

6Programmation Web

PHPFormulairesn°1 : TD création de ressources, token, requêtes vers un service distant

L'objectif du TD est de compléter la 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,
- l'accès au catalogue, pour obtenir le nom et le tarif des items commandés.

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

1. créer des commandes

Compléter le contrôleur associé à la route pour la création de commandes. 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 :

- 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" : "7-12-2018",
    "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",
    "livraison": {
```

```

    "date": "7-12-2018",
    "heure": "12:30"
  },
  "id": "e3786989-e0d2-4cfb-a72f-455ca4a16beb",
  "token":
    "b348142fc66bbbb4fae44a15ecd66460c1d12530a3d2eea404877364cac3e45a",
  "montant": 0
}
}

```

2. valider 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=b348142fc66bbbb4fae44a15ecd66460c1d12530a3d2eea404877364cac3e45a

```

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 :

- l'uri de l'item commandé,
- la quantité de l'item commandé

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

```

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

{
  "nom": "jean mi",
  "mail": "jm@gmail.com",
  "livraison": {
    "date": "7-12-2018",
    "heure": "12:30"
  },
  "items": [
    { "uri": "/sandwichs/s19002", "q": 1 },
    { "uri": "/sandwichs/s19001", "q": 2 },
  ]
}

```

```

    { "uri": "/sandwichs/s19004", "q": 3 }
  ]
}

```

Le traitement de cette requête nécessite d'interroger le service catalogue associé pour connaître le nm et le tarif de chaque item. On doit donc exécuter une requête GET pour chaque uri présente dans les données de création.

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.

La liste des items et le montant calculé sont ajoutés 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",
    "livraison": {
      "date": "7-12-2018",
      "heure": "12:30"
    },
    "id": "e3786989-e0d2-4cfb-a72f-455ca4a16beb",
    "token":
      "b348142fc66bbbb4fae44a15ecd66460c1d12530a3d2eea404877364cac3e45a",
    "montant": 36,
    "items": [
      { "uri": "/sandwichs/s19002", "q": 1 },
      { "uri": "/sandwichs/s19001", "q": 2 },
      { "uri": "/sandwichs/s19004", "q": 3 }
    ]
  }
}

```

4. finaliser la route d'accès à une commande

La route d'accès à une commande doit finalement retourner une ressource contenant la liste imbriquées des items commandés :

```

{
  "type": "resource",
  "links": {
    "self": "/commands/841241ac-3f38-4afb-bb8b-36963a2aadda/",
    "items": "/commands/841241ac-3f38-4afb-bb8b-36963a2aadda/items"
  },
  "command": {
    "id": "841241ac-3f38-4afb-bb8b-36963a2aadda",
    "livraison": "2018-10-30 19:00:00",
    "nom": "michou",
    "mail": "mich@to.fr",
    "status": 1,
    "montant": "36.00",
    "items": [

```

```
{
  {
    "uri": "/sandwichs/s19001",
    "libelle": "le bucheron",
    "tarif": "6.00",
    "quantite": 1
  },
  {
    "uri": "/sandwichs/s19002",
    "libelle": "jambon-beurre",
    "tarif": "5.25",
    "quantite": 2
  },
  {
    "uri": "/sandwichs/s19004",
    "libelle": "fajitas poulet",
    "tarif": "6.50",
    "quantite": 3
  }
]
}
```