

# **Cordic Discord Bot**

# Einleitung

- Discord: Online Chatplattform
- Jira: Projektverwaltung



# Anforderungen

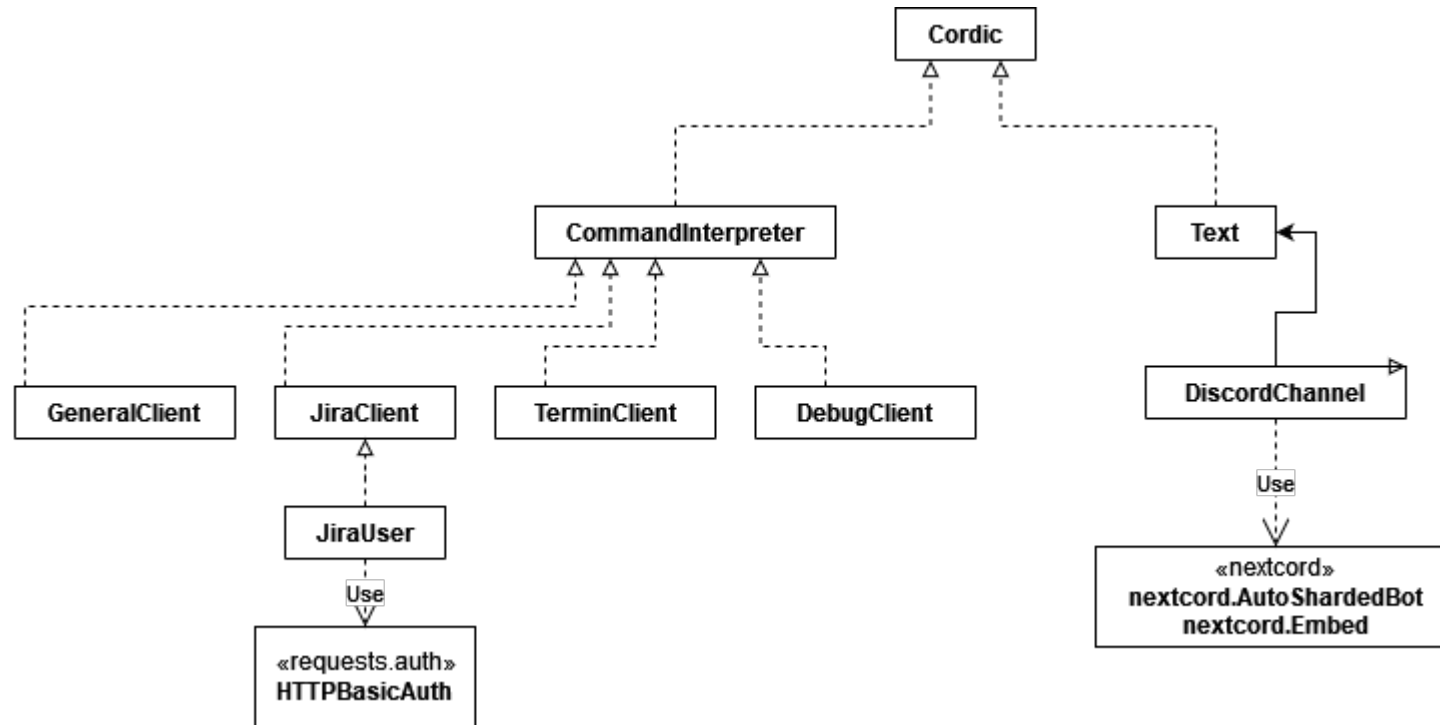
- Kommunikation an zentralisiertem Ort
- Jira Projektverwaltung mit Discord verbinden
- Termine verwalten (in Discord)

# Demo

# Verwendete Tools

- Python 3.9
- Nextcord
- SQLite3, SQLAlchemy
- Venv für Virtual Environment
- Git + GitHub

# Aufbau des Projektes



# Text (DiscordChannel)

- Definition des Cordic Befehls (Slash Commands)
- Besteht aus:
  - Befehl Typ (General, Jira, Termin. Debug)
  - Befehl
  - Argumente

# Command Interpreter

- Verarbeitung der Eingabedaten
- Leitet Eingabe an Command Clients weiter
- Command Clients: General, Jira, Termin, Debug
- Rückgabe: Dictionary für Ausgabe



# Command Interpreter

```
42     embedstructure = {
43         "title": "Error",
44         "description": "Could not find command: Maybe you are missing [t`/termin` or [j`/[jira` or [g`/^general`?",
45         "fields": None,
46         "color": COLOR_RED
47     }
48     output = embedstructure
49
50     if target_client is None:
51         return output
52
53     if command_type == "general":
54         output = await self.handle_general_commands(target_client, command, arguments, user, guild)
55     elif command_type == "debug":
56         target_client.init_bot(bot)
57         output = await self.handle_debug_commands(target_client, command, arguments, user, guild)
58     elif command_type == "jira":
59         output = self.handle_jira_commands(target_client, command, arguments, user)
60     elif command_type == "event":
61         output = self.handle_event_commands(target_client, command, arguments, user, channel)
62
63     return output
```

# General Client

- Setup Befehl:
  - Erlaubt Erstellen eines Projektes in Firmenserver



# Jira Client

- REST-API zu Jira
- Funktionen:
  - Verbinden mit Datenbank
  - Setup Domain
  - Tickets abfragen (Pro Nutzer, Details)
  - Projekte abfragen (Pro Nutzer/ Domain)
  - Kommentare von Discord in Jira

# Termin Client

- Nutzt SQL Datenbank
- Funktionen:
  - Erstellen von Terminen
  - Bearbeiten von Terminen
  - Löschen von Terminen
  - Automatisches Abrufen
  - Erinnerungsfunktion

# Probleme bei der Umsetzung

- Voice Input:
  - Entscheidung zwischen Autovervollständigung und Spracheingabe
  - Spracheingabe: Nicht essentiell für Ziel des Projektes
  - Ist in separatem Branch abgespeichert

# Quellen

- <https://stackoverflow.com/>
- <https://nextcord.rtf.d.io/>
- <https://github.com/>
- <https://docs.sqlalchemy.org/>
- <https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro/>