
Bot Discord

Réalisé par
Benoit Peutevynck

Dans le cadre du cours
Complément Informatique

Travail présenté à
Mr Gamelin

1 Introduction

Le but du projet été de créer un Bot Discord en utilisant l'API Discord. Nous avons donc décider de créer un bot qui permet de faire l'appel pour un discord utiliser dans le cadre du TéléTravail.

Le bot discord aura plusieurs fonctionnalité, la première étant de faire l'appel et lister les présent et les absent dans une base de données. La seconde est de savoir combien de fois un élève a était absent dans l'année scolaire. Et puis enfin, la dernier fonctionnalité est d'afficher un historique des élève présent et absent.

2 Ma partie

Ma partie consister a la créeaction de la base de données et la fonction qui permet de récupérer les nom des personnes qui check.

Pour la base de données on a utiliser l'ORM Sequelize que j'avais déjà utiliser. Cela permet de généraliser les commandes et la crée plus facile les tables.

J'ai utiliser mysql de Wamp pour crée une base de donnée. J'ai ensuite rentrer les information de connexion dans le projet.

```
const { Sequelize } = require('sequelize');

module.exports = new Sequelize('school', 'root', '', {
  dialect: 'mysql',
  host: 'localhost'
});
```

Ensuite, j'ai défini une table présent qui contiendra tout les personnes présent a un cours. Après avoir crée le model de la table je devais initialiser sequelize et la base de donnée dans le projet.

Pour cela j'ai du créer un fichier que j'importe dans le main qui va créer la table dans la base de donnée déjà créer.

```
const { DataTypes, Model } = require('sequelize');

module.exports = class Present extends Model {
  static init(sequelize) {
    return super.init({
      NomEleve: {
        type: DataTypes.STRING
      },
      Classe: {
        type: DataTypes.STRING
      },
      Date: {
        type: DataTypes.DATE
      },
      Cours: {
        type: DataTypes.STRING
      },
      Heure: {
        type: DataTypes.TIME
      }
    }, {
      tableName: 'Present',
      sequelize: sequelize
    })
  }
}
```

```
const Absent = require('../models/Absent');
const Present = require('../models/Present');
const db = require('../database');
module.exports = {
  init: () => {
    console.log("Mon BOT est Connecté");
    Present.init(db);
    Present.sync();
    Absent.init(db);
    Absent.sync();
    db.authenticate()
      .then(() => {
        console.log('BDD Done');
      }).catch((err) => console.log(err));
  }
}
```

L'autre partie de mon travail était de récupérer les nom des personnes qui check l'appel. Pour cela on créer une action quand une reaction valider et fait sur le message de l'appel.

On récupère la variable user dans laquelle contient l'id de personne qui a check il nous reste plus qu'à trouver son nickname grâce à son "user.id".

Il faut que les élèves renomme leurs pseudo dans le serveur sinon leurs nickname sera null. Ensuite je vérifie si le nickname n'est pas null et alors je le met dans une liste.

```
message.channel.send(exampleEmbed).then((question) => {
  question.react("✅").then(() => {
    const filter = (reaction) => reaction.emoji.name === "✅";
    question
      .awaitReactions(filter, { time: 8000 })
      .then((collected) => {
        collected.forEach((element) => {
          element.users.cache.each((user) => {
            //console.log("test 1 : " + user.username);
            const oui = message.guild.members.cache.get(user.id);
            if (oui.nickname !== null) {
              listUsers.push(oui.nickname);
            }
          });
        });
      });
  });
});
```

3 Bilan du travail

J'avais jamais utiliser l'API Discord, cela a été une bonne découverte même si j'ai perdu beaucoup de temps à trouver comment récupérer les personnes qui ont check à cause de ça mais collègues on pas pu commencer toute suite leurs travail.

J'ai beaucoup aimé travailler sur un bot Discord, j'ai acquis une bonne base pour la programmation d'un bot Discord en nodeJs.