BlueFire donations

BlueFire Donations
oAuth 2.0 Integration

The BlueFire oAuth integration allows you to create a custom payment solution where users can make donations and payments via the Sale API using their billing information stored in BlueFire. If you're new to oAuth 2.0, there is quite a bit of great information on the internet.

We implement two of the authentication flows described under oAuth 2.0. The first is for server-to-server applications and the second is client-side (JavaScript) applications.

In order to use oAuth, you must first create an API key in the BlueFire administration panel for your account. The API key must have permissions for oAuth and you must set the redirect URL and referrer hosts (for client-side applications).

Step 1

The first step is to send the user to BlueFire to login and give your web application permission to use their billing information.

URL Endpoint

https://bluefire-secure.com/go/accounts/oauth/

GET Parameters

Name	Type	Default	Description
client_id	String	None (field req.)	This is your oAuth client ID (can be retrieved by clicking "oAuth Info" on the API keys administration page).
redirect_uri	String	None (field req.)	This is the location of the redirect after user authentication and approval (or denial) and it must match one of the redirect URLs you've set for your API key.
response_type	String	'code'	What type of authorization you wish to receive, should be either 'code' (for server-to-server applications) or 'token' (for client-side applications).

Tip: Append the GET parameter 'reg' (with no value) to the URL in order to initially show the new-account registration page instead of the existing-account login page. While users can navigate between the login/register pages easily, it may be helpful to provide two different links to your users.

Step 2

After logging in the user is redirected to your site at the *redirect_uri* you specified in step 1. As a part of the redirect there are parameters added to the URI. The way these parameters are added is dependent on the *response_type* you've selected.

Returned Data (for *response_type* 'code')

When the user is redirected back to your website, after logging in, the following data is returned as query-string parameters.

Name	Type	Description
code	String	If the user authorized your application access, the code is provided here.
error	String	If there was an error, it is indicated here.

An example return URL after a user logs in:

https://yoursite.com/return/?code=CODE_WILL_BE_HERE

An example return URL if the user denies your application access:

https://yoursite.com/return/?error=access_denied

Returned Data (for *response_type* 'token')

When the user is redirected back to your website, after logging in, the following data is returned as parameters after the hash.

Name	Type	Description
access_token	String	If the user authorized your application access, the *access_token* is provided for use in step 4.
expires_in	Int	How long the *access_token* is valid for in seconds. If a transaction is not completed before the *access_token* expires, the user will have to re-authorize your application (step 1) and grant you a new code.
error	String	If there was an error, it is indicated here.

An example return URL after a user logs in:

https://yoursite.com/return/#access_token=TOKEN_WILL_BE_HERE&expires_in=86400

An example return URL if the user denies your application access:

https://yoursite.com/return/#error=access_denied

Step 3

This step is only relevant for a *response_type* of 'code'. In this step the server exchanges the returned code for an access token. To do this a POST request is sent to the BlueFire API oauth-token endpoint.

URL Endpoint

https://api.bluefire-secure.com/oauth-token

POST Parameters

Name	Type	Default	Description
code	String	None (field req.)	This is the code received in step 2.
client_id	String	None (field req.)	This is your oAuth client ID (can be retrieved by clicking "oAuth Info" on the API keys administration page).
client_secret	String	None (field req.)	This is your oAuth client secret (can be retrieved by clicking "oAuth Info" on the API keys administration page).

Returned Data

After performing the POST, the following data is returned JSON encoded.

Name	Type	Description
access_token	String	The *access_token* to be used in step 4.
expires_in	Int	How long the *access_token* is valid for in seconds. If a transaction is not completed before the *access_token* expires, the user will have to re-authorize your application (step 1) and grant you a new code.

Step 4

The server or JavaScript application can now retrieve the user's stored billing information by retrieving it from the API User endpoint with the access token. Please refer to the API User documentation for more information.

Step 5

The giving form can be modified to display the user's stored billing information (retrieved in step 4). If the user would like to submit the donation or payment to be billed to the stored information, simply pass the 'userToken' and 'userPaymentKey' from step 4 to the API Sale endpoint, as described in the documentation.

If the user does not have stored billing information we recommend still passing the `userToken` to the API Sale endpoint. This will make sure the transaction is still associated with the user (and show in their history) and BlueFire will make their new billing information available in the future.

We also recommend given the user two options after they are logged in: (1) The ability to log out (by causing your application to forget the *access_token*) and (2) a link to view their account settings and history on BlueFire (https://bluefire-secure.com/go/accounts/).