
PROJET JAVASCRIPT AS 2021

RENDU LE DIMANCHE 21 MARS À 23H59

VUE D'ENSEMBLE

Le but du projet est de créer une application Web (HTML/CSS + JavaScript) qui consommera l'API REST de commande de bières.

OBJECTIFS

Le projet contiendra au moins 3 pages HTML :

- une pour lister les bières disponibles à l'achat
- une pour lister notre historique de commandes
- une pour visualiser le détail d'une commande.

Toutes ces pages seront alimentées via des requêtes **Ajax**, pour récupérer les données depuis le serveur REST.

CARACTÉRISTIQUES

Les fonctionnalités du serveur REST sont accessibles en utilisant ces URL.

Fonctionnalité	Méthode HTTP	URL
Lister les bières	GET	http://localhost:9090/articles
Lister les commandes	GET	http://localhost:9090/commandes
Récupérer une commande	GET	http://localhost:9090/commandes/un_ID

Voici l'exemple d'une URL pour récupérer la commande portant l'ID 2 :

<http://localhost:9090/commandes/2>

UN MOT SUR L'API

Le **JAR** est à récupérer sur le Github habituel. Pour le lancer, il faut ouvrir un terminal (dans le dossier où se trouve le .jar) et exécuter la commande :

java -jar nom_du_jar_a_executer

```
$ java -jar magasin-api-rest.jar
```

Si ce message apparaît, c'est que le serveur tourne correctement.

```
Started MagasinApplication in 15.617 seconds (JVM running for 17.02)
```

Le serveur tourne en **localhost** sur le port **9090** et embarque sa propre base de données. Vous pouvez accéder à cette dernière via l'URL : **localhost:9090/h2**

Voici les identifiants de connection (attention, le mot de passe reste vide) :

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password:

Connect Test Connection

À savoir : la base de données est réinitialisée à chaque lancement du JAR.

BONUS

1. Vous pouvez ajouter toutes les améliorations que vous voulez.
2. Pour ceux qui aimeraient aller plus loin, il est possible d'ajouter des commandes en BDD, toujours via une requête **Ajax** vers le serveur REST.

Pour ce faire le serveur attend un JSON de ce type :

```
{
  "lignes": [
    {
      "article": {
        "id": 1,
        "designation": "kriek",
        "prix": 1.8
      },
      "quantite": 1
    },
    {
      "article": {
        "id": 2,
        "designation": "chouffe",
        "prix": 1.7
      },
      "quantite": 1
    }
  ]
}
```

à l'adresse : <http://localhost:9090/commandes>
et avec la méthode **POST**.