OAuth2 Authentication

OAuth2 is a protocol that allows applications to interact with blogs on WordPress.com and self-hosted WordPress sites running Jetpack. The primary goal of OAuth is to allow developers to interact with WordPress.com and Jetpack sites without requiring them to store sensitive credentials. Our implementation also allows users to manage their own connections.

If you are new to the world of OAuth, you can read more at http://oauth.net. If you are already familiar with OAuth, then all you really need to know about are the two authentication endpoints: the authorization endpoint and the token request endpoint. These endpoints are `https://public-api.wordpress.com/oauth2/authorize` and `https://public-api.wordpress.com/oauth2/token`

The same endpoints are used for WordPress.com blogs and Jetpack sites. Before you begin to develop an application, you will need a client id, redirect URI, and a client secret key. These details will be used to authenticate your application and verify that the API calls being made are valid. You can create an application or view details for your existing applications with our applications manager.

Receiving an Access Token

To act on a user's behalf and make calls from our API you will need an access token. To get an access token you need to go through the access token flow and prompt the user to authorize your application to act on their behalf.

Access tokens can be requested per blog per user or as a global token per user. In addition to the global tokens, there are certain endpoints (e.g. likes and follows) where you can use a user's token on any blog to act on their behalf.

On this page:

To begin, you will need to send the user to the authorization endpoint. Here's an example request:

```
https://public-api.wordpress.com/oauth2/authorize?
client_id=your_client_id&redirect_uri=&response_type=code&blog=1234
```

Required parameters:

- `client_id` should be set to your application's client ID as found in the applications manager.
- `redirect_uri` should be set to the URL that the user will be redirected back to after the request is authorized. The `redirect_uri` must match the one in the applications manager.
- `response_type` can be "code" or "token". "Code" should be used for server side applications where you can guarantee that secrets will be stored securely. These tokens do not expire. "Token" should be used for client-side applications. This is called "Implicit OAuth". Tokens currently last two weeks and users will need to authenticate with your app once the token expires. Tokens are returned via the hash/fragment of the URL.

Optional parameters:

- `blog` You may pass along a blog parameter (`&blog=`) with the URL or blog ID for a WordPress.com blog or Jetpack site. If you do not pass along a blog, or if the user does not have administrative access to manage the blog you passed along, then the user will be prompted to select the blog they are granting you access to.
- `scope`, if specified, can be set as "global" or "auth". If omitted, the authorization token will only grant you access to a single blog. You may request global scope (`&scope=global`) to access all blogs that the user has access to on WordPress.com, including any Jetpack blogs they have connected to their WordPress.com account. If you are specifying the scope as `global` then you should omit the `blog` parameter. The auth scope (`&scope=auth`) will grant you access to the `/me` endpoints only, and is primarily used for WordPress.com Connect.

Server / Code Authentication

The redirect to your application will include a code which you will need in the next step. If the user has denied access to your app, the redirect will include ?error=access_denied. Once the user has authorized the request, they will be redirected to the redirect_url. The request will look like the following:

```
https://developer.wordpress.com/?code=cw9hk1xG9k
```

Delete a link from current user's profile.

Get current user's connected applications.

Get one of current user's connected applications.

Delete one of current user's connected application access tokens.

Get information about current user's two factor configuration.

Sends a two-step code via SMS to the current user.

Get a list of the current user's likes.

Sites

Get a rendered shortcode for a site. Note: The current user must have publishing access.

Get a list of shortcodes available on a site. Note: The current user must have publishing access.

Get a rendered embed for a site. Note: The current user must have publishing access.

Get a list of embeds available on a site. Note: The current user must have publishing access.

Get information about a site.

Get a list of page templates supported by a site.

Get a list of post types available for a site.

Get number of posts in the post type groups by post status

Retrieve the active and inactive widgets for a site.

Activate a widget on a site.

Get detailed WordAds settings information about a site.

This is a time-limited code that your application can exchange for a full authorization token. To do this you will need to pass the code to the token endpoint by making a **POST** request to `https://public-api.wordpress.com/oauth2/token`.

```
$curl = curl_init( 'https://public-api.wordpress.co
curl_setopt( $curl, CURLOPT_POST, true );
curl_setopt( $curl, CURLOPT_POSTFIELDS, array(
    'client_id' => your_client_id,
    'redirect_uri' => your_redirect_url,
    'client_secret' => your_client_secret_key,
    'code' => $_GET['code'], // The code from the p
    'grant_type' => 'authorization_code'
) );
curl_setopt( $curl, CURLOPT_RETURNTRANSFER, 1);
$auth = curl_exec( $curl );
$secret = json_decode($auth);
$access_key = $secret->access_token;
```

You are required to pass client_id, client_secret, and redirect_uri for web applications. These parameters have to match the details for your application, and the redirect_uri **must** match the redirect_uri used during the Authorize step (above). grant_type has to be set to “authorization_code”. code must match the code you received in the redirect. If everything works correctly and the user grants authorization, you will get back a JSON-encoded string containing the token and some basic information about the blog:

```
{
    "access_token": "YOUR_API_TOKEN",
    "blog_id": "blog ID",
    "blog_url": "blog url",
    "token_type": "bearer"
}
```

You now have an access token which should be stored securely with the blog ID and blog URL. This access token allows your application to act on the behalf of the user on this specific blog. For an alternative example, check out our Node implementation.

Testing an application as the client owner

As the client owner, you can authenticate with the `password` `grant_type`, allowing you to skip the authorization step of authenticating, instead logging in with your WordPress.com username and password. Note that if you are using 2-step

Update WordAds settings for a site.

Get detailed WordAds earnings information about a site.

Get WordAds TOS information about a site.

Update WordAds TOS setting for a site.

Get WordAds stats for a site

Request streamlined approval to join the WordAds program.

Get a list of the current user's sites.

Search within a site using an Elasticsearch Query API.

Retrieve a widget on a site by its ID.

Update a widget on a site by its ID.

Deactivate a widget on a site by its ID. Will delete if already deactivated.

Get the custom header options for a site with a particular theme.

Get the custom header options for a site.

Set the custom header options for a site.

Posts

Get a single post (by ID).

Edit a post.

Get a single post (by slug).

Get a list of matching posts.

Create a post.

Delete a post. Note: If the trash is enabled, this request will send the post to the trash. A second request will permanently delete the post.

Restore a post or page from the trash to its previous status.

authentication (highly recommended), you will need to create an application password to be able to use the `password` `grant_type`.

```
$curl = curl_init( 'https://public-api.wordpress.co
curl_setopt( $curl, CURLOPT_POST, true );
curl_setopt( $curl, CURLOPT_POSTFIELDS, array(
    'client_id' => your_client_id,
    'client_secret' => your_client_secret_key,
    'grant_type' => 'password',
    'username' => your_wpcom_username,
    'password' => your_wpcom_password,
) );
curl_setopt( $curl, CURLOPT_RETURNTRANSFER, 1);
$auth = curl_exec( $curl );
$auth = json_decode($auth);
$access_key = $auth->access_token;
```

As noted above, this is only available to you as the owner of the application, and not to any other user. This is meant for **testing purposes only**.

Client/Implicit Oauth

Once the user authenticates their blog, they will be redirected back to your application. The token and user information will be included in the URL fragment.

https://developer.wordpress.com/#access_token=YOUR_API_TOKEN&expires_in=64800&token_type=bearer&site_id=blog_id

This token will allow you to make authenticated client-side calls using CORS/AJAX requests. The token currently only lasts two weeks. Use the expires_in fragment to detect when you should prompt for a refresh.

Validating Tokens

It can be helpful to be able to validate the authenticity of a token, specifically that it belongs to your application and the user you are authenticating. This is especially necessary when sending a token over the wire (e.g. mobile application sending token as login credentials to an API). To verify a token, use the `/oauth/token-info` endpoint, passing in the `token` and your `client_id`:

```
https://public-api.wordpress.com/oauth2/token-info?
client_id=your_client_id&token=your_token
```

Delete multiple posts. Note: If the trash is enabled, this request will send non-trashed posts to the trash. Trashed posts will be permanently deleted.

Restore multiple posts.

Get a list of posts across all the user's sites.

Get a list of the likes for a post.

Like a post.

Unlike a post.

Get the current user's like status for a post.

Get a list of the specified post's subscribers.

Get subscription status of the specified post for the current user.

Subscribe current user to be notified of the specified post's comments.

Unsubscribe the current user from the specified post.

Reblog a post.

Get reblog status for a post.

Search within a site for related posts.

Comments

Get a single comment.

Edit a comment.

Get a list of recent comments.

Get a list of recent comments on a post.

Create a comment on a post.

Create a comment as a reply to another comment.

Delete a comment.

Get comment counts for each available status

If the token provided was not authorized for your application, the endpoint will return an error. If the token is valid, you will get a JSON-encoded string with the user ID and scope of the token:

```
{
    "client_id": "your client ID",
    "user_id": "user ID",
    "blog_id": "blog ID",
    "scope": "scope of the token"
}
```

Making an API call

Our API is JSON-based. You can view all of the available endpoints at our API documentation. You can also make API calls with our legacy XML-RPC API. In order to make an authenticated call to our APIs, you need to include your access token with the call. OAuth2 uses a BEARER token that is provided in an Authorization header.

```
<!--?php
$access_key = 'YOUR_API_TOKEN';
$curl = curl_init( 'https://public-api.wordpress.com/
curl_setopt( $curl, CURLOPT_HTTPHEADER, array( 'Auth
curl_exec( $curl );
?-->
```

The above example would return information about the authenticated user.

```
jQuery.ajax( {
    url: 'https://public-api.wordpress.com/rest/v1/
    type: 'POST',
    data: { content: 'testing test' },
    beforeSend : function( xhr ) {
        xhr.setRequestHeader( 'Authorization', 'BEA
    },
    success: function( response ) {
        // response
    }
} );
```

The above example would create a new post. You can make similar calls to the other available endpoints.

Get the audit history for given comment

Get the likes for a comment.

Like a comment.

Remove your like from a comment.

Get your like status for a comment.

Kill comment likes

Taxonomy

Get a list of a site's categories.

Get a list of a site's tags.

Get information about a single category.

Edit a category.

Get information about a single tag.

Edit a tag.

Get information about a single term.

Edit a term.

Get a list of taxonomies associated with a post type.

Get a list of a site's terms by taxonomy.

Create a new category.

Create a new tag.

Delete a category.

Delete a tag.

Create a new term.

Delete a term.

Follow

List a site's followers in reverse chronological order.

Follow a blog.

Unfollow a blog.

Get blog following status for the current user.

Sharing

Get a list of a site's sharing buttons.

Edit all sharing buttons for a site.

Get a list of third-party services that WordPress.com or Jetpack sites can integrate with via keyring.

Get information about a single external service that WordPress.com or Jetpack sites can integrate with via keyring.

Get a list of publicize connections that the current user has set up.

Get a single publicize connection that the current user has set up.

Update a single publicize connection belonging to the current user.

Delete the specified publicize connection.

Get a list of all the keyring connections associated with the current user.

Get a single Keyring connection that the current user has setup.

Delete the Keyring connection (and associated token) with the provided ID. Also deletes all associated publicize connections.

Get a list of publicize connections that are associated with the specified site.

Get a single publicize connection that is associated with the specified site.

Update a single publicize connection belonging to the specified site.

Create a new publicize connection that is associated with the specified site.

Delete the specified publicize connection.

Get a list of external services for which sharing buttons are supported.

Freshly Pressed

Get a list of Freshly Pressed posts. (Note: Freshly Pressed has been retired. Please visit https://discover.wordpress.com to get the best content published across our network.)

Notifications

Set the timestamp of the most recently seen notification.

Mark a set of notifications as read.

Insights

Get a list of stats/metrics/insights that the current user has access to.

Get raw data for a particular graph.

Reader

Get default reader menu.

Get details about a feed.

Get a single post (by ID).

Get a list of posts from the blogs a user follows.

Get a list of posts from the blogs a user likes.

Get a list of posts from a tag.

Get a list of tags subscribed to by the user.

Get a list of trending tags.

Get details about a specified tag.

Get the subscribed status of the user to a given tag.

Subscribe to a new tag.

Unsubscribe from a tag.

Get a list of the feeds the user is following.

Follow the specified blog.

Unfollow the specified blog.

Get the ID and subscribe URL of one or more matching feeds by domain or URL.

Get a list of blog recommendations for the current user.

Stats

Get a site's stats

View a site's summarized views, visitors, likes and comments

View a site's top posts and pages by views

View the details of a single video

View a site's referrers

View a site's outbound clicks

View a site's views by tags and categories

View a site's top authors

View a site's top comment authors and most-commented posts

View a site's video plays

View a site's file downloads

View a post's views

View a site's views by country

View a site's followers

View a site's comment followers

Report a referrer as spam

Unreport a referrer as spam

View a site's publicize follower counts

View search terms used to find the site

View the total number of views for each post.

Get stats for Calendar Heatmap. Returns data with each post timestamp.

Media

Delete a piece of media. Note: Media is deleted and not trashed.

Get a single media item (by ID).

Edit basic information about a media item.

Get a list of items in the media library.

Upload a new piece of media.

Edit a media item.

Menus

Create a new navigation menu.

Update a navigation menu.

Get a single navigation menu.

Get a list of all navigation menus.

Delete a navigation menu

Batch

Run several GET endpoints and return them as an array.

Videos

Get the metadata for a specified VideoPress video.

Get the poster for a specified VideoPress video.

Upload and set a poster for a specified VideoPress video.

An Production Usage Guidelines Privacy Policy Terms of Service