_params->\_\_set('symbolList', **array**('GOOG','CSCO'));

$request\_params->\_\_set('detailFlag', 'All');

$out = $ac\_obj->getQuote($request\_params);

**Orders Library**

Orders library “OrderClient” provides interface for ETRADE Orders API. Here are the details for methods defined under this library.

**getOrderList()**

Provides list of all orders

Param : Account ID

Return: Orders list as xml or json string.

**previewEquityOrder()**

Provides preview equity order.

Param : getExpiryDateParams object.

Valid getExpiryDateParams properties.

accountId

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

stopLimitPrice

symbol

orderAction

priceType

routingDestination

marketSession

orderTerm

Return: response as xml or json string.

**placeEquityOrder()**

Provides place equity order.

Param : getExpiryDateParams object.

Valid getExpiryDateParams properties.

accountId

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

stopLimitPrice

symbol

orderAction

priceType

routingDestination

marketSession

orderTerm

Return: response as xml or json string.

**previewOptionOrder()**

Provide preview option order.

Param : OptionOrderRequest object.

Valid OptionOrderRequest properties:

accountId

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

stopLimitPrice

symbolInfo ( object )

orderAction

priceType

routingDestination

marketSession

orderTerm

Valid symbolInfo object properties:

symbol

callOrPut

strikePrice

expirationYear

expirationMonth

expirationDay

Return: response as xml or json string.

**placeOptionOrder()**

Provides expiry dates of a product.

Param : OptionOrderRequest object.

Valid OptionOrderRequest properties :

accountId

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

stopLimitPrice

symbolInfo ( object )

orderAction

priceType

routingDestination

marketSession

orderTerm

Valid symbolInfo object properties:

symbol

callOrPut

strikePrice

expirationYear

expirationMonth

expirationDay

Return: response as xml or json string.

**previewChangeEquityOrder()**

Provides preview change equity order

Param : changeEquityOrderRequest object.

Valid changeEquityOrderRequest properties.

priceType

orderTerm

accountId

orderNum

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

Return: response as xml or json string.

**placeChangeEquityOrder()**

Provides Place change equity order

Param : changeEquityOrderRequest object.

Valid changeEquityOrderRequest properties.

priceType

orderTerm

accountId

orderNum

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

Return: response as xml or json string.

**previewChangeOptionOrder()**

Provides perview change option order.

Param : changeOptionOrderRequest object.

Valid changeOptionOrderRequest properties :

stopLimitPrice

priceType

orderTerm

accountId

orderNum

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

Return: response as xml or json string.

**placeChangeOptionOrder()**

Provides place change option order.

Param : changeOptionOrderRequest object.

Valid changeOptionOrderRequest properties.

stopLimitPrice

priceType

orderTerm

accountId

orderNum

clientOrderId

limitPrice

previewId

stopPrice

allOrNone

quantity

reserveOrder

reserveQuantity

Return: response as xml or json string.

**cancelOrder()**

Provides cancel an order.

Param : CancelOrderRequest object.

Valid CancelOrderRequest properties.

accountId

orderNum

Return: response as xml or json string.

**Example usages:**

**Create a consumer object:**

$consumer = **new** etOAuthConsumer(‘KEY’,’SECRET’);

$consumer->oauth\_token = OAUTH\_ACCESS\_TOKEN;

$consumer->oauth\_token\_secret = OAUTH\_ACCESS\_TOKEN\_SECRET;

**Create an OrderClient object**

$ac\_obj = **new** OrderClient ($consumer);

**Get Order List**

$out = $ac\_obj->getOrderList(83491757);

**Preview Equity Order**

$request\_params = **new** EquityOrderRequest();

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('clientOrderId','123123');

$request\_params->\_\_set('limitPrice',300);

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice',300);

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity',4);

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity',0);

$request\_params->\_\_set('stopLimitPrice','');

$request\_params->\_\_set('symbol','AAPL');

$request\_params->\_\_set('orderAction','BUY');

$request\_params->\_\_set('priceType','LIMIT'

$request\_params->\_\_set('routingDestination','');

$request\_params->\_\_set('marketSession','REGULAR'

$request\_params->\_\_set('orderTerm','GOOD\_FOR\_DAY');

$request\_xml\_object = **new** PreviewEquityOrder($request\_params);

$out = $ac\_obj->previewEquityOrder($request\_xml\_object);

**Place Equity Order**

$request\_params = **new** EquityOrderRequest();

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('clientOrderId','asdfdsa12312');

$request\_params->\_\_set('limitPrice',300);

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice',300);

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity',4);

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity',0);

$request\_params->\_\_set('stopLimitPrice','');

$request\_params->\_\_set('symbol','AAPL');

$request\_params->\_\_set('orderAction','BUY'

$request\_params->\_\_set('priceType','LIMIT'

$request\_params->\_\_set('routingDestination','');

$request\_params->\_\_set('marketSession','REGULAR'

$request\_params->\_\_set('orderTerm','GOOD\_FOR\_DAY'

$request\_xml\_object = **new** PlaceEquityOrder($request\_params);

$out = $ac\_obj->placeEquityOrder($request\_xml\_object);

**Preview Option Order**

//Build option\_symbol\_obj

$option\_symbol\_obj = **new** OptionSymbol();

$option\_symbol\_obj->\_\_set('symbol','AAPL');

$option\_symbol\_obj->\_\_set('callOrPut','CALL');

$option\_symbol\_obj->\_\_set('strikePrice',115);

$option\_symbol\_obj->\_\_set('expirationYear','2011');

$option\_symbol\_obj->\_\_set('expirationMonth','11');

$option\_symbol\_obj->\_\_set('expirationDay','17');

//Build request\_params

$request\_params = **new** OptionOrderRequest();

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('clientOrderId',12345);

$request\_params->\_\_set('limitPrice',3);

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice','');

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity',4);

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity','');

$request\_params->\_\_set('symbolInfo',$option\_symbol\_obj);

$request\_params->\_\_set('stopLimitPrice','');

$request\_params->\_\_set('orderAction','BUY\_OPEN');

$request\_params->\_\_set('priceType','LIMIT');

$request\_params->\_\_set('marketSession','REGULAR');

$request\_params->\_\_set('orderTerm','GOOD\_FOR\_DAY');

$request\_params->\_\_set('routingDestination','');

$request\_xml\_object = **new** PreviewOptionOrder($request\_params);

$out = $ac\_obj->previewOptionOrder($request\_xml\_object);

**Place Option Order**

//Build option\_symbol\_obj

$option\_symbol\_obj = **new** OptionSymbol();

$option\_symbol\_obj->\_\_set('symbol','AAPL');

$option\_symbol\_obj->\_\_set('callOrPut','CALL'); $option\_symbol\_obj->\_\_set('strikePrice',115);

$option\_symbol\_obj->\_\_set('expirationYear','2011');

$option\_symbol\_obj->\_\_set('expirationMonth','11');

$option\_symbol\_obj->\_\_set('expirationDay','17');

//Build request\_params

$request\_params = **new** OptionOrderRequest();

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('clientOrderId',12345);

$request\_params->\_\_set('limitPrice',3);

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice','');

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity',4);

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity','');

$request\_params->\_\_set('symbolInfo',$option\_symbol\_obj);

$request\_params->\_\_set('stopLimitPrice','');

$request\_params->\_\_set('orderAction','BUY\_OPEN');

$request\_params->\_\_set('priceType','LIMIT');

$request\_params->\_\_set('marketSession','REGULAR');

$request\_params->\_\_set('orderTerm','GOOD\_FOR\_DAY');

$request\_params->\_\_set('routingDestination','');

$request\_xml\_object = **new** PlaceOptionOrder($request\_params);

$out = $ac\_obj->placeOptionOrder($request\_xml\_object);

**Preview Change Equity Order**

$request\_params = **new** changeEquityOrderRequest();

$request\_params->\_\_set('priceType','');

$request\_params->\_\_set('orderTerm','GOOD\_UNTIL\_CANCEL');

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('orderNum',162);

$request\_params->\_\_set('clientOrderId','asdfse');

$request\_params->\_\_set('limitPrice','');

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice','');

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity','');

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity','');

$request\_xml\_object = **new** PreviewChangeEquityOrder($request\_params);

$out = $ac\_obj->previewChangeEquityOrder($request\_xml\_object);

**Place Change equity Order**

$request\_params = **new** changeEquityOrderRequest();

$request\_params->\_\_set('priceType','');

$request\_params->\_\_set('orderTerm','GOOD\_UNTIL\_CANCEL');

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('orderNum',162);

$request\_params->\_\_set('clientOrderId','asdfse');

$request\_params->\_\_set('limitPrice','');

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice','');

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity','');

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity','');

$request\_xml\_object = **new** PlaceChangeEquityOrder($request\_params);

$out = $ac\_obj->placeChangeEquityOrder($request\_xml\_object);

**Preview Change Option Order**

$request\_params = **new** changeOptionOrderRequest();

$request\_params->\_\_set('stopLimitPrice','');

$request\_params->\_\_set('priceType','');

$request\_params->\_\_set('orderTerm','GOOD\_UNTIL\_CANCEL');

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('orderNum',162);

$request\_params->\_\_set('clientOrderId','asdfse');

$request\_params->\_\_set('limitPrice','');

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice','');

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity','');

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity','');

$request\_xml\_object = **new** PreviewChangeOptionOrder($request\_params);

$out = $ac\_obj->previewChangeOptionOrder($request\_xml\_object);

**Place Change Option Order**

$request\_params = **new** changeEquityOrderRequest();

$request\_params->\_\_set('priceType','');

$request\_params->\_\_set('orderTerm','GOOD\_UNTIL\_CANCEL');

$request\_params->\_\_set('accountId',83405188);

$request\_params->\_\_set('orderNum',162);

$request\_params->\_\_set('clientOrderId','asdfse');

$request\_params->\_\_set('limitPrice','');

$request\_params->\_\_set('previewId','');

$request\_params->\_\_set('stopPrice','');

$request\_params->\_\_set('allOrNone','');

$request\_params->\_\_set('quantity','');

$request\_params->\_\_set('reserveOrder','');

$request\_params->\_\_set('reserveQuantity','');

$request\_xml\_object = **new** PlaceChangeEquityOrder($request\_params);

$out = $ac\_obj->placeChangeEquityOrder($request\_xml\_object);

**Cancel Order**

$request\_params = **new** CancelOrderRequest();

$request\_params->\_\_set('accountId',83491757);

$request\_params->\_\_set('orderNum',262);

print\_r($request\_params);

$request\_xml\_object = **new** CancelOrder($request\_params);

print\_r($request\_xml\_object);

$out = $ac\_obj->cancelOrder($request\_xml\_object);