BetBurger

Google API

* To create a Google Cloud Console project and enable the Google Sheets API, you can follow these steps:
* 1. Open your web browser and go to the Google Cloud Console (https://console.cloud.google.com).
* 2. If you haven't already, sign in to your Google account. If you don't have a Google account, create one.
* 3. Once you're signed in, you'll be redirected to the Google Cloud Console homepage. Click on the project drop-down menu at the top of the page, near the Google Cloud Platform logo. This menu displays your existing projects, if any.
* 4. Click on the "New Project" button located on the top-right corner of the page. A dialog box will appear.
* 5. In the dialog box, enter a name for your project. Choose a unique and descriptive name that represents the purpose of your project.
* 6. (Optional) Select an organization from the drop-down menu if you want the project to be associated with an organization. If you don't have an organization, you can skip this step.
* 7. (Optional) Choose a location for your project by clicking on the "Location" drop-down menu. Select the region or zone that is most appropriate for your project. If you're unsure, you can choose the default location.
* 8. After entering the project name and optionally selecting an organization and location, click on the "Create" button. Google Cloud Console will create your new project.
* 9. Once the project is created, you'll be redirected to the project dashboard. In the left navigation menu, click on "APIs & Services" and then select "Library" from the submenu.
* 10. In the "Library" section, search for "Google Sheets API" using the search bar.
* 11. Click on the "Google Sheets API" result that appears in the search results.
* 12. On the Google Sheets API page, click on the "Enable" button. This action will enable the API for your project.
* 13. You may also need to set up credentials to authenticate your API requests. In the left navigation menu, click on "Credentials" under the "APIs & Services" section.
* 14. Click on the "Create credentials" button and select the appropriate credential type for your use case. For accessing the Sheets API, you may choose "OAuth client ID."
* 15. Follow the instructions to configure your credentials, including specifying authorized JavaScript origins and redirect URIs if required.
* 16. Once you've completed the credential setup process, you'll receive the necessary credentials, such as a client ID and client secret.
* You have now created a Google Cloud Console project and enabled the Google Sheets API. You can use the provided credentials to authenticate your application and start interacting with the API to read, write, and manipulate spreadsheets.
* Certainly! After creating the project and enabling the Google Sheets API, you can download the JSON file containing your project's credentials. Here's how you can do it:
* 1. Make sure you are in the Google Cloud Console and on your project's dashboard.
* 2. In the left navigation menu, click on "Credentials" under the "APIs & Services" section.
* 3. On the Credentials page, you will see a list of your project's credentials. Look for the credential you created earlier, which should be an OAuth client ID.
* 4. In the OAuth client ID section, locate the credential you want to download and click on the name of the credential to open its details.
* 5. In the credential details, scroll down to the "OAuth 2.0 Client IDs" section. You should see a download icon (downward arrow) next to the credential name.
* 6. Click on the download icon to download the credentials JSON file to your computer.
* The downloaded JSON file contains the necessary information, such as the client ID, client secret, and other authentication details, which you can use in your application to authenticate and interact with the Google Sheets API.
* Make sure to keep the credentials JSON file secure and don't share it publicly or expose it in your code repositories, as it contains sensitive information that grants access to your project.
* Name the json file to google.json
* In Json file you get a email named as iam.gserviceaccount.com
* Copy email
* Go Spread sheets and create one Name it top with “Reports” and in bottom “Reportsheet”
* Click on share on sheet and on add paste email and give editor or complete permission & Done
* On sheet each column type your heading as follow

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Person (username) | Date | Event Name | Bookie | Event Start On | Time to Event | Market | Current Odds | Last Accepted Odds | Placed Odds | Amount |

* On url of the page , below url after “d” you see
* An Alpha numeric code “1fNbP9cwvZmEZRCzrpksmCYqNWWS1wAhkzaWHggPDmhk”
* <https://docs.google.com/spreadsheets/d/1fNbP9cwvZmEZRCzrpksmCYqNWWS1wAhkzaWHggPDmhk/edit#gid=0>
* Copy it and paste in Discord Bot.py at “Google\_url\_Key”
* Make sure google.json and bot.py should same folder

Discord Server and Discord Bot

To create a Discord server and a Discord bot, follow these steps:

Creating a Discord Server:

1. Go to the Discord website (https://discord.com) and sign in to your Discord account. If you don't have an account, you can create one for free.

2. After signing in, you'll be on the Discord homepage. On the left-hand side, you'll find a column with a list of servers you're a part of. At the bottom of this column, click on the "+" button to create a new server.

3. A pop-up window will appear with two options: "Create a Server" and "Join a Server." Choose "Create a Server."

4. Enter a name for your server in the "Server Name" field. Choose a name that represents the purpose or topic of your server.

5. Optionally, you can upload an image as the server's icon by clicking on the square next to the server name. This step is not required and can be skipped.

6. Once you've entered the server name and, if desired, added an icon, click on the "Create" button.

7. Congratulations! You have successfully created your Discord server. You can customize the server further by adding channels, roles, and configuring various settings.

Creating a Discord Bot:

1. Go to the Discord Developer Portal website (https://discord.com/developers/applications) and sign in using your Discord account.

2. After signing in, you'll be on the Developer Portal homepage. Click on the "New Application" button at the top-right corner of the page.

3. In the "Create an Application" dialog box, enter a name for your bot in the "Name" field. Choose a name that represents the purpose or functionality of your bot.

4. Once you've entered the “BetBurgerBot” bot name, click on the "Create" button.

5. You'll be redirected to the application settings page for your bot. In the left navigation menu, click on "Bot" under the "Settings" section.

6. On the Bot page, click on the "Add Bot" button.

7. A confirmation dialog box will appear, asking if you want to create a bot for your application. Click on "Yes, do it!" to proceed.

8. Once the bot is created, you'll see a section with the bot's username and a "Token" field. The token is a secret key that you'll need to authenticate your bot and interact with the Discord API.

9. IMPORTANT: Keep the bot token secure and don't share it publicly or expose it in your code repositories. It grants access to your bot and should be treated as sensitive information.

10. Customize your bot further by configuring its name, avatar, permissions, and other settings according to your requirements.

To add your bot to your Discord server, you'll need to generate an OAuth2 URL. Here's how:

1. In the left navigation menu, click on "OAuth2" under the "Settings" section.

2. In the "Scopes" section, select the "bot" checkbox.

3. Scroll down to the "Bot Permissions" section and choose the permissions your bot requires on your server. For example, if your bot needs the ability to read and send messages, select the "Send Messages" and "Read Messages" permissions.

4. After selecting the necessary permissions, a generated URL will appear in the "Scopes" section. Click on the "Copy" button next to the URL to copy it to your clipboard.

5. Open a new browser tab or window and paste the copied URL into the address bar. Press Enter to go to the URL.

6. On the webpage that opens, select your Discord server from the drop-down menu. This will authorize the bot to join your

server with the specified permissions.

7. Complete the authorization process by following the on-screen instructions.

Congratulations! You have successfully created a Discord server and a Discord bot. The bot is now added to your server and ready to be programmed with the desired functionality using the bot token you obtained.

Above you Generated a token copy and paste in DiscordBot.py at bottom

At run(“you discord token”)

**Now all Token are created now starting the bot**

* **Before what files are there in project folder**
  + - * **DiscordBot.py**
      * **Scraper.py**
      * **Credits.py**
      * **Chromedriver.exe**

**All Python files has some requirement**

* + - **Pip install gspread**
    - **Pip install selenium**
    - **Pip install sqlite3**
    - **Pip install discord**
    - **Pip install regex**
    - **Pip install bs4**
    - **Pip install asyncio**

**File Structure**

* You see four files in Folder
  + - * Chromedriver.exe
      * DiscordBot.py
      * Scraper.py
      * Credits.py
* Chromedriver.exe
  + - If system or server is it is enough ,if
    - Server is another os like mac,ubuntu you need to download there file extension Chomedriver
    - Go to search for chromedriver then go to first link of google and download on latest version
* Discordbot.py
  + - It is for All Discod Bot controls
    - Run it by “python discordbot.py”
    - Then after little time gap it show “BetBurgerBot is ready”
    - Then by discord interface send a direct msg to bot as
    - “@STARTACTIONSMSG143”
    - In Terminal you see async action are started .
    - Now Bot is Done to do all its task’s
* Scraper.py
  + - Run it “python scraper.py”
    - After little time it show “Login Successful”
    - If login not successful it show “login failed” and exit automatic
    - If it login Successful no Problem deployment has Successful
    - If Login Failed
    - Reasons for Login failed is Just when scraper type pass and email it ask captacha and bot can’t do it .
    - So for it ,you login with that credits in credits to bet burger.com
    - If captcha showed then solve it and signout
    - Then final re run the start command “python scraper.py”
    - It show login successful and Deployment is successful
* Note First run the Scraper ,After Deployment of scraper ,Run the DiscordBot.py
* Bot usage , new user need to connect to server by name or link
* Then send a direct message to Bot ,hi or hello
* After it he/she , receive all Bets periodically
* Reply on a bet ,with 2.39-400 or 300-400 format make a order with comfirmation

The End

**Project by Rohith Yarramala.**

**4/7/2023**