

# Backend-Discord-GUI

---

## Model - JSON Structure & Layout

- JSON Structure
  - Master-Element ( Name: cmd ) | Dictionaries
    - Title | String - e.g. "Test1"
    - Aliases | List (Strings) - e.g. ["t1", "t"]
    - Reference | String (ID) - e.g. "001"
      - This references the unique link ID to show the processes of commands, which will be stored in a list. e.g. ["ChannelText", "AwaitResponse"...], this is the way we will control the flow of commands that a user chooses for a particular entry of text.
  - Master-Element ( Name: references ) | Dictionaries
    - ID | String (ID) - e.g. "001"
      - This is where the Reference ID will link.
    - Commands | List (Strings) - e.g. ["ChannelText", "AwaitResponse", "ChannelText"]
- Layout
  - Master-Window ( Name: main.py | Loading Window, start-up, etc... ) | File
    - Tk() Window | Geo: "300x300"; Title: None; overriddenirect: True; | Class
    - Start-Up-Assets | Create any files needed, basic start-up | Class
  - Master-Application ( Name: application.py | Main Window, UI Front-End | File
    - Tk() Window | Geo: "500(+)x500(+)"; Title: Discord-GUI | Class
    - Create Window (Tk()) - Use: Create a command
    - Log Window (Tk()) - Use: Show a simple terminal
    - Watch Window (Tk()) - Use: Watch ongoing data changes to database etc.
    - Settings Window (Tk()) - Use: Change every in-app setting

Monday, 13 January 2020

- Master-Bot ( Name: bot.py | Backend Environment for bot to work off | File
  - Act as a API, use a form of 'GET', 'POST' e.g. Call a class / function
  - Create Command exists here.
  - Direct access to JSON Files, and any database files
  - Should control the essentials of the bot, e.g. on\_ready() etc...