

Filtres, tri et pagination

Une collection peut contenir des milliers d'éléments. Le client doit pouvoir n'en demander qu'une partie, dans l'ordre qui l'intéresse. Tout passe par les **paramètres de query** : la ressource reste la même, on ne change que la vue.

Les filtres

Un paramètre par critère, avec le nom du champ :

```
GET /v1/cocktails?alcohol_free=true
GET /v1/cocktails?tag=classic&max_price=10
GET /v1/orders?status=pending
```

Pas de filtre dans le chemin : `/cocktails/alcohol-free` laisse croire qu'il s'agit d'une ressource à part, et ne se combine pas avec les autres filtres.

Le tri

Un paramètre `sort` avec le nom du champ. Un `-` devant inverse l'ordre, et on peut trier sur plusieurs champs séparés par des virgules :

GET /v1/cocktails?sort=price	prix croissant
GET /v1/cocktails?sort=-price	prix décroissant
GET /v1/cocktails?sort=-price,name	prix décroissant, puis nom

Un tri par défaut doit toujours exister (souvent par `id` ou par date) : sans lui, l'ordre peut changer d'une page à l'autre et des éléments apparaissent en double.

La pagination

Par numéro de page (offset)

```
GET /v1/cocktails?page=2&per_page=5
```

- ☐ Simple, on peut sauter directement à la page 5.
- ☐ Si des éléments sont ajoutés pendant qu'on parcourt, on voit des doublons ou on en rate.
- ☐ Lent sur de très grosses tables (la base doit compter les lignes à sauter).

Par curseur

```
GET /v1/orders?limit=20
GET /v1/orders?limit=20&cursor=eyJpZCI6NDJ9
```

Le serveur renvoie un **curseur** opaque qui marque la position atteinte. C'est le choix de Stripe, Slack ou Twitter.

- ☐ Stable même si des données arrivent, et rapide.
- ☐ Impossible de sauter à une page précise.

Ce que la réponse doit contenir

Le client doit savoir où il en est et comment continuer. Deux approches, souvent combinées :

- des métadonnées et des liens dans le body ;
- le header `Link` (RFC 8288), utilisé par exemple par GitHub.

```
{
  "items": [ ... ],
  "page": 2,
  "per_page": 5,
  "total": 13,
  "_links": {
    "self": {"href": "/v1/cocktails?page=2&per_page=5"},
    "first": {"href": "/v1/cocktails?page=1&per_page=5"},
    "prev": {"href": "/v1/cocktails?page=1&per_page=5"},
    "next": {"href": "/v1/cocktails?page=3&per_page=5"},
    "last": {"href": "/v1/cocktails?page=3&per_page=5"}
  }
}
```

```
Link: </v1/cocktails?page=3&per_page=5>; rel="next", </v1/cocktails?page=3&per_page=5>;
rel="last"
```

Fixez toujours une **limite maximale** à `per_page` . Sinon, quelqu'un finira par demander `?per_page=1000000` , et votre serveur s'en souviendra.

En résumé

Besoin	Convention
Filtrer	<code>?champ=valeur</code>
Trier	<code>?sort=- champ1, champ2</code>
Paginer	<code>?page=&per_page=</code> ou <code>?cursor=&limit=</code>
Naviguer	<code>_links</code> et/ou header <code>Link</code>

Revision #3
Created 2 October 2026 20:00:48 by Thibaud FRICHET
Updated 2 October 2026 21:13:59 by Thibaud FRICHET