Conception & Développement d'Applications et Services Web/Serveur

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.

0. préparation

Créer le service "catalogue" qui propose une api pour parcourir, sélectionner et accéder aux produits proposés par le vendeur de sandwichs.
Il s'agit d'un service distinct du service de prise des commandes et qui utilise sa propre base de données de type mongodb ou sql.
Des données initiales en format sql ou en format tableau json sont disponibles dans arche.
Charger dans un premier temps uniquement le 1er jeu de données.

1. collection

Programmer la requête qui permet d'obtenir la collection complètes des sandwichs.
L'objet json retourné doit contenir des méta-données incluant notamment le nombre d'éléments dans la collection, et un tableau de sandwichs. Chaque sandwich est décrit par sa référence, nom, type_pain et prix, et contient un lien vers la ressource sandwich correspondante.Les sandwichs sont triés par nom.
Le json aura donc la forme suivante :

```
{
        "type": "collection",
        "count": 158,
        "date": "10-11-2020",
        "sandwichs": [
                { "sandwich": {
                        "ref": "s2004",
                        "nom": "le bucheron",
                        "type_pain": "baguette campagne",
                        "prix": "6.00"
                        },
                 "links": {
                        "self": {"href": "/sandwichs/s2004/" }
                        }
                 },
                { "sandwich": {
                        "ref": "s2005",
                        "nom": "jambon-beurre",
                        "type_pain": "baguette",
                        "prix": "5.25"
                        },
                  "links": {
                        "self": {"href": "/sandwichs/s2005/" }
                        }
                },

       ...
         ]
}
```

2. filtrage

On souhaite pouvoir filtrer les sandwich selon le type de pain. 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 /sandwichs` retourne tous les sandwichs enregistrés, et la requête :
`GET /sandwichs?t=baguette` retourne tous les sandwichs dont le type de pain contient "baguette" .
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. Pour réaliser des tests plus complets et réalistes, charger le 2nd jeu de données disponible dans arche.

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 /sandwichs`: par défaut, retourne la 1ère page, c'est à dire les 10 premiers éléments,
- `GET /sandwichs?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 /sandwichs?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 /sandwichs?page=4&size=15
```


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


 "sandwichs": [
```

```
        ...
  ]
}
```