## **Introduction**

A common service for catching webhooks and performing a wide variety of automated workflows is Zapier. Zapier is extremely user-friendly and the perfect place to start truly understanding the power of Webhooks. Here's a quick tutorial how to set up an OrderCloud.io webhook that gets caught by a "Zap":

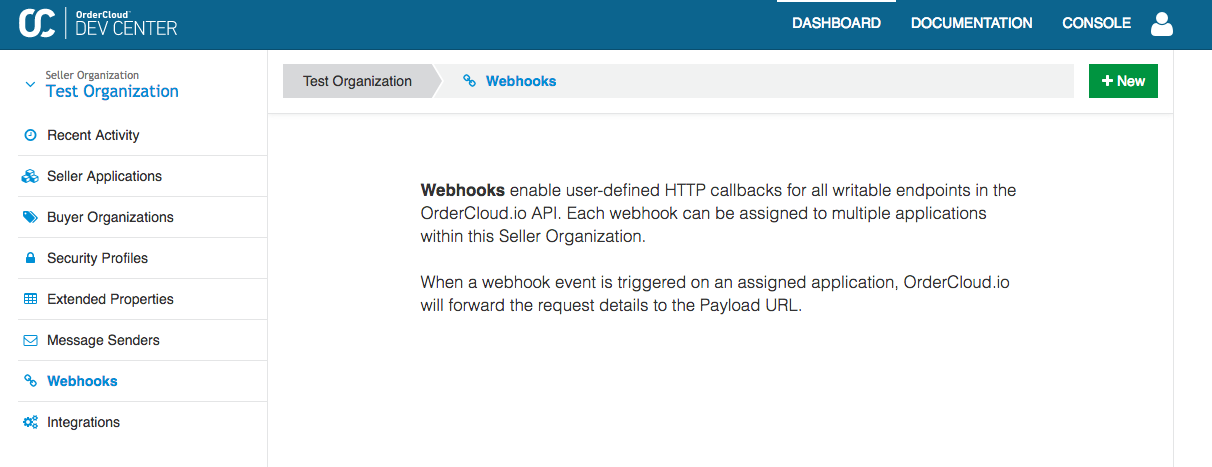
## **1. Create the Hook URL**

First of all, if you don't have one already - go ahead and [register for a Zapier account](https://zapier.com/sign-up/). It's free and the trial lasts 14 days - plenty of time to explore and play! Zapier is self-described as "... your Swiss Army knife for the web. It's one of those tools you can constantly come back to for automating mindless tasks.” With that in mind, let's [make a simple Zap](https://zapier.com/app/editor) that will catch our OrderCloud.io webhook.

Once logged in to your Zapier account, click the "Make a Zap" button at the top of the page. Next, search for "Webhooks for Zapier," and click the "Catch Hook" radio button. Click continue, and continue again (we won't pick off a child key for this example). This will produce a URL that we will register in the OrderCloud.io dashboard. Copy this URL for the next step.

## **2. Register Your Hook with OrderCloud**

Back in the OrderCloud.io dashboard, jump to the "Webhooks" page and click the green "New" button (you will need the Admin Webhooks permission).



### **Create New Webhook**

**Name**, **description** and **Secret** can be whatever you like. Use the recently copied Zap URL as the **Payload URL**. When "Append Route" is checked, the triggered route will be appended to your base URL. Turning on “Inherit Http Method” will use the HTTP method of the route that triggered the Webhook event.

**Trigger Events** represent the API calls that will trigger this webhook. For this example, we'll be using the [CreateNewUser](https://devcenter.ordercloud.io/docs/api#Users_Create) endpoint. This means that each time a user is created, this webhook will forward the user information to our Zap. The endpoints are categorized by API resource: first select **Users** and then **Create New User** followed by pressing "Add".

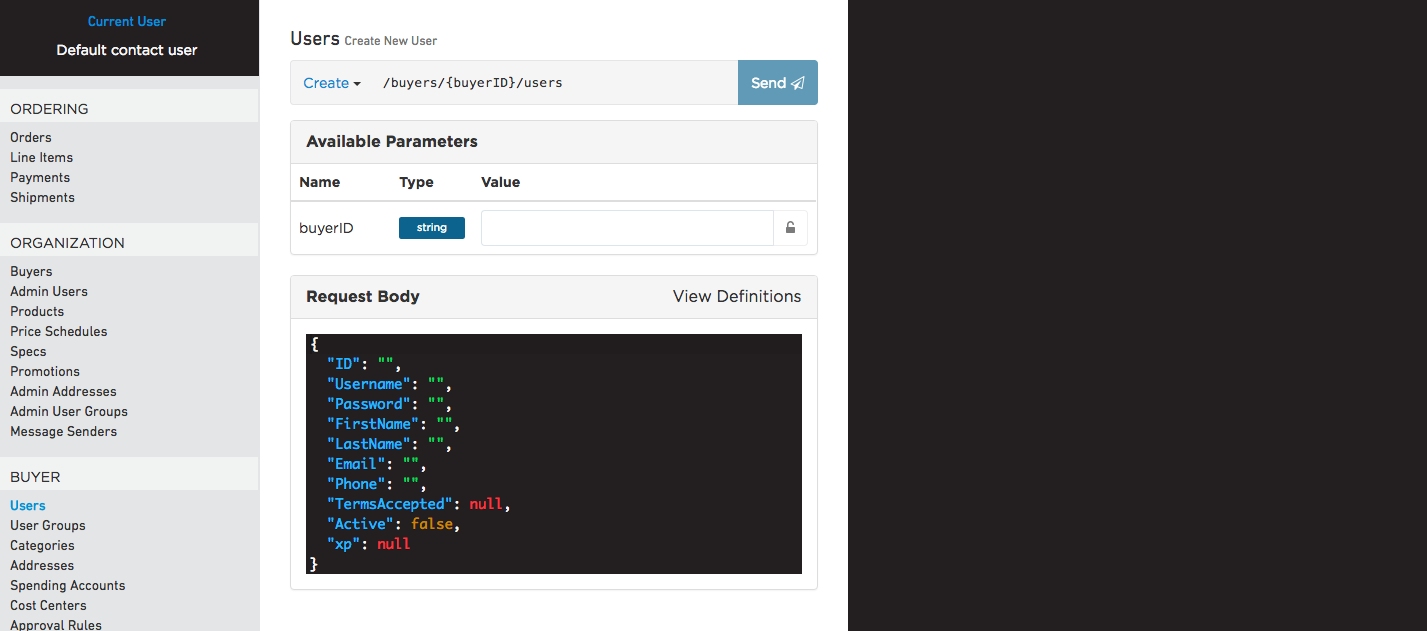
**Elevated roles** are a used for passing along an access\_token with higher permissions than the client that triggered the webhook has. For instance, if the initiating user only had the UserAdmin role but our Zap needed to list Categories and perform some other action, we would add CategoryReader to the elevated claims. For this example we will choose **FullAccess** here and save the webhook.

### **Assign Webhook**

Before we can test our webhook, we first need to tell OrderCloud which application should watch for our webhook event. Choose an application (most likely a Seller application, any app will do as long as you continue to use it for the duration of this guide) and open the "Webhooks" tab, click "Add", enable the webhook we just created, and confirm the change.

## **3. Test Your Webhook**

The Zapier dashboard will prompt us to test our webhook by calling the URL that was generated in **Step 1**. We will do one better and test the OrderCloud.io webhook system at the same time. Return to Zapier and click *"okay, I did this"* to start the test. Once the loading indicator pops up, open the [API Console](http://localhost:3000/docs/guides/getting-started/dashboard-and-api-console#ApiConsole) with the application that has your webhook and [Create a New User](https://devcenter.ordercloud.io/docs/api#Users_Create). (remember this is the route that we configured Zapier to listen to in Step 1.)



If everything has been configured correctly, we should be able to create a new User and see that triggered in Zapier. Shortly after you create a user on OrderCloud, the Zapier dashboard should notify you that your test was successful. Zapier will provide some guidance if the Test fails, you may have to reattempt creating a new user in the API Console if this happens.

If all goes well we will have caught the Create New User event from our OrderCloud.io application!

## **4. Perform an OrderCloud.io Action from Zapier**

Now we are able to set up an "Action" step. To keep it simple, let's use our Zap to update the newly created user with a different last name. Once again we will choose "Webhooks by Zapier". Because this is an Action step, we are presented with different options. Click "show more options", select Custom Request, and hit continue.

Here's where it get's interesting. Zapier will use your last successful test data to help you model your action template. Set the HTTP method to PATCH. The **URL** will use data from the original request (**Step 1: Catch Hook**) for route parameters. Start by typing **"https://api.ordercloud.io/v1/buyers/"** - now, click the icon to the right of the input and you should be able to find an option for **Route Params Buyer ID**. Continue your URL with **"/users/"** - now find **Request Body ID** in the dropdown on the right. This is the ID of the user that was created on OrderCloud.io during our test in Step 2.

Hopefully you're beginning to see just how useful Zapier can be when working with OrderCloud webhooks. Our **URL** is now set up to PATCH the user created in **Step 1: Catch Hook**. But what will we PATCH it with? Skip down to the **Data** text box in the custom request form. This is where we can enter the request body for our action. Write in a simple request body:

{  
 "LastName": "ZippityDoDa"  
}

One last piece of configuration. We need to provide Zapier an access\_token it can use to make the request. Under **Headers** enter in a key of **Authorization** and a value of **"Bearer "** (don't forget the space). Similar to how we set up our **URL**, click the dropdown on the right of the value field and find **User Token**. This will pass through the generated token which has our Elevated Claims (FullAccess) from earlier. Let's also add a **Content-Type** header of **application/json**.

Finally, hit "Continue" and verify the request looks like this:

Method  
PATCH  
  
URL  
https://api.ordercloud.io/v1/buyers/SAZBUYER/users/test  
  
Data  
{  
 "LastName":"ZippityDoDa"  
}  
  
Unflatten  
yes  
  
Headers  
application/json  
Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...

## **5. Test the Action Step**

Finalize your Zap in the Zapier interface (one final test for the Action step), turn it on, and try creating another user in the [API Console](http://localhost:3000/docs/guides/getting-started/dashboard-and-api-console#ApiConsole). Shortly after, GET the user you just created. If everything worked correctly, their last name should now read **"ZippityDoDa"**.

**Congratulations!** You've successfully created a Zap-based OrderCloud.io webhook that can alter data in the origin application.