***Documentation back Strapi***

**1)Accéder à l'interface administrateur**

**Installation**

1. Commencez par cloner le dépot si ce n'est pas déja fait et accédez ensuite à son dossier.

git clone https://github.com/djalexkidd/vacances.git

cd vacances/

1. Accédez au dossier contenant Strapi.

cd strapi/

1. Téléchargez les dépendances nécéssaires.

npm i

1. Strapi est démarré automatiquement après l'installation, utilisez cette commande pour rédémarrer Strapi.

npm run develop

Pour accéder à Strapi, entrez l'adresse <http://localhost:1337/admin/> dans un navigateur internet.

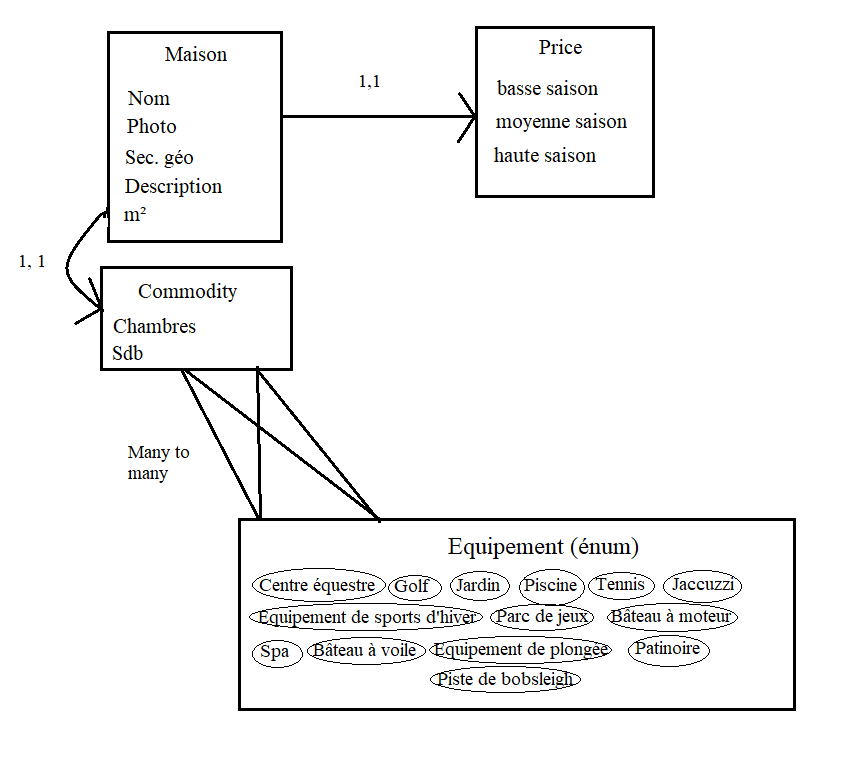
**Identifiants par défaut**

E-mail: [gustave.m@gmail.com](mailto:gustave.m@gmail.com)

Mot de passe: Password01

Changez ensuite le mot de passe pour des raisons de sécurité évidentes.

**2)Structure de la base de données**

****

La base de données est composé de 4 tables : Maison, Price, Commodity et Equipement.

La table Maison est la table principale. Elle regroupe les informations principales présentant les maisons dans la base de données : son nom, sa photo, son secteur géographique, sa description et sa surface en m².

La table maison possède 2 relations. Elle est liée à les tables Price et Commodity. La table Price stocke les différents prix de la maison en question suivant les saisons touristiques.

La table commodity stocke le nombre de chambres et de salles de bain de la maison.

Initialement, les équipements était une liste des équipements dans le JSON qui me servais de modèle. J'ai dans un premier temps donné le format "Rich Text" à la propriété Equipement dans le but de donner une belle présentation aux différents agréments de la maison.

Mais cette présentation textuelle ne permet pas de réaliser des recherches suivant les différents équipements. Mon enseignant m'a appris une manière plus appropriée de gérer les listes avec Strapi.

La table Equipement stocke un énum du même nom qui stocke les différents équipements possibles à l'intérieur des maisons. Mais le souci d'un énum, c'est qu'il impose le choix d'un seul élément. Pour remédier à cette situation, on relie les tables Commodity et Equipement par une relation "Many to many" (car une liste de commodités peut avoir plusieurs équipements et qu'un même équipement peut apparaître dans la liste de commodités de plusieurs maisons).

**3)Utilisation de l'interface graphique de Strapi**

Pour créer une nouvelle table à votre base de données (ou modifier une table existaente), cliquez sur l'onglet "**Content-Type Builder**". Vous pourrez ainsi gérer la structure de vos tables.

Pour créer une nouvelle table, cliquez sur "Create new collection type". Vous verrez une petite interface dans laquelle vous devrez choisir un nom à votre table (par exemple "Maison" ou "price". A droite du nom que vous aurez choisi, Strapi vous affichera le singulier et le pluriel de ce mot qu'il aura générés. Vous pouvez les modifier si nécessaires. Mais retenez-les, ils seront essentiels lors de la création de endpoints dans les URL et l'utilisation des services d'API.

Vous verrez ensuite une table vide. Pour lui donner des attributs, cliquez sur "Add another field to this collection type".

En cliquant sur ce bouton, vous devrez choisir le type ou le format de l'attribut à créer (texte, number, booléan… ainsi que rich text et énum comme je l'ai évoqué dans le chapitre précédent avec la table Equipement).

Chaque type de données possède des settings de base et des settings avancés. Parmis les settings avancés, on peut trouver "paramètre obligatoire" ou "longueur minimale ou maximale".

Il faut créer une table suivant les caractéristiques logique de ce qu'elle décrit. Par exemple une maison possède un nom (texte), une photo (l'url de la photo est une chaîne de caractères, donc un texte) et une surface en m² (un nombre).

Comme dans les bases de données en SQL, vous aurez besoin de créer des relations entre les tables. Il vous voudra évidemment au minimum 2 tables existentes. Ensuite ajoutez dans la table mère une relation avec la table enfant. Pour cela, cliquez sur l'objet "relation" dans "Add another field to this collection type".

Vous devrez choisir les cardinalités de la relation (unique dans les 2 sens, multiple à unique, many to many). Regardez le MCD de votre base de données si besoin.

Une bonne habitude à prendre, quand on réalise une modification que l'on souhaite pérenne, cliquez sur "Save", puis "Publish" si vous le voyez".

Maintenant, une fois la structure des tables créées, il faut remplir la base de données en créant des éléments. Cliquez sur **"Content Manager".**

Vous verrez le nom de tables déjà présentes (ou que vous aurez créées). En cliquant sur le nom d'une table (par exemple Maison), vous remplirez les informations d'une table maison de la base de données.

Imaginons que vous voulez créer la maison "Villa des Pins" dans la base de de données. Il vous faudra d'abord remplir la table maison pour ajouter les 1ères informations à son sujet. Puis il faudra créer les autre tables qui lui sont liées (car cette maison possède des commodités, des équipements ainsi que des prix variant suivant les saisons touristiques). Reliez les tables par des relations cohérentes.

(Note : quand vous créerez des éléments à votre base de données, vous devrez relier les tables mères à leurs tables filles. Pour cela, allez dans l'onglet "relation" (tout à droite en bas en dessous de l'onglet information).

Dans **Media Library**, vous pourrez stocker les différents médias (photos, vidéos…) devant être stockées dans votre base de données sous forme d'URL.

Dans **Settings**, vous pourrez gérer les utilisateurs et les rôles.

Dans les onglets de "Adminsitration Panel", vous pourrez gérer les rôles et les utilisateurs.

Dans "Roles" vous pourrez voir les 3 rôles par défaut : author, editor et super-admin. Vous pourrez les modifier en cliquant sur "edit". Vous pourrez gérer les différents privilèges des différents rôles.

En revanche si vous voulez ajouter de nouveau rôles, il vous faudra opter pour la version Enterprise de Strapi.

Dans "Users", vous pourrez voir les utilisateurs de votre base de données. Vous pouvez créer de nouveaux utilisateurs en cluant sur "Invite new user".

Vous devrez donc donner son nom, son email et son rôle. Si l'adresse mail est valide, il recevra cette invitation dans sa boite aux lettres.

Dans "User & Permission Plugin", vous pouvez préciser certains détails sur les rôles. En plus des 3 rôles précédemment cités, les rôles "Public" et "Authentified" existent.

Le rôle public représente ce que peuvent faire tous les utilisateurs. Dans notre cas, nous voulons que les utilisateurs doivent s'identifier avant de pouvoir louer les maisons proposées. Dans Public on leur interdit tout sauf le droit d'envoyer un mail aux admins (pour réaliser une demande de parrainage).

Les utilisateurs "authentified" doivent s'identifier (mettre leur identifiant et leur mot de passe pour utiliser le site). Cette authentification leur permet de profiter des offres de location de maisons de luxe.

**4)Pusher les modifications avec Strapi**

Quand on travaille en équipe, on peut être plusieurs à travailler sur une même page ou un même élément. Pour éviter les conflits, il est préférable de réaliser un "git pull".

Voici la procédure à suivre pour pusher vos modifications.

1. Quitter le serveur Strapi
2. Lancer le "Git bash here"
3. Faire un "git pull" (on peut recevoir les dernières modifications des autres), puis "git commit", puis "git push"
4. Si jamais Vi (l'éditeur de texte par défaut de Strapi) vous cause quelques soucis, tapez :wq (pour écrire et quitter)

**5)Utilisations de PostMan avec Strapi**

Il est possible d'utiliser PostMan avec l'interface graphique. Mais il est également possible d'utiliser Postman (surtout si on doit travailler en équipe en télétravail et que vous n'avez pas accès à l'interface administrateur).

Sur Postman, avec le rôle "authentified", il faut préciser écrire la route http://localhost:1337/api/auth/local

puis écrire dans le JSON du req.body son identifiant et son mot de passe.

Postman vous proposera un jeton jwt valable jusqu'à la prochaine connexion.

Le jeton valable change donc à chaque connexion. Son début est identique mais sa fin change. Tout cela est fait pour compliquer la tâche aux pirates.

Quand on doit utiliser un jeton jwt avec Postman, il faut aller dans "Headers", créer une clé "Authorization" avec comme valeur "Bearer <contenu du jeton>".

On peut également réaliser des recherches en utilisant des services d'API via le navigateur ou Postman.

Par exemple, on peut taper [<port>/api/maisons?populate=\*](http://localhost:1337/api/maisons?populate=*) pour chercher le contenu de toutes les tables "maison".

Dans la 1ère ligne, vous pouvez modifier "maisons" par "prices" ou "commodities" (mettez les noms au pluriel).