

# README

---

## Documentation d'utilisation du test technique - chez Nestor

- Etiam sit amet est
  - Aenean iaculis laoreet arcu
  - Curabitur vulputate viverra pede
  - Nulla rutrum commodo ligula
  - Sed tellus suscipit in aliquam
    - Nunc ut lectus
    - Donec quis nunc

## **1. Rappel du sujet**

Le but de l'exercice est de créer un module de réservation de chambre pour le back office de chez Nestor : Nous gérons u parc d'appartements. Nos clients, pour réserver une chambre dans l'un de nos appartements, doivent remplir un certain nombre d'informations afin de valider leur réservation.

Il est question d'exposer une API qui permettra à un client de réserver une chambre, ainsi que de récupérer toutes les données du client, de la chambre, de l'appartement et tout ce qui semblera utile à l'utilisation des données par interface utilisateur. Pour chaque modèle, il faut qu'on puisse créer, lire, modifier et supprimer (CRUD) des données.

Il est important de respecter les règles suivantes :

- Un appartement contient au moins une chambre
- Un client ne peut pas réserver plusieurs chambres en même temps.
- L'adresse email d'un client est unique
- Lorsqu'une chambre a été réservé par un client, elle ne peut plus être réservée.
- Une chambre peut être réservé si et seulement si le client a rempli toutes ses informations.

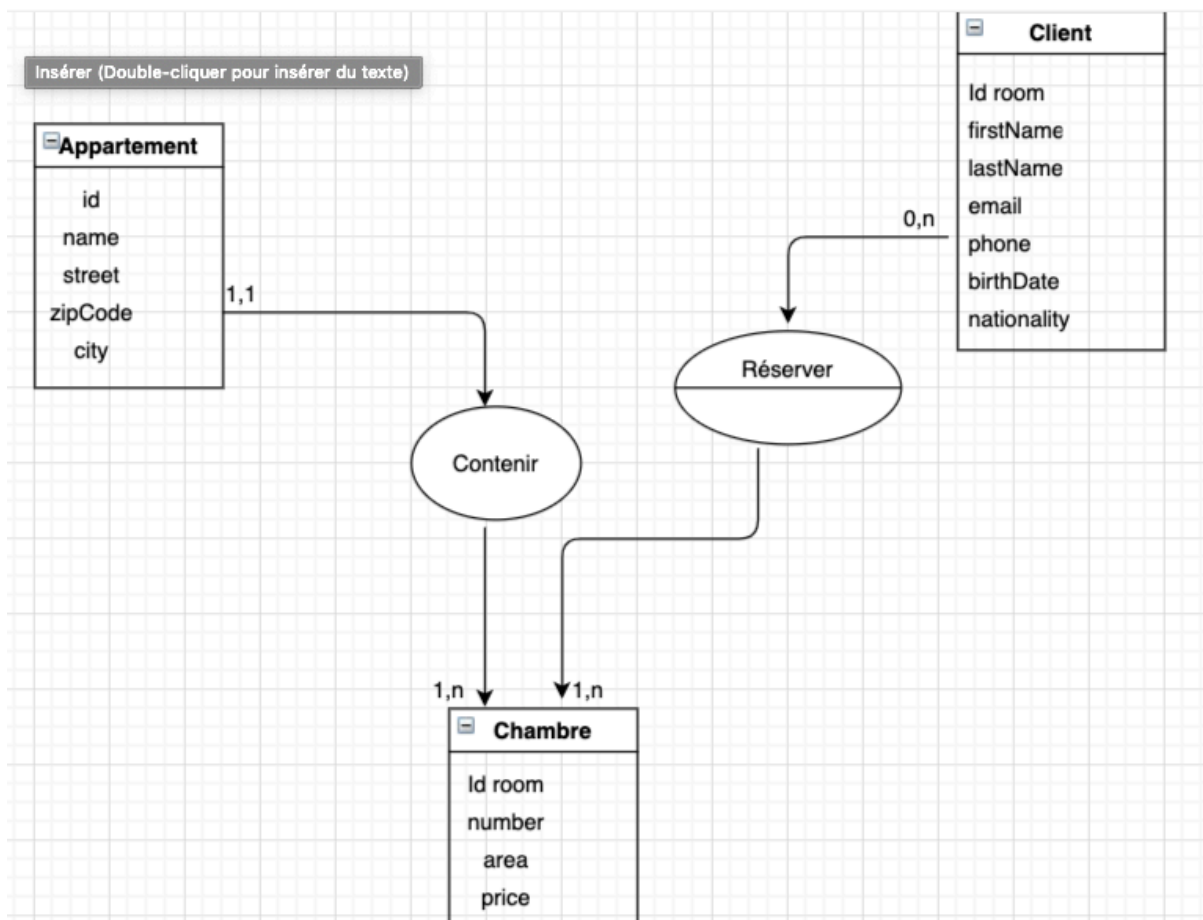
## **2. Base de données**

### **2.1. Conditions de conception**

Il est important de respecter les règles suivantes :

- Un appartement contient au moins une chambre
- Un client ne peut pas réserver plusieurs chambres en même temps.
- L'adresse email d'un client est unique
- Lorsqu'une chambre a été réservé par un client, elle ne peut plus être réservée.
- Une chambre peut être réservé si et seulement si le client a rempli toutes ses informations.

## 2.1. Modèle conceptuel de données



On peut comprendre du MCD :

- Un appartement contient une ou plusieurs chambres
- Zéro ou plusieurs clients peuvent réserver une ou plusieurs chambres

## 2.3. Importer le dump de la BD

En informatique, le terme **dump** désigne généralement une copie brute (sans transformation) de données d'un périphérique à un autre ; par exemple pour effectuer une sauvegarde de données, les afficher sur un terminal, ou faire persister les données d'une mémoire vive. Il se distingue de la sauvegarde, qui désigne des copies structurées.

On peut importer le dump d'une base de données sql vers MySQL de plusieurs façons, cependant, ce tutoriel n'indique que celles liées à l'interface phpmyadmin et au CMD.

Avant de d'importer une base de données, il est question de mettre un environnement en place.

## Mise en place de l'environnement de travail

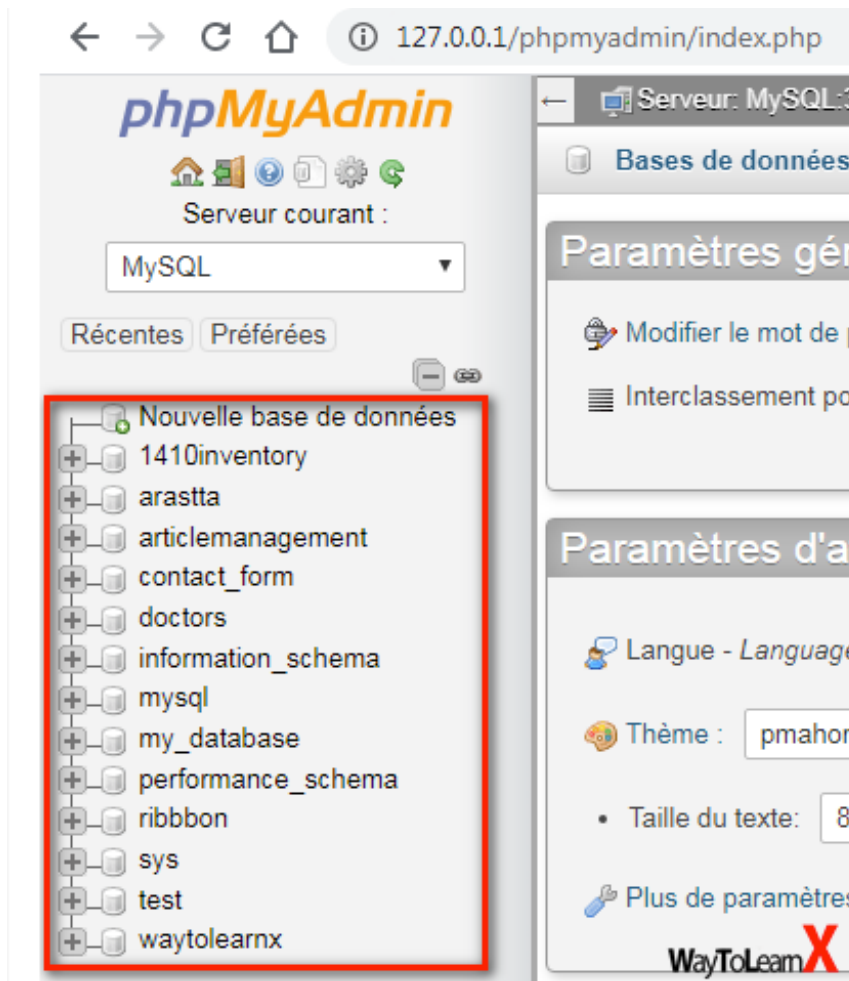
- Télécharger Xampp de votre système d'exploitation (windows, Mac OS X, etc ...)
- Installer correctement Xampp
- Démarrer Xampp et ses différents services (Apache et MySQL)
- Activer la connection ssh ou pas dépendant des réseaux qui voudront accéder aux services Xampp démarré
- Ouvrir l'interface Xampp dans un navigateur en localhost ou via l'adresse IP d'accès indiqué par Xampp
- Cliquez sur phpmyadmin et s'authentifier
- Il est nécessaire que login = 'root' et password = ''

## Tutoriel d'importation du fichier .sql

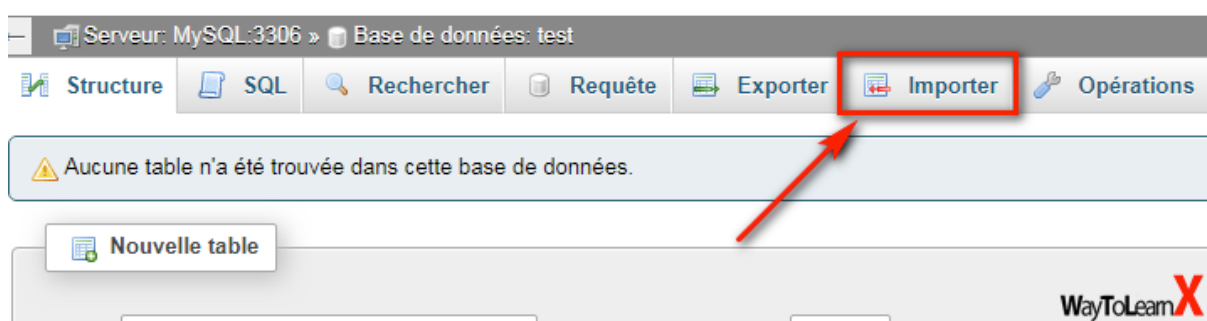
1. Assurez-vous que la base de données dont vous avez besoin, a déjà été créée. Si ce n'est pas le cas, veuillez d'abord créer la base de données :



2. Dans phpMyAdmin, sélectionnez votre base de données dans le menu à gauche.



3. Cliquez sur « Importer » dans le menu en haut.



4. Cliquez sur le bouton « Choisir un fichier » à côté de « Parcourir les fichiers : ».



## Importation dans la base de données « test »

### Fichier à importer :

Le fichier peut être compressé (gzip, bzip2, zip) ou non.

Le nom du fichier compressé doit se terminer par **[format].[compression]**. Exemple: **.sql.zip**

Parcourir les fichiers : Choisir un fichier Aucun fichier choisi (Taille maximale : 128 Mio)

Il est également possible de glisser-déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier : utf-8 ▼

5. Naviguez jusqu'à vous trouverez le fichier SQL et cliquez sur « Ouvrir ». Si c'est un fichier zippé, veuillez le décompresser d'abord.

6. Cliquez sur le bouton « Exécuter » en bas. Attendez pendant l'importation de votre base de données. En fonction de la taille, cela peut prendre quelques minutes.

Vous devriez recevoir un message comme celui-ci :

L'importation a réussi, X requêtes exécutées.

Si vous recevez plutôt une erreur, veuillez essayer la méthode sur la ligne de commande ci-dessous.

### Importation via la ligne de commande

Pour importer un fichier existant dans MySQL, vous devrez créer une nouvelle base de données. C'est là que le contenu du fichier SQL sera importé.

Après que vous avez créé la base de données.

A partir de la ligne de commande, vous pouvez importer le fichier SQL avec la commande suivante :

```
mysql -u username -p new_db < file.sql
```

- **username** est le nom d'utilisateur avec lequel vous pouvez vous connecter à la base de données
- **new\_db** est le nom de la base de données fraîchement créée

- **file.sql** est le fichier SQL à importer, situé dans le répertoire courant

Si la commande est exécutée avec succès, elle ne produira aucune sortie.

Si des erreurs se produisent pendant le processus, mysql les affichera sur le terminal. Vous pouvez vérifier que la base de données a été importée en vous connectant à nouveau au shell MySQL et en inspectant les données. Cela peut être fait en sélectionnant la nouvelle base de données avec **USE new\_db** et en utilisant ensuite **SHOW TABLES;**.

### 3. API

Technologies utilisées :

- Express JS
- IDE: Visual Studio Code

Utiliser le framework Express JS consiste à définir un environnement favorable au déploiement d'une API Express JS.

Environnement de travail Express JS:

- Télécharger et Installer Node et NPM dépendant du système d'exploitation utilisé
- Tester l'installation nodejs et npm
- **> node -v**
- **> npm -v**
- Créer un dossier **node-express** et se placer dedans
- Utiliser NPM pour installer Express et ajouter les dépendances nécessaires à la racine du dossier projet créé
- Utiliser la commande **npm init y** pour créer un fichier package.json pour l'API
- Ajouter les dépendances : body-parser, express et path.  
**npm install body-parser express path**
- Modifier le fichier package.json pour démarrer l'application en utilisant node :  
« scripts »: {  
 « start »: nodemon,  
 « test »: "echo \"Error: no test specified\" && exit 1"  
}

- Démarrer npm en utilisant la commande **npm start** et l'écouter via un navigateur sur le port localhost précisé.
- Puis arrêter si tout fonctionne et pour tester notre projet, créons un fichier app.js et démarrons le à l'aide de node, **node app.js**

### Structure de l'API

- Importation des bibliothèques et librairies nécessaires pour travailler
- Création de l'application express
- Création des routes
- Envoi d'un message de confirmation fonctionnel
- Mise en route du serveur sur un port donné

### Les routes créées

- **/apartment** : Récupère tous les appartements et leurs informations
- **/room** : Récupère toutes les chambres et leurs caractéristiques
- **/apartment/:id** : Récupère les informations de la chambre dont l'identifiant est indiqué
- **/room/:id\_room** : Récupère les informations de la chambre dont l'identifiant est indiqué
- **/client** : Récupère tous les clients enregistrés ou ajoute les informations d'un client
- **/client/:id** : Récupère ou supprime ou modifie les informations du client dont l'identifiant est indiqué
- **/reservation/:email/:id\_room/:motif** : Enregistre une réservation
- **/reservation/:id\_room** : Supprime ou annule une réservation

## 4. Utiliser Postman pour tester l'API

Utilisation de Postman:

- Télécharger Postman dépendant du SE utilisé
- Installer Postman
- Créer un compte si vous n'en avez pas ou vous connecter avec un compte Google
- Accéder à l'interface de test des API

### Les requêtes HTTP

C'est la fonctionnalité la plus triviale à laquelle répond cet outil, pouvoir faire des requêtes HTTP simplement. L'interface graphique rend l'exercice aussi facile que surfer sur un navigateur. Il suffit de renseigner l'**URL (la route complète)** et la méthode **HTTP**



Les méthodes HTTP diffèrent suivant l'opération souhaité et l'API sur laquelle vous faites votre opération. Voici les méthodes HTTP utilisées pour l'API Express JS de réservation de chambres et leurs fonctions:

- **GET** : les requêtes GET sont celles effectuées par un navigateur lorsque vous entrez une URL dans laquelle la barre de navigation. Elles ont but d'aller récupérer les données dans la BD.
- **POST**: les requêtes POST ont pour but d'envoyer de l'information et contenue dans le body de la requête, vers le serveur.
- **PUT**: les requêtes PUT vont écraser une ressource avec de nouvelle donnée, là aussi présente dans le body de la requête. Elle est utilisée pour mettre à jour les données à condition qu'on soit capable de fournir la ressource mise à jour dans son intégrité.
- **DELETE**: comme son nom l'indique, la requête DELETE sert à effacer une ressource.