

## TD2 : connexion à la base de données, gestion des erreurs

L'objectif du TD est de connecter votre micro-api à une base de données et mettre en place une gestion des codes d'erreurs retournés dans les réponses.

**Contexte** : on réalise le service de prise de commandes pour l'application de vente de sandwiches. Une commande est composée de plusieurs items, chaque item contenant simplement le tarif et la quantité commandées, et une référence vers un élément dans un catalogue. Des données SQL sont disponibles dans Arche.

### 1. Base de données

Compléter les 2 routes programmées lors du TD1 pour que les données soient issues de la base de données SQL. La structure des objets json produits doit rester la même. Utiliser les données fournies pour créer votre base SQL.

### 2. Gestion des erreurs

On souhaite mettre en place une gestion des cas d'erreur dans notre micro api. Une erreur est signalée à l'émetteur de la requête au travers d'une réponse http contenant :

- un code de retour correspondant à un cas d'erreur (4xx, 5xx),
- un body contenant un objet json de description de l'erreur

```
{  
  "type": "error",  
  "error": 404,  
  "message": "ressource non disponible : /commandes/ABCDE-12345/"  
}
```

#### a. Erreur 404

Programmer les cas correspondant à l'erreur 404.

Lorsque la ressource demandée n'existe pas, l'api retourne une erreur de type 404 NOT FOUND. C'est le cas par exemple lorsque l'id indiqué dans la requête ne correspond pas à une ressource existante.

#### b. Erreur 400

Programmer les cas correspondant à l'erreur 400.

Lorsque la requête est mal formée, l'api retourne une erreur de type 400 BAD REQUEST. C'est le cas notamment lorsque l'URI indiquée n'est pas connue de l'API.

Par exemple : GET /command/ABCDE-12345 doit retourner un code 400

#### c. Erreur 500

Programmer les cas correspondant à l'erreur 500.

Une erreur de type 500 Internal Server Error est retournée en cas d'erreur d'exécution au sein du serveur. Ce cas peut correspondre par exemple à une exception levée à l'exécution, ou une erreur lors de la connexion à la base de données.

#### c. Erreur 405

Programmer les cas correspondant à l'erreur 405.

Une erreur de type 405 Method Not Allowed est retournée lorsque la requête concerne une méthode HTTP qui n'est pas prévue sur la route demandée, la route étant cependant valide pour d'autres méthodes.

exemple : POST /commandes/ABCDE-12345 doit retourner une erreur 405