

**Sujet d'épreuves des Finales Nationales
de la 47^e Compétition des Métiers**

MÉTIER N°09

**SOLUTIONS LOGICIELLES
EN ENTREPRISE**

Module 2

Documentation API

Soumis par :

Paul NGO, Expert WorldSkills France

Briac DELAIGUE, Expert Adjoint WorldSkills France

TABLE DES MATIERES

INTRODUCTION	2
PRESENTATION DE L'API	2
UTILISATION DE L'API	3

INTRODUCTION

Ce document a pour objectif d'expliquer le fonctionnement de l'API du module2.
Pour vous aider dans la compréhension de l'API, le code source de celle-ci vous sera fourni.

PRESENTATION DE L'API

L'API à utiliser est au format REST. Cela correspond à un ensemble de principes et de conventions qui permettent à des applications informatiques de communiquer entre elles de manière simple et efficace.

Elle repose sur le protocole HTTP et utilise des méthodes standard, telles que GET, POST, PUT et DELETE, pour effectuer des opérations qui sont représentées sous forme d'URL.

Les APIs REST sont conçues pour être légères, évolutives et faciles à comprendre. Elles utilisent le format JSON pour structurer les données échangées.

UTILISATION DE L'API

2.1 Swagger UI

Pour vous aider à tester et à comprendre l'API sans regarder le code source vous retrouverez une interface web Swagger-UI disponible à l'url <http://ip-de-votre-server:8080/swagger-ui/index.html> une fois votre API démarré.

items-controller ^	
PUT	/api/v1/items/prices v
POST	/api/v1/items/prices v
DELETE	/api/v1/items/prices v
GET	/api/v1/items v
auth-controller ^	
POST	/api/v1/login v
dates-controller ^	
GET	/api/v1/dates v
booking-controller ^	
GET	/api/v1/bookings/item v

2.2 Collection Postman

De la même manière si vous préférez utiliser le logiciel Postman pour tester votre API vous retrouverez une collection importation dans le dossier /doc de l'API.

WS FR 2023 - Session 2	
POST	2.3 - Login
GET	2.4, 2.5 - List items
DEL	2.5 - Items prices
PUT	2.5 - Items prices
POST	2.5 - Items prices
GET	2.5 - List booking by item id
GET	2.5 - List dates

3.1 – Présentation des routes

Méthode à utiliser pour la connexion (2.3)

POST /api/v1/login

Format du corps du message :

```
{
  "username": "string",
  "password": "string"
}
```

Format du retour :

```
{
  "id": 0,
  "guid": "string",
  "username": "string"
}
```

Méthode à utiliser pour lister les locations (2.4)

GET /api/v1/items

```
[
  {
    "id": 0,
    "guid": "string",
    "name": "string",
    "itemPrices": [
      {
        "id": 0,
        "guid": "string",
        "date": "2023-09-09T09:58:04.062Z",
        "price": 0,
        "policyId": 0,
        "policyName": "string"
      }
    ]
  }
]
```

Méthode à utiliser pour supprimer un prix d'un hébergement (2.5)

DELETE /api/v1/items/prices

Format de réponse : 200 OK

Méthode à utiliser pour mettre à jour un prix d'un hébergement (2.5)

PUT /api/v1/items/prices

Format du corps du message :

```
{
  "id": 0,
  "date": "2023-09-09T10:00:07.846Z",
  "price": 0,
  "policyId": 0
}
```

Format du retour :

```
{
  "id": 0,
  "guid": "string",
  "date": "2023-09-09T10:00:07.849Z",
  "price": 0,
  "policyId": 0,
  "policyName": "string"
}
```

Méthode à utiliser pour ajouter un prix à un hébergement (2.5)

POST /api/v1/items/prices

Format du corps du message :

```
{
  "itemId": 0,
  "date": "2023-09-09T10:01:12.611Z",
  "price": 0,
  "policyId": 0
}
```

Format du retour :

```
{
  "id": 0,
  "guid": "string",
  "date": "2023-09-09T10:01:12.613Z",
  "price": 0,
  "policyId": 0,
  "policyName": "string"
}
```

Méthode permettant de lister les réservations et leurs détails d'un hébergement.

GET /api/v1/bookings/item

Format du retour :

```
[
  {
    "bookingId": 0,
    "bookingGuid": "string",
    "userId": 0,
    "bookingDate": "2023-09-09T10:02:08.566Z",
    "couponId": 0,
    "transactionId": 0,
    "amountPaid": 0,
    "price": {
      "id": 0,
      "guid": "string",
      "date": "2023-09-09T10:02:08.566Z",
      "price": 0,
      "policyId": 0,
      "policyName": "string"
    },
    "refund": true
  }
]
```

Méthode permettant de lister les dates de jours fériés.

GET /api/v1/dates

Format du retour :

```
[
  {
    "id": 0,
    "date": "2023-09-09T10:03:09.363Z",
    "holiday": true
  }
]
```