

Cordic Discord Bot

Einleitung

- Discord: Online Chatplattform
- Jira: Projektverwaltung



Anforderungen

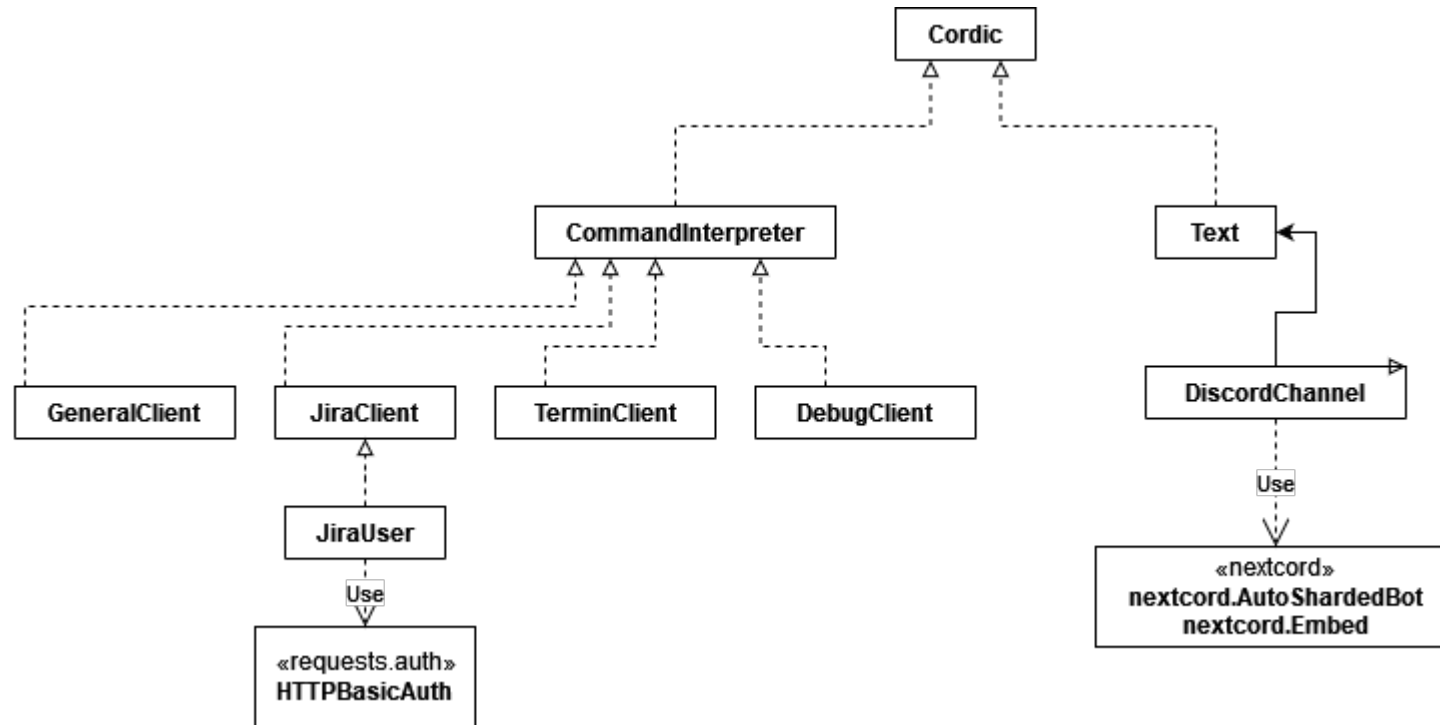
- Kommunikation an zentralisiertem Ort
- Jira Projektverwaltung mit Discord verbinden
- Termine verwalten (in Discord)

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

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/>