Vendor Services in API-NG

When & How to Use Vendor Services

There are two separate methods available to Vendors.

- **Vendor Services API** - Available to licensed software vendors who want to provide Betfair Exchange customers access to their certified software using a subscription based system.
- **Vendor Web API** - Available to licensed software vendors who want to develop **web applications** using the Betfair API for access by Betfair Exchange customers.

Vendor Services API

The Vendor Services API allows licensed and certified Software Vendors to manage the permissioning of Betfair accounts to their API-NG application/s.

With the Vendor Services API, you can:

- Grant permission for users to access your application
- Set the permission to expire after a specific date
- Store arbitrary information against each subscription
- View existing and historical application subscriptions

- For details of the operations to manage subscriptions, please see the Accounts section of the API-NG Reference Guide.

There are two key processes that Vendor will need to following when using the Vendor Services in API-NG:

- Subscribing a Customer.
- Checking a Customers Subscription History.

Each of these processes are detailed below.

Subscribing a Customer

Preconditions:

- Customer has a Betfair account.
- Vendor has a Betfair vendor account, the App Key for their app, and their (Vendor) session token

Example flow:

1. Customer visits Vendor's web site and decides to sign up for the Vendor's app.
2. The Customer purchases a 12 month subscription to the app from the Vendor's web site.
3. The Vendor's website server then calls the Account API-NG operation **getApplicationSubscriptionToken**, passing in the Vendor's App

Key, a valid Session Token for the Vendor's Betfair account (to prove that they own that App Key) and the length of the subscription required (**365 days in this scenario**).

4. Betfair returns a Subscription Token in the form of **ABCD-EFGH-IJKL** to the Vendor
5. The Subscription Token is then provided to the instance of the Vendor's app used by the Customer.

   This may happen in a number of ways, including:

   i) Emailed from Vendor to Customer, who then types the Subscription Token into the app.
   ii) Associated with the Customer's app instance by the Vendor, so that the token can be cited by the app based on Customer login to the app.

6. The Vendors app requires the Customer to login to Betfair using the Interactive Login method. See sample code Interactive Login sample code here
7. Betfair returns Customer's Session Token to the app.
8. The subscription is then activated by the app calling **activateApplicationSubscription** passing in the **Subscription Token** and a valid **Session Token** for the Customer. **Please note:** the X-Application header is not required for the **activateApplicationSubscription** request.

Scenario 1: Customer signs up for a subscription to a Vendor's app
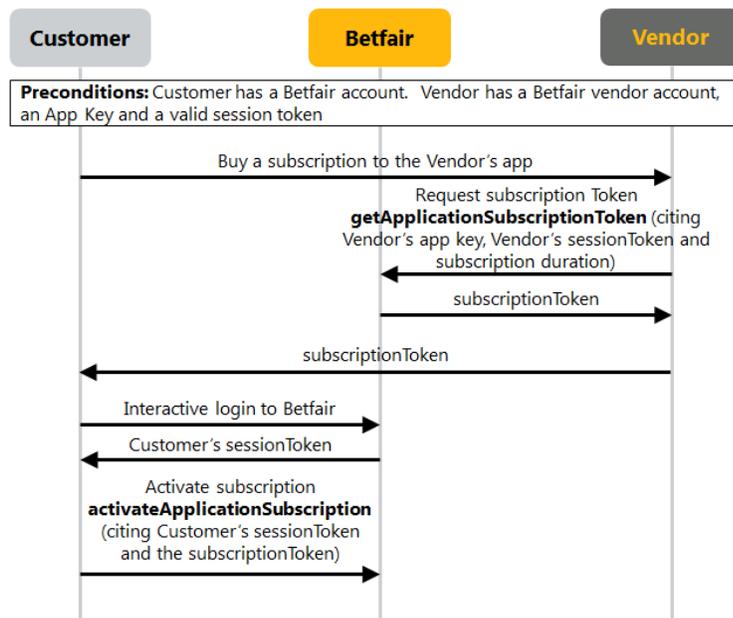
Checking a Customers Subscription History

In this scenario the Vendor wishes to check the customer's subscription status to establish if the customer has:

1. An existing active subscription.
2. An expired subscription.
3. A cancelled subscription.
4. No current or previous subscription history (i.e. Customer is entirely new).

This can be done within the application itself (client side) or by Vendor on their app server (server side). Each process is explained in more detail below:

Client side process:

1. Vendor's app client requires Customer to login to Betfair using the Interactive Login method.
2. Betfair returns Customer's Session Token to the Vendor's app client.
3. Vendor's app client calls **getApplicationSubscriptionHistory** citing Vendors App Key in the request body and the customer's Session

Token in the X-Authentication header.

4. Betfair returns Customer's subscription history. If an empty list is returned then the customer has no current or previous subscription history (i.e. Customer is entirely new)
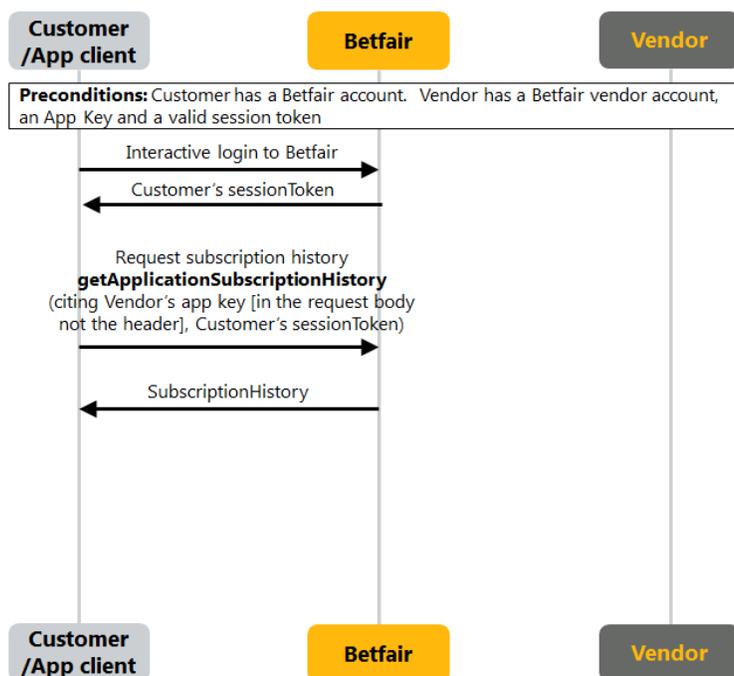
Server side process:

1. Vendor's app client requires Customer to login to Betfair using Interactive Login method.
2. Betfair returns Customer's Session Token to the Vendor's app client.
3. Vendor's app client calls **getVendorClientId** citing Customer's session token from step 1.
4. Betfair returns Customer's vendorClientId.
5. Vendor's App client sends Customer's vendorClientId to Vendor's app server.
6. Vendor's app server calls **getApplicationSubscriptionHistory** citing Vendor's Session Token, Vendor's App Key, and the vendorClientID from step 4.
7. Betfair returns Customer's complete history for the Vendor's app (as identified by the app key cited in step 5). If an empty list is returned then the customer has no current or previous subscription history (i.e. Customer is entirely new).

Scenario 3: Vendor wishes to check a Customer's subscription history (using Vendor's App Server)
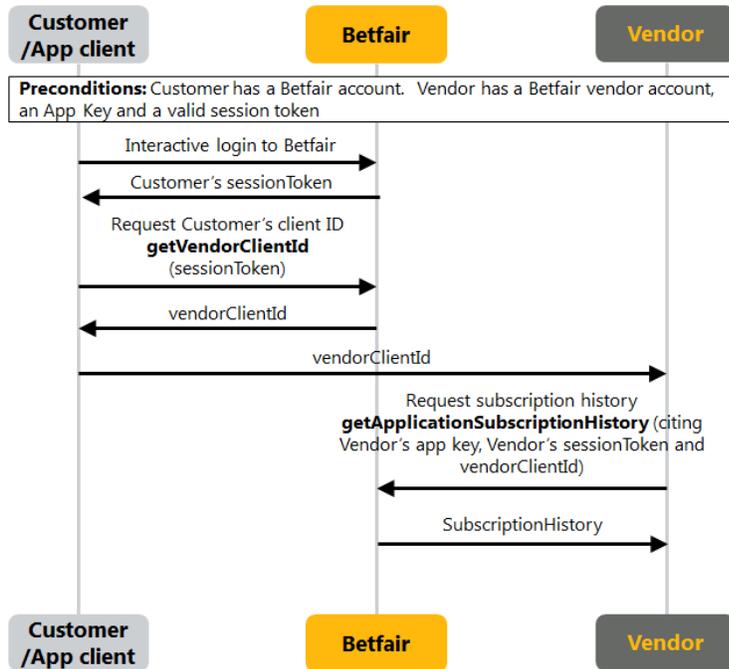
Vendor Web API

The Vendor Web API is available to licensed Software Vendors who are creating **web based applications** These operations enable the web application to carry out operations on the users behalf using the OAuth2 protocol.

Operations

The following operations are available via the Vendor Web API

- token
- revokeAccessToWebApp
- isAccountSubscribedToWebApp

OAuth 2 Flow - Creating an Access Token

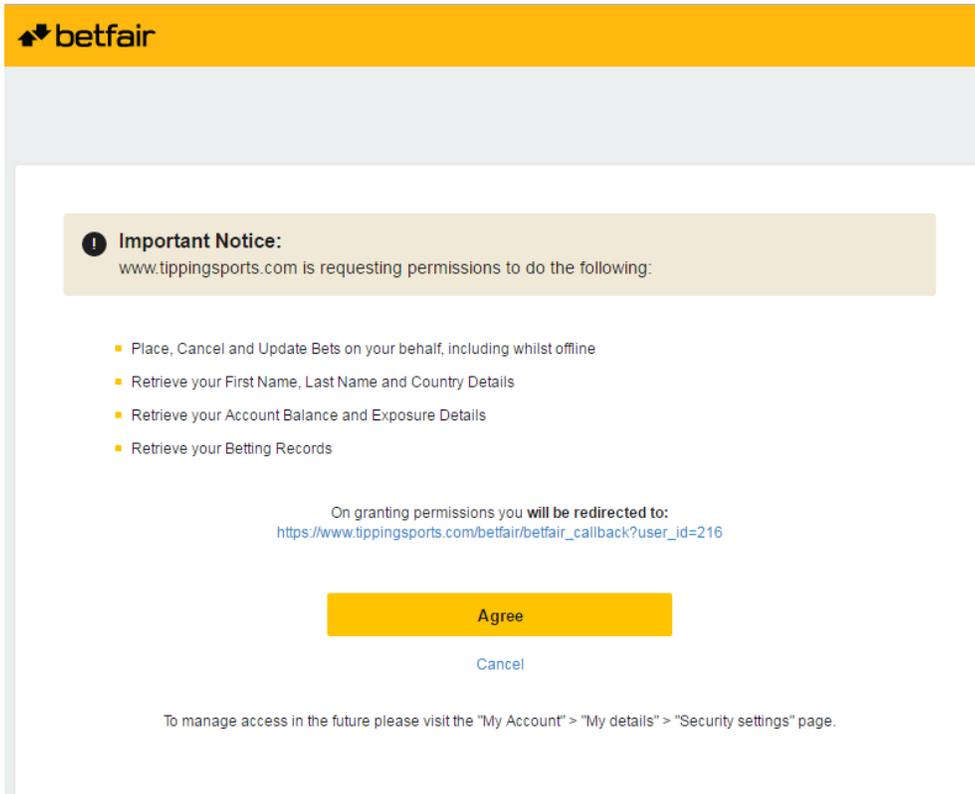The user authentication via web apps is underpinned by the OAuth 2 protocol.

You will need to redirect the user to the Betfair login page (identitysso.betfair.com/view/vendor-login), along with up to three parameters:

- **client_id (mandatory):** this is your vendor ID, returned to you by the **getDeveloperAppKeys** call provided you have been registered as a web vendor
- **response_type**: this needs to be set to **'code'**, to indicate you are requesting an authorization code (see later)
- **redirect_uri (optional):** this will be appended to the URL you provided **upon sign up to the Vendor Program** when we redirect the user back to your site after login. This needs to be URL encoded.

Example call - Creating An Access Token

```
https://identitysso.betfair.com/view/vendor-login?client_id=123456&response_type=code&redirect_uri=newjoiner
```

The user will log in and be presented with a message asking them to allow you to perform Betfair calls on their behalf, as shown below.

Should they accept this message, they will be redirected to your site (using your **redirect URL + any optional parameters**). Critically, we will include an **authorization code** as a parameter, under the parameter name "code".

NOTE: The **authorization code** will be valid for a single use for 10 minutes.

Example redirect

```
https://mywebsite.com/newjoiner?code=12345
```

You will need to propagate this code to your back-end, from which you will have to exchange it for an access token.

The **access token** will allow you to use the Betfair API on the user's behalf. To obtain the **access token** you will need to call the token operation on the Accounts API

Example Token Request

The token call takes the following parameters:

Request Headers

- **Header**: 'X-Application' : 'your app key'
- **Header**: 'X-Authentication' : 'your session token'

Request Body

- **client_id:** your vendor ID
- **grant_type**: in this context, this will have to be set to 'AUTHORIZATION_CODE'

- **code**: the code you were returned
- **client_secret**: your secret, obtained from GetDeveloperAppKeys

Response Parameters

- **access_token**: the access token, used to call Betfair on the user's behalf
- **token_type**: meta data for the access token (see 'Making calls on the user's behalf')
- **expires_in**: how long the access token will be valid for (in seconds)
- **refresh_token**: a token that can be used to create a new access token (see 'Using the refresh token')
- **application_subscription**: contains the vendor client ID, a unique identifier for a user. Can also contain some subscription related information (See 'Legacy Subscriptions')

IMPORTANT
To protect sensitive information such as your app key and secret, it is important that the token operation only be called from server to server.

Example token request - REST

```
X-Application: 'your App key'
X-Authentication : 'your session token'
Accept:  application/json
Content-Type:  application/json


Endpoint


https://api.betfair.com/exchange/account/rest/v1.0/token/


Request Body


{"client_id"="4534","grant_type":"AUTHORIZATION_CODE","code":"-22a1-121510
08-000007cb61","client_secret":"bc183d-f5-40dc-82a6-d97681


Response


{"access_token":"KeOi+kyg2RvDK4HM+W46CvSnP5w=","refresh_token":"50d76117-7
f85-375v-a38f-ffb332713f93","application_subscription":{"vendor_client_id"
:"456238"},"token_type":"BEARER","expires_in":"14400"}
```

```
                    Example token request - JSON

X-Application: 'your App key'
X-Authentication : 'your session token'
Accept:  application/json
Content-Type:  application/json


Endpoint


https://api.betfair.com/exchange/account/json-rpc/v1


Request


{"jsonrpc": "2.0", "method": "AccountAPING/v1.0/token", "params":
{"client_id":"CLIENTID","grant_type":"AUTHORIZATION_CODE","code":"CODE","c
lient_secret":"CLIENTSECRET"}, "id": 1 }
```

OAuth 2 Flow - Using the Refresh Token

When the access token expires, it is possible to create a new one without any user input, using the refresh token. This is done with the same call that was used to create it originally, the token operation, but with a different set of parameters, e.g grant_type REFRESH_TOKEN.

Header: 'X-Application' : 'your app key'

Header: 'X-Authentication' : 'your session token'

client_id: your vendor ID, obtained from GetDeveloperAppKeys

grant_type: 'REFRESH_TOKEN'

refresh_token: the refresh token for the **user**

client_secret: your secret, obtained from GetDeveloperAppKeys

This will return the same information as the original call:

access_token: the access token, used to call Betfair on the user's behalf

token_type: meta data for the access token (see 'Making calls on the user's behalf')

expires_in: how long the access token will be valid for (in seconds)

refresh_token: the refresh token remains the same

application_subscription: contains the vendor client ID, a unique identifier for a user

```
                    Example Request - REST

https://api.betfair.com/exchange/account/rest/v1.0/token?client_id=VENDOR_
ID&client_secret=VENDOR_SECRET&grant_type=REFRESH_TOKEN&refresh_token=50d7
6117-7f85-375v-a38f-ffb332713f93
```

You can user the 'expires_in' value to determine when the access token will stop being valid. Alternatively, if calls made with the access token start returning an INVALID_SESSION error, it is likely that the token has expired.

Making API Calls On The Users Behalf

Once you have an access token, you can start calling the Betfair API on the user's behalf based on their actions on your site.

You will need to populate headers in a different way to a standard API call:

X-Application: your application key

Authorization: token_type + " " + access_token

The Authorization header needs to be a concatenation of the token type and the access token (both returned by the token call), separated by a space.

Example:

Betfair will use the access token to determine which user the calls are being made for.

These calls also have to be from back-end to back-end.

> The getAccountStatement operation isn't available via the Vendor Web API

User Revocation

The user may choose to revoke the permissions previously granted to your web application. This will invalidate the access token and destroy your refresh token. Any subsequent calls to Betfair using the access token, or any attempt to generate a new one using the refresh token will result in either an **INVALID_SESSION** exception or **UNEXPECTED_ERROR** respectively.

Legacy Subscriptions

The way subscriptions are handled for web applications differ greatly from the way they were for desktop based applications. The subscription token model is no longer enforced, however you may still choose to create and manage subscriptions using the existing API calls.

Please note that in this model, Betfair will no-longer prevent requests for a user with an expired or cancelled subscription. You can choose what action to take based on information contained in the application_subscription field returned by the token operation.

The available operations are the following:

- getApplicationSubscriptionToken
- activateApplicationSubscription
- cancelApplicationSubscription
- updateApplicationSubscription
- listApplicationSubscriptionTokens
- getApplicationSubscriptionHistory

If you do use these calls to manage subscriptions, the token call will return information on the most relevant subscription (i.e. the subscription with the latest expiration date) as part of the application_subscription object.

Flow Diagram
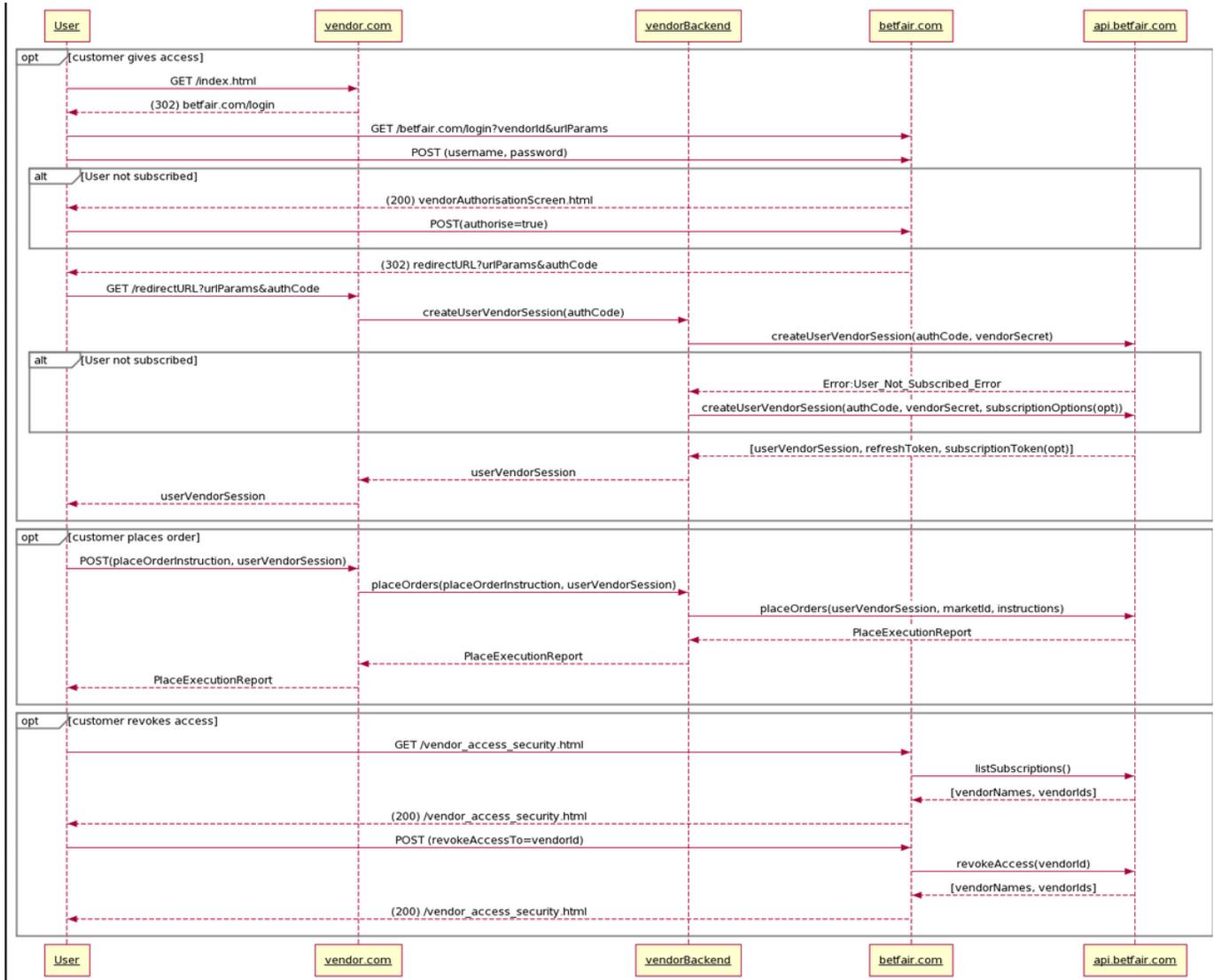
Sample Code

Sample code that demonstrates the web apps interaction is available via https://github.com/betfair/sample-web-app-vendor/