

# Etat de l'API

Last edited by Maelan LE BORGNE 4 years ago

# Etat de l'API

Le traitement asynchrone introduit la notion d'état de l'API. En effet il faut empêcher une solution d'initier une nouvelle session de synchronisation si une session est déjà en cours. Il faut également que la solution sache quand elle doit attendre la fin du traitement de sa synchro et quand est ce que le package d'instruction est prêt à être télécharger. Pour cela, l'API dispose de 3 états par type de communication, que la solution vient interroger avant chaque appel pour savoir ce qu'il convient de faire (voir GET /v4/type\_de\_communication/statut).

L'état d'une solution pour un type de synchro correspond à une ligne dans la table **epack\_manager.api\_status**, identifiée par le couple (**solution\_id,task\_code**). La classe correspondante est [ApiStatus](#).

- **solution\_id** identifie la solution
- **task\_code** correspond au type de synchro
  - 1 => synchro des données
  - 2 => synchro des températures sondes
  - 3 => synchro messagerie
- **status\_code** correspond au status
  - 0 => Prêt à traiter une synchro
  - 1 => Traitement en cours / en attente
  - 2 => Package d'instruction prêt a être télécharger / en attente d'acquitement
- **last\_updated** correspond à la date de dernière mise à jour de l'état (voir [réinitialisation des synchro stagnantes](#) )

Un [Worker](#) vient chaque minute remettre à 0 le **status\_code** des synchros stagnantes. On considère qu'une opération de synchronisation stagne si elle reste plus de x minutes dans l'état "Traitement en cours / en attente" (status\_code = 1). Le nombre de minutes est de 30 par défaut et se configure dans la clé OTHER\_CONST.STALE\_STATUS\_TIME\_LIMIT du fichier app/config/config.yaml)