

## PayPal API for osConcert. February 2022

Setup via admin is as before – if you are going to use the pseudo-cron order cancelling then you should set the 'awaiting' status to anything other than the initial one

Suggested statuses:

<b>Set Successful Order Status</b> Set the status of successful orders made with this payment module to this value. Delivered = download tickets.	Delivered ▼
<b>PayPal Set Order Status</b> Set the status of pending orders made with this payment module to this value - <b>do not use</b> Delivered!	Preparing [PayPal] ▼
<b>PayPal Set Awaiting Order Status</b> Set the status of orders with incomplete payment - <b>do not use</b> Delivered!	Awaiting [PayPal] ▼
<b>Set Failed/Error Order Status</b> Set the status of failed/error orders made with this payment module to this value. Suggested value is PayPal Error/Cancelled - <b>do not use</b> Delivered!	PayPal Error/Cancelled ▼

Operation is as follows:

### 1. Customer checks out of store

Order is created and saved as 'preparing'. Customer's browser gets sent to PayPal.

### 2. Customer completes payment at PayPal

Initially the payment is authorised within the PayPal system. This is notified to the store by either the customer returning to the store or via the Webhook.

Both of these will set the status to 'awaiting'.

### 3. Payment Capture

In order to release tickets, the payment must be captured. We now send a secure request to PayPal to confirm the data received and secure the capture. We do not rely on unsolicited incoming data for security reasons – we will always check.

Both customer return and webhook will attempt capture & email independently.

Once payment is captured the order status moves to 'delivered' and the order email & tickets are sent.

### 4. Emails

The order invoice and pdf tickets (if setup) are only sent upon a full capture – in other circumstances customers should refer to their online account pages. This circumvents the problem where customers in receipt of an email (minus tickets) when their payment is not confirmed believe that payment has been completed when it is still pending.

## 5. Cron

If the pseudo-cron cancellation system is going to be used with this module then it is strongly recommended that the 'awaiting' status is setup independently of the 'pending' status. That way authorised but not captured payments will be moved to 'awaiting' and the order is not then cancelled by the cron.

## 6. Timescales & Pending Orders

The live and test systems on PayPal are not uniformly aligned therefore it is difficult to suggest timescales – however if a payment is still 'pending' after 60 minutes then a cancellation may be in order. Webhook calls are coming though in testing almost immediately for authorisation but taking up to 30 minutes for capture.

No webhook calls have been seen for customer's manually cancelling at PayPal. It is conceivable that an order can be abandoned at any point whilst the customer is 'away' from the store and those orders will remain at 'pending' and should either be manually verified/cancelled or cron'd.

## 7. Webhooks

The main purpose of the webhook is to 'listen' for delayed payment confirmations and then capture the amount, update the order status to 'delivered'. It will only perform this function for notifications from PayPal that are marked PAYMENT.SALE.COMPLETED.

A secondary function is in receipt of PAYMENTS.PAYMENT.CREATED (a customer has authorised payment). Here it will move the order status to 'awaiting' and note the order.

Any other messages from PayPal will simply be noted in the order comments.

## Webhooks – setup at PayPal

Click on the name of your App (here "q"). Scroll down until you see the area below and click "Add Webhook"

**LIVE WEBHOOKS**

Configure webhooks to notify your app when certain events occur. To configure a webhook, define your webhook listener URL and a list of events for which to listen. You can configure up to ten webhooks. Each webhook can subscribe to either specific events or all events. To learn more about webhooks, see [webhooks notifications](#).

Webhook	Webhook ID	Events tracked
This app has no webhooks.		

[Add Webhook](#)

Fill in the Live Webhook URL as [https://...../paypal\\_api\\_webhook.php](https://...../paypal_api_webhook.php), select "all events" and save.

### LIVE WEBHOOKS

**⚠ Important:** Make sure that you entered the correct URL to receive webhooks when events occur in your account. You must keep the URL up-to-date and are responsible for the safety and security of the data that you transmit through webhooks.

Webhook URL

Event types

All events [?](#)