

# Manuel d'exploitation

Projet de gestion d'une chaîne d'hôtels

Jérémie HENRI | Quentin PLA

## Aperçu du système

Notre système est composé de trois éléments majeurs : la base de données, l'application web pour les clients et enfin l'application de gestion JAVA pour les salariés de la chaîne d'hôtels.

### Base de données

La base de données a été conçue avec le système MariaDB (version 10.4.12) avec un encodage de type UTF-8. Elle est gérée grâce à l'interface phpMyAdmin permettant de la mettre à jour rapidement et facilement.

### Application Web

Réalisée grâce au logiciel node.js (version 13.8.0) en utilisant l'infrastructure d'application web Express côté serveur et la bibliothèque React côté interface. Elle permet aux clients d'avoir une vue d'ensemble sur la liste des chambres disponibles de la chaîne d'hôtels, les réservations qu'ils ont effectuées, la liste de leurs factures et une interface de gestion de leur profil afin d'avoir la possibilité de modifier des informations erronées. Afin d'optimiser le rendu, nous avons utilisé la bibliothèque Socket.IO permettant d'effectuer une communication bidirectionnelle en temps réel entre les clients web et le serveur, permettant ainsi d'éviter le rafraichissement de la page et donc une navigation plus agréable pour le client.

### Application JAVA

Nous avons décidé d'utiliser le langage JAVA (sdk 11.0.5) afin de la réaliser car c'est un langage orienté objet très largement utilisé pour la réalisation d'application. Elle est destinée aux salariés de la chaîne d'hôtels afin de gérer les réservations effectuées, les occupations en cours, les factures disponibles et avoir une vue globale sur l'ensemble des hôtels et services mis en place. Afin de générer l'application nous avons utilisé le moteur Gradle (version 6.4.1) permettant de réaliser des tests unitaires pour assurer un fonctionnement sans bug. L'application a été réalisée grâce à l'environnement de développement JetBrains IntelliJ Idea très populaire pour sa facilité d'utilisation et les fonctionnalités qu'il propose.

## Mise en service du système

### Conditions préalables à la mise en service

Afin de mettre en service le système, il est nécessaire de disposer d'un hébergeur web et de base de données. Pour cela nous avons utilisé les hébergeurs gratuits alwaysdata, afin d'héberger la base de données et heroku afin d'héberger l'application web.

Pour débiter la mise en service, il faudra créer un compte sur les deux plateformes gratuitement à partir de ces liens :

- [Inscription Alwaysdata](#)
- [Inscription Heroku](#)

## Déroulement de la mise en service

Une fois la création des comptes terminées, connectez-vous avec vos identifiants.

### Mise en service de la base de données

L'initialisation de la base de données s'effectue sur la plateforme Alwaysdata. Veuillez suivre les étapes suivantes après s'être connecté :

- Cliquer sur l'onglet « **MySQL** » situé dans la liste nommée « **Bases de données** ».
- Cliquer sur le bouton « **+ Ajouter une base de données** ».
- Saisir le nom que vous souhaitez et cocher le paramètre « **tous les droits** » en-dessous de « **Permissions** » puis terminer en cliquant sur « **Valider** ».
- Afin d'accéder à l'interface phpMyAdmin, cliquer sur le lien « **Gestionnaire MySQL : phpMyAdmin** ».
- Pour trouver votre identifiant, cliquer sur l'onglet en rose « **UTILISATEURS** » et copiez la suite de chiffres située en-dessous de « Nom ». Concernant le mot de passe, saisir le même que celui utilisé pour créer le compte.
- Une fois sur l'interface phpMyAdmin, cliquer sur votre base de données à gauche.
- Cliquer ensuite sur l'onglet « **Import** » situé dans les onglets horizontaux.
- Télécharger le fichier [Database.sql](#) qui servira à initialiser la base de données.
- Cliquer sur « **Parcourir** » et sélectionner le fichier téléchargé.
- Cliquer sur le bouton « **Go** » situé en bas de page pour initialiser la base de données.

### Mise en service de l'application web

L'initialisation du site web se fait sur la plateforme Heroku. Veuillez suivre les étapes suivantes après s'être connecté :

- Se rendre sur le [panneau de gestion des applications](#).
- Cliquer sur le bouton « **New** » situé à droite.
- Cliquer sur l'option « **Create new app** ».
- Saisir le nom de l'application, choisir Europe comme région et finaliser en cliquant sur le bouton « **Create app** ».
- Cliquer sur l'application tout juste créée puis se diriger dans l'onglet « **Deploy** ».
- Sélectionner la méthode de déploiement « **Heroku Git** ».
- Cloner le projet dans le dossier que vous souhaitez avec la commande : « **git clone -b application-web https://github.com/quentin-pla/Gestion-Bibi-Hotels.git** ».
- Modifier le fichier « **react-ui/src/database/dbconnection.js** » pour modifier les champs « **base\_de\_donnees** », « **identifiant** », « **mot\_de\_passe** » et « **host\_mysql** » avec ceux de votre base de données.
- Ouvrir un terminal au dossier de l'application clonée.
- Taper les commandes dans le terminal dans l'ordre comme indiqué sur le site, en sautant la commande « **cd my-project** » (déjà placé dans le dossier de l'application).

### Mise en service de l'application JAVA

Afin d'exécuter l'application JAVA, il est nécessaire d'installer des compléments, voici les liens permettant de les télécharger :

- [SDK JAVA 11.0.5](#) (se rendre dans la section Java SE Development Kit 11.0.5 et choisir le fichier en fonction de votre système d'exploitation).
- Pour modifier la base de données liée à l'application, vous devrez modifier les champs « **host\_mysql** », « **base\_de\_donnees** », « **identifiant** », « **mot\_de\_passe** » dans le fichier :  
« **src/main/java/org/gestion\_bibi\_hotels/database/DatabaseConnection.java** ».
- Pour compiler l'exécutable, taper la commande dans un terminal au dossier :  
« **./gradlew jlink** », il sera situé ici : « **build/image/bin/nom\_du\_projet** ».

## Assurance de la qualité après la mise en service

Après la mise en service de l'application web, elle sera accessible tout le temps du moment que le client dispose d'une connexion internet.

Concernant la base de données elle sera également fonctionnelle en continu.

L'application JAVA nécessite une connexion internet car elle se connecte à la base de données. Sachant que les services proposés pour l'hébergement sont gratuits, des limites peuvent être appliquées, ainsi des ralentissements peuvent se faire ressentir du côté clientèle, si un grand nombre de connexions sont établies simultanément.

## Interruption ou arrêt du système

### Application web

Le service Heroku propose une option de maintenance permettant ainsi de rendre l'application web hors-ligne pendant une période voulue. Pour activer cette option il suffit de se rendre dans l'onglet « **Settings** » sur l'interface de l'application et d'activer l'option de maintenance. Ainsi les clients ne pourront plus se rendre sur le site web.

### Base de données

Elle ne peut être interrompue ou arrêtée.

### Application JAVA

Afin d'interrompre l'application JAVA, il suffit de fermer le programme.

## Gestion des modifications

### Application web

Si vous souhaitez mettre à jour l'application web, après avoir modifier le code voulu, pour mettre à jour le site en ligne, suivre la procédure suivante :

- Ouvrir un terminal au dossier de l'application
- Taper la commande : « **git add .** » qui va permettre de sélectionner tous les fichiers mis à jour.
- Taper ensuite : « **git commit -m «votre message»** » pour historiser les modifications.

- Pour envoyer les données vers Heroku : « **git push heroku master** ».

## Base de données

Pour mettre à jour la base de données, il suffit d'accéder à l'interface phpMyAdmin (voir la procédure ci-dessus) et d'insérer, modifier ou supprimer les tuples souhaités.

## Application JAVA

L'utilisation du moteur Gradle est nécessaire pour compiler l'application. La procédure d'installation est disponible [ici](#). Après avoir effectué les modifications souhaitées, suivre la procédure suivante afin de générer un exécutable :

- Ouvrir un terminal au dossier de l'application.
- Vérifier que la compilation s'effectue correctement en tapant la commande :  
« **./gradlew** ».
- Vérifier que l'application s'exécute en tapant : « **./gradlew run** ».
- Générer le fichier exécutable avec la commande : « **./gradlew jlink** ».
- Après que la commande s'est terminée correctement, se diriger dans le dossier  
« **build/image/bin** » pour trouver l'exécutable.

## Dispositions de sécurité

En ce qui concerne l'application web, le logiciel Node.js est déjà robuste ainsi nous avons juste ajouté la dépendance « **bcrypt** » permettant de crypter le mot de passe des clients lors de leur inscription et ainsi renforcer la sécurité.

Pour la base de données, de nombreux triggers ont été réalisés afin d'éviter des erreurs d'insertion ou de mise à jour de données.