Module_347 : Labo 3

Auteur : Morgane Cachin & Dylan Diaz

Date : 26.12.2023

Table des matières

Architecture App

1) Base de données : Utilise MySQL avec une interface web phpmyadmin
2) Back-end : Application phyton, se connecte à une base de données et se connecte à un bot discord et exécute les requêtes
3) Front-end : Bot Discord

Architecture Docker

L'application est composée de trois services Docker :

Nos environnements utilisent la même structure mais chacun possède ses propres conteneurs et il se différencient avec les suffixes ***dev*** et ***prod***

1) ***Db_dev/prod*** : Ce service gère votre base de données MySQL. Il utilise l'image Docker officielle de MySQL et stocke les données dans le volume ***db_data_***.
2) ***phpmyadmin_dev/prod***: Ce service fournit une interface utilisateur web pour gérer votre base de données. Il utilise l'image Docker officielle de phpMyAdmin et dépend du service ***db_dev/prod***.
3) ***botdiscord_dev/prod*** : Ce service gère votre bot Discord. Il est construit à partir d'un Dockerfile nommé ***dockerfile_dev/prod*** dans le même répertoire que votre fichier ***docker-compose.dev/prod.yml.*** Il dépend également du service ***db_dev/prod***.

Quoi dans quel conteneur

1) Le conteneur ***db_dev/prod*** contient votre base de données MySQL.
2) Le conteneur ***phpmyadmin_dev/prod*** contient l'interface utilisateur web de phpMyAdmin pour gérer votre base de données.
3) Le conteneur ***botdiscord_dev/prod*** contient votre bot Discord.

Environnements et les caractéristiques les différentiant

Environnement Dev

Nous avons choisi un environnement de dev ou il sera possible de changer directement le code source depuis votre PC. Et avec nodemon l'application redémarrera pour appliquer les changements effectués.

Il y aura aussi une base de données en MySQL pré-peuplé avec des données fictives qui sert à contrôler que l'application soit bien connectée à la base de données est qu'elle est accès aux données.

Environnement Prod

Puis nous avons un environnement de production, qui aura lui aussi une base de données MySQL pré-peuplé avec des données réelles et une interface web phpmyadmin

HOW TO :

Pour réaliser ce projet, il y a plusieurs prérequis.

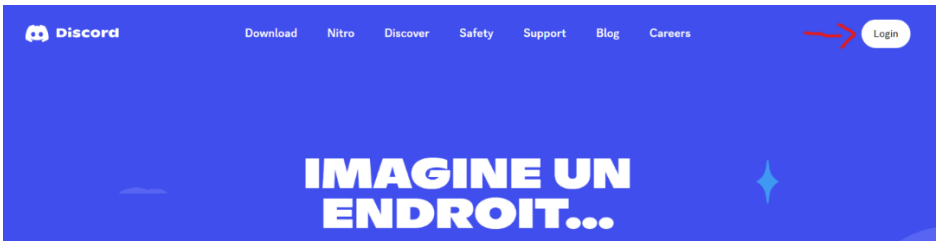
Discord :

Création compte :

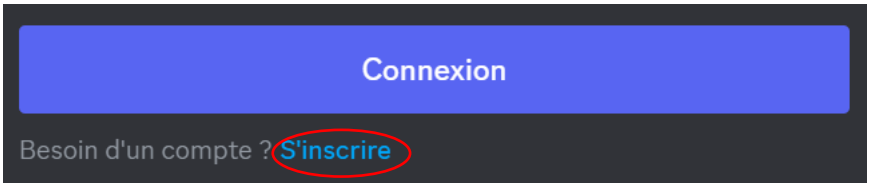
Discord est une application de communication, elle peut être utilisée comme application de bureau ou sur web.

Vous pouvez télécharger l'application avec ce lien : https://discord.com/download

Ou vous connecter directement sur la page web en cliquant sur le bouton « Login » en haut à droite : https://discord.com/

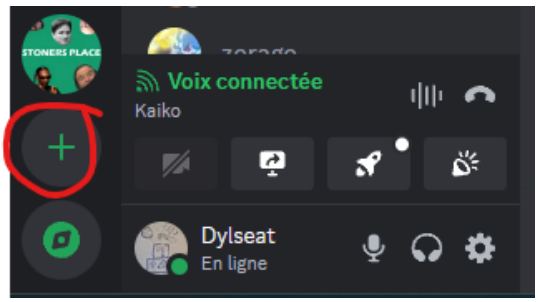


Si vous n'avez pas encore de compte, une fois sur la page de login cliquez sur le lien « s'inscrire » en dessous du bouton « Connexion »



Création serveur :

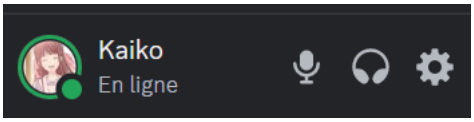
Pour utiliser le bot vous allez avoir besoin d'un serveur. Vous pouvez le créer en cliquant sur le bouton « + » à la fin de la liste des serveurs :



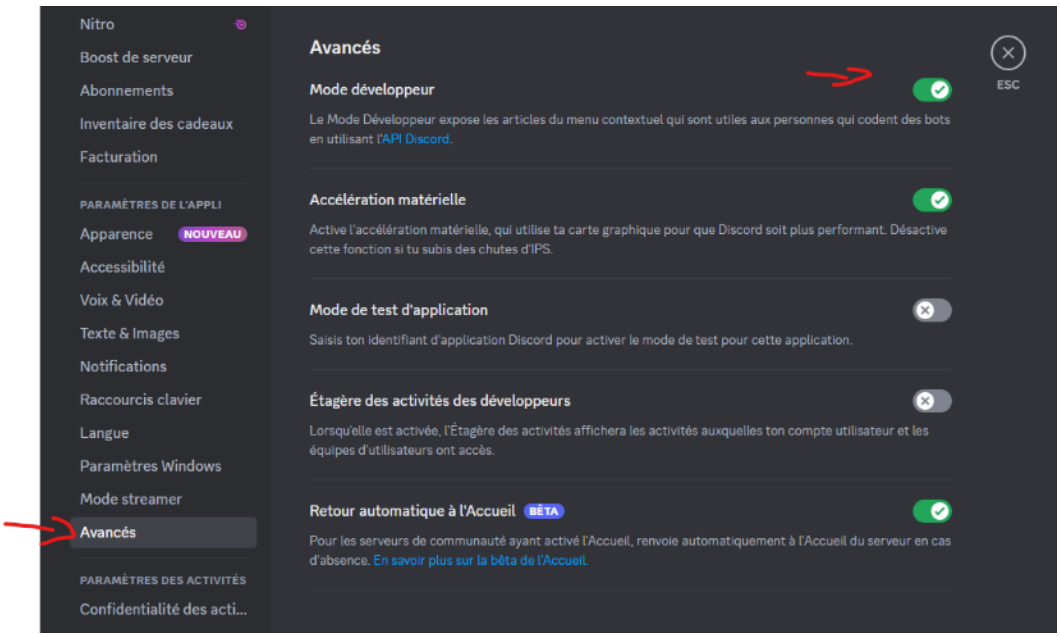
Un popup s'ouvrira pour configurer votre serveur. Cliquez sur « Créer le mien » et choisissez le type de votre serveur pour vos amis ou pour une communauté. Définissez le nom et une image (pas obligatoire) et cliquez sur le bouton Créer.

Création du bot :

Une fois le serveur créé au tour du bot, pour ce faire :
Il faut accéder au paramètre qui se trouve à côté de votre profil en bas à gauche de l'écran :

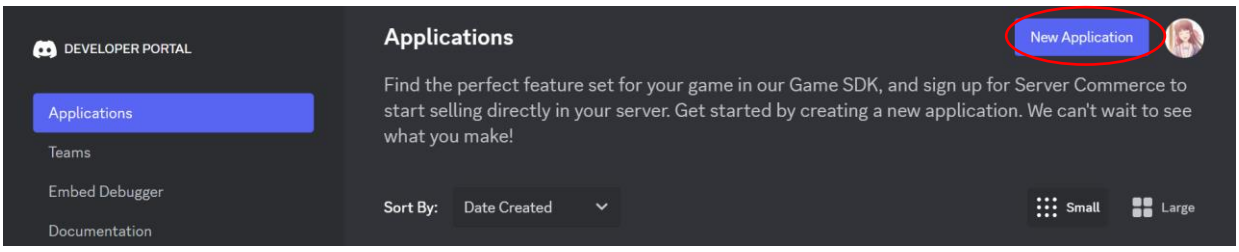


Ensuite, il faut activer le paramètre « Mode développeur » que vous retrouverez dans les options avancées de l'application :

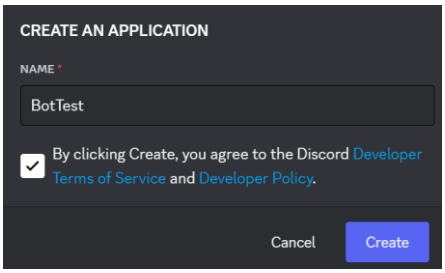


Maintenant, vous pouvez accéder au portail de développeur discord soit en cliquant sur « API Discord » (image précédente) ou cliquez sur ce lien : https://discord.com/developers/applications

Sur l'onglet Application cliquez sur le bouton « New Application »



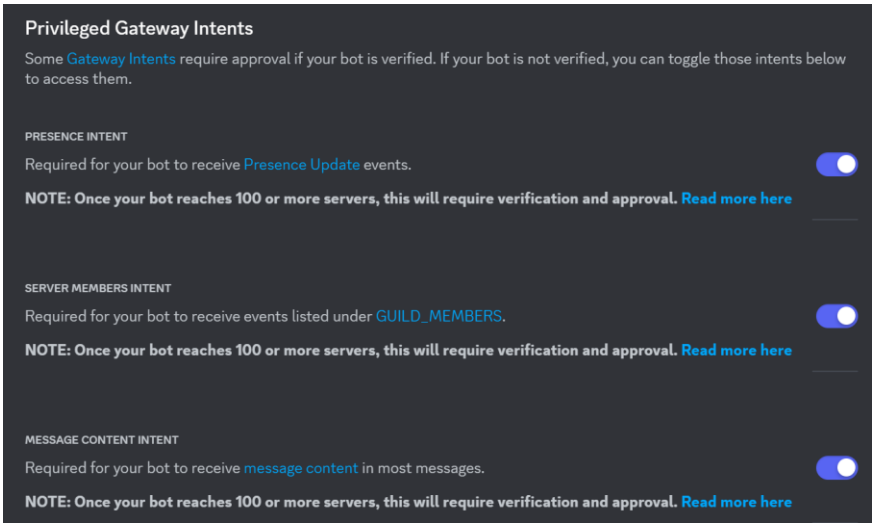
Donnez un nom à votre bot cochez la case et cliquez sur le bouton Create :



Tout d'abord, allez dans l'onglet « Bot ».

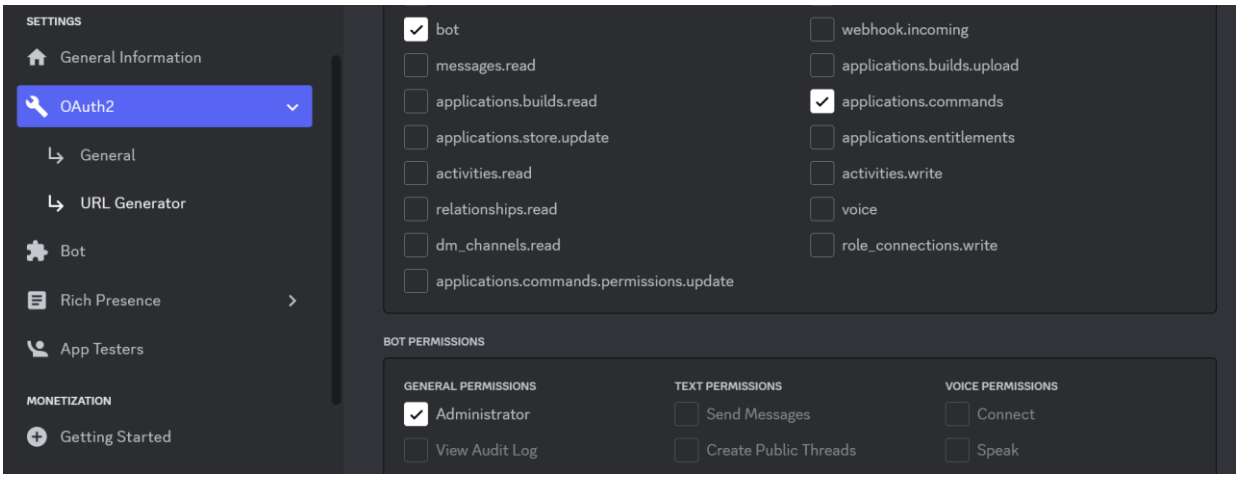
Il vous faudra créer un token. Pour en créer un, vous devez cliquer sur le bouton « Reset Token ». Conservez précieusement ce token pour la suite et ne le partagez pas.

Ensuite, activez ces trois options :

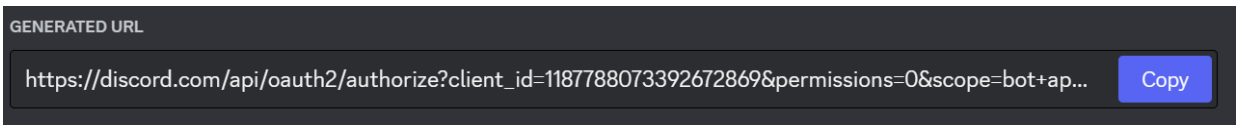


N'oubliez pas de sauvegarder vos changements.

Pour relier le bot au serveur discord il faut créer une URL. Pour cela, allez sur l'onglet « OAuth2 » puis sur « URL Generator ». Suite à cela, dans scopes, sélectionnez les options « bot » et « applications.commands » et dans Bot permission sélectionnez « Administrator ».



L'URL générée se trouve en bas de la page, copiez et collez l'url dans un nouvel onglet de votre navigateur.



Sur cette nouvelle page, sélectionnez le serveur sur lequel vous voulez ajouter le bot et cliquez sur le bouton « autoriser ».

Docker :

Maintenant que votre bot à accès au serveur, créer un fichier à la racine du projet s'appelant « .env » et copier le token comme ceci : TOKEN=***tokenquevousvenezdegénérer***

Ouvrez le terminal et déplacez-vous à la racine du projet ou ouvrez-le directement à la racine. Et lancez l'application Docker Desktop

Pour lancez le projet en environnement de développement, vous devez utiliser cette commande docker dans le terminal : « *docker-compose -f docker-compose.dev.yml up -d »*

Vous pouvez modifier l'intervalle des messages dans le code source (main.py) à la ligne 16 en notifiant la durée voulu dans les parenthèses (seconds=chiffre, minutes=chiffre, hours=chiffre)

```
16  @tasks.loop()  # Définissez l'intervalle de temps que vous voulez ici
```

À la suite de ces modifications, on peut constater que dans les logs du conteneur que nodemon redémarre le conteneur avec les nouvelles modifications.

```
[nodemon] restarting due to changes...
[nodemon] starting `python -u Scripts/main.py`
```

Les données utilisées dans cet environnement sont fictives pour vérifier que le bot accède à toutes les colonnes.

Pour arrêter le projet, utilisez la commande suivante :
« *docker-compose -f docker-compose.dev.yml down* »

Pour lancez le projet en environnement de production, vous devez utiliser cette commande docker dans le terminal : « *docker-compose -f docker-compose.prod.yml up -d* »

Il ne peut pas y avoir de modification du code avec cet environnement.

Dans cet environnement nous utilisons de vraies données.

Pour arrêter le projet, utilisez la commande suivante :
« *docker-compose -f docker-compose.prod.yml down* »