

# Corrigé

P1C3 - Décryptez le fonctionnement d'une API



## 1. Le verbe HTTP et son adéquation

Le verbe à utiliser est `GET`. L'objectif est de \*récupérer\* des données existantes sans créer, modifier ni supprimer quoi que ce soit. `GET` est le verbe adapté chaque fois que vous lisez des informations.

## 2. Décomposition de la requête

Verbe : `GET`

Endpoint : `[URL de base MockAPI]/orders`

Paramètre : `status=pending`, ajouté à l'URL sous la forme `?status=pending`

En-tête d'authentification : `X-API-Key: [clé fournie dans les ressources]`

Dans n8n, vous saisissez l'URL complète dans le champ endpoint, sélectionnez `GET` dans le menu verbe, et ajoutez l'en-tête dans la section *Headers*.

## 3. Code 401 : que s'est-il passé ?

Le code `401` signifie *Unauthorized* : la requête a été rejetée car le serveur n'a pas pu vous identifier. Les causes les plus fréquentes sont : la clé d'API absente de l'en-tête, une clé incorrecte ou expirée, ou un nom d'en-tête mal orthographié. Vérifiez que la clé est bien transmise dans le bon en-tête, tel qu'indiqué dans les ressources du chapitre.

## 4. Code 200 : que faire ensuite ?

Le code `200` confirme que la requête a réussi. Le serveur a renvoyé la liste des commandes avec le statut `pending`. Dans n8n, cette réponse est disponible en sortie du nœud *HTTP Request* : vous pouvez la brancher directement au nœud suivant de votre workflow, par exemple pour mettre à jour Airtable, envoyer une alerte dans Slack ou déclencher un email Mailchimp pour chaque commande en attente.