

Développement d'un BOT Discord

Rapport de projet LATEX

7 Avril 2021

par

Gwendal, Benoit, Florian H, Antoine C, Maxime

Professeur : Antoine Gamelin

IUT de Calais

Table des matières

Chapitre 1 Présentation du projet	1
1.1 Objectif	1
1.2 Le BOT	1
1.2.1 Présentation	1
1.2.2 Fonctionnalités	2
1.3 Documentation	2
1.3.1 Prérequis	2
1.3.2 Création d'une application Discord	2
1.3.3 Inviter son BOT sur son serveur Discord	3
1.3.4 Lancement du BOT	4
1.3.5 Les commandes importantes	4
1.4 Système meteo par Maxime	5
1.4.1 Présentation de la fonctionnalité	5
1.4.2 Algorithme simplifié de la fonctionnalité	5
1.4.3 Difficultés rencontrées	5
1.4.4 Ce que j'ai appris	5
1.4.5 Résultat de la fonctionnalité	5

Chapitre 1

Présentation du projet

Lors de notre année LP DIM, il était demandé dans le module "Complètement informatique" de réaliser un projet de développement afin de rédiger un rapport avec \LaTeX

1.1 Objectif

L'objectif de ce projet est de se connecter à une API et exécuter des requêtes. Nous avons fait le choix d'utiliser l'API Discord afin d'y développer un robot répondant à nos requêtes.

1.2 Le BOT

1.2.1 Présentation

Afin de développer un BOT, il faut créer une application depuis l'espace développeur de Discord

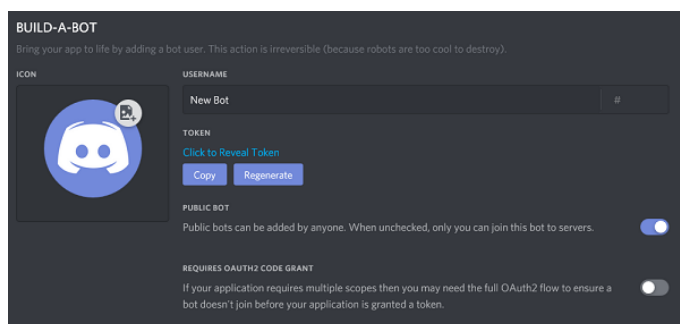


FIGURE 1.1 – Page de création d'une application BOT.

Nous avons choisis d'utiliser le langage JAVA car une librairie complète est disponible gratuitement afin de développer son BOT facilement : il s'agit de JDA disponible à cette adresse : <https://github.com/DV8FromTheWorld/JDA>

1.2.2 Fonctionnalités

- Générateur de memes
- Utilisation de photos personnalisées de chats
- Système de météo
- Photo du jour de la NASA
- Photo de la NASA en fonction d'une date définie par l'utilisateur

1.3 Documentation

1.3.1 Prérequis

- Java 11+
- Compilateur Java
- Discord Developpers : suivre le sous chapitre "Création d'une application Discord".
- Editeur de code si vous souhaitez modifier des features.

1.3.2 Création d'une application Discord

Avant de récupérer les fichiers du BOT, il est nécessaire de récupérer un token. Ce token sera accessible en créant votre application dans le portail Discord Developpers.

1. Ouvrez et connectez-vous au portail des développeurs Discord : https://discord.com/login?redirect_to=%2Fdevelopers%2Fapplications%2F.
2. Cliquez sur New Application.
3. Entrez le nom du bot (exemple BotIUT) et confirmez la création en cliquant sur le bouton Créer.
4. Déplacez vous maintenant dans l'onglet Bot
5. Cliquez sur Add Bot.
6. Confirmez en cliquant sur Yes, do it !.
7. Sur la page principal de votre application, dans la rubrique Token, cliquez sur "Copy"

Gardez précieusement votre token en la collant par exemple dans un fichier .txt

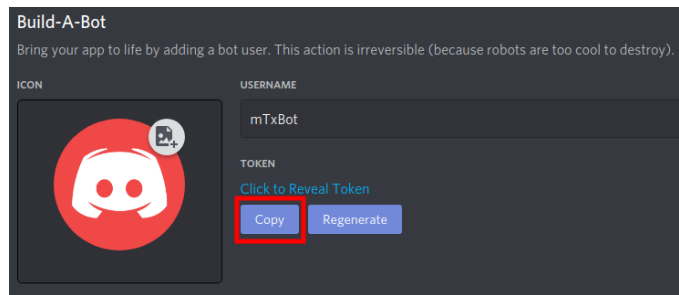


FIGURE 1.2 – Récupération du token.

1.3.3 Inviter son BOT sur son serveur Discord

Nous vous conseillons de créer un serveur privée via le client Discord pour inviter et tester l'application.

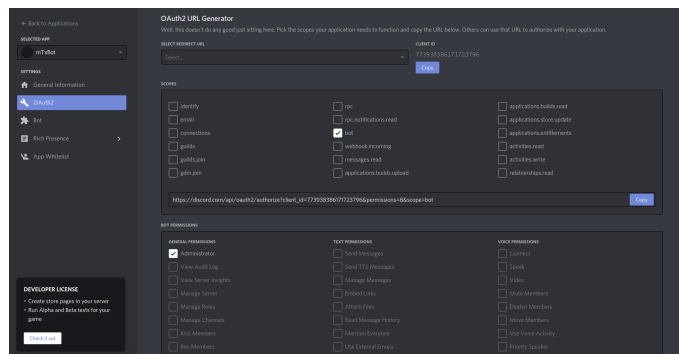


FIGURE 1.3 – Invitation de l'application sur le serveur.

1. Dans le portail développeurs, cliquez sur l'onglet OAuth2.
2. Dans la section scopes de OAuth2 URL Generator, sélectionnez bot.
3. Une nouvelle section Bot Permissions doit apparaître juste en dessous, cliquez sur Administrator dedans.
4. Cliquez sur Copy dans scopes pour récupérer le lien d'invitation du bot.
5. Ouvrez le lien que vous venez de copier dans votre navigateur.
6. Sélectionnez le serveur discord sur lequel ajouter votre bot.
7. Validez les permissions en cliquant sur Autoriser.

1.3.4 Lancement du BOT

Maintenant que notre BOT est sur notre serveur, il faut l'activer.

1. Ouvrez un terminal bash
2. Dirigez vous dans le dossier du projet :

```
cd <chemin_du_projet>
```

3. Compilez en executant la commande :

```
javac -cp .:* Bot.java
```

4. Lancez le BOT avec la commande :

```
java -cp .:* Bot <votre_token>
```

1.3.5 Les commandes importantes

Votre Bot est désormais en ligne, dans un salon public de votre serveur, exécutez cette commande pour connaître les différentes fonctionnalités disponibles :

```
/MM help
```

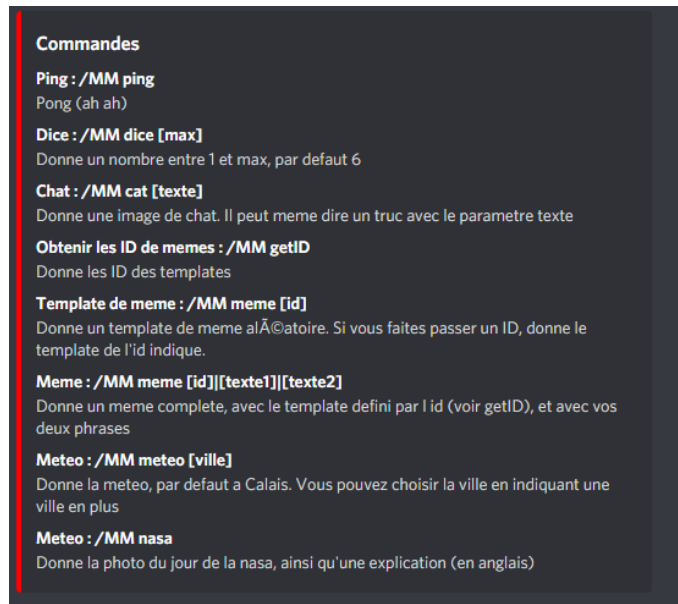


FIGURE 1.4 – Liste des fonctionnalités prêtes à l'emploi.

1.4 Système meteo par Maxime

1.4.1 Présentation de la fonctionnalité

Le but de cette fonctionnalité était à l'aide d'une API, de pouvoir prendre les fonction d'une ville par ça meteo, soit la temperature, le levé du soleil et son couché

Alors j'ai donc fait mes petites recherche pour pouvoir m'aider via internet de comment faire

1.4.2 Algorithme simplifié de la fonctionnalité

```
Si la commande contient meteo et si elle contient un
argument supplémentaire , on appelle l'api_avec_comme_ville
l'argument, sinon si la commande contient meteo sans argument
supplementaire , on donne la meteo de calais
```

1.4.3 Difficultés rencontrées

J'ai rencontré une difficulté. C'était de pouvoir signaler une erreur si la ville existé pas, au debut je n'ai pas vraiment compris comment faire, Du a mon manque de connaissance d'API cela ma pris un peu de temps a comprendre Difficultés d'adaptation de langage aussi, un moment que le java n'était pas dans mes reperes et donc devoir se remettre dedans

1.4.4 Ce que j'ai appris

J'ai appris pas mal de choses, premiere le principe d'une API et a quoi cela sert réellement, je me suis beaucoup instruit deçu. Ensuite, j'ai appris a mieux utiliser discord, son interface et son efficacité a pouvoir mettre aussi facilement un bot. Cela ma aussi permis de improve dans les domaines qu'on a vu, api, java et LATEX

1.4.5 Résultat de la fonctionnalité

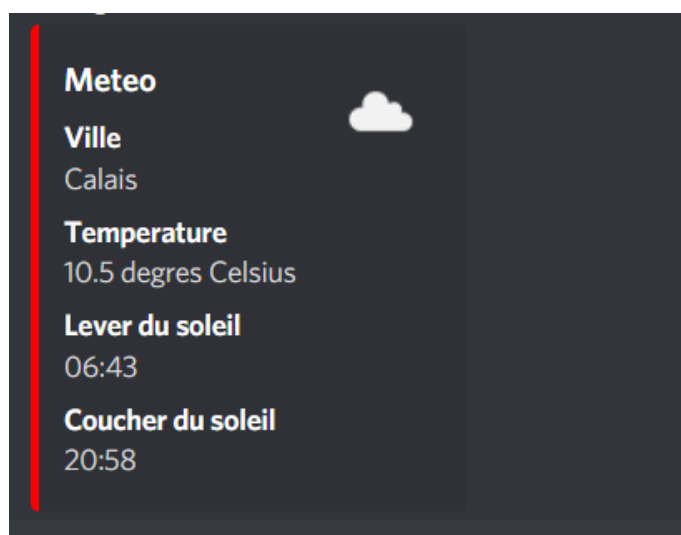


FIGURE 1.5 – Image du résultat attendu.