

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 Affichage de la photo du jour de la Nasa par Antoine C	5
1.4.1 Présentation de la fonctionnalité	5
1.4.2 Algorithme simplifié de la fonctionnalité	5
1.4.3 Difficultés que j'ai rencontré	5
1.4.4 Ce que ce projet m'a apporté	6
1.4.5 Résultat de la fonctionnalité	6

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

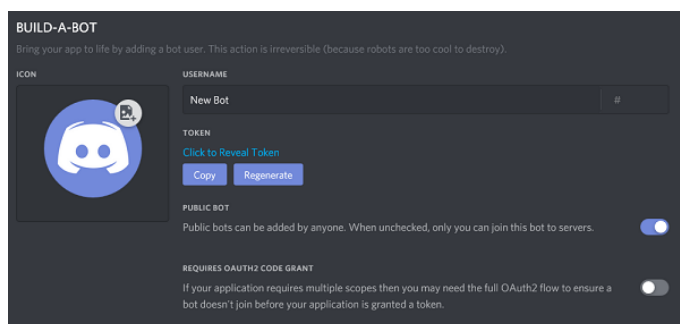


FIGURE 1.1 – Page de création d'une application BOT.

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

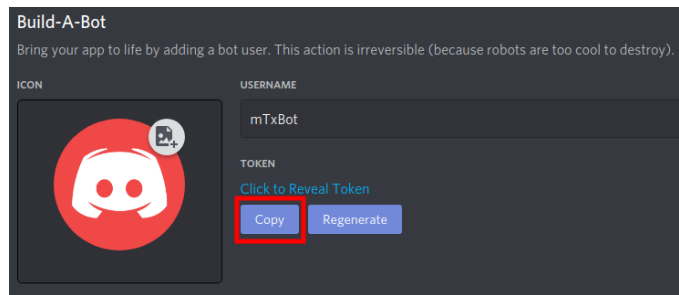


FIGURE 1.2 – Récupération du token.

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.

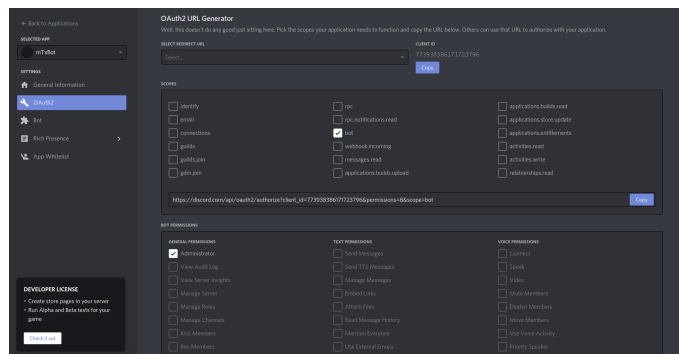


FIGURE 1.3 – Invitation de l'application sur le serveur.

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 Affichage de la photo du jour de la Nasa par Antoine C

1.4.1 Présentation de la fonctionnalité

L'objectif de ma fonctionnalité était d'afficher la photo du jour de la Nasa ainsi un petit texte explicatif en anglais.

Après quelques recherche sur internet j'ai pu découvrir le site des différentes API de la Nasa qui est disponible a l'adresse : <https://api.nasa.gov>

Sur la documentation de l'API j'ai vu que pour récupérer les informations de l'image il fallait ajouter a l'URL :

1. Le paramètre "/planetary/apod"

1.4.2 Algorithme simplifié de la fonctionnalité

```
Si la commande executee est /MM nasa < date du jour (YYYY-MM-DD) >  
Alors Afficher la photo et le texte explicatif correspondant a la date  
entree
```

```
Sinon Si la commande executee est /MM nasa  
Alors Afficher la photo et le texte explicatif correspondant a la date  
du jour
```

1.4.3 Difficultés que j'ai rencontré

La principale difficulté que j'ai rencontrée était de gérer le fait de pouvoir appeler l'API lorsque la date est différente de la date actuelle, car je ne savais pas quel format utilisé, après quelque recherche internet, j'ai trouvé la solution.

```
JSONObject jsonNasa = (JSONObject) (parser.parse(picture));  
}
```

D'un autre coté, j'ai eu quelques prone varblemes avec le fichier JSON, je ne savais pas comment stocker le fichier JSON dans un fichier, j'ai trouver la solution par la suite :

```
if (date != "") {  
    serv = serv + "&date=" + date;  
}
```

1.4.4 Ce que ce projet m'a apporté

Durant ce projet j'ai pu revoir certaines notions sur les API que j'avais vu l'an dernier telle que la gestion des variables, l'appel d'une API avec la méthode GET. Ensuite grâce à ce projet, j'ai pu découvrir et pratiquer le Latex pour rédiger des rapports, langage que je ne connaissait absolument pas avant .

1.4.5 Résultat de la fonctionnalité

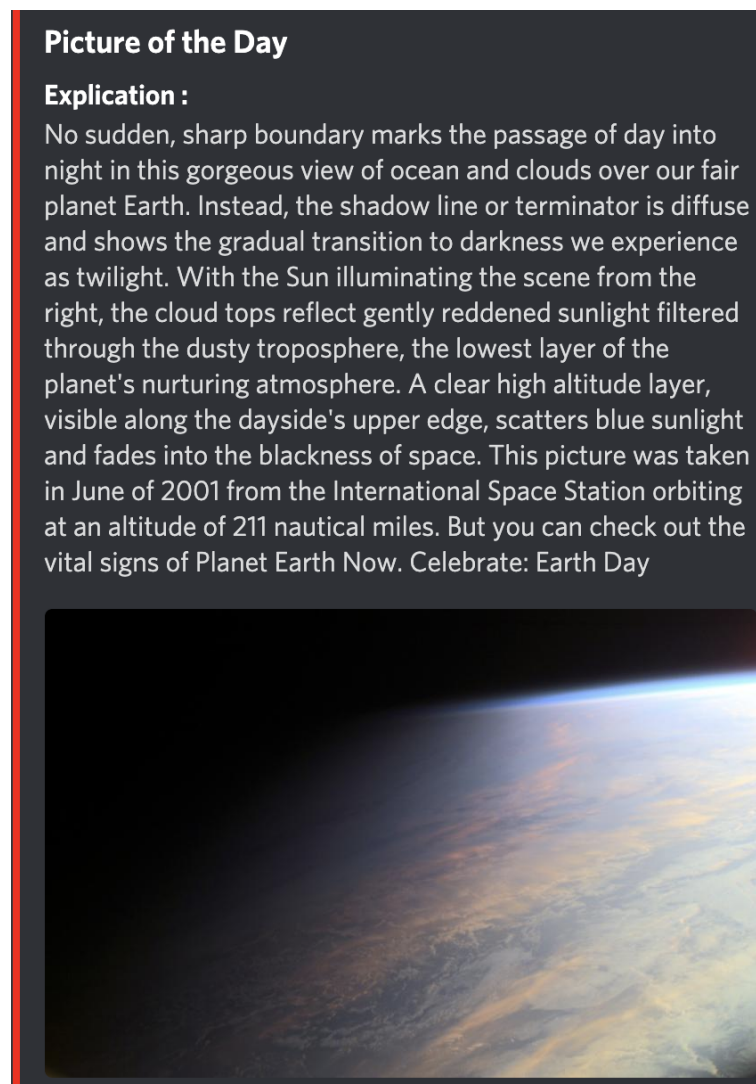


FIGURE 1.5 – Résultat de la commande /MM nasa 2021-04-22