

Configuration of the Login by Autho WordPress Plugin

[Autho](#)

By default, new installations of Login by Autho run the Setup Wizard and ask for an app token and attempt to setup all necessary components within your Autho tenant. This includes:

- Creating a new Application using your site name with the correct app type and URLs
- Creating a database Connection for this Application for storing users
- Creating an application grant for the system Autho Management API
- Creating a new user for the WordPress administrator running the wizard

Once this process is complete, your tenant is set up correctly and ready to accept signups and logins.

The Setup Wizard must run to completion for your site to be setup correctly. If the Wizard fails for any reason before the "setup successful" screen, check the plugin error log at **wp-admin > Autho > Error Log** and the steps below to determine the issue.

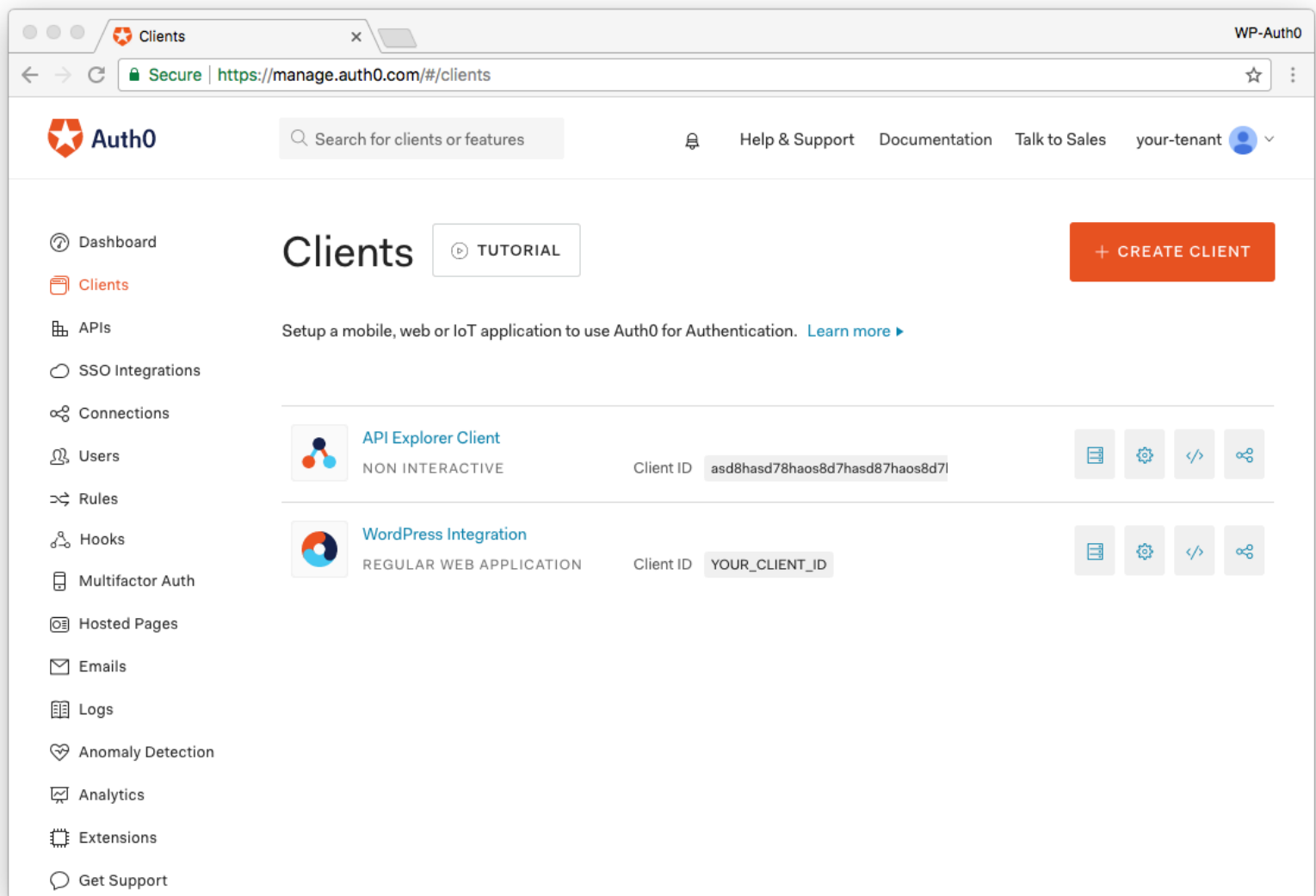
It can be helpful, if you're having any issues with logging in or creating accounts, to walk through the screens for each section below to confirm your setup.

You'll need to be logged into your Autho account before starting the steps below. If you don't have one yet, [create one here](#).

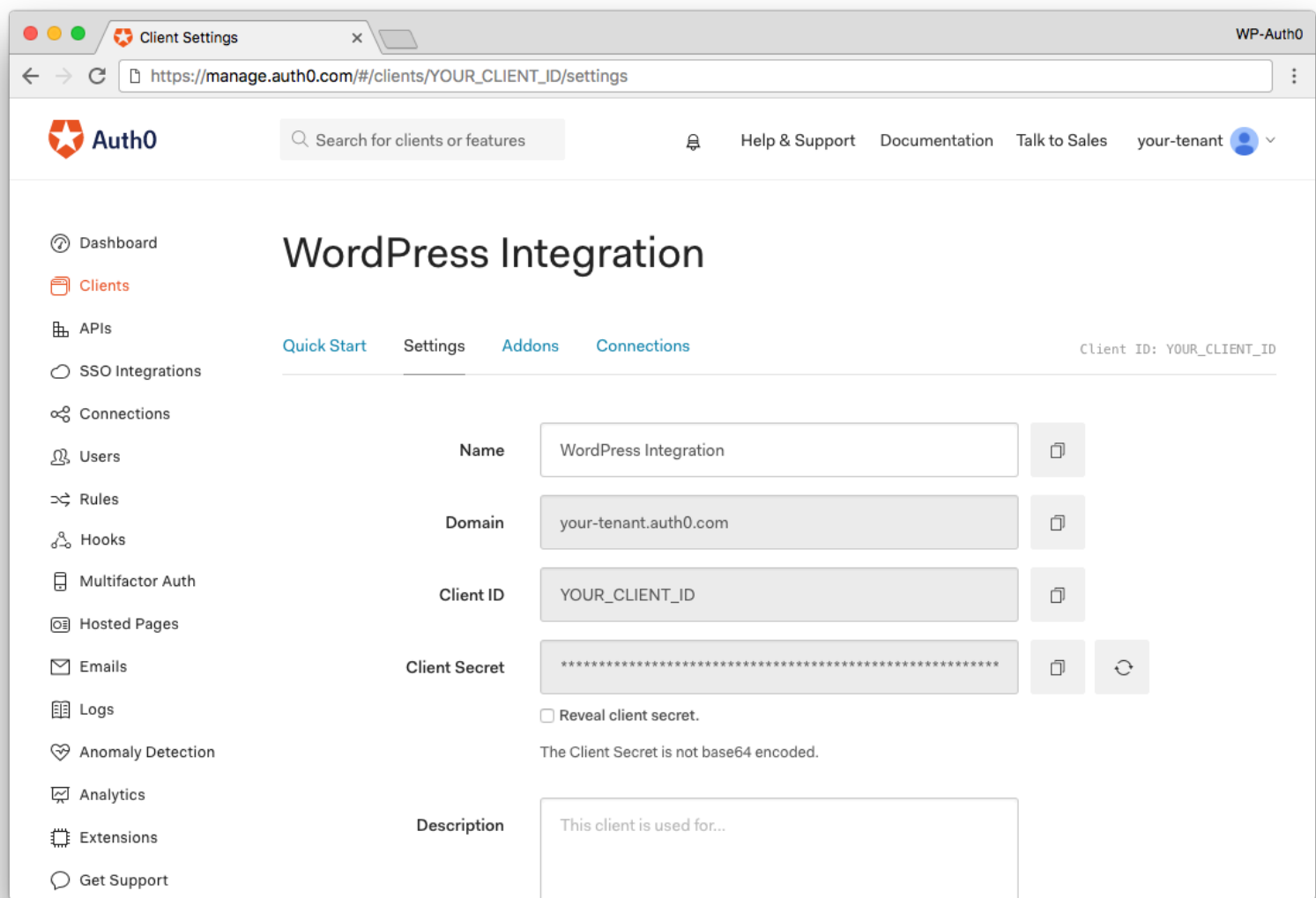
First, we'll check for the Application created for your WordPress site.

1. Navigate to the [Applications](#) page and look for an application that is similar to your site name. If you don't find one, it means that an

Application was not created by the Wizard. Restart the Setup Wizard or create a new Application manually by clicking **Create Application**. Enter a name for the application, select **Regular Web Applications**, and click **Create**.



2. Click on the name to get to the **Settings** tab. You will see your Domain, Client ID, and Client Secret, which are used in **wp-admin > Auth0 > Settings** to make a connection to Auth0.



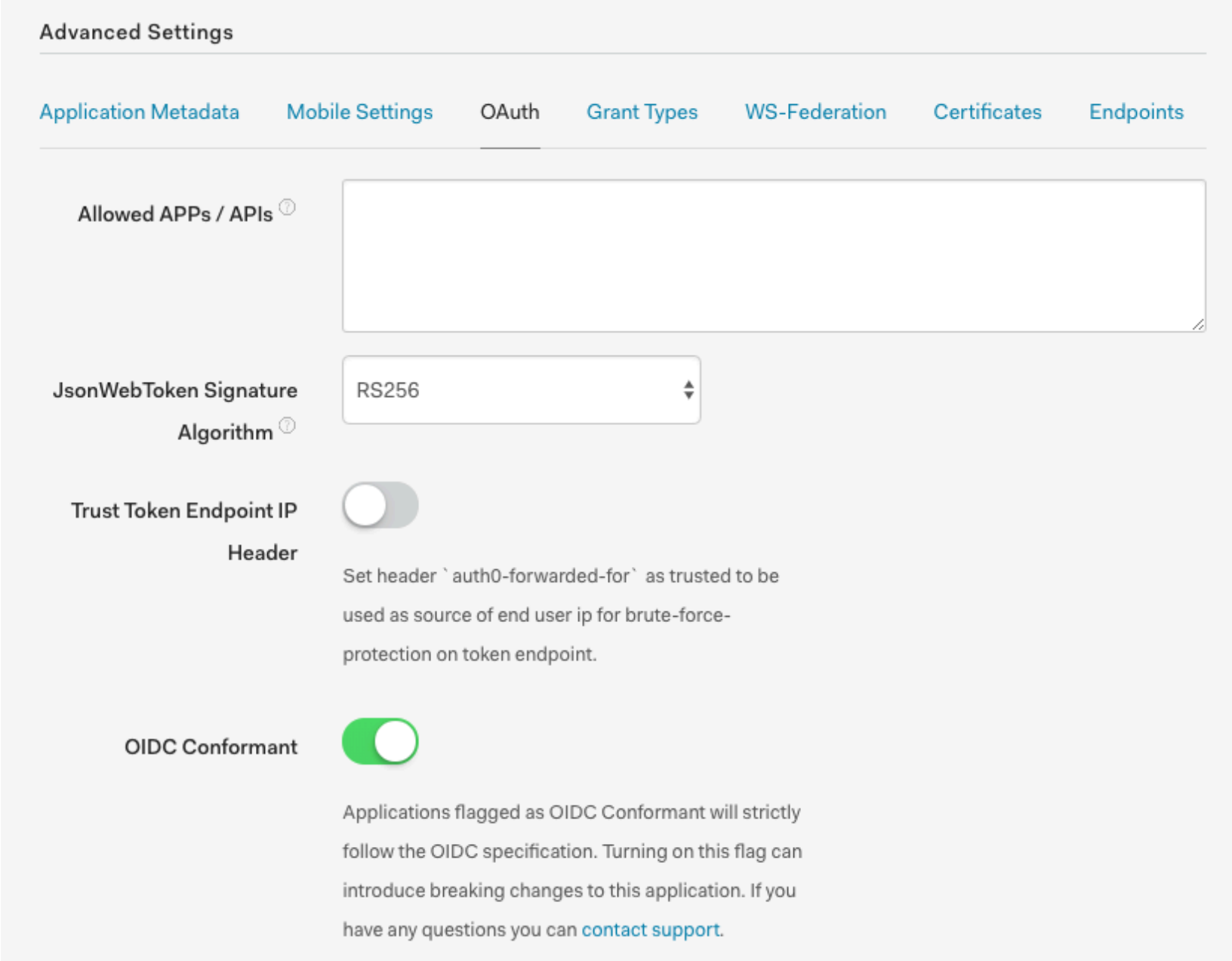
3. **Application Type** must be set to **Regular Web Application** and **Token Endpoint Authentication Method** must be set to **Post**
4. Scroll down to **Allowed Callback URLs** and provide the WordPress site URL with `?auth0=1` appended:

`https://yourdomain.com/index.php?auth0=1`

5. Enter your WordPress site's home domain (where the WordPress site appears) and, if different, site domain (where wp-admin is served from) in the **Allowed Web Origins** field
6. Enter your WordPress site's login URL in the **Allowed Logout URLs** field
7. Leave the **Allowed Origins (CORS)** field blank (it will use the **Allowed Callback URLs** values from above)

Make sure to match your site's protocol (http or https) and use the site URL as a base, found in **wp-admin > Settings > General > WordPress Address (URL)** for all URL fields above.

8. Scroll down and click the **Show Advanced Settings** link, then the **OAuth** tab and make sure **JsonWebToken Signature Algorithm** is set to RS256. If this needs to be changed later, it should be changed here as well as in wp-admin (see Settings > Basic below).
9. Turn on **OIDC Conformant**.



The screenshot shows the 'Advanced Settings' interface for an OAuth2 provider. At the top, there are tabs for 'Application Metadata', 'Mobile Settings', 'OAuth', 'Grant Types', 'WS-Federation', 'Certificates', and 'Endpoints'. The 'OAuth' tab is currently selected. Below the tabs, there are several settings:

- Allowed APPs / APIs**: A large empty text area for listing authorized applications.
- JsonWebToken Signature Algorithm**: A dropdown menu currently set to 'RS256'.
- Trust Token Endpoint IP Header**: A toggle switch that is currently turned off. Below it, a description reads: 'Set header `auth0-forwarded-for` as trusted to be used as source of end user ip for brute-force-protection on token endpoint.'
- OIDC Conformant**: A toggle switch that is currently turned on (green). Below it, a description reads: 'Applications flagged as OIDC Conformant will strictly follow the OIDC specification. Turning on this flag can introduce breaking changes to this application. If you have any questions you can [contact support](#).'

10. Click the **Grant Types** tab and select at least **Implicit**, **Authorization Code**, and **Client Credentials**.

Advanced Settings

Application Metadata Mobile Settings OAuth Grant Types WS-Federation Certificates Endpoints

GRANTS

☒ Implicit

☒ Authorization Code

☒ Refresh Token

☒ Client Credentials

☐ Password

☐ MFA

SAVE CHANGES

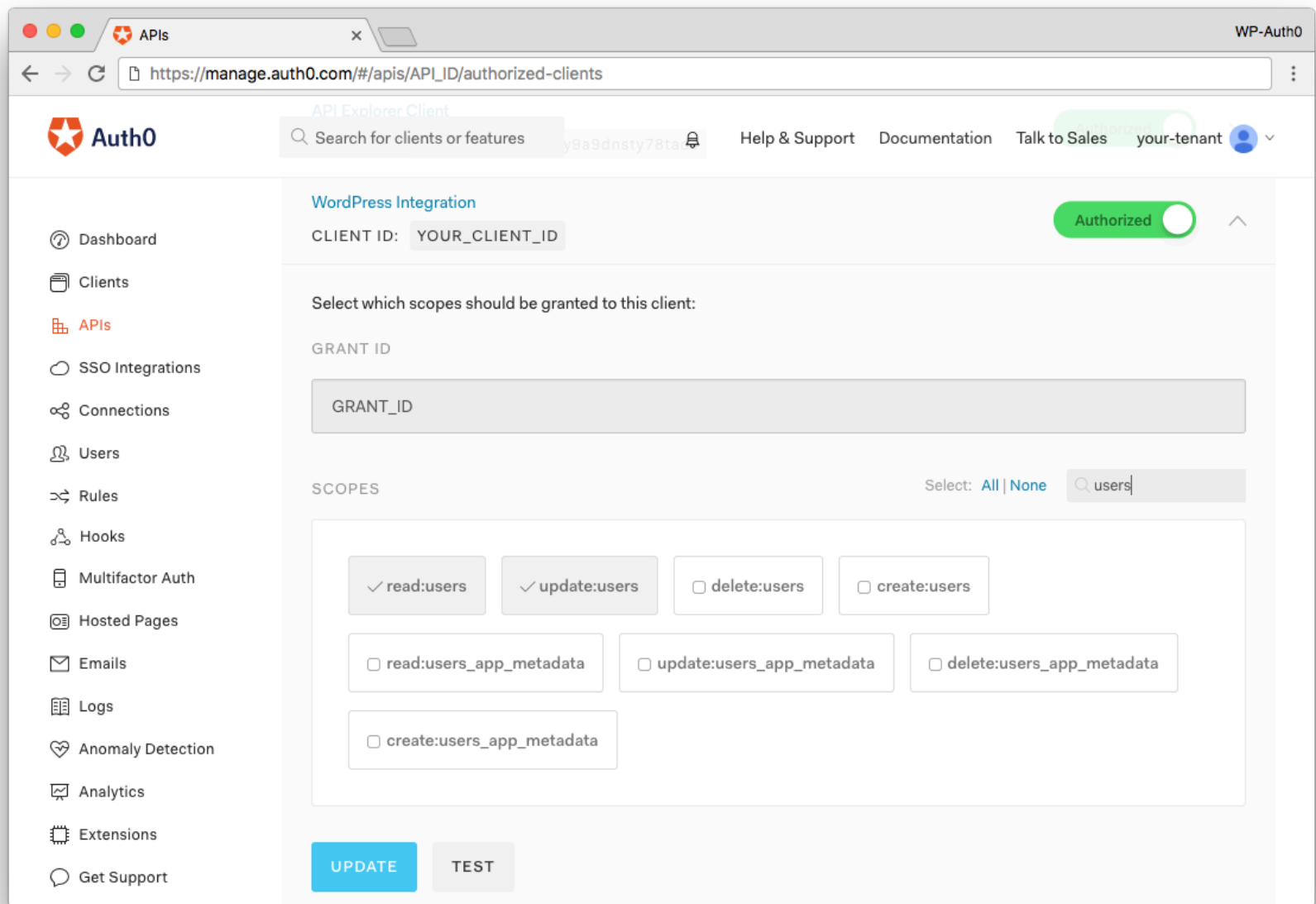
11. Click **Save Changes** if anything was modified.

Authorize the Application for the Management API

In order for your WordPress site to perform certain actions on behalf of your Autho tenant, you'll need to authorize the Application created above to access the Management API. This is not required but will enable retrieving complete user data on login (including `user_metadata` and `app_metadata`), email and password changes for users, and email verification re-sending when verified emails are required.

1. Make sure your Application allows the Client Credentials grant (step 10 in the section above)
2. Navigate to the [APIs](#) page
3. Click on **Autho Management API**, then the **Machine to Machine Applications** tab
4. Look for the WordPress Application and click **Unauthorized** to grant access

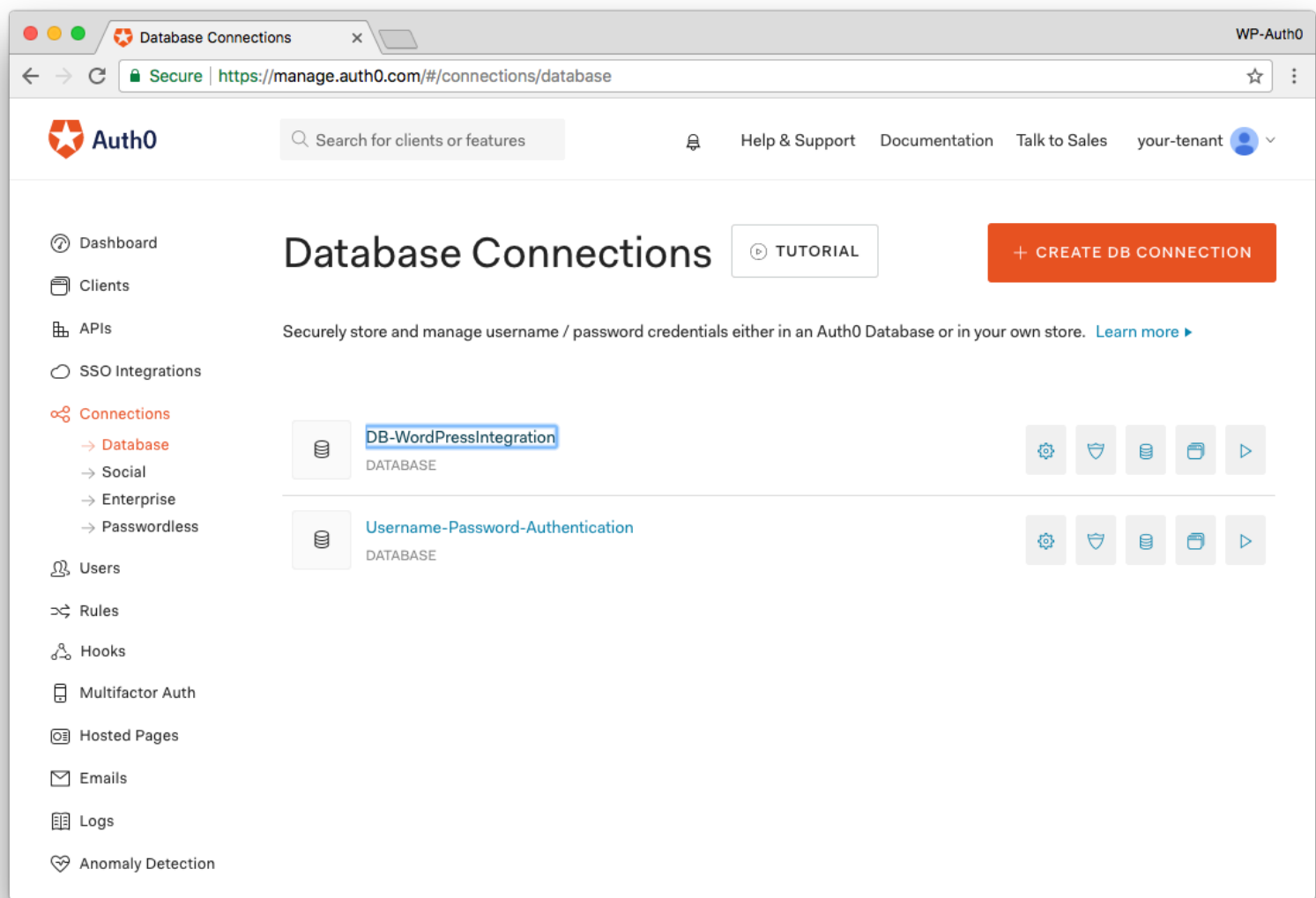
5. In the panel that appears, select only the `read:users` and `update:users` scopes and click **Update** (you can search using the **Filter scopes** field)



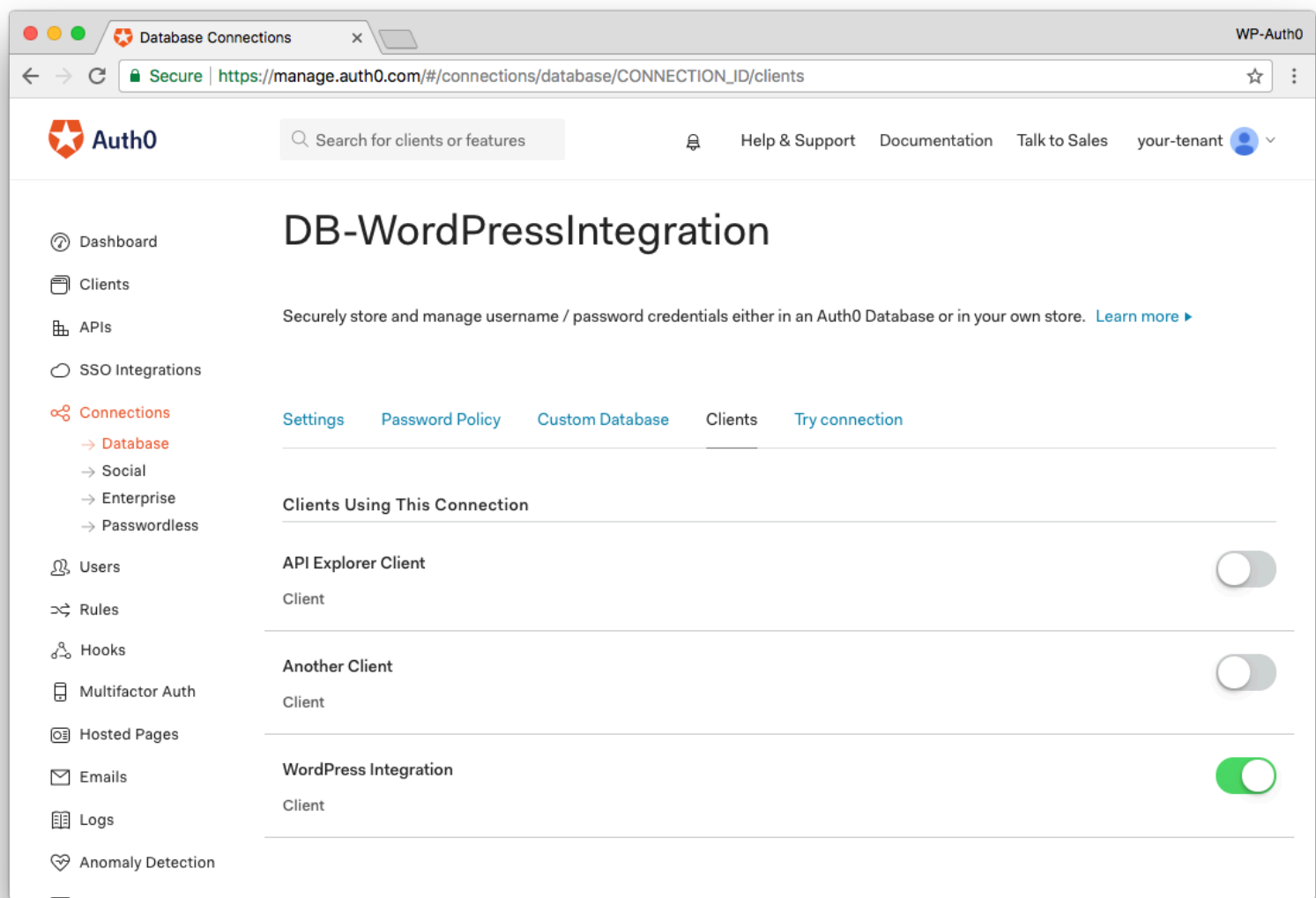
Database Connection setup

Database Connections enable the typical username and password login seen on most sites. This type of Connection is not required and can be skipped if you're using *passwordless* or social logins only.

1. If you used the wizard during setup, navigate to the [Connections > Database](#) page and look for a Connection that has a similar name to the Application setup above. Otherwise, you can create a new Connection, use an existing Connection, or use the default **Username-Password-Authentication**. Click an existing Connection name to view settings or click **Create DB Connection** and follow the steps.



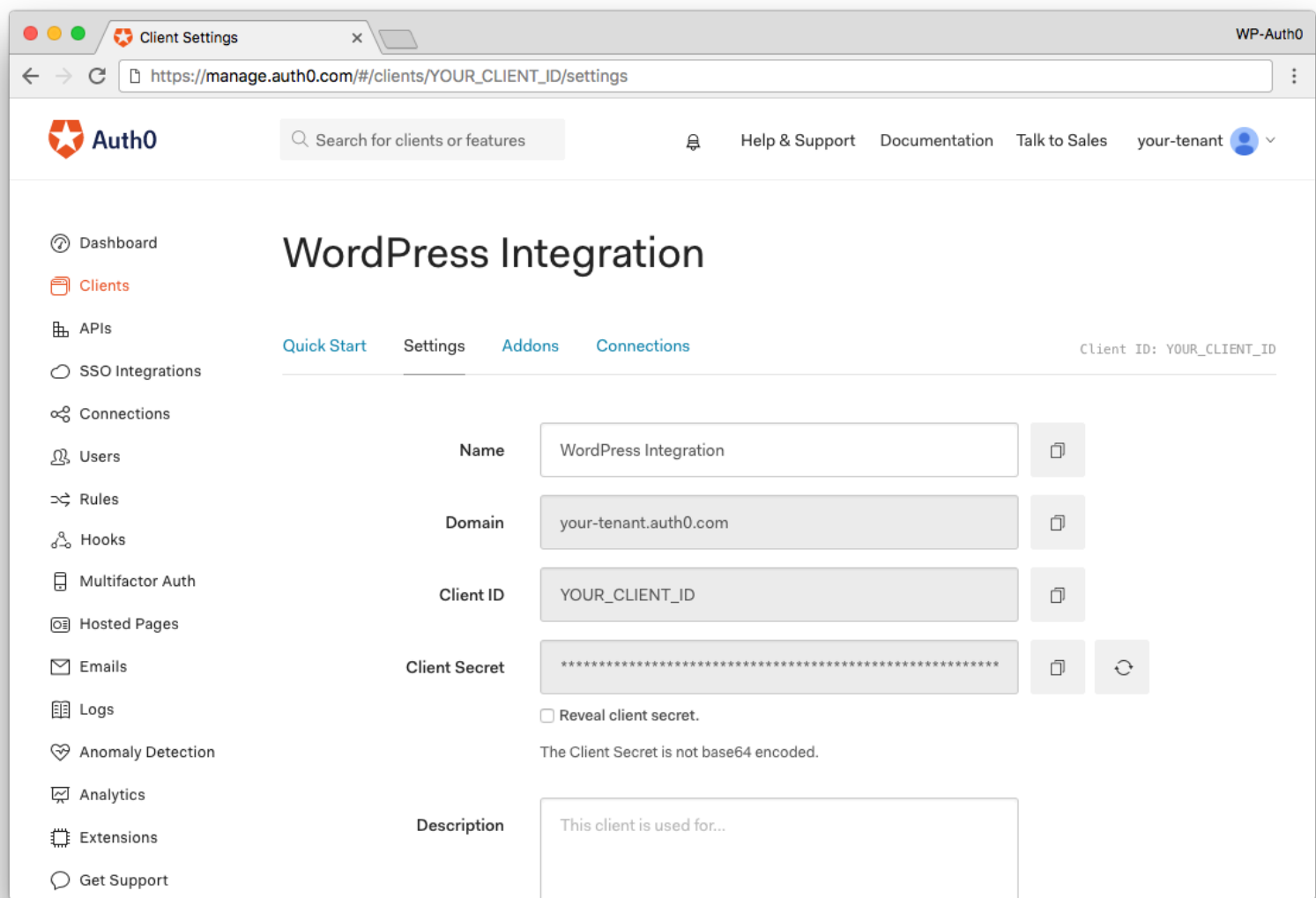
2. Click the **Applications** tab and activate the Application created above.



See our [dedicated page on Social Connections](#) for detailed information on how to activate and configure these login methods.

Update Auth0 settings in WordPress

1. Go to back to the [Applications](#) page and select the Application created above.



2. In a new tab/window, log into wp-admin for your WordPress site and go to **wp-admin > Auth0 > Settings**.
3. Click on the **Basic** tab.
4. Copy **Domain**, **Client ID**, and **Client Secret** from your Auth0 Application page to your WordPress settings using the Copy to Clipboard buttons next to each field.
5. Make sure **Application Signing Algorithm** matches the Application's Advanced > OAuth setting.
6. Scroll down and click **Save Changes**.

PHP Constant Setting Storage

Plugin settings can be saved to the database (default) or they can be set using a specifically named PHP constant. This will allow for sensitive data like the

client secret, API token, and migration token to be stored more securely (assuming that file they are defined in is [stored securely](#)).

The constant **must** be defined before the plugin is loaded or it will not be used. This should happen in your `wp-config.php` file or in a [must-use plugin](#). If the constant is defined in your theme's `functions.php` or in a plugin that loads after Autho, the value will be ignored.

The PHP constants are defined like so:

```
define( 'AUTH0_ENV_CLIENT_SECRET', 'YOUR_CLIENT_SECRET_HERE' );
```

The default constant name should be `AUTH0_ENV_` followed by the option name to override in all caps (the prefix can be modified with the `auth0_settings_constant_prefix` filter [explained here](#)). All plugin options that can be overridden and their keys can be found in the `WP_Auth0_Options::defaults()` [method](#).

Note: The `migration_token` value is generated by the plugin when user migration is turned on. If there is already a value in the admin, make sure to set the constant to the same value. If that value needs to change, it also must be changed in the custom scripts for the database Connection being used in the Autho dashboard.

The settings field will change its display based on this new value and show the constant being used for reference. This value will be used everywhere in the plugin automatically.

Important: Saving the settings page after setting a constant value will validate the constant-set values (but not change them) and delete them from the options array being saved to the database. If you are just testing this

functionality, do not save settings in the WordPress admin page until you're ready to delete that value.

All sites in a WordPress multi-site network will use the same constant value making this an easy way to setup a network using a single Application and database Connection.

- **Domain:** The Domain copied from the Application settings in your dashboard. Option name is `domain`.
- **Custom Domain:** The Custom Domain for your tenant, if one is configured. More information on Custom Domains [here](#). Option name is `custom_domain`.
- **Client ID:** The Client ID copied from the Application settings in your dashboard. Option name is `client_id`.
- **Client Secret:** The Client Secret copied from the Application settings in your dashboard. Option name is `client_secret`.
- **Client Secret Base64 Encoded:** Whether or not the Client Secret is Base64 encoded; it will say below the Client Secret field in your Autho dashboard whether or not this should be turned on. Option name is `client_secret_b64_encoded`.
- **JWT Signature Algorithm** The algorithm used for signing tokens from the Advanced Application Settings, OAuth tab; default is RS256. Option name is `client_signing_algorithm`.
- **JWKS Cache Time (in minutes):** How long the JWKS information should be stored when using the RS256 JWT Signature Algorithm. Option name is `cache_expiration`.
- **Original Login Form on wp-login.php:** Provides ways to access or block the core WordPress login page. Option name is

wordpress_login_enabled. Login page code option name is wle_code.

- **Never** will not allow the core WordPress login form to display.
 - **Via a link under the Autho form** will display a link to the WordPress core login form directly below the Autho embedded one on wp-login.php. The login page can also be accessed directly by adding ?wle to the login URL.
 - **When "wle" query parameter is present** will allow the login page to be accessed directly by adding ?wle to the login URL. This will bypass the Universal Login Page redirect.
 - **When "wle" query parameter contains specific code** will allow the login page to be accessed directly by adding ?wle= plus a code to the login URL. The code is generated automatically and will be shown below the controls for this setting. This will bypass the Universal Login Page redirect.
-
- **Allow Signups:** User signup will be available only if the WordPress *Anyone can register* option is enabled. You can find this setting under **Settings > General > Membership**.
 - **Universal Login Page:** Redirects the wp-login.php page to the Universal Login Page for *Single Sign-on (SSO)* authentication using all active Connections for this Application. Option name is auto_login.
 - **Auto Login Method:** A single, active connection to use for authentication when **Universal Login Page** is turned on. Leave this blank to show all active Connections on the Universal Login Page. Option name is auto_login_method.
 - **Single Logout:** Enable this option to log out of Autho when logging out of WordPress. Option name is singlelogout.
 - **Single Sign-On (SSO):** *This option is deprecated and will be removed in the next major. Please use the Universal Login Page option to enable*

SSO Enable this option to attempt SSO on the `wp-login.php` page.

Option name is `sso`.

- **Override WordPress Avatars:** Forces WordPress to use Autho avatars. Option name is `override_wp_avatars`.

This section was changed from "Appearance" to "Embedded" to reflect the fact that these settings only affect Autho login forms embedded on the WordPress site. Options here do not affect the Universal Login Page (see [this page](#) for customization options).

- **Passwordless Login:** Enable this option to turn on Passwordless login on all embedded Autho login forms. Passwordless connections are managed in the Autho dashboard and at least one must be active and enabled on this Application for this to work. Option name is `passwordless_enabled`.
- **Icon URL:** Sets the icon above the embedded Autho login form. Option name is `icon_url`.
- **Form Title:** Sets the title of the embedded Autho login form. Option name is `form_title`.
- **Enable Gravatar Integration:** When user enters their email, their associated Gravatar picture is displayed in the embedded Autho login form. Option name is `gravatar`.
- **Login Name Style:** Selecting **Email** will require users to enter their email address to login. Set this to **Username** if you do not want to force a username to be a valid email address. Option name is `username_style`. Option name is `client_secret_b64_encoded`.
- **Primary Color:** Information on this setting is [here](#). Option name is `primary_color`.

- **Language:** Information on this setting is [here](#). Option name is `language`.
- **Custom Signup Fields:** This field is the JSON that describes the custom signup fields for Lock. The should be a in the form of JSON and allows the use of functions for validation. [More info on custom signup fields here](#). Option name is `custom_signup_fields`.
- **Extra Settings:** A valid JSON object that includes options to call Lock with. This overrides all other options set above. For a list of available options, see [Lock: User configurable options](#) (e.g.: `{"disableResetAction": true }`). Option name is `extra_conf`.
- **Use Custom Lock JS URL:** When turned off, WordPress will use the latest tested version of Lock (Auth0 embedded login form) automatically. When turned on, administrators can provide a custom Lock URL to use. Option name is `custom_cdn_url`.
- **Custom Lock JS URL:** A valid URL pointing to a version of Lock. This field will be automatically hidden when **Use Custom Lock JS URL** is turned off. Option name is `cdn_url`.
- **Connections to Show:** List here each of the identity providers you want to allow users to login with. If left blank, all enabled providers will be allowed. (See [connections {Array}](#) for more information.) Option name is `lock_connections`.

If you have enabled Passwordless login, you must list here all allowed social identity providers. (See [.social\(options, callback\)](#) for more information.)

- **Require Verified Email:** If set, requires the user to have a verified email to log in. This can prevent some Connections from working properly if they do not provide an email address or an `email_verified`

flag in the user profile data. Option name is `requires_verified_email`.

- **Skip Strategies:** If Require Verified Email is turned on, this setting will display. This field accepts strategy names to skip the verified email requirement on login and account association. This should only be used for strategies that do not provide an `email_verified` flag.
- **Remember User Session:** By default, user sessions live for two days. Enable this setting to keep user sessions live for 14 days. Option name is `remember_users_session`.
- **Login Redirection URL:** If set, redirects users to the specified URL after login. This does not affect logging in via the `[auth0]` shortcode. Option name is `default_login_redirection`. To change the redirect for the shortcode, add a `redirect_to` attribute, like so:

```
[auth0 redirect_to="http://yourdomain.com/redirect-here"]
```

- **Force HTTPS Callback:** Enable this option if your site allows HTTPS but does enforce it. This will force Auth0 callbacks to HTTPS in the case where your home URL is not set to HTTPS. Option name is `force_https_callback`.
- **Auto Provisioning:** Should new users from Auth0 be stored in the WordPress database if new registrations are not allowed? This will create WordPress users that do not exist when they log in via Auth0 (for example, if a user is created in the Auth0 dashboard). Option name is `auto_provisioning`.

If registrations are allowed in WordPress, new users will be created regardless of this setting.

- **User Migration:** Enabling this option will expose the Auth0 migration web services. However, the Connection will need to be manually

configured in the [Autho dashboard](#). For more information on the migration process, see our [documentation page on user migrations](#). The **Generate New Migration Token** button can be used to replace the saved token with a new one. Make sure to have your database Connection configuration page open to the **Custom Database** tab so you can replace the existing token with the new one in both scripts. Option name is `migration_ws`. Migration token option name is `migration_token`.

- **Migration IPs Whitelist:** Only requests from listed IPs will be allowed access to the migration webservice. Option name is `migration_ips_filter`.
- **Implicit Login Flow:** If enabled, uses the [Implicit Flow with form_post](#) protocol for authorization in cases where the server is without internet access or behind a firewall. Option name is `auth0_implicit_workflow`.
- **Valid Proxy IP:** List the IP address of your proxy or load balancer to enable IP checks for logins and migration web services. Option name is `valid_proxy_ip`.
- **Autho Server Domain:** The Autho domain, it is used by the setup wizard to fetch your account information. Option name is `auth0_server_domain`.

More information on the Login by Autho WordPress plugin:

[How does it work?](#)

[Install the plugin](#)

[JWT API authentication](#)

[Troubleshooting](#)

[Extend the plugin](#)