

## TD4 : collection, filtrage et pagination

L'objectif du TD est de programmer des requêtes portant sur des collections et comportant du filtrage de valeurs et de la pagination.

**Contexte :** On réalise un service de suivi de commandes prises en ligne pour un vendeur de sandwich. On programme une route qui permet au point de vente de lister les commandes pour les afficher sur les terminaux afin d'organiser leur fabrication et livraison.

### 0. préparation

Il faut créer un service distinct du service catalogue. Ce service utilise sa propre base de données, distincte de la base de données du catalogue. Importer dans cette base le schéma et les données fournies.

### 1. collection

Programmer la requête qui permet d'obtenir la collection complète des commandes.

L'objet json retourné doit contenir des méta-données incluant notamment le nombre d'éléments dans la collection, et un tableau de commandes. Chaque commande est décrite par son id, nom du client, date de création et date de livraison attendue, état, et contient un lien vers la ressource commande correspondante. Les commandes sont triées par date de livraison (croissante) et date de création (croissante).

Le json aura donc la forme suivante :

```
{
  "type": "collection",
  "count": 1502,
  "commands": [
    {
      "command": {
        "id": "e0369177-dbc6-4811-a8fc-cf5c353ac59b",
        "nom": "Rodriguez",
        "created_at": "2018-10-21 17:39:41",
        "livraison": "2018-10-21 17:44:26",
        "status": 2
      },
      "links": {
        "self": { "href" : "/commands/e0369177-dbc6-4811-a8fc-cf5c353ac59b/" }
      }
    },
    {
      "command": {
        "id": "afd9f7d8-83db-4000-a1b7-7c9011bc48f4",
        "nom": "Bruneau",
        "created_at": "2018-10-21 17:39:54",
        "livraison": "2018-10-21 17:46:38",
        "status": 4
      },
      "links": {
        "self": { "href" : "/commands/afd9f7d8-83db-4000-a1b7-7c9011bc48f4/" }
      }
    },
    ...
  ]
}
```

### 2. filtrage

On souhaite pouvoir filtrer les commandes selon leur état : lister les commandes payées, les commandes

en cours de fabrication, les commandes prêtes à être livrées. Pour cela, on met en place un paramètre optionnel dans l'uri collection sur les commandes, permettant de sélectionner un état.

Ainsi, la requête :

GET /commands retourne toutes les commandes enregistrées, et la requête :

GET /commands?s=2 retourne toutes les commandes qui sont dans l'état "2".

L'objet json retourné aura la même forme que dans le cas de l'uri non filtrée.

### 3. pagination

On souhaite mettre en place un mécanisme de pagination permettant de parcourir la collection lorsqu'il y a un grand nombre de données. Le mécanisme de pagination doit être utilisable en même temps que les filtres.

Le principe consiste à ajouter dans l'uri un numéro de page. Par défaut, on considère que les pages contiennent 10 éléments.

Ainsi, l'uri :

- GET /commands: par défaut, retourne la 1ère page, c'est à dire les 10 premiers éléments,
- GET /commands?page=3 : retourne la 3ème page, c'est à dire, 10 éléments à partir du 21ème.

Les méta-données de la réponse indiquent :

- count : le nombre d'éléments total répondant à la requête (indépendamment de la pagination)
- size : le nombre d'éléments dans la page courante

### 4. pagination avancée

Compléter le mécanisme de pagination avec les fonctionnalités suivantes :

1. on peut indiquer la taille des pages souhaitées dans la requête, qui remplace alors la taille par défaut : GET /commands?page=3&size=15 retourne 15 éléments à partir du 31ème.
2. si le numéro de page demandé est < 0, retourne la 1ère page,
3. si le numéro de page demandé est supérieur au nombre de pages, retourne la dernière page.
4. des liens permettant de parcourir les pages sont ajoutés dans la réponse.

GET /commands?page=4&size=15

```
{
  "type": "collection",
  "count": 1502,
  "size": 15,
  "links": {
    "next": {
      "href": "/commands/?page=5&size=15"
    },
    "prev": {
      "href": "/commands/?page=3&size=15"
    },
    "last": {
      "href": "/commands/?page=94&size=15"
    },
    "first": {
      "href": "/commands/?page=1&size=15"
    }
  },
  "commands": [
    ...
  ]
}
```