Schwab & API Security

Schwab is committed to providing the highest standards of digital security and privacy for our Clients. Using the OAuth 2 authorization framework, Schwab is able to provide secure, delegated access over HTTPS to devices, APIs, servers and applications using access tokens in place of Client credentials.

OAuth 2 Overview

Schwab employs the OAuth 2 protocol to secure services from unauthorized access. OAuth 2 is the second iteration of this IETF traditional client-server authentication framework and a current standard for RESTful API security. Our implementation adheres to current IETF standards. This open standard framework operates over HTTPS effectively replacing *username+password* with an encrypted token to access **User** data. Official IETF RFC articles may be found here:

- OAuth 2 - https://tools.ietf.org/html/rfc6749
- Bearer Token - https://tools.ietf.org/html/rfc6750

Bearer tokens are used for the OAuth **authorization_code** Grant Type

Three Legged Workflow

Three Legged OAuth is a workflow *(Flow)* that allows Users to grant an **App** permission to access to **Protected Resources**, such as their account information, without disclosing credentials. OAuth directs Users to Schwab's Login Micro Site (**LMS**) to perform the Consent and Grant (**CAG**) process. Here, the User may select and authorize the accounts they wish to be shared with the Third-Party Application *(Application)*. Upon completion of the CAG process, the User is redirected back to the Application.

Key Terms

-

App

OAuth registration is managed by Apps on the Dev Portal. Here, these Apps are owned by a Company and used internally to manage Application access to Protected Resource data. App elements and attributes include:

-

Client ID & Client Secret

These string values are unique to an App and are generated when it gets approved and registered with the OAuth server. When an App is authorized using the OAuth Flow, these elements help to identify and control what access that App has to Protected Resources data going forward. Permissions to use access an API Product is also tied to an App and its corresponding Client ID. The Client Secret will never be exposed outside of the OAuth Flow and App management in the Dev Portal. Client Secret values should always be kept confidential and stored securely by an Application.

Callback URL

OAuth uses the Callback URL ("*redirect_uri*") to redirect the User and OAuth Flow back to the Application when necessary. This will be used as the URL "host" of that Application's landing page.

- Callback URLs must be HTTPS.
- Multiple URLs are supported by separating each with a comma.
- There is a 255 character limit on this field including all URLs listed.
- Local host Callback URL can be: https://127.0.0.1

Display Name

The Display Name is established when the App is created in the Dev Portal and shown on screen to the User performing CAG activities. This helps to ensure that consent is granted to the appropriate App.

Environment

Apps may exist in either Sandbox (test data access) or the Production environment (live data access). See the "Creating an App" and "Promoting an App to Production" sections for more information.

- The Trader API Sandbox environments will be available later this year.

Product Subscription

Apps may only subscribe to a single API Product, for example, Trader API - Individual.

Third-Party Application (*User-Agent / "Application*"):

This will represent any website, stand-alone application or other HTTP platform that uses an OAuth Bearer token to access Protected Resource data on behalf of a User.
Please note – this is completely different than the "App" as defined here.

CAG - Consent and Grant

Using LMS, Schwab Users will provide their approval of Application access and select the accounts they wish to link.

LMS - Login Micro Site

A website for Users to log into Schwab directly from an Application to perform CAG activities.

LOB - Line of Business

Owner of an API Product or functional grouping of APIs in Schwab's Dev Portal. Examples: Data Aggregation Services, Tax Services, etc. Companies may request access to API Products owned by the LOB.

- **Roles**

IETF's OAuth 2 framework defines four Roles, such as the Resource Owner (User), referenced throughout this OAuth documentation.

- **User**

User is the Protected Resource Owner that authorizes Application access to their information. User may be referenced interchangeably with: a Schwab Client, the Resource Owner, End User, or App User. Further IETF OAuth definition of Resource Owner can be found at: https://tools.ietf.org/html/rfc6749#section-1.1

- **Token**

Several types of Tokens are used in OAuth 2 Flows. All Tokens are simply string values representing attributes such as scope, lifetime and other information that is used for different purposes.

  - ***Access Token:***

To enhance API security, Apps will use an Access Token to access a User's Protected Resources. This is used in place of their *username+password* combination. A Trader API access token is valid for 30 minutes after creation.

  - ***Bearer Token:***

A Bearer token is the Access Token in the context of an API call for Protected Resource data. It is passed in the Authorization header as "Bearer {access_token_value}."

  - ***Refresh Token:***

The Refresh Token renews access to a User's Protected Resources. This may be done before, or at any point after the current, valid **access_token** expires. When they do expire, the corresponding Refresh Token is used to request a new Access Token as opposed to repeating the entire Flow. This token is provided along with the initial Access Token and should be stored for later use.

A Trader API refresh token is valid for 7 days after creation. Upon expiration, a new set refresh token must be recreated using the authorization_code Grant Type authentication flow (CAG/LMS).

Three Legged Flow Entities

The primary entities involved in the Three Legged OAuth Flow are the following:

- **Resource Owner** *(User)* - Schwab Client or User that owns and grants access to Protected Resources
- **OAuth Client** *(App)* - This is the App, living in the Dev Portal. Using its Client ID and Client Secret, it requests access to Protected Resources on behalf of the User.
- **User-Agent** *(3rd-party application)* - The Resource Owner will use this application, or website, to interact with Schwab APIs and access Protected Resources
- **Authorization Server** *(OAuth server)* - OAuth server that authenticates OAuth Clients and issues Tokens
- **Resource Server** - Schwab server that hosts our Users' Protected Resources, such as financial account information

OAuth Flow - Sequence Diagram

Step 1: App Authorization

This endpoint authorizes a specific App to access to Protected Resources on behalf of the Resource Owner (User).

An Application passes the parameters of a registered App to direct the Flow to LMS. Once CAG activities are completed in LMS, an Authorization Code ("code") is returned in the landing URL following a redirect. The "code" parameter will be used in Step 2 to create the initial set of Refresh and Access Tokens. The Callback URL will be the "host" of this Application's landing page.

Following the CAG activities:

- An Authorization Code will be provided and can be exchanged for an Access Token in the next step.
- This Access Token can be used to call API Product endpoints for Protected Resource data after rest of this Flow is completed. An Access Token is valid for 30 minutes on the Trader API.
- Once a Refresh Token is invalidated or expired, the CAG activities will need to be completed again to restart the OAuth flow. A Refresh Token is valid for 7 days.

Request Template - Authorization URL

https://api.schwabapi.com/v1/oauth/authorize?client_id={CONSUMER _KEY}&redirect_uri={APP_CALLBACK_URL}

Response Template - Final landing URL

https://{APP_CALLBACK_URL}/?code={AUTHORIZATION_CODE_GENERATED}&session={SESSION_ID}

The website will be redirected to a 404 page, but the address bar will contain "code" needed for the next step.

Step 2: Access Token Creation

POST https://api.schwabapi.com/v1/oauth/token

This first POST call to the //oauth/token endpoint exchanges the **"code" (authorization_code)**, returned above, for the initial **"access_token"**. This is used to access Protected Resources from an API Product. An Access Token is valid for 30 minutes on the Trader API.

The "code" within this request must be URL decoded prior to making the request. For example, this should end in '@' instead of '%40' when used in the request.

"Access Token" - Request Example (CURL)

```
{curl -X POST \https://api.schwabapi.com/v1/oauth/token \-H 'Authorization: Basic
{BASE64_ENCODED_Client_ID:Client_Secret} \-H 'Content-Type: application/x-www-form-urlencoded'
\-d 'grant_type=authorization_code&code=
{AUTHORIZATION_CODE_VALUE}&redirect_uri=https://example_url.com/callback_example'}
```

Response Example (body)
Example - Access Token Response

```
{  "expires_in": 1800,  //Number of seconds access_token is valid for  "token_type": "Bearer",
"scope": "api",  "refresh_token": "{REFRESH_TOKEN_HERE}", //Valid for 7 days  "access_token":
"{ACCESS_TOKEN_HERE}", //Valid for 30 minutes  "id_token": "{JWT_HERE}"}
```

A Trader API access token is valid for 30 minutes. A Trader API refresh token is valid for 7 days.

Step 3: Make an API Call

API Product calls use the following authorization header format:
Authorization: Bearer {access_token}
The Application supplies the access_token value after the Bearer keyword like the example below:
Authorization: Bearer I0.kC95zyI039S-YTEw=

Step 4: Refresh an Access Token (with existing Refresh Token)

POST https://api.schwabapi.com/v1/oauth/token

An OAuth Refresh Token functionality renews access to a User's Protected Resources before, or soon after, the current **access_token** expires.

"Refresh Token" - Request Example (cURL)

```
curl -X POST \https://api.schwabapi.com/v1/oauth/token \-H 'Authorization: Basic
{BASE64_ENCODED_Client_ID:Client_Secret} \-H 'Content-Type: application/x-www-form-urlencoded'
\-d 'grant_type=refresh_token&refresh_token={REFRESH_TOKEN_GENERATED_FROM_PRIOR_STEP}
```

Response Example (body)
Example - Refresh Token Response

```
{  "expires_in": 1800,  //Number of seconds access_token is valid for  "token_type": "Bearer",
"scope": "api",  "refresh_token": "{REFRESH_TOKEN_HERE}", //Valid for 7 days  "access_token":
"{NEW_ACCESS_TOKEN_HERE}",//Valid for 30 minutes  "id_token": "{JWT_HERE}"}
```

Should I Refresh or Restart OAuth?

The Refresh Token step (Step 4) can be executed *before an Access Token expires*. Certain conditions and edge-cases exist where your application may need to restart the OAuth Flow as opposed to attempting to use the "Refresh Token" step.

The Refresh Token step (Step 4) will no longer be available once a Refresh Token is expired after 7 days or invalidated (e.g., User password reset). If the refresh token is no longer valid, App Authorization (Step 1) and Access Token Creation (Step 2) must be repeated to restart the OAuth Flow.

Place Order Samples

Below, you will find examples specific to orders for use in the Schwab Trader API POST and PUT Order endpoints. Order entry will only be available for the assetType 'EQUIT' and 'OPTION as of this time.

Trader API applications (Individual and Commercial) are limited in the number of PUT/POST/DELETE order requests per minute per account based on the properties of the application specified during registration or follow-up process. Throttle limits for orders can be set from zero (0) to 120 requests per minute per account. Get order requests are unthrottled. Contact TraderAPI@schwab.com for further information.

Options and their Symbology:
Options symbols are broken down as:
Underlying Symbol (6 characters including spaces) | Expiration (6 characters) | Call/Put (1 character) | Strike Price (5+3=8 characters)

Option Symbol: XYZ 210115C00050000
Stock Symbol: XYZ
Expiration: 2021/01/15
Type: Call
Strike Price: $50.00

Option Symbol: XYZ 210115C00055000
Stock Symbol: XYZ
Expiration: 2021/01/15
Type: Call
Strike Price: $55.00

Option Symbol: XYZ 210115C00062500
Stock Symbol: XYZ
Expiration: 2021/01/15
Type: Call
Strike Price: $62.50

Instruction for EQUITY and OPTION

Instruction	EQUITY (Stocks and ETFs)	Option
BUY	ACCEPTED	REJECT

SELL	ACCEPTED	REJECT
BUY_TO_OPEN	REJECT	ACCEPTED
BUY_TO_COVER	ACCEPTED	REJECT
BUY_TO_CLOSE	REJECT	ACCEPTED
SELL_TO_OPEN	REJECT	ACCEPTED
SELL_SHORT	ACCEPTED	REJECT
SELL_TO_CLOSE	REJECT	ACCEPTED

Buy Market: Stock

Buy 15 shares of XYZ at the Market good for the Day.

```
{  "orderType": "MARKET",  "session": "NORMAL",  "duration": "DAY",  "orderStrategyType":
"SINGLE",  "orderLegCollection": [    {      "instruction": "BUY",      "quantity": 15,
"instrument": {        "symbol": "XYZ",        "assetType": "EQUITY"      }    }  ]}
```

Buy Limit: Single Option

Buy to open 10 contracts of the XYZ March 15, 2024 $50 CALL at a Limit of $6.45 good for the Day.

```
{  "complexOrderStrategyType": "NONE",  "orderType": "LIMIT",  "session": "NORMAL",  "price":
"6.45",  "duration": "DAY",  "orderStrategyType": "SINGLE",  "orderLegCollection": [    {
"instruction": "BUY_TO_OPEN",      "quantity": 10,      "instrument": {        "symbol": "XYZ
240315C00500000",        "assetType": "OPTION"         }    }  ]}
```

Buy Limit: Vertical Call Spread

Buy to open 2 contracts of the XYZ March 15, 2024 $43 Put and Sell to open 1 contract of the XYZ March 15, 2024 $45 Put at the Market good for the Day.

```
{ "orderStrategyType": "SINGLE",  "orderType": "MARKET",  "orderLegCollection": [    {
"instrument": {        "assetType": "OPTION",        "symbol": "XYZ   240315P00043000"    },
"instruction": "SELL_TO_OPEN",      "quantity": 1    },    {      "instrument": {
"assetType": "OPTION",        "symbol": "XYZ   240315P00045000"      },      "instruction":
"BUY_TO_OPEN",      "quantity": 2    }  ],  "complexOrderStrategyType": "CUSTOM",  "duration":
"DAY",  "session": "NORMAL"}
```

Conditional Order: One Triggers Another

Buy 10 shares of XYZ at a Limit price of $34.97 good for the Day. If filled, immediately submit an order to Sell 10 shares of XYZ with a Limit price of $42.03 good for the Day. Also known as 1st Trigger Sequence.

```
{  "orderType": "LIMIT",  "session": "NORMAL",  "price": "34.97",  "duration": "DAY",
"orderStrategyType": "TRIGGER",  "orderLegCollection": [    {      "instruction": "BUY",
"quantity": 10,      "instrument": {        "symbol": "XYZ",        "assetType": "EQUITY"
}    }  ],  "childOrderStrategies": [    {      "orderType": "LIMIT",      "session":
"NORMAL",      "price": "42.03",      "duration": "DAY",      "orderStrategyType": "SINGLE",
"orderLegCollection": [        {          "instruction": "SELL",          "quantity": 10,
"instrument": {            "symbol": "XYZ",            "assetType": "EQUITY"          }
}      ]    }  ]}
```

Conditional Order: One Cancels Another

Sell 2 shares of XYZ at a Limit price of $45.97 and Sell 2 shares of XYZ with a Stop Limit order where the stop price is $37.03 and limit is $37.00. Both orders are sent at the same time. If one order fills, the other order is immediately cancelled. Both orders are good for the Day. Also known as an OCO order.

```
{  "orderStrategyType": "OCO",  "childOrderStrategies": [    {      "orderType": "LIMIT",
"session": "NORMAL",      "price": "45.97",      "duration": "DAY",      "orderStrategyType":
"SINGLE",      "orderLegCollection": [        {          "instruction": "SELL",
"quantity": 2,          "instrument": {            "symbol": "XYZ",            "assetType":
"EQUITY"          }        }      ]    },    {      "orderType": "STOP_LIMIT",      "session":
"NORMAL",      "price": "37.00",      "stopPrice": "37.03",      "duration": "DAY",
"orderStrategyType": "SINGLE",      "orderLegCollection": [        {          "instruction":
"SELL",          "quantity": 2,          "instrument": {            "symbol": "XYZ",
"assetType": "EQUITY"          }        }      ]    }  ]}
```

Conditional Order: One Triggers A One Cancels Another

Buy 5 shares of XYZ at a Limit price of $14.97 good for the Day. Once filled, 2 sell orders are immediately sent: Sell 5 shares of XYZ at a Limit price of $15.27 and Sell 5 shares of XYZ with a Stop order where the stop price is $11.27. If one of the sell orders fill, the other order is immediately cancelled. Both Sell orders are Good till Cancel. Also known as a 1st Trigger OCO order.

```
{  "orderStrategyType": "TRIGGER",  "session": "NORMAL",  "duration": "DAY",  "orderType":
"LIMIT",  "price": 14.97,  "orderLegCollection": [    {      "instruction": "BUY",
"quantity": 5,      "instrument": {        "assetType": "EQUITY",        "symbol": "XYZ"
}    }  ],  "childOrderStrategies": [    {      "orderStrategyType": "OCO",
"childOrderStrategies": [        {          "orderStrategyType": "SINGLE",          "session":
"NORMAL",          "duration": "GOOD_TILL_CANCEL",          "orderType": "LIMIT",
"price": 15.27,          "orderLegCollection": [            {              "instruction":
"SELL",              "quantity": 5,              "instrument": {                "assetType":
"EQUITY",                "symbol": "XYZ"              }            }          ]        },
{          "orderStrategyType": "SINGLE",          "session": "NORMAL",          "duration":
"GOOD_TILL_CANCEL",          "orderType": "STOP",          "stopPrice": 11.27,
"orderLegCollection": [            {              "instruction": "SELL",
"quantity": 5,              "instrument": {                "assetType": "EQUITY",
"symbol": "XYZ"              }            }          ]        }      ]    }  ]}
```

Sell Trailing Stop: Stock

Sell 10 shares of XYZ with a Trailing Stop where the trail is a -$10 offset from the time the order is submitted. As the stock price goes up, the -$10 trailing offset will follow. If stock XYZ goes from $110

to $130, your trail will automatically be adjusted to $120. If XYZ falls to $120 or below, a Market order is submitted. This order is good for the Day.

```
{  "complexOrderStrategyType": "NONE",  "orderType": "TRAILING_STOP",  "session": "NORMAL",
"stopPriceLinkBasis": "BID",  "stopPriceLinkType": "VALUE",  "stopPriceOffset": 10,
"duration": "DAY",  "orderStrategyType": "SINGLE",  "orderLegCollection": [    {
"instruction": "SELL",      "quantity": 10,      "instrument": {        "symbol": "XYZ",
```