LiveWebinar

REST API Webhooks (Pabbly Connect Manual)

Spis treści

Opis 3
Eventy dostępne w LiveWebinar REST Webhooks 4
Metody HTTP dostępne w LiveWebinar REST Webhooks 5
Dodawanie webhooków z informacją o nowych rejestracjach/leadach 6
Dodawanie webhooków z informacją o nowych nagraniach 12
Dodawanie pozostałych rodzajów Webhooków 17

Opis

Interfejs LiveWebinar REST Webhooks pozwala na szybkie i łatwe przesyłanie informacji o nowo utworzonych pokojach, zarejestrowanych użytkownikach, sesjach, nowych subkontach i nagraniach.

Dzięki naszej nowej funkcjonalnośći REST Webhooks masz dostęp do RESTowego interfejsu, który jest intuicyjny i wygodny w użytkowaniu. Interfejs znacznie ułatwia integrację z naszym API, a także redukuje czas poświęcony na czytanie dokumentacji.

W jaki sytuacjach używać REST Webhooks?

- Jeśli chcemy tworzyć różnego rodzaju automatyzacje z użyciem LiveWebinar
- Jeśli chcemy zintegrować LiveWebinar z inną platformą wykorzystującą REST Webhooks
- Jeśli chcemy zmniejszyć ilość zapytań do serwera i zoptymalizować komunikację między API LiveWebinar a inną aplikacją

W tym poradniku pokażemy Ci jak używać interfejsu RESTWebhooks na przykładzie platformy Pabbly Connect.

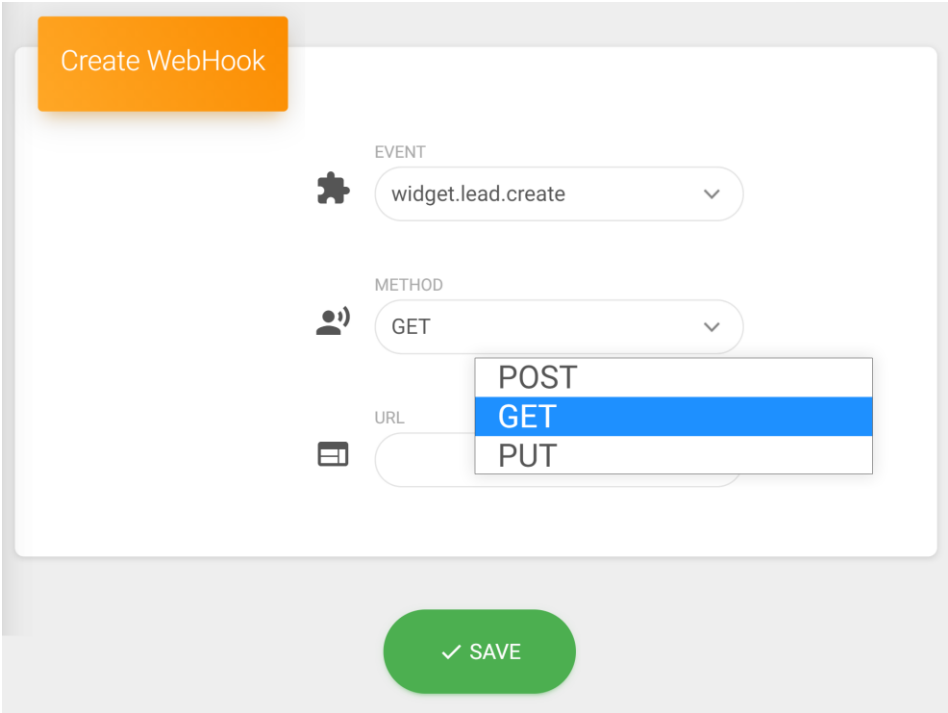
Eventy dostępne w LiveWebinar REST Webhooks

Listę dostępnych Eventów znajdziemy tuż po wejściu w **Integracje → REST Webhooks** z głównego menu na naszym koncie. Dzielą się one na typ danych, które zostaną wysłane z LiveWebinar na wskazany endpoint w następujących przypadkach:

- **Po utworzeniu eventu** – wybieramy jeśli chcemy przesłać informację o każdym nowo utworzonym pokoju.
- **Po zaktualizowaniu pokoju** – wybieramy jeśli chcemy przesłać informację o każdorazowym zaktualizowaniu szczegółów danego wydarzenia.
- **Po aktywowaniu eventu** – wybieramy jeśli chcemy przesłać informację o każdorazowym aktywowaniu pokoju (który wcześniej był zdezaktywowany).
- **Po dezaktywowaniu eventu** – wybieramy jeśli chcemy przesłać informację o każdorazowym dezaktywowaniu pokoju (aby go dezaktywować przechodzimy do zakładki **Pokoje/Wydarzenia** i przy wybranym pokoju klikamy przycisk z zębatką i wybieramy **Deaktywuj**).
- **Po usunięciu eventu** – wybieramy jeśli chcemy przesłać informację o każdorazowym usunięciu wydarzenia.
- **Po dodaniu zarejestrowanego (użytkownika) do eventu** – wybieramy jeśli chcemy przesłać informację o każdym nowym zarejestrowanym uczestniku.
- **Po dodaniu leada do eventu** – wybieramy jeśli chcemy przesłać informację o każdym nowym uczestniku który wypełnił Formularz informacyjny.
- **Po wystartowaniu sesji** – wybieramy jeśli chcemy przesłać informację o każdorazowym rozpoczęciu sesji w pokoju webinarowym.
- **Po zatrzymaniu sesji** – wybieramy jeśli chcemy przesłać informację o każdorazowym zastopowaniu sesji w pokoju webinarowym.

Metody HTTP dostępne w LiveWebinar REST Webhooks

W LiveWebinar REST Webhooks masz do wyboru 3 podstawowe metody HTTP, które spotykane są sporej większości REST API:



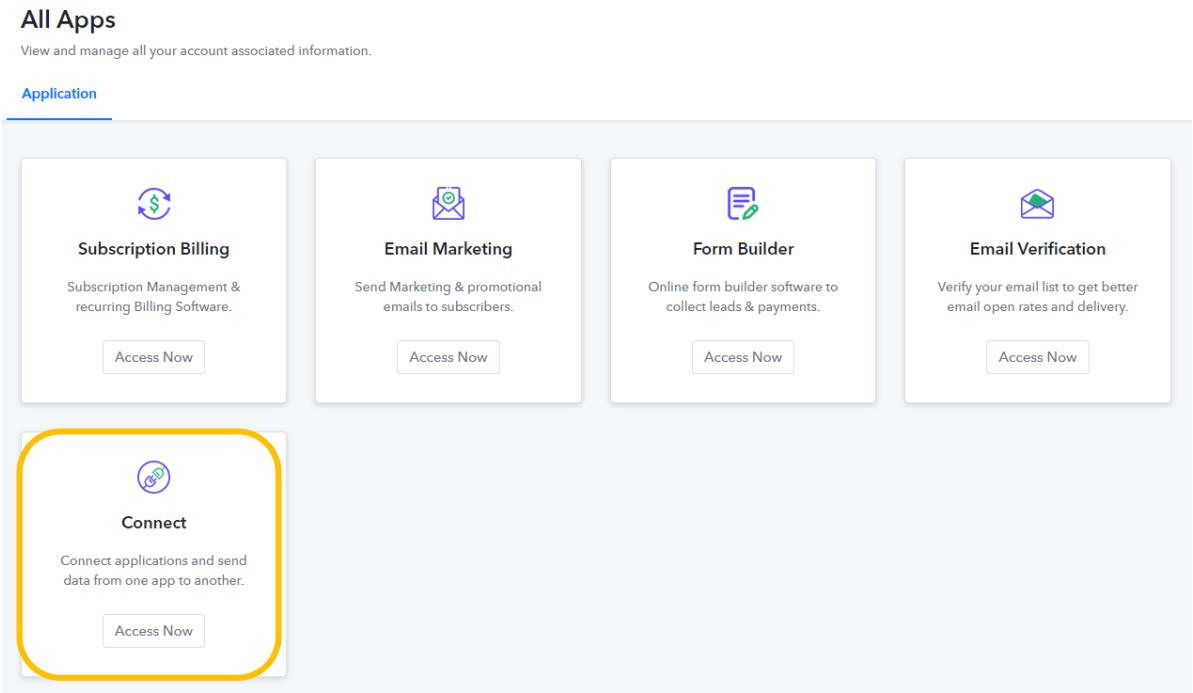
- GET – jak sama nazwa mówi służy do pobierania danych. Tutaj wystarczy podać odpowiedni endpoint. Bardzo częstym przypadkiem użycia metody GET jest weryfikacja czy dany zasób istnieje czy też nie.

Przykładowe zapytanie może wyglądać następująco:
`http://example.com/api/resource?per_page=10&page=2`

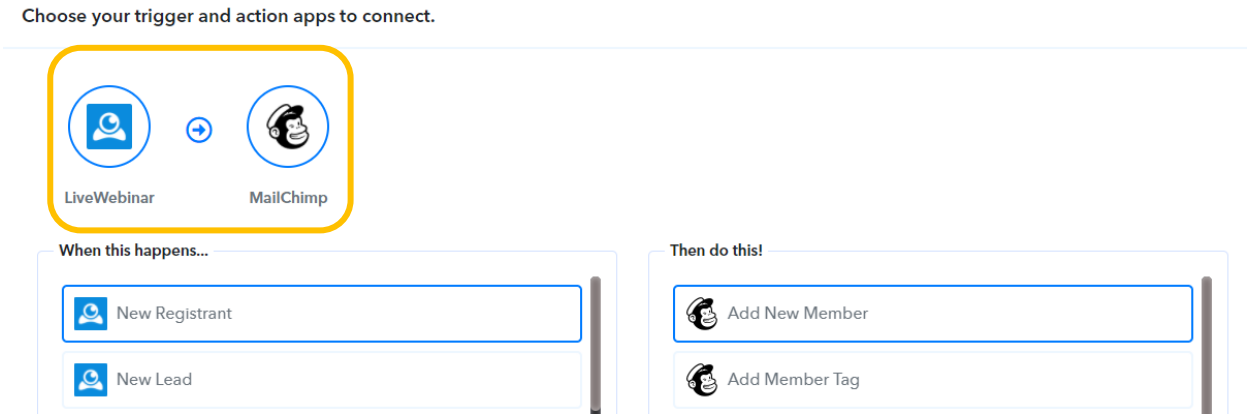
- POST – służy tworzeniu i przesłaniu nowych danych. W tym przypadku konieczne jest już stworzenie ciała (body), w którym przekażemy dane do naszego REST API. W odpowiedzi powinien zostać zwrócony nagłówek HTTP Location z adresem do nowo utworzonego zasobu.
- PUT – bardzo podobna do metody POST. Główną cechą odróżniającą PUT jest to, że zapytanie PUT musi wskazywać na konkretny zasób. PUT jest używany głównie do aktualizowania istniejących zasobów. Wartym wspomnienia jest fakt, że PUT nadpisuje cały zasób.

Dodawanie webhooków z informacją o nowych rejestracjach/leadach

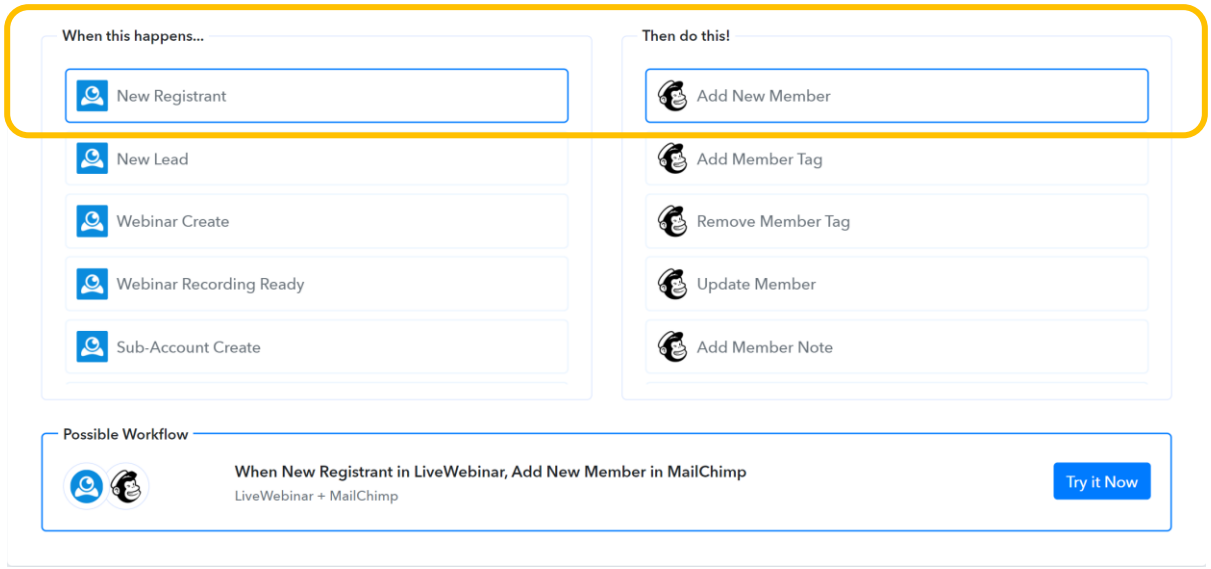
1. Aby dodać webhook musimy napierw wygenerować URL dopasowany do danego Eventu w LiveWebinar. W tym celu logujemy się do naszego konta w Pabbly Connect i wybieramy opcję Connect.



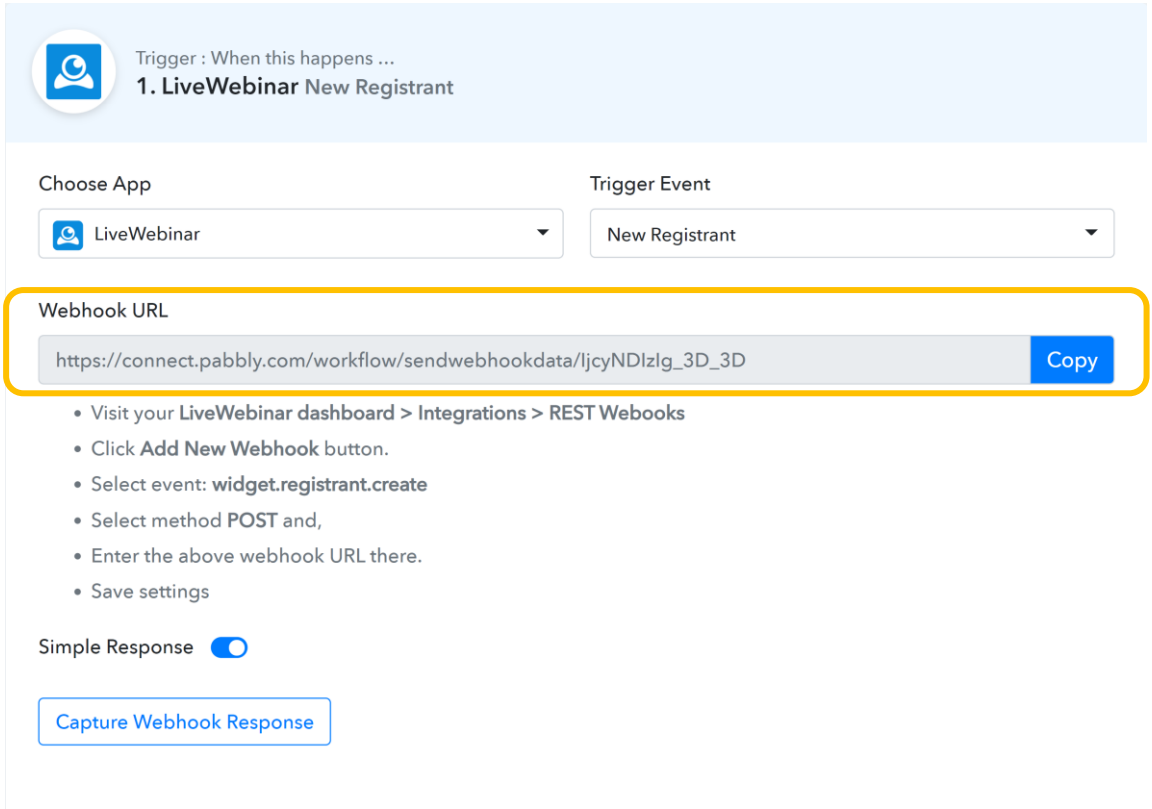
2. Następnie jako Trigger App wybieramy LiveWebinar zaś jako tzw. Action App, czyli aplikację w której będziemy wywoływać dane działanie, wybieramy dowolną platformę do automatyzacji marketingu (w tym konkretnym przykładzie będzie to Mailchimp, ale może to być również ActiveCampaign, MailerLite, GetResponse, ConvertKit itp.)



3. Następnie dobieramy tzw. Trigger Event w LiveWebinar (**New Registrant**) oraz Action Event w Mailchimpie, w tym przypadku będzie to „**Add new member**”. W ten sposób utworzymy automatyzację, która prześle dane każdego nowo zarejestrowanego uczestnika z LiveWebinar na nasze konto w Mailchimp.

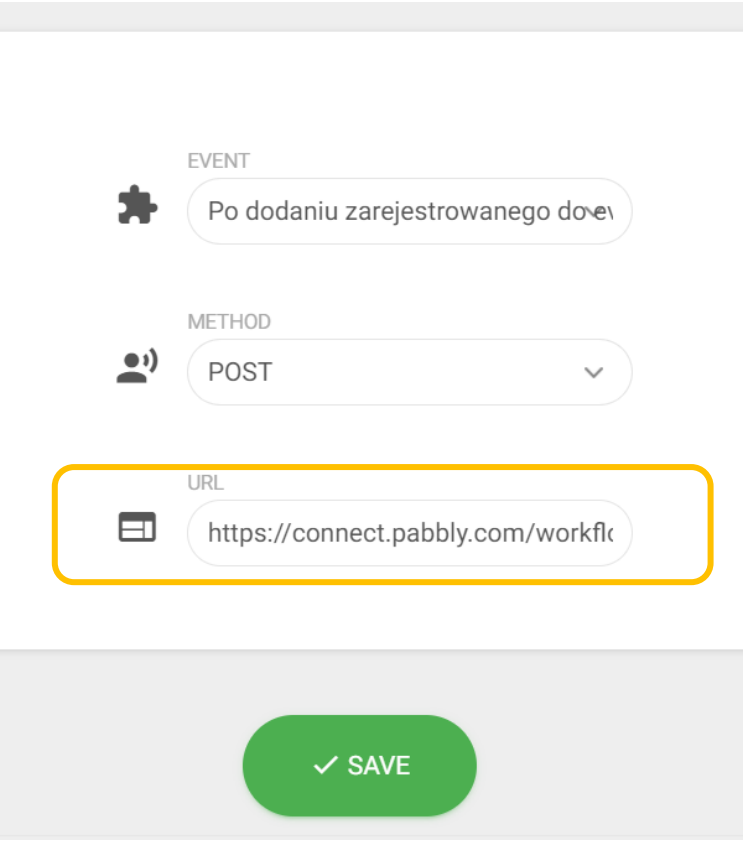


4. Po wybraniu odpowiednich Trigger oraz Action Events klikamy przycisk „**Try it Now**”. Następnie kopiujemy Webhook URL i wklejamy go do interfejsu REST Webhooks w LiveWebinar.

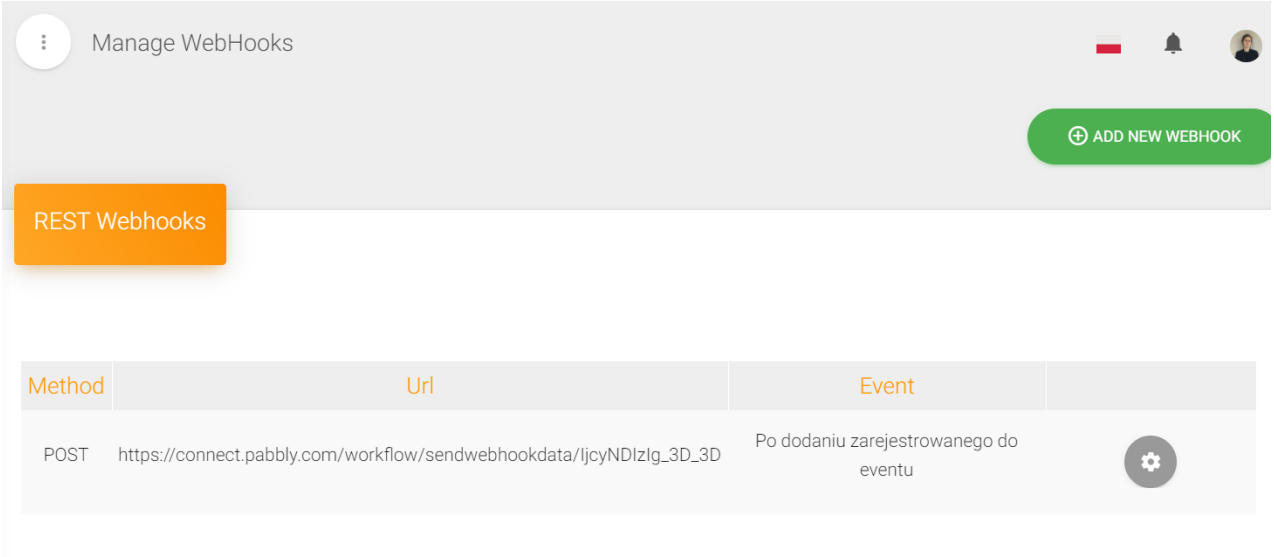


Postępujemy zgodnie z instrukcjami wypisanymi w punktach pod linkiem.

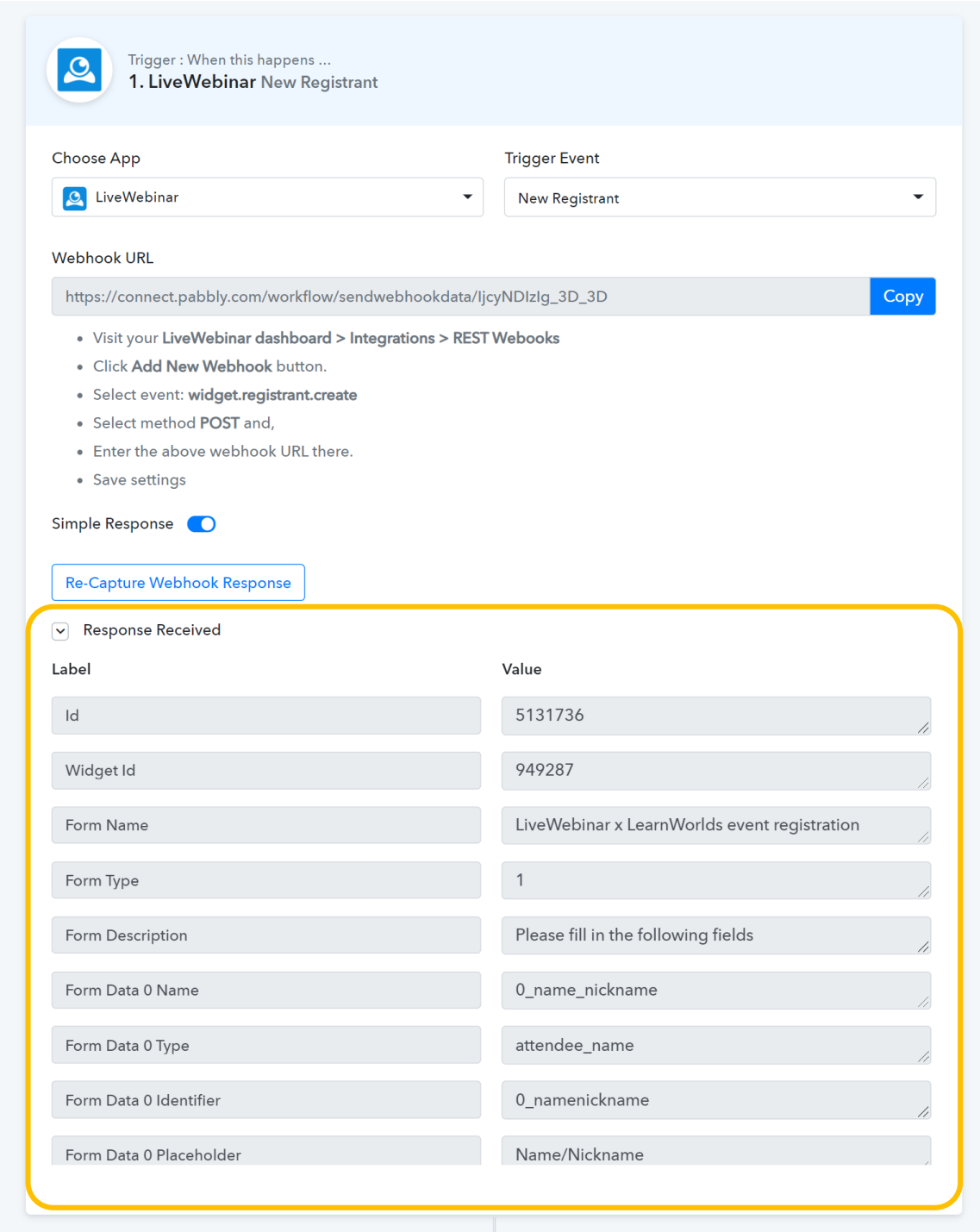
5. Wchodzimy na swoje konto w LiveWebinar, wybieramy „Integracje” i przechodzimy do „REST Webhooks”. Wybieramy wydarzenie „Po dodaniu zarejestrowanego (użytkownika) do eventu”. Jako metodę wybieramy „POST” i wklejamy dopiero co wygenerowany Webhook URL z Pabbly Connect. Następnie klikamy przycisk „**Zapisz**”.



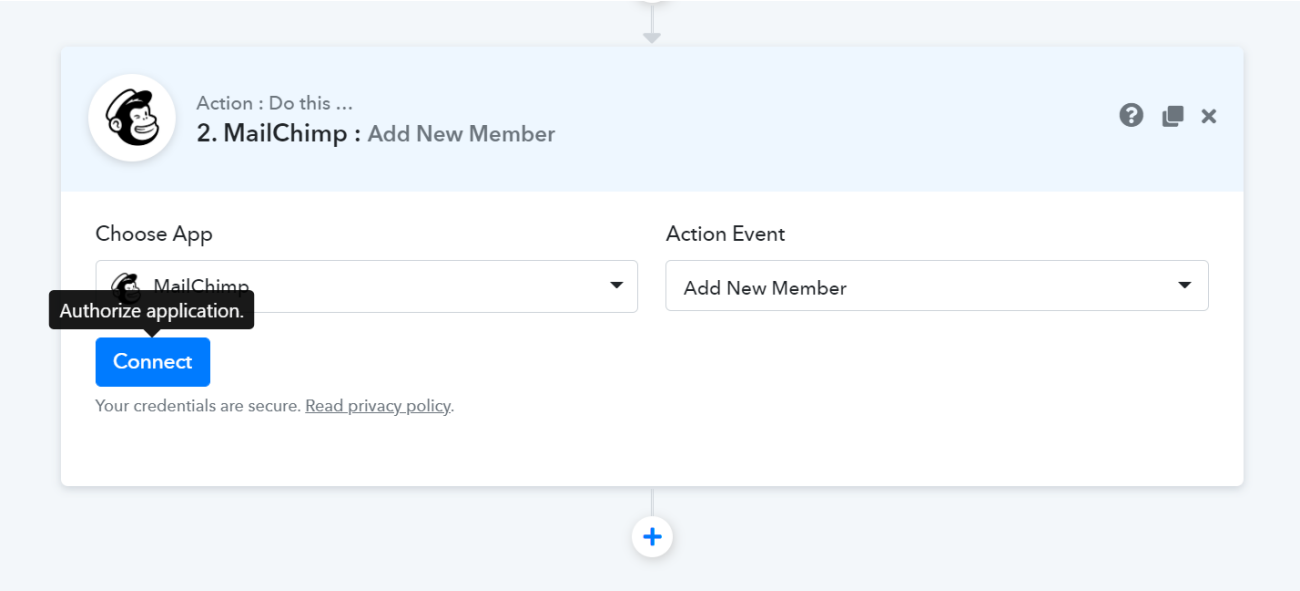
6. Po kliknięciu “**Save**” (Zapisz) zobaczymy na naszej liście webhooków nowo dodaną pozycję.



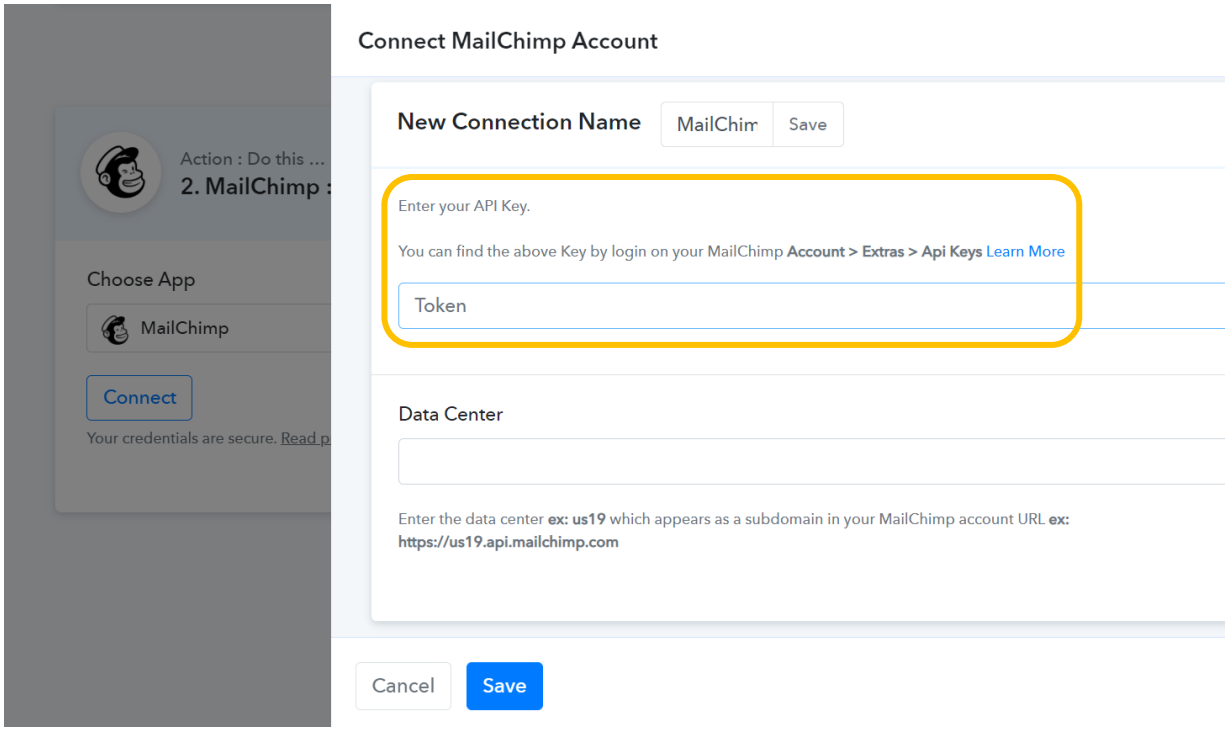
7. Następnie musimy wykonać akcję, którą przed chwilą wskazaliśmy dodając webhook. W tym celu wystarczy zarejestrować się jako uczestnik na dowolny pokój z podłączonym formularzem rejestracyjnym. W ten sposób wyślemy pierwszy pakiet danych (czyli dane zarejestrowanego uczestnika) do Pabbly Connect, przez co nasza automatyzacja będzie mogła je „złapać” następnym razem. Odpowiedź otrzymaną od LiveWebinar powinniśmy po chwili zobaczyć w Pabbly Connect (pojawi się status „Response Received”).



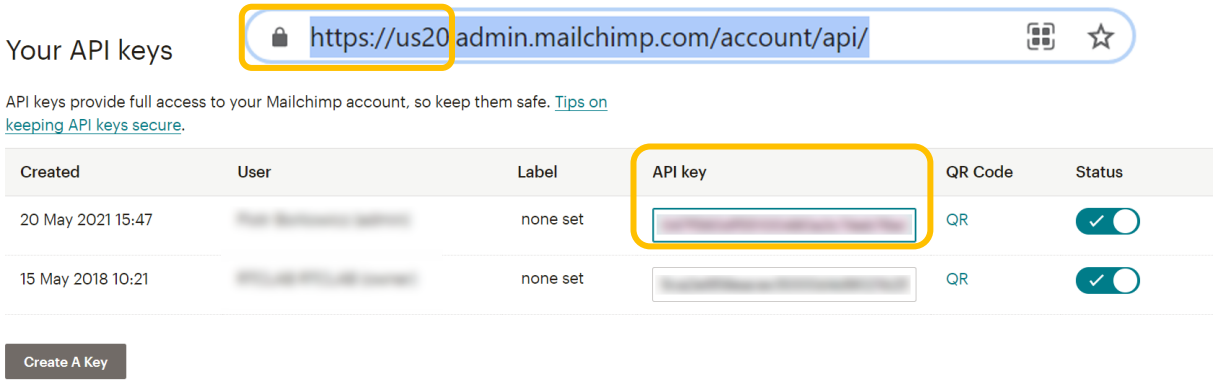
8. Następnie przechodzimy na zakładkę Action i klikamy przycisk „**Connect**”.



9. Po kliknięciu przycisku „**Connect**" otworzy nam się nowa zakładka, w której musimy wkleić API Key, który odnajdziemy na naszym koncie Mailchimp. Aby znaleźć API Key musimy wejść w **Account** → **Extras** a następnie w **API Keys**.



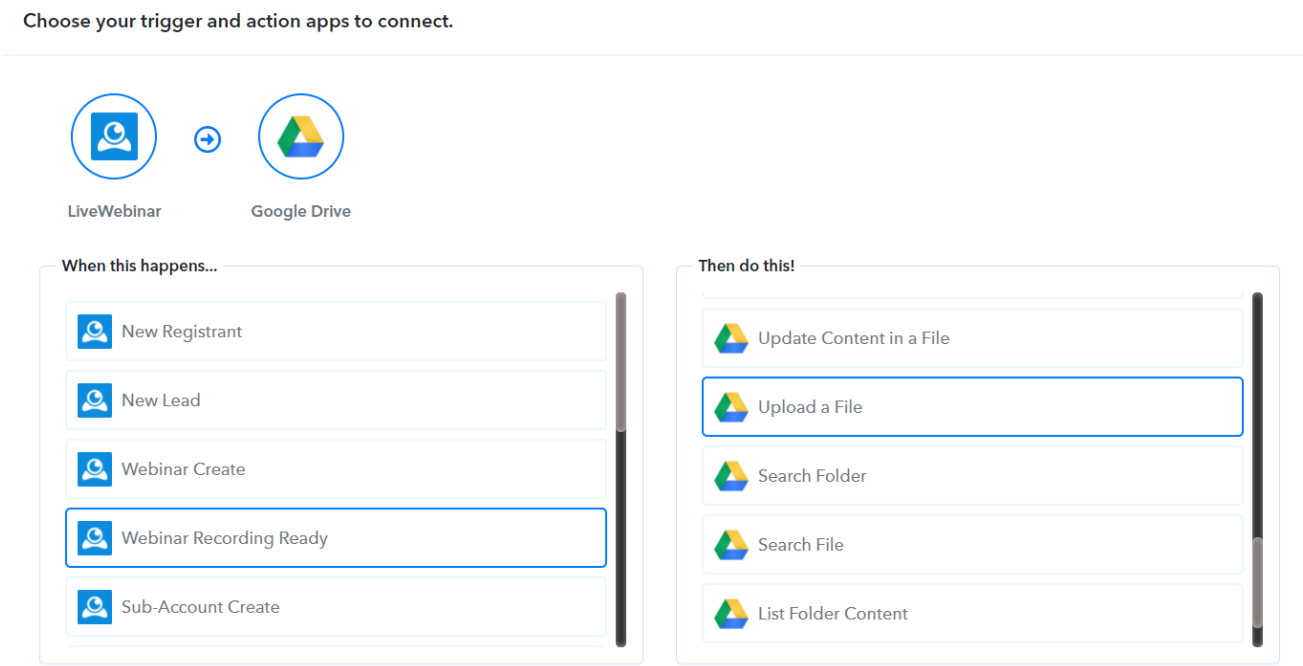
10. Kiedy wejdziemy we wspomnianą ścieżkę na naszym koncie Mailchimp należy kliknąć przycisk „**Create a Key"** a następnie skopiować wartość pola **API Key** w pole **Token** zaznaczone na powyższym screenie.



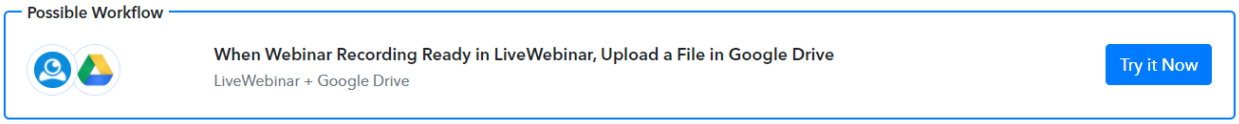
11. Wracamy na kartę z Pabbly Connect i w pole Data Center wpisujemy wartość, którą odnajdziemy w naszym URLu (wyróżniona na screenie powyżej). W tym przypadku będzie to **us20** i klikamy „**Save**”.

Dodawanie webhooków z informacją o nowych nagraniach

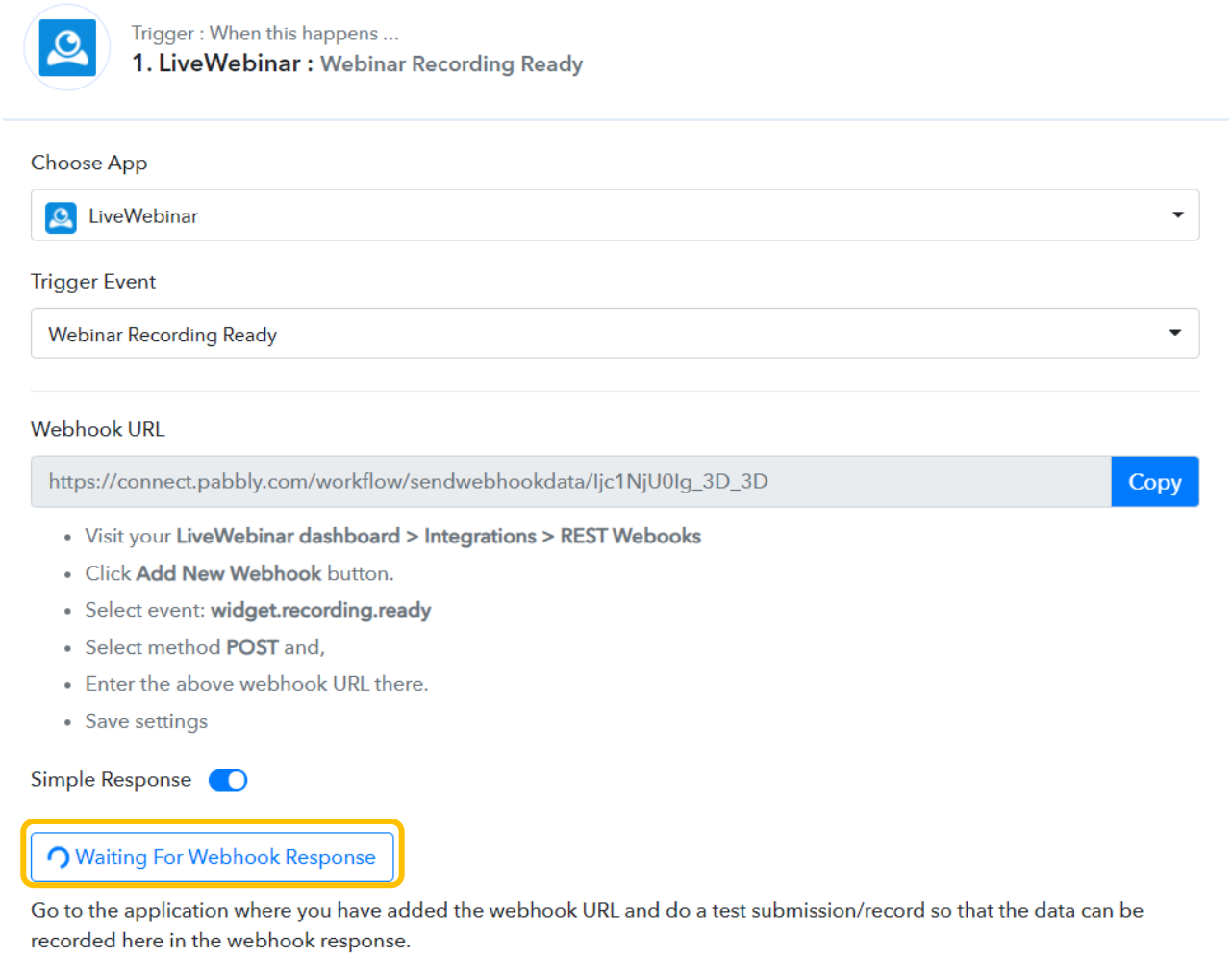
1. Powtarzamy krok pierwszy z poprzedniego rozdziału, czyli tworzymy nowy „workflow”, jako Trigger App wybieramy LiveWebinar z triggerem „**Webinar recording ready**” zaś jako tzw. Action App wybieramy Google Drive z Eventem „**Upload a File**” (może to być również Dropbox lub podobna platforma np. Microsoft OneDrive).



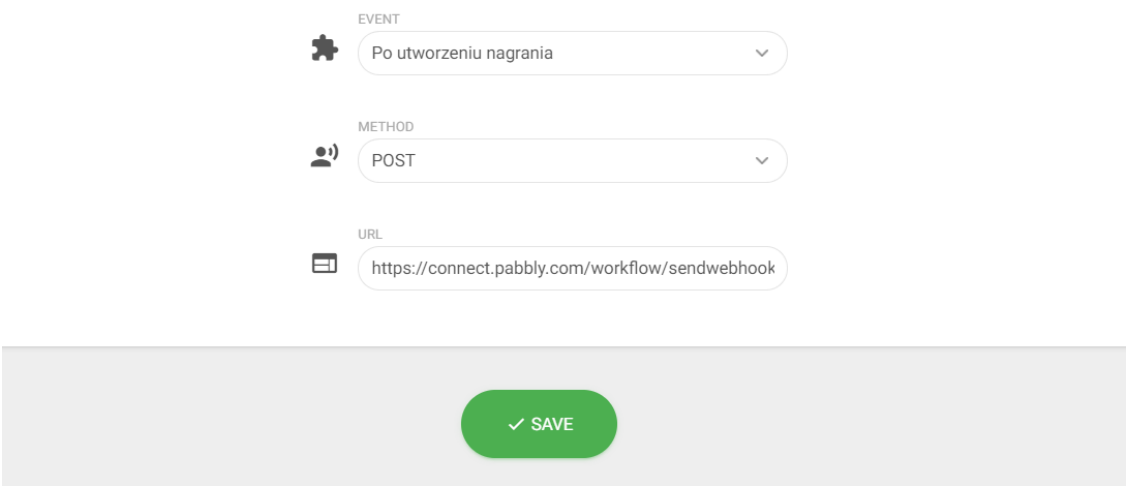
2. Tym sposobem z każdym razem kiedy nagranie będzie gotowe Pabbly Connect zauploaduje je na nasz Google Drive. Klikamy przycisk „**Try it Now**”.



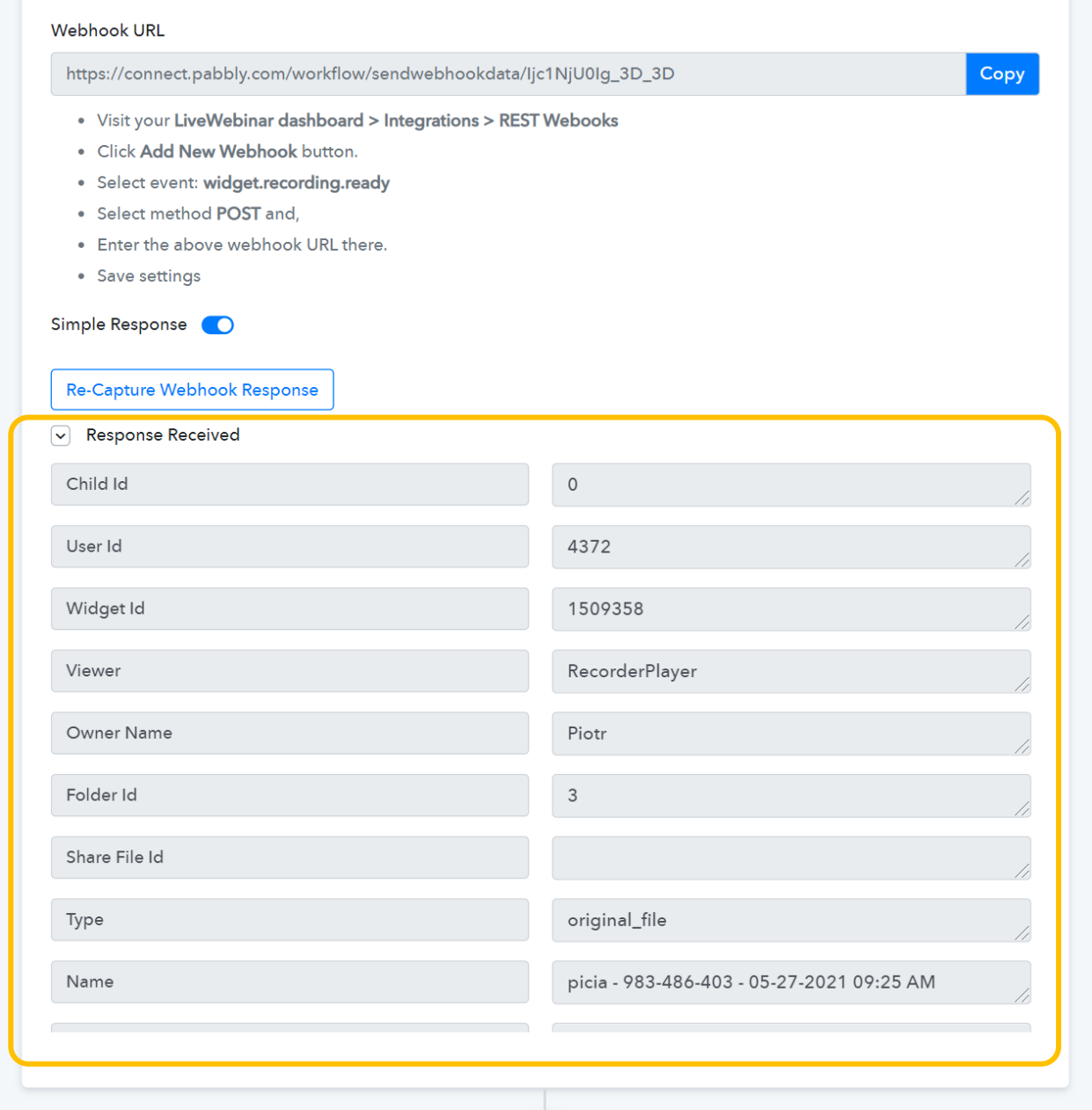
3. Kopiujemy wygenerowany Webhook URL. W tym momencie Pabbly Connect będzie czekał na odpowiedź z LiveWebinar, świadczy o tym animowana ikona i status „**Waiting for the Webhook Response**”. Musimy teraz przejść do panelu Webhooków w LiveWebinar i dodać nowego webhooka, który wyśle odpowiednie dane do Pabbly Connect.



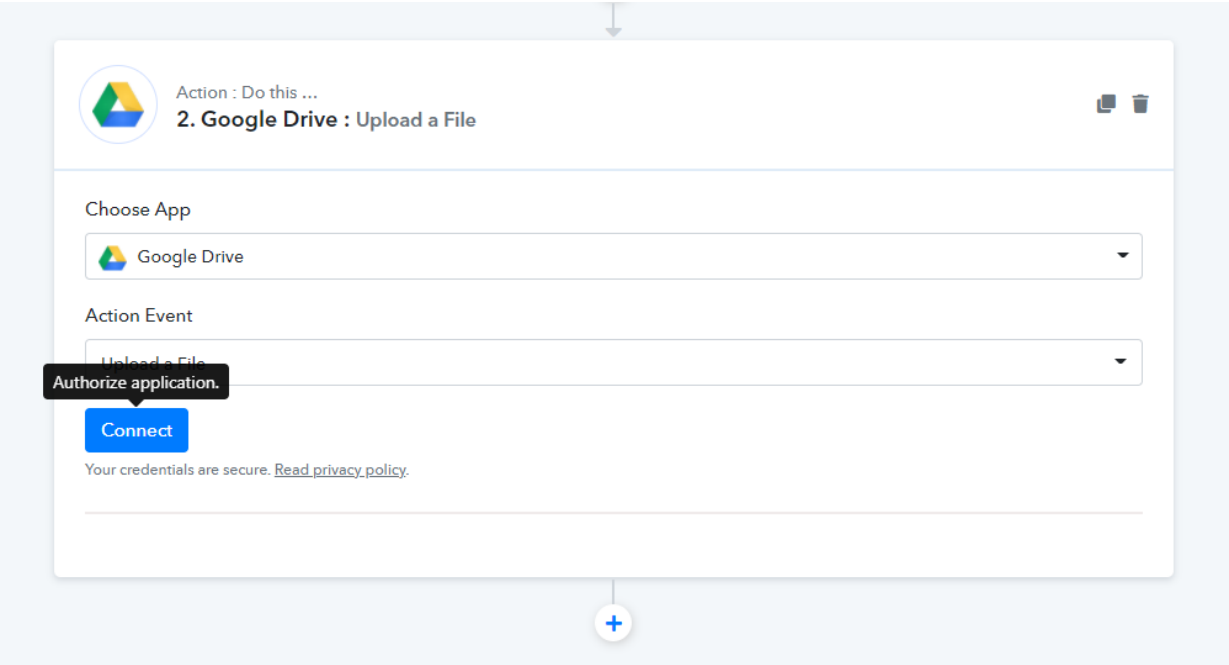
4. Dodajemy nowego Webhooka zgodnie z instrukcją, wchodzimy w Integracje → REST Webhooks, klikamy „**Dodaj nowego Webhooka**”, jako metodę wybieramy **POST** a jako Event „**Po utworzeniu nagrania**”. Następnie klikamy przycisk „**Save**”



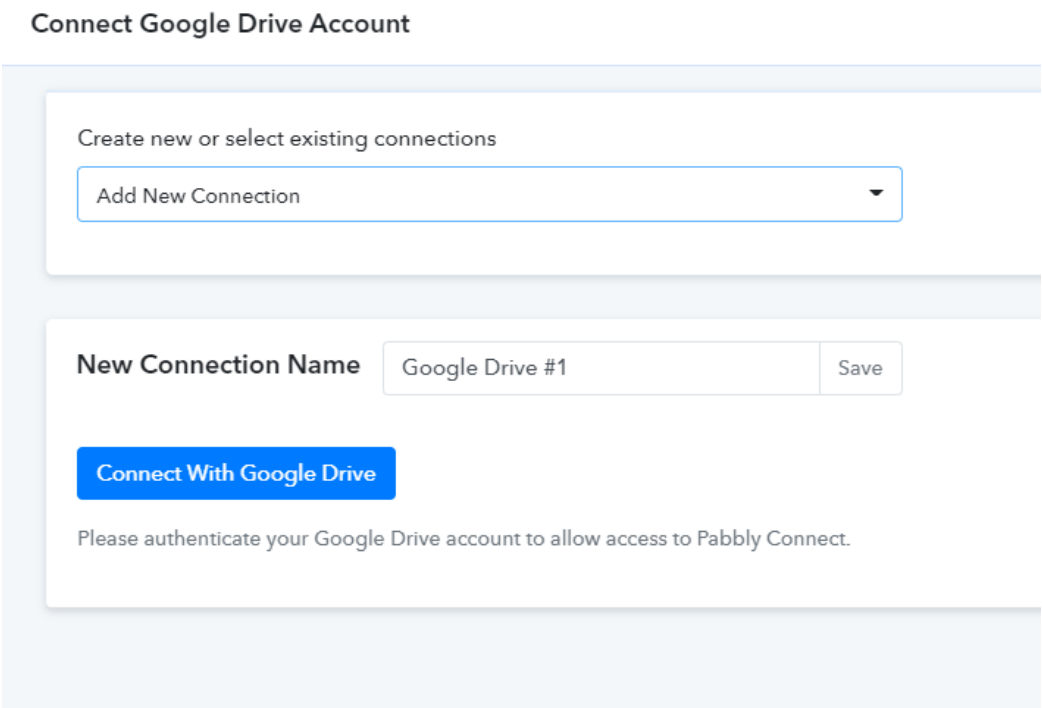
5. Po dodaniu Webhooka na listę musimy wykonać Event, na którym opiera się przed chwilą utworzony Webhook. W tym przypadku będzie to utworzenie nagrania. Musimy zatem wejść na dowolny pokój, wystartować, po czym zastopować nagrywarkę. W ten sposób Pabbly Connect nauczy się przechwytywać nowe nagrania i będzie w stanie uploadować je na nasz Google Drive. Kiedy utworzymy nagrania w pierwszej części naszego workflow powinna się pojawić taka tabelka:



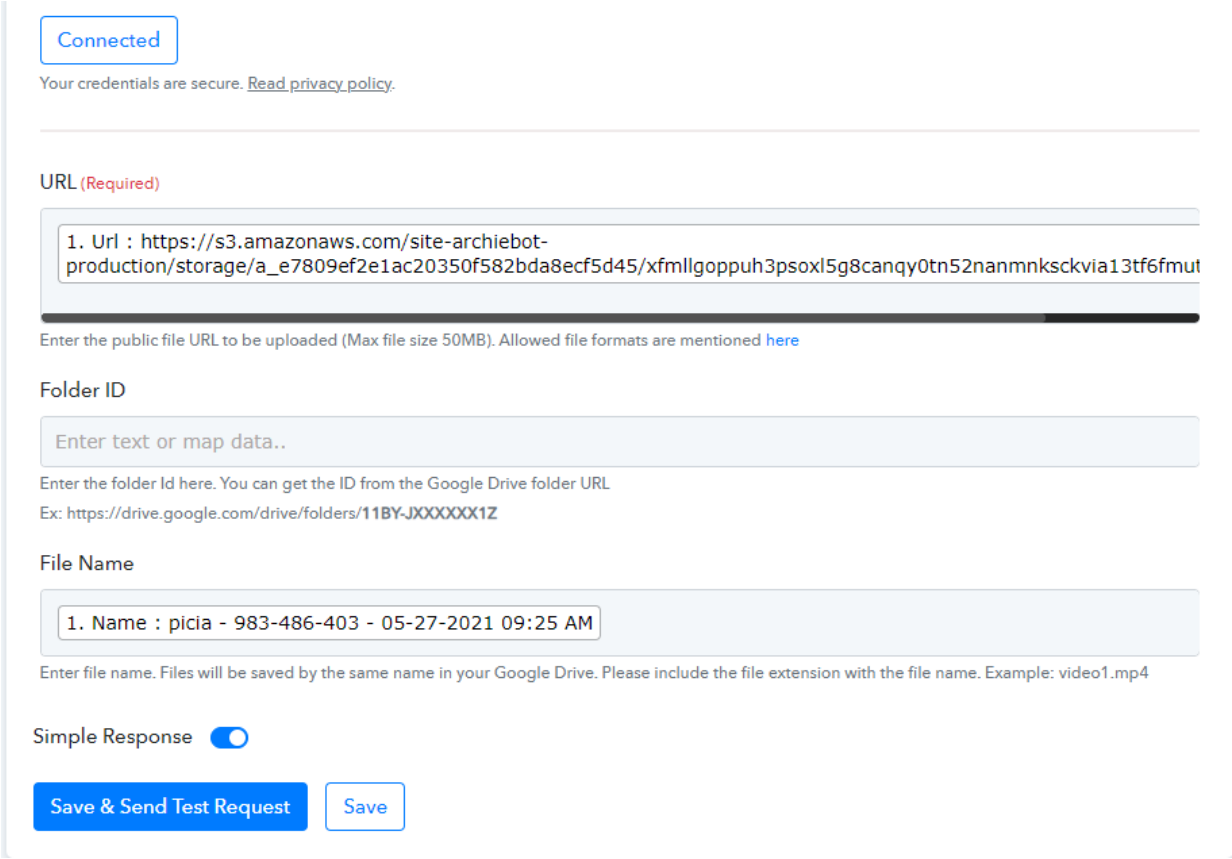
6. Schodzimy w dół po czym klikamy przycisk „**Connect**”.



7. Następnie musimy pozwolić PabblyConnect na dostęp do naszego dysku Google, w tym celu klikamy „**Connect with Google Drive**”, logujemy się i klikamy przycisk „**Zezwól**”.



8. W obowiązkowym **polu URL** z rozwijalnej listy wybieramy „**1. Url:**” w ten sposób wskazujemy naszego Google Drive bezpośrednią ścieżkę do utworzonego nagrania. W polu **File Name** możemy wybrać opcję „**1. Name:**” dzięki czemu zachowany oryginalną nazwę nagrania.



9. Następnie klikamy przycisk „**Save & Send Test Request**”, aby Pabbly Connect przetestował utworzoną przez nas automatyzację. Kiedy „**Response Result Status**” otrzyma wartość „**success**” naciskamy przycisk „**Save**”.

Dodawanie pozostałych rodzajów Webhooków

Do pozostałych rodzajów webhooków można zaliczyć te, które będą dotyczyły Eventów wiążących się ze zmianą statusu jakiegoś pokoju, subkonta, czy sesji. Zaliczamy do nich webhooki oparte na następujących „triggerach”:

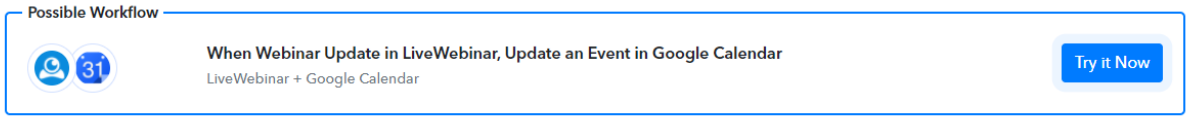
- **Po utworzeniu eventu** – wybieramy jeśli chcemy przesłać informację o każdym nowo utworzonym pokoju.
- **Po zaktualizowaniu pokoju** – jeśli chcemy przesłać informację o każdorazowym zaktualizowaniu szczegółów danego wydarzenia.
- **Po aktywowaniu eventu** – jeśli chcemy przesłać informację o każdorazowym aktywowaniu pokoju (który wcześniej był zdezaktywowany).
- **Po dezaktywowaniu eventu** – jeśli chcemy przesłać informację o każdorazowym dezaktywowaniu pokoju.
- **Po usunięciu eventu** – jeśli chcemy przesłać informację o każdorazowym usunięciu wydarzenia.
- **Po dodaniu subkonta** – jeśli chcemy przesłać informację o każdym nowo utworzonym subkoncie.
- **Po zaktualizowaniu subkonta** – jeśli chcemy przesłać informację o każdorazowym zaktualizowaniu szczegółów danego subkonta.
- **Po aktywowaniu subkonta** – jeśli chcemy przesłać informację o każdorazowym aktywowaniu danego subkonta.
- **Do dezaktywowaniu subkonta** – jeśli chcemy przesłać informację o każdorazowym dezaktywowaniu danego subkonta.
- **Po usunięciu subkonta** – jeśli chcemy przesłać informację o każdorazowym usunięciu subkonta.

Pozostaje pytanie do jakich platform moglibyśmy przesłać informację o wyżej wymienionych zdarzeniach, poniżej prezentujemy krótkie zestawienie dotyczące tej kwestii:

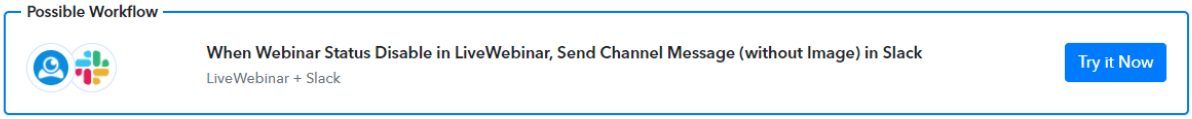
Rodzaj Webhooka	**Zalecana platforma/narzędzie**
Aktywacja/deaktywacja, aktualizacja statusu wydarzenia	• Kalendarze (Google Calendar) • Komunikatory (Slack, Hangouts), • Product Management Software (Jira, Asana, Clickup, Monday.com)

Przykłady:

- Przy każdorazowej aktualizacji wydarzenia w LiveWebinar dokonuje się update tego wydarzenia w Google Calendar.



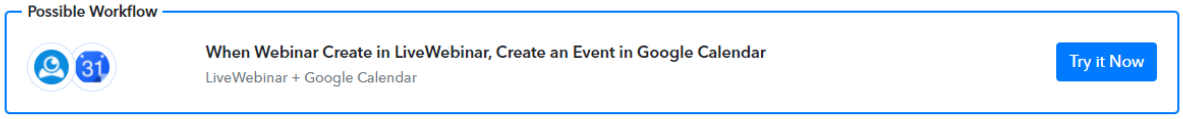
- Kiedy wydarzenie w LiveWebinar zostanie deaktywowane możemy poinformować o tym nasz zespół wysyłając wiadomość na wybrany kanał Slack.



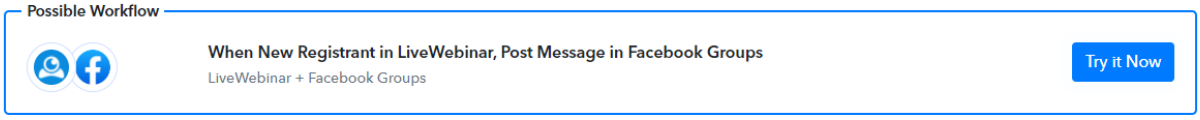
Rodzaj Webhooka	Zalecana platforma/narzędzie
Nowe wydarzenie/usunięcie wydarzenia	• Kalendarze (Google Calendar) • Platformy Mediów Społecznościowych (Facebook, Linked.in) • Komunikatory (Slack, Hangouts), • Product Management Software (Jira, Asana, Clickup, Monday.com)

Przykłady:

- Każdy nowe wydarzenie w LiveWebinar powoduje utworzenie odpowiadającego mu wydarzenia w Google Calendar.



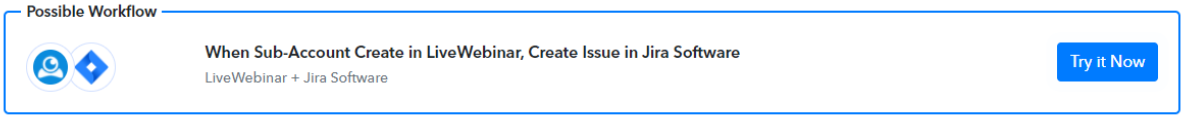
- Każdy nowe wydarzenie w LiveWebinar powoduje publikację posta na kanałach mediów społecznościowych.



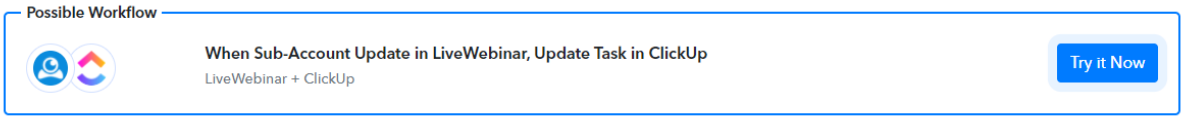
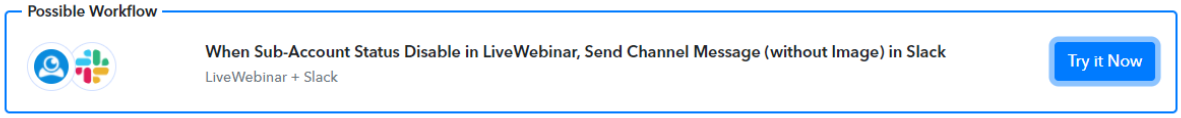
Rodzaj Webhooka	Zalecana platforma/narzędzie
Dodanie/usunięcie subkonta, aktualizacja lub aktywacja/deaktywacja subkonta)	• Komunikatory (Slack, Hangouts), • Product Management Software (Jira, Asana, Clickup, Monday.com)

Przykłady:

- Każde dodanie subkonta tworzy zadanie w naszej platformie do zarządzania projektami.



- Aktualizacja subkonta aktualizuje odpowiadający mu task w Clickup.



- Deaktywacja Subkonta powoduje wysłanie wiadomości do całego kanału na Slacku.