REST API Webhooks
(Pabbly Connect Manual)
# Table of Contents

Description ........................................................................................................................................... 3
Events Available with LiveWebinar REST Webhooks ........................................................................ 4
HTTP Methods Available in LiveWebinar REST Webhooks ............................................................... 5
Adding Webhooks with Information About New Registrations/Leads .............................................. 6
Adding Webhooks Containing Information About New Recordings .................................................. 12
Adding Other Types of Webhooks ...................................................................................................... 17
The LiveWebinar REST Webhooks interface allows you quickly and easily send information about nearly created rooms, registered attendees, sessions, new subaccounts, and recordings.

Thanks to the new REST Webhooks function, you have access to the intuitive and easy to use REST interface. The interface makes integration with our API much easier, and also reduces time lost on reading manuals and other documentation.

In which situations should you use REST Webhooks?

- In situations where you want to create multiple types of automations with LiveWebinar
- If you’d like to integrate LiveWebinar with another platform with the help of RESTWebhooks
- If you want to decrease the number of requests to the server and optimize communication between LiveWebinar’s API and another application.

In this manual we’ll show you how to use the RESTWebhooks interface on the example of Pabbly Connect.
Events Available with LiveWebinar REST Webhooks

The list of available events can be found right after entering Integration REST Webhooks from the main menu on our account. They contain information about new data that will be sent from LiveWebinar to the indicated endpoint in the following cases:

- **After creating an event** - choose if you want to send information about each newly created room.
- **After updating a room** - select if you want to send information about each update of the details of a given event.
- **After activating the event** - choose if you want to send information about each activation of the room (which was previously deactivated).
- **After deactivating the event** - choose if you want to send information about each deactivation of the room (to deactivate it, go to the Rooms/Events tab and click the gear button next to the selected room and select Deactivate).
- **After deleting an event** - select if you want to send information about each deletion of an event.
- **After adding a registered (user) to the event** - select if you want to send information about each new registered participant
- **After adding a lead to the event** - select if you want to send information about each new participant who completed the Leads Form.
- **After starting the session** - select if you want to send information about each session start in the webinar room.
- **After stopping the session** - select if you want to send information about each stopping the session in the webinar room.
HTTP Methods Available in LiveWebinar REST Webhooks

LiveWebinar’s REST Webhooks lets you choose between 3 basic HTTP methods that are quite commonly found in most REST API options:

- **GET** – as the name says for data retrieval. Here, a suitable endpoint is enough. Very often GET method verifies whether a given resource exists or not.

  An example query can generate:
  
  http://example.com/api/resource?per_page=10&page=2

- **POST** – method used for is used to create and send new data. In this case, there is already an existing (body) in which we will pass data to our REST API. The response should return an HTTP Location header with the address to the newly created resource.

- **PUT** - very similar to the POST method. PUT is a distinguishing feature of this method is that the PUT query must point to a specific resource. It is worth mentioning that PUT overwrites the entire resource.
1. To add a webhook, we first need to generate a URL that matches a given LiveWebinar Event. For this purpose, we log in to our Pabbly Connect account and select the Connect option.

2. Then, as the Trigger App, choose LiveWebinar and as the so-called. Action App, i.e. the application in which we will trigger the action, we choose any marketing automation platform (in this particular example it will be Mailchimp, but it can also be ActiveCampaign, MailerLite, GetResponse, ConvertKit, etc.)

3. Then we choose the so-called Trigger Event in LiveWebinar (New Registrant) and Action Event in Mailchimp, in this case it will be "Add new member". This way, we will create an
automation that will send the data of each newly registered LiveWebinar participant to our Mailchimp account.

After selecting the appropriate Trigger and Action Events, click the "Try it Now" button. Then we copy the Webhook URL and paste it into the REST Webhooks interface in LiveWebinar. Then we follow the instructions in the points below the link.

- Visit your LiveWebinar dashboard > Integrations > REST Webhooks
- Click Add New Webhook button.
- Select event: widget.registrant.create
- Select method POST and:
- Enter the above webhook URL there.
- Save settings

Capture Webhook Response
4. Enter your LiveWebinar account, select "Integration" and go to "REST Webhooks". Select the event "After adding a registered (user) to the event". Select "POST" as the method and paste the newly generated Webhook URL from Pabbly Connect. Then click the "Save" button.

5. After clicking “Save” we will see the newly added item on our list of webhooks.

6. Then we need to perform the action that we have just indicated by adding the webhook. For this purpose, it is enough to register as a participant for any room with an attached registration form. In this way, we will send the first data package (that is, the data of a registered participant) to Pabbly Connect, so that our automation will be able to "catch" it the next time. After a while, you should see the response received from LiveWebinar in Pabbly Connect (the "Response Received" status will appear).
7. Then go to the Action tab and click the "Connect" button.
8. After clicking the "Connect" button, a new tab will open, in which we must paste the API Key, which we will find in our Mailchimp account. To find the API Key, go to Account à Extras and then API Keys.

9. When we enter the mentioned path in our Mailchimp account, click the "Create a Key" button and then copy the value of the API Key field into the Token field marked in the above screen.
10. Go back to the Pabbly Connect card and enter the value found in our URL (highlighted in the screen above) into the Data Center field. In this case it will be **us20** and click "Save".
Adding Webhooks Containing Information About New Recordings

1. Repeat the first step from the previous chapter. To create a new "workflow", select LiveWebinar with the trigger "Webinar recording ready" as the Trigger App and the so-called Action App select Google Drive with the "Upload a File" Event (it can also be Dropbox or a similar platform, eg Microsoft OneDrive).

2. This way, each time the recording is ready, Pabbly Connect will upload it to our Google Drive. Click the "Try it Now" button.

3. Copy the generated Webhook URL. At this point, Pabbly Connect will be waiting for a response from LiveWebinar, as evidenced by the animated icon and the "Waiting for the Webhook Response" status. We now need to go to the Webhook panel in LiveWebinar and add a new webhook that will send the relevant data to Pabbly Connect.
4. To add a new Webhook according to the instructions, go to Integrations - REST Webhooks, click "Add new Webhook", select POST as the method and "After creating a recording" as the Event. Then click the "Save" button.
5. After adding the Webhook to the list, we must execute the Event on which the Webhook created just now is based. In this case it will be creating a recording. So we have to go to any room, take off, and then stop the recorder. This way Pabbly Connect will learn to capture new recordings and be able to upload them to our Google Drive. When we create recordings in the first part of our workflow, the following table should appear:

6. Go down and click the "Connect" button.
7. Then we must allow PabblyConnect to access our Google Drive, to do this, click "Connect with Google Drive", log in and click the "Allow" button.

8. In the mandatory URL field, select "1. URL:" this is how we indicate to our Google Drive the direct path to the created recording. In the File Name field we can choose the option "1. Name: ", thus retaining the original name of the recording.
9. Then click the "Save & Send Test Request" button for Pabbly Connect to test the automation we have created. When "Response Result Status" receives the value "Success" press the "Save" button.
Adding Other Types of Webhooks

The other types of Webhooks include those that will apply to Events related to the change of the status of a room, sub-account or session. These include Webhooks based on the following “triggers”:

- **After creating an event** - choose if we want to send information about each newly created room
- **After updating a room** - if we want to send information about each update of the details of a given event
- **After activating the event** - if we want to send information about each activation of the room (which was previously deactivated)
- **After deactivating the event** - if we want to send information about each deactivation of the room
- **After deleting an event** - if we want to send information about each deletion of an event
- **After adding a sub-account** - if we want to send information about each newly created sub-account
- **After updating the sub-account** - if we want to send information about each update of the details of a given sub-account
- **After activating a sub-account** - if we want to send information about each activation of a given sub-account
- **To deactivate a sub-account** - if we want to send information about each deactivation of a given sub-account
- **After deleting a sub-account** - if we want to send information about each deletion of a sub-account

The question remains to which platforms we could send information about the above-mentioned events, below we present a short summary of this issue:

<table>
<thead>
<tr>
<th>Type of Webhook</th>
<th>Recommended Platform/Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation / deactivation, event status update</td>
<td>• Calendars (Google Calendar)</td>
</tr>
<tr>
<td></td>
<td>• Instant messaging (Slack, Hangouts),</td>
</tr>
<tr>
<td></td>
<td>• Product Management Software (Jira,</td>
</tr>
<tr>
<td></td>
<td>Asana, Clickup, Monday.com)</td>
</tr>
</tbody>
</table>

**Examples:**

- Every time an event in LiveWebinar is updated, the event is updated in Google Calendar
• When an event in LiveWebinar is deactivated, we can let our team know by sending a message to the selected Slack channel

<table>
<thead>
<tr>
<th>Type of Webhook</th>
<th>Recommended Platform/Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>New event / deletion of event</td>
<td>• Calendars (Google Calendar)</td>
</tr>
<tr>
<td></td>
<td>• Social Media Platforms (Facebook, LinkedIn)</td>
</tr>
<tr>
<td></td>
<td>• Instant messaging (Slack, Hangouts),</td>
</tr>
<tr>
<td></td>
<td>• Product Management Software (Jira, Asana, Clickup, Monday.com)</td>
</tr>
</tbody>
</table>

Examples:

• Each new LiveWebinar event creates a corresponding event in Google Calendar.

• Each new LiveWebinar event causes the post to be published on social media channels.
### Type of Webhook vs. Recommended Platform/Tool

<table>
<thead>
<tr>
<th>Type of Webhook</th>
<th>Recommended Platform/Tool</th>
</tr>
</thead>
</table>
| Adding / removing a subaccount, updating or activating / deactivating a subaccount | • Messenger (Slack, Hangouts),  
• Product Management Software (Jira, Asana, Clickup, Monday.com)             |

**Examples:**

- Each addition of a sub-account creates a task in our project management platform

```
Possible Workflow
When Sub-Account Create in LiveWebinar, Create Issue in Jira Software
LiveWebinar + Jira Software
```

- Updating a sub-account updates the corresponding task in Clickup

```
Possible Workflow
When Sub-Account Update in LiveWebinar, Update Task in ClickUp
LiveWebinar + ClickUp
```

- Deactivating a Sub-account sends a message to the entire channel on Slack

```
Possible Workflow
When Sub-Account Status Disable in LiveWebinar, Send Channel Message (without Image) in Slack
LiveWebinar + Slack
```