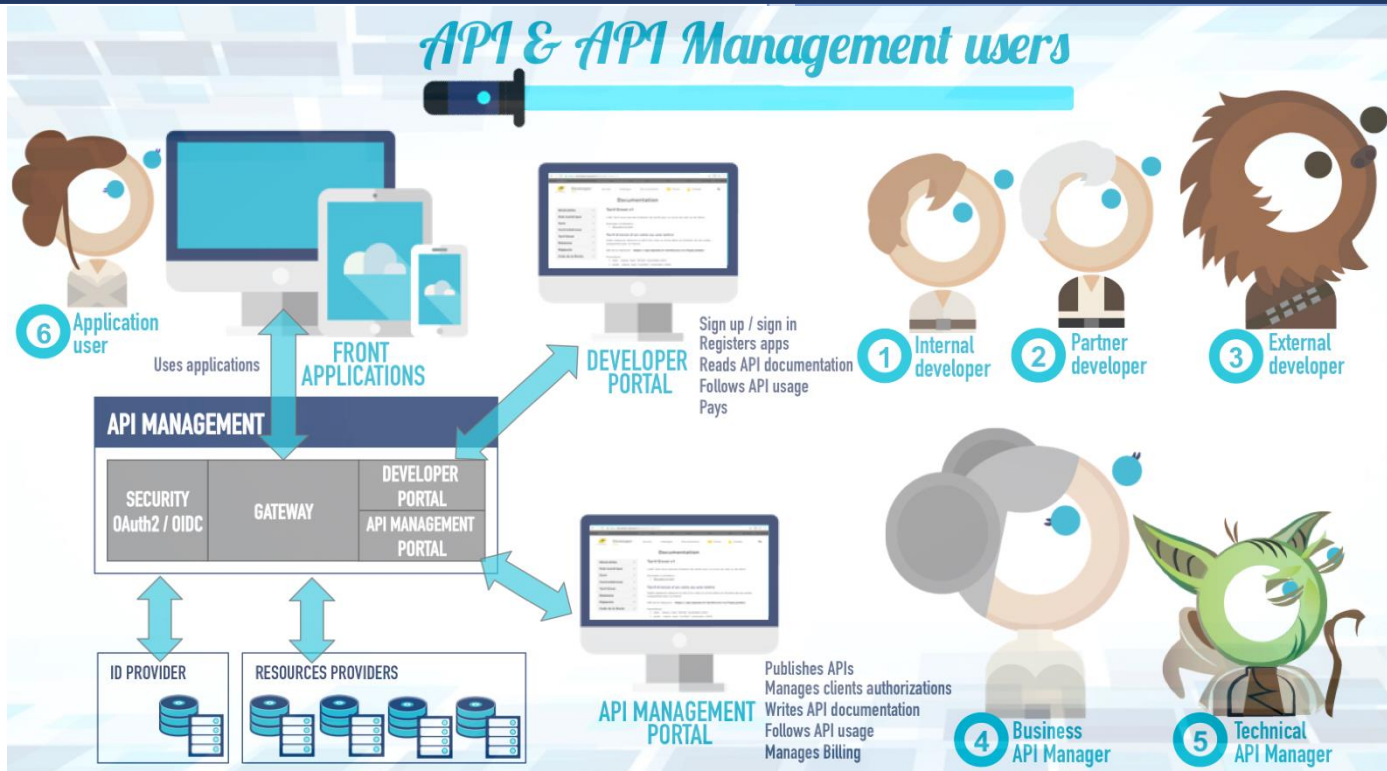


## Réalisation d'une API REST – Partie Base de Données



Laetitia LEBOHEC

Axel Laforge

Antonia Deville

## TABLE DES MATIERES

<b>Description de la Base de données .....</b>	<b>2</b>
Mise en place de la Base de Données : .....	2
<i>Choix de la Base de Données .....</i>	<i>2</i>
<i>SCRIPT pour chacune des tables .....</i>	<i>2</i>
Importation des données dans une table : .....	6
<i>Changements nécessaires.....</i>	<i>6</i>

## DESCRIPTION DE LA BASE DE DONNEES

### MISE EN PLACE DE LA BASE DE DONNEES :

#### CHOIX DE LA BASE DE DONNEES

Phpmyadmin : phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.



Nous souhaitons à la base utiliser mongoDB, or nous ne parvenions pas à le faire fonctionner. De plus, nous avions tous des petites notions avec phpmyadmin, de ce fait nous nous sommes très vite tournés vers cette application web pour les systèmes de gestion de base de données.

Plus facile à gérer que MongoDB, elle était aussi connue par tous les membres de notre groupe, ce qui nous donnait un peu plus confiance.

#### SCRIPT POUR CHACUNE DES TABLES

Dans un premier temps, il a fallu créer la base de données dans phpmyadmin se nommant « stid\_a17antoniad\_codesi » contenant 3 tables :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
BEERS	Afficher Structure Rechercher Insérer Vider Supprimer	5 901	InnoDB	latin1_swedish_ci	2,5 Mio	-
CATEGORIES	Afficher Structure Rechercher Insérer Vider Supprimer	11	InnoDB	latin1_swedish_ci	16 Kio	-
STYLES	Afficher Structure Rechercher Insérer Vider Supprimer	141	InnoDB	latin1_swedish_ci	16 Kio	-
3 tables	Somme	6 053	InnoDB	latin1_swedish_ci	2,5 Mio	0 o

- BEERS
- CATEGORIES
- STYLES

---

## CREATION DE LA STRUCTURE DE LA TABLE STYLES

```
DROP TABLE IF EXISTS `STYLES`;  
  
CREATE TABLE IF NOT EXISTS `STYLES` (  
  ID_STYLE INT NOT NULL,  
  ID_CATEGORIE INT(3),  
  STYLE_NAME VARCHAR(50),  
  PRIMARY KEY (`ID_STYLE`)  
)
```

### La table STYLES :

Contient :

- 141 individus (lignes)
- 3 variables (champs/colonnes)
- ID\_STYLE = Clé primaire

**ID\_STYLE** = Identifiant de l'individu et donc du style,

**ID\_CATEGORIE** = Identifiant de la catégorie pour chaque style,

**STYLE\_NAME** = Nom du style.

Suppression de la variable sur le fichier « **LAST\_MOD** » qui ne nous intéresse pas.

---

## CREATION DE LA STRUCTURE DE LA TABLE CATEGORIES

```
DROP TABLE IF EXISTS `CATEGORIES`;  
  
CREATE TABLE IF NOT EXISTS `CATEGORIES` (  
    ID_CATEGORIE INT(3) not null,  
    CATEGORIE_NAME VARCHAR(30),  
    PRIMARY KEY (`ID_CATEGORIE`)  
)
```

### La table CATEGORIES :

- 11 individus (lignes)
- 2 variables (champs/colonnes)

Suppression de la variable « **LAST\_MOD** » qui ne nous intéresse pas.

**ID\_CATEGORIE** = Identifiant de la catégorie,

**CATEGORIE\_NAME** = Nom de la catégorie.

---

## CREATION DE LA STRUCTURE DE LA TABLE BEERS :

```
DROP TABLE IF EXISTS `BEERS` ;

CREATE TABLE IF NOT EXISTS `BEERS` (
  NAME VARCHAR(100),
  ID_BEER INT(11) NOT NULL,
  ID_BREWERY INT(5),
  ID_CATEGORIE INT(3),
  ID_STYLE DOUBLE,
  ALCOHOL_BY_VOLUME FLOAT(20,6),
  DESCRIPTION VARCHAR(4000),
  ADD_USER VARCHAR(20),
  LAST_MOD VARCHAR(30),
  STYLE_NAME VARCHAR(50),
  CATEGORIE_NAME VARCHAR(30),
  BREWERY VARCHAR(100),
  ADDRESS VARCHAR(100),
  CITY VARCHAR(50),
  STATE VARCHAR(50),
  COUNTRY VARCHAR(50),
  COORDINATES VARCHAR(35),
  WEBSITE VARCHAR(100),

  PRIMARY KEY (`ID_BEER`)
)
```

### La table BEERS :

- 5901 individus (lignes)
- 18 variables (champs/colonnes)

Suppression des variables « ID\_BREWERY », « ADD\_USER », « LAST\_MOD », « CATEGORIE\_NAME », « ADDRESS », « CITY », « STATE », « COUNTRY », « COORDINATES », « WEBSITE », « INTERNATIONAL\_BITTERNESS », « STANDARS\_REFERENCE\_METHOD », « UNIVERSAL\_PRODUCT\_CODE », et « FILE... », qui ne nous intéresse pas

**NAME** = Nom de la bière.

**ID\_BEER** = Identifiant de la bière.

**ID\_CATEGORIE** = Identifiant de la catégorie pour chaque style.

**ID\_STYLE** = Identifiant de l'individu et donc du style.

**ALCOHOL\_BY\_VOLUME** = Degrés d'alcool.

**DESCRIPTION** = Description de la bière.

**BREWERY** = Brasserie.

**COUNTRY** = Pays.

- Le fichier CSV contenait **5973** lignes, sans compter la ligne de champs. La suppression de **72 lignes** a été forcée, en effet ces individus contenaient de mauvaises données ; id impossible, doublons, lignes n'ayant aucune donnée.

## IMPORTATION DES DONNEES DANS UNE TABLE :

### Fichier à importer :

Le fichier peut être comprimé (gzip, bzip2, zip) ou non.  
Le nom du fichier comprimé doit se terminer par `[format].[compression]`. Exemple: `.sql.zip`

Parcourir :  `open-beer-...types.csv` (Taille maximum: 2 048Kio)

Vous pouvez également faire glisser et déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier :

### Importation partielle :

☒ Permettre l'interruption de l'importation si la limite de temps configurée dans PHP est sur le point d'être atteinte. (Ceci pourrait aider à importer des fichiers volumineux, au détriment du respect des transactions.)

Ignorer ce nombre de requêtes (pour SQL), à partir du début :

### Autres options :

☒ Activer la vérification des clés étrangères

### Format :

Note : si le fichier contient plusieurs tables, elles seront combinées.

### Options spécifiques au format :

☐ Mettre à jour les données lorsque des clés dupliquées sont trouvées lors de l'importation (Ajouter ON DUPLICATE KEY UPDATE)

Colonnes séparées par :

Colonnes entourées par :

Caractère d'échappement :

Lignes terminées par :

Nom des colonnes :

Une fois que les structures entre les tables ont été faites il faut importer les données contenues dans les fichiers CSV. Pour cela sur phpmyadmin nous importons en prenant soin de vérifier que la première ligne ne soit pas récupérée comme des données, et que le séparateur est bien en adéquation avec le fichier utilisé.

## CHANGEMENTS NECESSAIRES

Nous avons ensuite changé d'hébergeur afin d'y avoir tous accès plus facilement. Nous avons donc mis la base de données sur alwaysdata.



La base de données est toujours sur phpmyadmin mais cette fois sur un compte public où nous pouvons avoir accès. La base de données se nomme shiisoo\_beers.

Elle est disponible sur cette adresse :

[https://phpmyadmin.alwaysdata.com/phpmyadmin/db\\_structure.php?server=1&db=shiisoo\\_beers&token=495414c821de2b3e15ee1b4771ef2705](https://phpmyadmin.alwaysdata.com/phpmyadmin/db_structure.php?server=1&db=shiisoo_beers&token=495414c821de2b3e15ee1b4771ef2705)