All about tokens and accounts

Deployteq - Suzanne Martens - 2026-02-23 - Comments (0) - Data, Webhooks & API's

The Deployteq webhook offers the possibility to receive data from external systems. Check our documentation for installing and using the Webhook or follow the e-learning about Webhooks and Deployteq!

App Token

Authentication for the webhook can be done in various ways. When installing the webhook via the installation wizard, you will be asked in step 2 which authentication method you want to choose. Here, you can choose between Bearer Token and Basic Authentication. In both cases, the created token (or password) only gives you access to that specific webhook. We call this an App token. This means that if you create multiple webhooks, each webhook will have its own token, which is linked to the user account of the person who created the webhook.

Integration Token or API Token

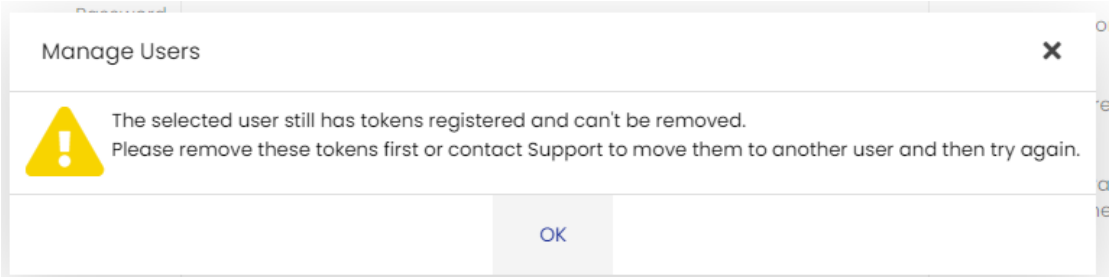
When you create an integration with multiple webhooks, it might be wise to use a token created on a user, an integration token. Such a token can be used for all webhooks, making token management much easier. A user with access to user management can create a new user with a token. The steps are explained in this guide.

What Happens if an Account Becomes Inactive?

A token (both App and API) is always linked to an active Deployteq user account. It can happen that the user account to which the token is linked has become inactive after 60 days of no activity, or that the user has left the company and you want to delete the account. In the first case, the associated token(s) will no longer work and cannot be used to authenticate that webhook. You will receive a 401 error on your webhook call.

The account in question can be reactivated by a user with access to user management (read more about reactivating an account here).

The token linked to the account can then also be used again to successfully authenticate the webhook. If the user to whom the tokens are linked has left the company, you will likely want to delete the account. The system will warn you if you try to delete a user with active tokens. Deletion is not possible as long as tokens are linked to it.



You can deny the user access to the system by changing the password of the account in question as an administrator. The account will remain active, and the webhooks will continue to function. You can then follow the solution described below.

Solution

To prevent webhooks from malfunctioning due to inactive user accounts, we always recommend adding the token to the App Token System User. In most cases, this is done automatically when setting up a new webhook. This account can never become inactive.

Note

The use of a token for a webhook call is seen as a login attempt by the system. If webhook calls are made regularly, this user account will also remain active automatically.

If you are already in a situation where a webhook is not functioning due to an inactive user, you can resolve this by creating a new token on another user account and using this new token for the existing webhook.

Note: This will also impact the systems that connect to this webhook, as you will need to update the token there as well.