Authenticating to the Black Diamond API

Authorization Overview

The Black Diamond API uses the Authorization Code Flow from the OAuth2.0 specification for authentication and authorization. This eliminates the need to store any user-level credentials and is the ideal solution for SSO-only users across multiple, integrated platforms**.** If you are not able to support the Authorization Code Flow we can also support the resource owner password credentials grant, the details of which can be found here.

OAuth is an open standard which provides client applications with secure authorization to server resources. Black Diamond's OpenID configuration schema can be referenced here: https://login.bdreporting.com/.well-known/openid-configuration

Authorization Process

Criteria needed to properly authenticate and authorize with Black Diamond:

- **Client ID** - The client ID is an identifier that represents the client application accessing a server resource. This is similar to a person's username but is at the integration level. This is the same for all API calls by a partner or client.
- **Client Secret** - Client Secret is similar to a password and should be treated as such. It is used to authenticate a client application at the Client ID level.
- **Registered Redirect URI** - This is the endpoint where Black Diamond will post the code that can be traded for an access token. Multiple redirect URI's can be held on a single Client ID. It must be a fully-qualified domain name and be a https address
- **State** - This can be any value that represents and is unique to the end-user attempting to authorize.

Step 1: Authorize the User

Send the user to authorize against Black Diamond's identity server with a GET call to the endpoint below with additional query parameters. If the user is already logged into Black Diamond with an active session, they will not be prompted for credentials.

GET: https://login.bdreporting.com/connect/authorize

Query Parameters should include:

- response_type = code
- client_id =
- state =
- scope = offline_access+api
- redirect_uri =

Sample request URL:

https://login.bdreporting.com/connect/authorize?response_type=code&client_id=ClientID&state=YourSpecifiedState&scope=offline_access+api&redirect_uri=YourRedirectURI

Step 2: Black Diamond Returns a Code to the Redirect URI

If the user successfully authenticates, Black Diamond will send a code back to the redirect URI specified in the authorization call, along with the scope and state (to identify which user the code relates to).

If the user is already logged into Black Diamond with an active session, they will not be prompted for credentials and the code will be sent back to the redirect URI.

Sample GET to callback:

https://oauth.pstmn.io/v1/callback?code=FD549B3DA94716F85BC6940122383E777F1683D6AE319791A13FC97BD2AD2221&scope=offline_access+api&state=Test

Step 3: Trade the Code for an Access Token

Once the code is returned, use it to obtain an access token for the user by making a POST call to:

https://login.bdreporting.com/connect/token

Request Header should include:

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

The request body should include the following key-value pairs:

- grant_type = authorization_code
- client_id = xxxxxxxxx
- client_secret = xxxxxxxxx
- code = {code_from_redirect}
- redirect_uri = registered redirect URI

The response body will contain an access token and refresh token, such as:

{

"access_token": "10385E4673CFDD99D9313FF063EF8608F29BEA480B36B80E8BD5E0E9F2A74C97",

"expires_in": 600,

"token_type": "Bearer",

"refresh_token": "80D17602BFC6BA034F4365827C9FB3A7EA47A68924124A12A716B557B838C33C",

"scope": "offline_access api"

}

Step 4: Make an API Call

Using the access token received in the previous step and your unique Subscription Key from the developer portal you can call any available Black Diamond API endpoint. Your subscription key can be found in the My Profile page, and either the primary or secondary keys can be used. In each call to the API you will need to include two headers:

- **Bearer Token:** The client must use the "Bearer" authentication scheme, in order to supply the access token. The "Bearer" authentication scheme consists of supplying a valid access token, prepended by the word "Bearer" and a space, as the authorization header's value. Please refer to RFC6750 for further information on the authorization request header field.
- **Subscription Key:** In addition to the Authorization header, each call must contain a header that passes your primary or secondary subscription key for your account on the Black Diamond developer portal. The format of the header must include "Ocp-Apim-Subscription-Key" as the key and your subscription key as the value.

Example Headers:

KEY	VALUE
Authorization	Bearer 07F2470EDE05C3764A4C6B58FD8934E54FE81464449DC90AD34ADF3863566CB3
Ocp-Apim-Subscription-Key	a1b4a536e65db4bce8c098cdb148276de

Things to Note

- The access token you receive is valid for 10 minutes.
- The refresh tokens received in the response in step 3 are credentials which can be used to retrieve a new access token.
- The refresh token is valid for 14 days and can be used to generate a new access token after the current access token expires. See section below.
- When the refresh token is used to acquire a new access token, the call returns both an access token (valid for 10 minutes) and a brand new refresh token (valid for 14 days). You must store and use the new refresh token going forward as the original refresh token expires when a new one is generated.
    - If the refresh token expires before a new one is generated, the user would need to re-authorize through the Authorization Code flow.

Authorization Using a Refresh Token:

The lifetime of an access token is limited. The expiration time will be indicated in the response from the Black Diamond Authentication Server. Along with the expiration time, the response will contain a refresh token. Refresh tokens are credentials which can be used to retrieve future new access tokens instead of making the user authenticate again. For more information on refresh tokens, please refer to RFC6749 Section 1.5.

Once the access token expires, use the refresh token to obtain a new access token by making a POST call to:

https://login.bdreporting.com/connect/token

The request body should include:

- grant_type = refresh_token
- client_id = xxxxxxxxx
- client_secret = xxxxxxxxx
- refresh_token – xxxxxxxxx

The response body will contain an access token and refresh token, such as:

{

"access_token": "10385E4673CFDD99D9313FF063EF8608F29BEA480B36B80E8BD5E0E9F2A74C98",

"expires_in": 600,

"token_type": "Bearer",

"refresh_token": "80D17602BFC6BA034F4365827C9FB3A7EA47A68924124A12A716B557B838C33B",

"scope": "offline_access api"

}

Things to Note

- The access token is valid for 10 minutes.
- The refresh token returned in the response is valid for 14 days and can be used to generate a new access token after the access token expires
- ****Important Note**** When the refresh token is used to acquire a new access token, the call returns both an access token (valid for 10 minutes) and a refresh token (valid for 14 days). **The refresh token has a rolling 14 day window so generating a new one is not necessary until the previous one has expired after 14 days on inactivity.**