**DOCUMENTATION TECHNIQUE BOT DISCORD**

**load\_data-\_from\_json(file\_name)** :

Cette fonction prend en argument un nom de fichier et tente de lire et charger les données JSON à partir de ce fichier. Si le fichier n'existe pas, la fonction retourne un dictionnaire vide. Cette fonction est utilisée pour charger les historiques des commandes des utilisateurs depuis un fichier JSON.

**save\_data\_to\_json(file\_name, data)** :

Cette fonction prend en argument un nom de fichier et des données à sauvegarder. Elle écrit les données au format JSON dans le fichier spécifié. Cette fonction est utilisée pour sauvegarder les historiques des commandes des utilisateurs dans un fichier JSON.

**get\_random\_motivation\_quote():**

Cette fonction asynchrone effectue une requête HTTP GET pour récupérer une citation motivante aléatoire à partir de l'API zenquotes.io. Si la requête réussit, la fonction retourne la citation et son auteur sous la forme d'une chaîne de caractères. Sinon, elle retourne un message d'erreur.

**send\_motivation\_quote():**

fonction asynchrone récupère une citation motivante aléatoire en appelant la fonction get\_random\_motivation\_quote() et envoie cette citation dans un canal Discord spécifique en utilisant l'ID du canal.

**current\_datetime(ctx):**

Cette fonction récupère la date et l'heure actuelles, les formate en une chaîne de caractères lisible et envoie cette chaîne dans le contexte de la commande (ctx).

**before\_any\_command(ctx):**

Cette fonction est appelée avant l'exécution de n'importe quelle commande. Elle ajoute la commande à la file d'attente du bot et envoie un message indiquant que la commande est en attente d'exécution.

**on\_command\_completion(ctx):**

Cette fonction est appelée lorsqu'une commande est terminée. Elle sauvegarde les historiques des commandes des utilisateurs dans un fichier JSON.

**send\_motivation\_quote() :**

la planification de l'envoi de citations motivantes est réalisée à l'aide de la bibliothèque APScheduler. La fonction send\_motivation\_quote() est planifiée pour s'exécuter tous les jours à 6h00 du matin.

**save\_polls() :**

Cette fonction sauvegarde les données des sondages dans un fichier JSON nommé 'polls.json'. Les données sont sauvegardées avec un formatage pour une meilleure lisibilité.

**load\_polls() :**

Cette fonction charge les données des sondages à partir d'un fichier JSON nommé 'polls.json'. Si le fichier n'existe pas, elle renvoie un dictionnaire vide.

**on\_reaction\_add :**

Cet événement est déclenché lorsqu'une réaction est ajoutée à un message. Il vérifie si la réaction concerne un sondage, puis met à jour les votes de l'utilisateur en conséquence et sauvegarde les données dans le fichier JSON.

**on\_reaction\_remove :**

Cet événement est déclenché lorsqu'une réaction est supprimée d'un message. Il vérifie si la réaction concerne un sondage, puis met à jour les votes de l'utilisateur en conséquence.