TD5 : création de ressources, token, validation des données

L'objectif du TD est de réaliser la route de création d'une commande en incluant :
- la gestion d'un token, généré lors de la création de la commande et utilisé pour autoriser l'exécution de toutes les requêtes suivantes concernant cette commande,
- la validation des données transmises lors de cette création

Puis, compléter les routes d'accès ou de modification d'une commande pour vérifier la présence et la validité tu token.

Ces différentes routes sont à programmer dans le service de prise de commandes en ligne.

1. créer des commandes
Créer la route pour la création de commandes. Dans un premier temps, on ne fait pas de contrôle sur les données reçues (on doit néanmoins filtrer les données reçues pour se prémunir contre l'injection, mais on suppose qu'elles sont présentes et complètes).
On ne traite pas non plus la liste des items commandés. Le montant de la commande est donc mis à 0.

La création d'une commande doit répondre aux caractéristiques suivantes :
- les données sont transmises en json,
- l'identifiant d'une commande est un uuid,
- la création d'une nouvelle commande conduit à la génération d'un token unique et cryptographique, retourné dans la réponse, et qui sera utilisé pour valider toutes les requêtes ultérieures concernant cette commande,

Ainsi, la requête de création de commande suivante :

```
POST /commandes HTTP/1.1
Content-Type: application/json

{
      "nom" : "jean mi",
      "mail": "jm@gmal.com",
      "livraison" : {
            "date": "7-12-2021",
            "heure": "12:30"
      }
}
```

conduit à la réponse suivante :

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /commandes/e3786989-e0d2-4cfb-a72f-455ca4a16beb

{
    "commande": {
        "nom": "jean mi",
        "mail": "jm@gmal.com",
        "date_livraison": "2021-12-07 12:30",
        "id": "e3786989-e0d2-4cfb-a72f-455ca4a16beb",
        "token":
           "b348142fc66bbbb4fae44a15ecd66460c1d12530a3d2eea404877364cac3e45a",
        "montant": 0
    }
}
```

2. contrôle et validation les requêtes concernant une commande

Compléter le contrôleur associé à la route pour accéder à une commande.
Le traitement doit procéder à la vérification du token : le système doit vérifier la présence et la valeur du token transmis lors de cette requête. On doit prévoir 2 modes de transport du token :

- transport dans l'url,
- transport dans un header applicatif.

Ainsi, la transmission du token permettant l'accès à une commande peut se faire ainsi :

```
GET /commandes/e3786989-e0d2-4cfb-a72f-455ca4a16beb?token=b348142fc66bbbb4fae
44a15ecd66460c1d12530a3d2eea404877364cac3e45a
```

ou :

```
GET /commandes/e3786989-e0d2-4cfb-a72f-455ca4a16beb
X-lbs-token: b348142fc66bbbb4fae44a15ecd66460c1d12530a3d2eea404877364cac3e45a
```

3. Traiter les items commandés
On met maintenant en place le traitement des items commandés. Ces items sont transmis sous la forme d'un tableau dans la requête de création de la commande. Chaque élément du tableau contient :

- le libelle de l'item,
- l'uri de l'item commandé,
- la quantité de l'item commandé
- le tarif de l'item

Ainsi, une requête de création de commande aura la forme suivante :

```
POST /commandes HTT/1.1
Content-Type: application/json

{
   "nom" : "jean mi",
   "mail": "jm@gmal.com",
   "livraison" : { "date": "7-12-2021","heure": "12:30" },
   "items" : [
      { "uri": "/sandwiches/6", "q": 3,"libelle": "panini","tarif": 6.00 },
      { "uri": "/sandwiches/1", "q": 2,"libelle": "bucheron","tarif": 6.00}
   ]
}
```

La requête crée la commande et son token, enregistre les items commandés dans la base de données et calcule le montant total de la commande.
Le montant calculé est indiqué dans la réponse :

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /commandes/e3786989-e0d2-4cfb-a72f-455ca4a16beb

{
    "commande": {
        "nom": "jean mi",
        "mail": "jm@gmal.com",
        "date_livraison": "2021-12-21",
        "id": "e3786989-e0d2-4cfb-a72f-455ca4a16beb",
        "token":
           "b348142fc66bbbb4fae44a15ecd66460c1d12530a3d2eea404877364cac3e45a",
        "montant": 36
    }
}
```

4. valider les données de création d'une commande

Mettre en place un mécanisme de validation des données reçues pour la création d'une commande.
Il faut valider :

- présence obligatoire du nom, mail et date/heure de livraison,
- format des noms-prénoms : chaîne de caractères alphabétiques,
- format de l'@mail
- format de la date, et validité de la date (date future uniquement).
- présence du tableau d'items

Vous utiliserez la librairie php respect/validation (https://github.com/Respect/Validation) et son intégration dans Slim grâce au middleware davidepastore/slim-validation (https://github.com/DavidePastore/Slim-Validation).