Configure EZFacility for API Access

EZFacility (EZF) is our CRM solution. When an RFID card is tapped on a Station, the Station reads from the CRM service to verify the credentials and access rights associated with the clientID that is stored on the RFID card.

The RFID Station communicates with internet resources via the Particle.io cloud service called webhooks.

This short document explains how to configure the various systems to allow a Particle.io webhook to access EZF.

API Client vs EZF User

The EZF API uses OAuth to handle authentication. An external application presents credentials and obtains an OAuth token that is good for some period of time (currently one hour). This application passes this token to EZF with every API call.

To obtain an OAuth token you need four pieces of information.

- EZFacility API Client and Client Secret
- EZFacility instance User and Password with the appropriate permissions

It is easy to confuse these two credentials, and that is the source of a lot of frustration. As you read the rest of this document please keep these two sets of credentials in mind.

Sandbox or Production?

An important concept is that EZF provides you with both your Production CRM instance (of course) and a “sandbox” instance that contains some data, but is completely disconnected from your production instance. Each instance has its own credentials. Any work you do using the sandbox credentials are not charged to your account and can not affect your production instance.

EZF also has a very nice web page that lets you test the API calls to get a feeling for how they work. Unfortunately the web page for your Production instance looks identical to the web page for your Sandbox instance; the only difference is in the URL. BE VERY CAREFUL.

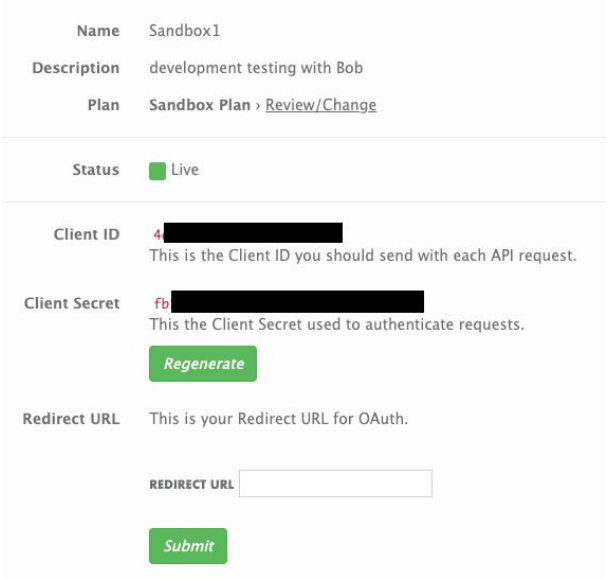
In addition, it seems that while an API Client can be designated for Sandbox, if paired with the correct URL and EZFacility User, it can modify your Production instance. BE VERY, VERY CAREFUL.

On top of all this, the API Client information is encoded in a way that is impossible for a human to verify on sight. Did we already mention: BE CAREFUL?

For this reason we strongly suggest you have different EZFacility Users in your Production and Sandbox account, and that you name them in a way that indicates Production vs Sandbox. For example, “SandboxUser1” and “ProductionUser1”. The EZF User credentials are easy for you to verify on visual inspection and are specific to the instance that hosts them.

Step by Step

1. Establish an EZFacility API account.
    a. At https://developers.ezfacility.com/
2. Create an API Client
    a. Log on to https://developers.ezfacility.com/
    b. Create an Application
        i. Sandbox or Live
        ii. Wait about 24 hours for an activation confirmation from EZF
    c. Once activated you can return to the developers site, click on the Application name and obtain two important pieces of information: Client ID and Client Secret.

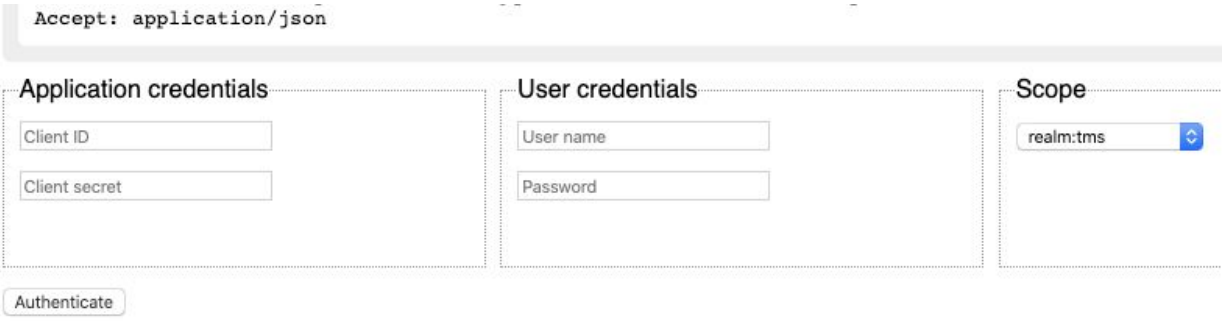


3. Create an EZFacility User in the correct system
    a. Choose Production or Sandbox and log in
        i. Production: https://tms.ezfacility.com
        ii. Sandbox: https://tms-ua.ord.ezfacility.com/
    b. Go to Administration / User Administration / Role Administration

i. Create a role called API Checkin
ii. Remove all permissions from this role except:
    1. Exempt From Login IP Address Restriction
    2. View Client List
    3. View Clients

c. Go to Administration / User Administration / User Administration
i. Add a user with the name API Checkin.
    1. Generate a password.
    2. Assign this user the role API Checkin
ii. Add a user with the name API ReadOnly
    1. Generate a password
    2. Assign this user the role Read Only

4. Test in the API sandbox page.
    a. We will assume you are testing sandbox credentials.
    b. Go to https://api.sandbox.ezfacility.com/swagger/ui/index#/
    c. Enter the four pieces of credential info in the boxes shown



    d. Click Authenticate and wait for a pop up that says “ok”
    e. If you don’t get the OK pop up, then something is not right.

5. Authentication String
    a. Now you will build the string necessary for web access. You will need the API Client ID and Client Secret.
    b. Use a text editor (not a WYSIWYG editor) like Notepad to create the following string: Client ID:Client Secret
    c. B64 encode this string. (You can use a web based tool: https://www.base64encode.org/)
    d. Record this string and use it with the instructions on configuring your Particle.io webhooks.
    e. Example:
        i. Client ID: `4538976`
        ii. Client Secret: `874f9a76cd0325efb4a3ab98c`
        iii. Create: `4538976:874f9a76cd0325efb4a3ab98c`
        iv. B64 encoded:
        `NDUzODk3Njo4NzRmOWE3NmNkMDMyNWVmYjRhM2FiOThj`