

Overview: Webhooks



Last Modified on 03/27/2024 3:53 pm EDT

What Are Webhooks?

Webhook is a way for a user of a system to tap into the workflow and listen for specific events. When an event of a subscribed webhook occurs, the system will send your assigned listener (the "hook") the event information. To receive this notification, you have to let the system know where your listener lives by adding a publicly accessible callback url upon first subscribing.

For example, I want to listen for when the Updox system receives a new fax and create a service to do so, listening at a particular endpoint. I then subscribe to the `inbound_fax` webhook, providing the url of my listener's location. Now, whenever a fax is sent to Updox, I receive a JSON message containing the information I need to go get further information about the fax.

Managing and Using Webhooks

How to Subscribe

To get a list of available active webhook events that can be subscribed to, make a POST request to <https://updoxqa.com/api/io/WebHookEventList> using your application credentials and note the `id`s of the applicable events.

Sample Request

```
1{
2  "auth": {
3    "applicationId": "<applicationId>",
4    "applicationPassword": "<applicationPassword>"
5  }
6}
```

Sample Response

```
1{
2  "successful": true,
3  "responseMessage": "OK",
4  "responseCode": 2000,
5  "events": [
6    {
7      "id": "inbound_direct",
8      "description": "Inbound Direct Message"
9    },
10   {
11     "id": "inbound_fax",
12     "description": "Inbound Fax"
13   },
14   {
15     "id": "inbound_mdn",
16     "description": "Inbound MDN"
17   },
18   {
19     "id": "outbound_fax",
20     "description": "Outbound Fax"
21   }
22 ]
23}
```

Once you know the webhook events you want to subscribe to, you will call a POST request to

<https://updoxqa.com/api/io/WebHookSave>.

There are two levels of webhook subscription: account-level and vendor-level. If you want the event type to fire on a single account in your organization for a specific callback url, you must specify the **accountId** in the **auth** block of the request. Otherwise, if you want the callback url to receive all webhooks of that event type for your organization (i.e. tied to your vendor id), then you will leave the **accountId** empty or **NULL**.

Sample Request	Sample Response
<pre>1{ 2 "auth": { 3 "applicationId": "<applicationId>", 4 "applicationPassword": "<applicationPassword>", 5 "accountId": "", 6 "userId": "" 7 }, 8 "eventId": "inbound_fax", 9 "url": "https://updoxqa.com/wh/callback/inbound_fax", 10 "active": "true" 11}</pre>	<pre>1{ 2 "successful": true, 3 "responseMessage": "OK", 4 "responseCode": 2000, 5 "webhook": { 6 "eventId": "inbound_fax", 7 "url": "http://updoxqa.com/wh/callback/inbound_fax", 8 "active": true, 9 "added_date": "2016-02-05T21:10:20Z", 10 "modified_date": "2016-03-24T18:14:48Z" 11 } 12}</pre>

And that's it! Callbacks to your URL will now be triggered on the appropriate events and will match the payloads specified above.

How to Update a URL

At some point, you may need to change which URL webhook events are delivered to. To update the URL, make a POST request to <https://updoxqa.com/api/io/WebHookSave>. This request will be the same as the original request, but with the new URL.

After the change, all new events will be delivered to the new URL. It's important to note that any previously queued events will still go to the old URL, no matter where they are in the retry process. If you need existing events to be re-queued with a new URL, please [contact Updox Partner Support](#) after updating to the new URL.

How to Handle Failed Webhooks

If your server is having issues receiving Webhooks, there will be a need to requeue the Webhooks when the issue is resolved.

Webhook Status Workflow

The initial HTTP POST call to your specified callback url is made at the time of the event.

- A successful call is any call that returns any 2xx status.
- A call will be aborted for any 4xx call, meaning we will not try to call this again.
- A call will be retried for any 5xx status. The retry is delayed on a Fibonacci scale (1, 1, 2, 3, 5, 8, 13, 21, 34, and 55) in minutes. It will retry every 55 minutes for another 15 times until it is aborted.

Webhook Retry

To assist with this process we provide the following API calls:

<https://updoxqa.com/api/io/WebHookFailuresGet> - This API will retrieve from our logs the webhook IDs that have a not NULL, non 2xx http status received from your server.

```
1{
2  "auth": {
3    "applicationId": "<applicationId>", 4  "applicationPassword": "<applicationPassword>",
5  "accountId": "",
6  "userId": ""
7  },
8  "eventId": "inbound_fax",
9  "url": "https://updoxqa.com/wh/callback/inbound_fax",
10 "active": "true"
11}
```

To retry these IDs:

<https://updoxqa.com/api/io/WebHookRetry> or **<https://updoxqa.com/api/io/WebHookVendorRetry>** can be used to retrigger the webhook after the issue has been resolved.
