Afin de faire l’interface d’admin il nous fallait choisir un framework.

Notre groupe a choisi d’utilisé Laravel pour découvrir ce framework tout en étant guidé par 1 personne du groupe qui maitrisé/utilisé le framework.

Quelque prérequis était utile de faire avant d’installer l’interface, on a dû faire la connexion à la base de données afin de pouvoir installer notre interface d’admin.

On a dû pour faire marcher l'interface d'administration de rajouter à toutes nos tables des champs « created\_at « et « update\_at » car l'interface qu'on utilise a besoin de ces 2 éléments pour créer des objets ou modifier des objets

Après avoir fait cette étape, on a fait les différents téléchargements via Composer de notre interface d’admin nommé « Backpack » qui est une collection de packages Laravel qui aide à créer des interfaces d’administration personnalisé.

Pour utiliser Backpack, on a dû suivre un tuto qui permet d’afficher les tables de la base de données, à pouvoir faire les opérations CRUD sur les tables mais aussi à faire les filtres.

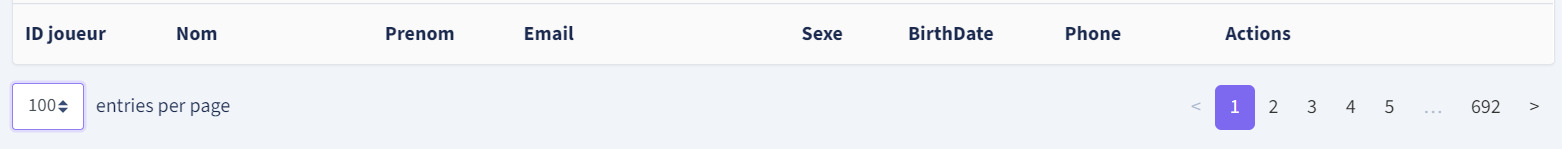
Le résultat de l’interface d’amin se trouve dans les lignes suivantes accompagné de capture d’écrans

La liste est ordonnable comme on peut le voir sur l’exemple ci-dessous. La liste ici est ordonnée de façon croissante de l’id joueur mais elle peut être inversable c’est-à-dire de façon décroissante.

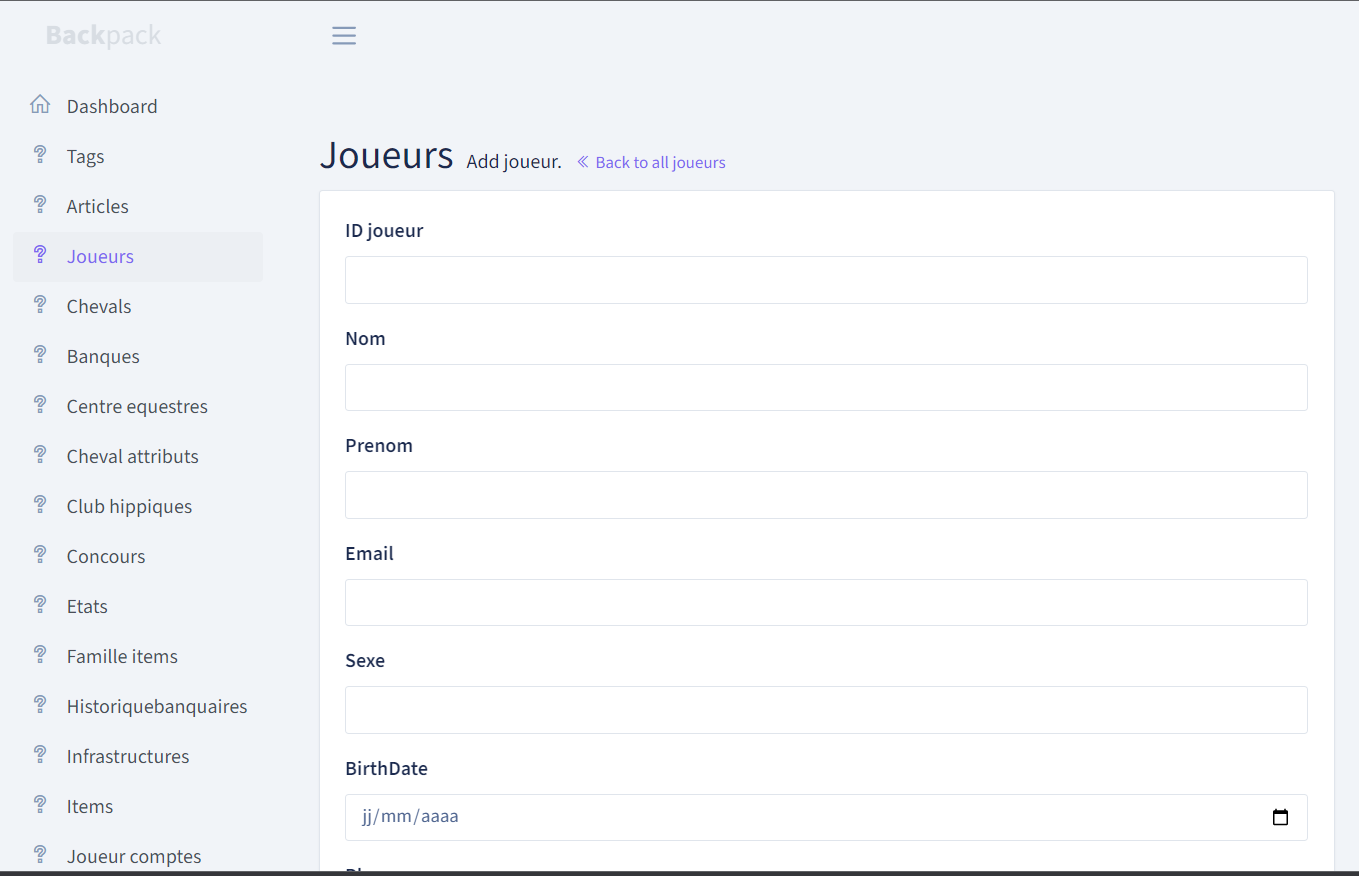
Une image contenant table

Description générée automatiquement

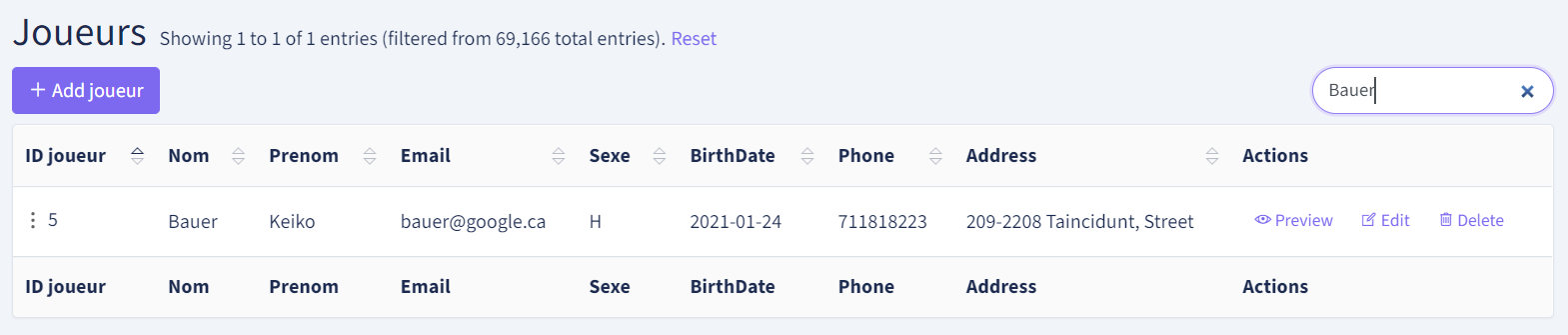
Les listes d’objets sont bien sûre paginé ici on peut voir qu’on a mis 100 objets par page et on 692 listes de page avec chacune encore 100 objets dedans



Voici le formulaire de l’interface d’admin qui permet de créer ou modifier des objets. Le formulaire valide à chaque création ou modification que tous les inputs soient remplis et vérifient aussi la clé primaire.



Nous avons la possibilité de recherché un objet dans ce cas-là on cherche un joueur nommé « Bauer »

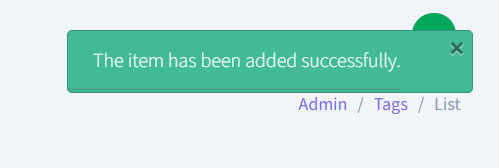


On peut aussi voir les informations de chaque objets (exemple ci-dessous)

Une image contenant table

Description générée automatiquement

Nous avons du coup des messages très rapide qui sont la afin de prévenir l’utilisateur que l’action a été parfaitement terminé/réussi (exemple en capture d’écran ci-dessous ) .



Une image contenant texte

Description générée automatiquement