

Callback HTTP sur IJITRACK

Guide utilisateur

Guide utilisateur:

Date de publication 2026-03-06

Table des matières

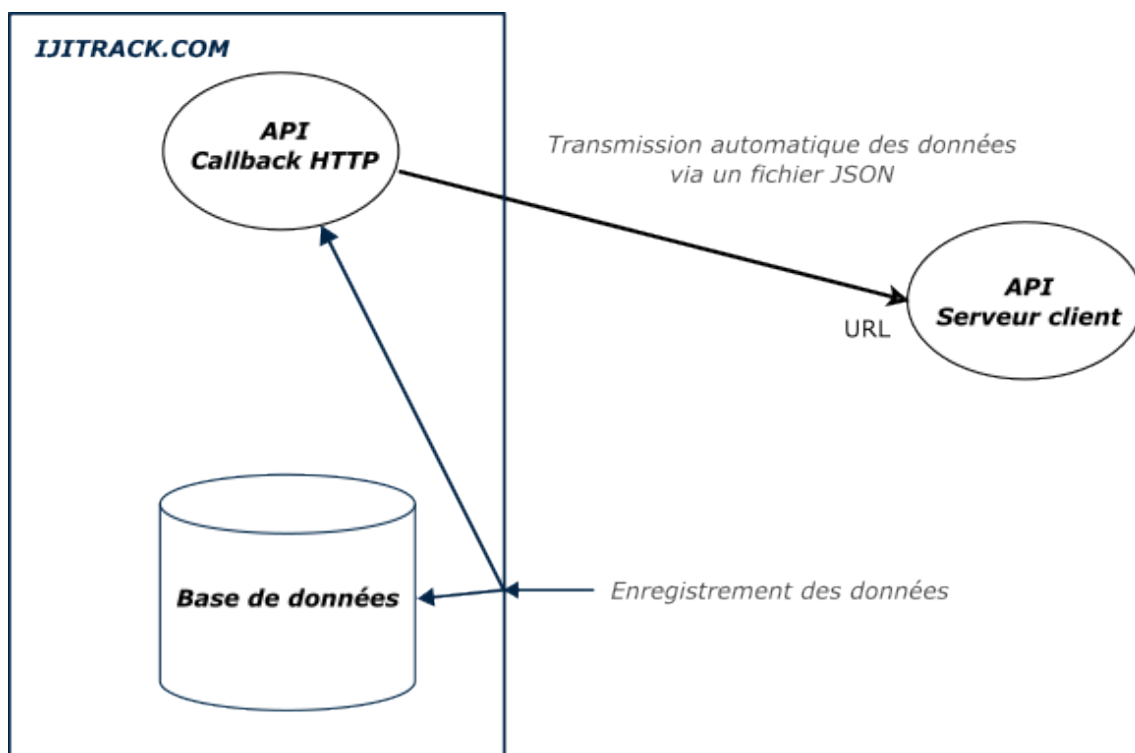
1. Création d'un callback HTTP sur Ijitrack	4
1.1. Callback HTTP (Webhook)	4
1.1.1. Principe	4
1.1.2. Création d'un callback HTTP de type POST	5
<i>Callback source - Configuration de l'évènement</i>	<i>5</i>
<i>Callback content - Configuration de la Requête HTTP POST</i>	<i>6</i>
<i>Options du callback</i>	<i>6</i>
1.1.3. Test du callback	7
1.1.4. Tester une réception de données en PHP	8
1.1.5. Tester en ligne	8

Chapitre 1. Création d'un callback HTTP sur Ijitrack

1.1. Callback HTTP (Webhook)

1.1.1. Principe

Le Callback HTTP (type Webhook) permet d'établir une communication entre 2 API afin de transmettre automatiquement des données de Ijitrack vers un serveur client.



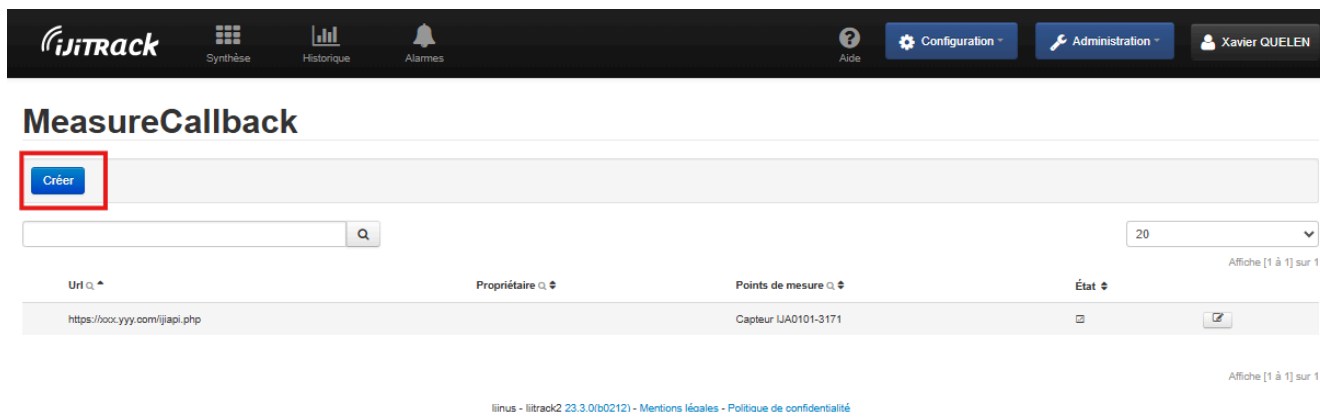
Pour configurer un Callback, on transmet une URL unique à l'API d'Ijitrack et on précise l'**événement déclencheur**. Ensuite, l'envoi des données utiles vers l'**URL** du Callback se fait automatiquement dès que l'événement spécifié survient.

C'est le serveur d'Ijitrack qui envoie au client une demande HTTP POST unique dès que les données sont disponibles, et non le serveur client qui envoie des requêtes en continu.

- **Événement déclencheur** : une nouvelle mesure qui se produit dans Ijitrack sur un objet ou un groupe d'objets pour un type de donnée sélectionné (voir [Callback source](#)).
- **URL du Callback** : Ijitrack envoie une requête à une URL prédéfinie. L'application réceptrice écoute les données entrantes à cette URL (voir [Callback content](#)).
- **Transmission des données** : les données associées à l'événement déclencheur sont envoyées sous forme de charge utile (format JSON) à l'URL du callback via une requête HTTP POST (voir [Callback content](#)).

1.1.2. Création d'un callback HTTP de type POST

- Se connecter à Ijitrack, avec votre compte, puis aller sur la page <https://www.ijitrack.com/measurecallback/list>.
- Pour créer un nouveau callback, cliquer sur le bouton **Créer**.



MeasureCallback

Créer

Url  Propriétaire  Points de mesure  État 

https://xxx.yyy.com/ijapi.php	Capteur IJA0101-3171	OK
-------------------------------	----------------------	----

Affiche [1 à 1] sur 1

Ijinus - Ijitrack2 23.3.0(b0212) - Mentions légales - Politique de confidentialité

Callback source - Configuration de l'évènement

Sur chaque nouvelle mesure sur : Définit l'évènement qui déclenche la création et l'envoi du fichier JSON vers l'URL du serveur. Il se fait soit sur un :

- **Groupe** : L'envoi est réalisé à chaque nouvelle mesure pour tous les capteurs associés au **groupe** sélectionné.
- **Objet** : Seules les données du **point de mesure** sélectionné seront transmises

Type de données :

- Sélectionner le ou les types de données à transférer sur le serveur, par exemple *Batterie* et *distance*.



On peut définir autant de types que nécessaire, celles-ci devront être définies dans **élément du contenu de la requête**.

Callback source

Sur chaque nouvelle mesure sur

Object
 

Points de mesure

Capteur IJA0101-3171
 

Type de données

Batterie
 
 Distance [45]
 

Exemple de callback paramétré pour une transmission pour chaque nouvelle mesure reçue au point de mesure "Capteur IJA0101-3171" pour les données "Batterie" et "Distance"

Callback content – Configuration de la Requête HTTP POST

Callback content

Méthode

POST

Url

https://xxx.yyy.com/ijiapi.php

avant le contenu

{

élément du contenu de la requête

{ "sn": "{sn}", "name": "{name}", "date": "{date:yyyy-MM-dd HH:mm:ss}" {if_0{ , "BATTERY":{0} }} {if_1{ , "DISTANCE":{1} }} }

après le contenu

}

Méthode : Sélectionner POST. Les méthodes GET et PUT sont disponible mais non recommandées.

URL : Saisir l'URL de votre serveur. Exemple : https://xxx.yyy.com/api

-> Les données seront envoyées à cette adresse.



Comme pour **élément du contenu de la requête**, il est possible d'insérer les données directement dans l'url via les {...}.

Exemple :https://xxx.yyy.com/api/{sn}/mesures

avant le contenu

- Dans le cadre d'un POST json, mettre {.

élément du contenu de la requête

Exemple 1 :{ "sn": "{sn}", "name": "{name}", "date": "{date:yyyy-MM-dd HH:mm:ss}" {if_0{ , "BATTERY":{0} }} {if_1{ , "DISTANCE":{1} }} }

Dans cet exemple, on transfère le numéro de série du logger (IJI---), le nom du capteur, la date courante, la batterie pour le type de donnée 0, la distance pour le type de donnée 1. La condition (**if_x**) permet de n'envoyer la donnée que si elle est présente.



Pour l'écriture de la requête, Il est important de suivre le même ordre que celui de la sélection de **type de données**.

Type de données

Batterie ✕ Distance [45] ✕

après le contenu

- Dans le cadre d'un POST json, mettre }.

Options du callback

Callback options

Http content-type

MeasureCallback.field.maxSend

Personnalisé

Http content-type

- Pour un POST JSON, saisir **application/json**

maxSend

Défini le nombre maximum de données par envoi. (exemple : 10).

Authentification (Optionnel)

Si nécessaire, on peut renseigner le type, le login et mot de passe.

Authentification

Utiliser l'authentification http

☒ oui

Type

SPNEGO

SPNEGO

NTLM

Kerhis

DIGEST

KERBEROS

BASIC

Login

Mot de passe

05/03/2026, 07:55:01

☐ Faire la requête http

1.1.3. Test du callback

Pour visualiser les données au format programmé :

- Sélectionner la période désirée, puis **Save & launch test**.

Exemple :

Send to Url : `https://xxx.yyy.com/api`

```
{
  "sn": "{sn}",
  "name": "{name}",
  "date": "{date:yyyy-MM-dd HH:mm:ss}"
  {if_0{ , "BATTERY":{0} }}
  {if_1{ , "DISTANCE":{1} }} } }
```

En cochant **Faire la requête http**, les données sont envoyées à l'URL configurée plus haut.

Test callback

17/03/2026, 08:57:22

18/03/2026, 08:57:22

☒ Faire la requête http

Save & Launch test

1.1.4. Tester une réception de données en PHP

Le script suivant enregistre dans data.txt les données qui lui sont transmises.

```
$json_str = file_get_contents('php://input'); $fp = fopen('data.txt', 'a'); echo "Json STR : $json_str";
fwrite($fp, date("F j, Y, g:i a") . " : Data : ". $json_str. "\n\n"); fclose($fp); die(""); ?>
```

Exemple: log pour un callback configuré en **Objet**

```
March 6, 2026, 10:01 pm : Data : {{ "sn": "IJIA101-3171", "name":"Capteur IJIA101-3171","date":"2026-03-06
18:00:00" ,"BATTERY": 3.45,"DISTANCE":2654.0 }}
```

```
March 6, 2026, 10:01 pm : Data : {{ "sn": "IJIA101-3171", "name":"Capteur IJIA101-3171","date":"2026-03-06
19:00:00" ,"BATTERY": 3.45,"DISTANCE":2657.0 }}
```

```
March 6, 2026, 10:01 pm : Data : {{ "sn": "IJIA101-3171", "name":"Capteur IJIA101-3171","date":"2026-03-06
21:00:00" ,"BATTERY": 3.45,"DISTANCE":2649.0 }} ...
```

Exemple: log pour un callback configuré en Groupe

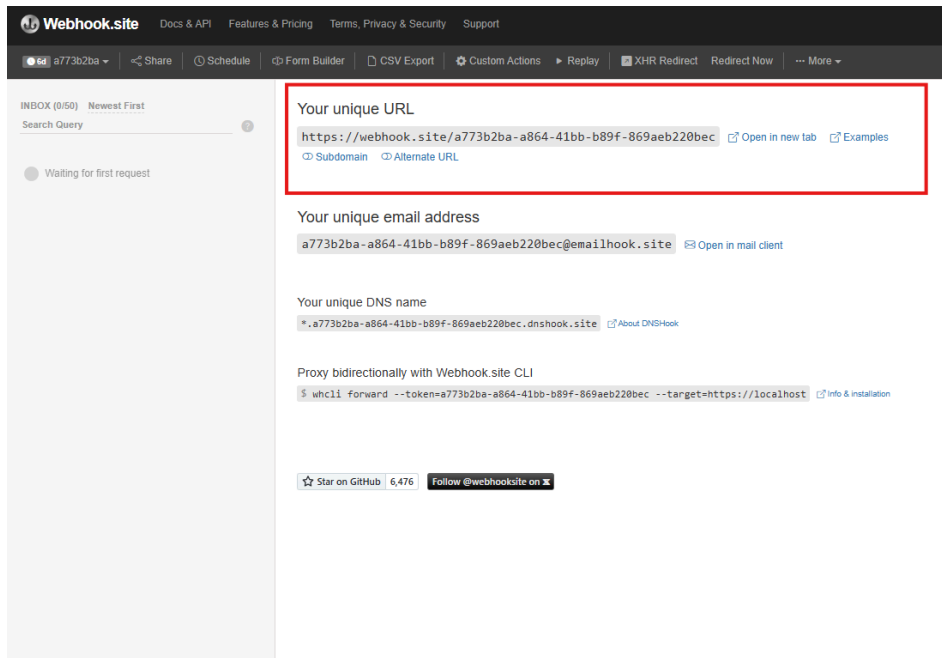
Il y a un seul appel pour toutes les mesures :

```
March 6, 2026, 10:01 pm : Data : {{ "Mesure tous objets, sn": "IJC0101-00000322",
"name":"Capteur IJC0101-00000322","date":"2026-03-07 00:00:32" ,"BATTERY":3.45 }}{ "Mesure tous objets,
sn": "IJC0101-00000326", "name":"Capteur IJC0101-00000322","date":"2026-03-06 15:00:00","BATTERY":3.45,
"DISTANCE":2710.0 }}{ "Mesure tous objets, sn": "IJC0101-00000326", "name":"CAPTEUR
IJC0101-00000326","date":"2026-03-06 16:00:00","BATTERY":3.45, "DISTANCE":2705.0 }...
```

1.1.5. Tester en ligne

Pour tester le Callback, il est possible d'utiliser des services en ligne tel que *Webhook.site* :

- Remplacer le lien donné par le site en tant qu'URL du callback.



The screenshot shows the Webhook.site interface. At the top, there's a navigation bar with links like 'Docs & API', 'Features & Pricing', 'Terms, Privacy & Security', and 'Support'. Below this is a toolbar with various actions like 'Share', 'Schedule', 'Form Builder', 'CSV Export', 'Custom Actions', 'Replay', 'XHR Redirect', 'Redirect Now', and 'More'. The main content area displays the 'Your unique URL' as `https://webhook.site/a773b2ba-a864-41bb-b89f-869aeb220bec`, with links to 'Open in new tab' and 'Examples'. Below this, it shows the 'Your unique email address' as `a773b2ba-a864-41bb-b89f-869aeb220bec@emailhook.site` with a link to 'Open in mail client'. Further down, it displays the 'Your unique DNS name' as `*.a773b2ba-a864-41bb-b89f-869aeb220bec.dnshook.site` with a link to 'About DNSHook'. At the bottom, there's a section for 'Proxy bidirectionally with Webhook.site CLI' showing a command: `$ whcli forward --token=a773b2ba-a864-41bb-b89f-869aeb220bec --target=https://localhost` with a link to 'Info & Installation'. The footer includes a GitHub star button and a social media follow button for @webhooksite.