

Bot Aufsetzen:

Einen Bot erstellen welcher „MESSAGE CONTENT INTENT“ besitzt.

Einen Token erstellen und in der script.py den Token in der TOKEN-Variable speichern.

Den Bot mittels einer URL einladen, welche dem bot „Read Messages/View Channels“ und „Send Messages“ Rechte gibt.

Der Discord-Server benötigt für jeden Kurs, der korrigiert werden soll, einen eignen Kanal mit einem eindeutigen Namen z.B. für ST1 „ST1-Korrektur“ und für ST2 „ST2-Korrektur“.

Um eine Ausgabe von Links zu ermöglichen, benötigt der Bot für jeden Kanal, der genutzt wird, eine eigene Listen-Datei der CampusIDs und Links. Diese Datei muss denselben Namen wie der genutzte Kanal haben, im „.md“ Format sein und im gleichen Verzeichnis wie der Bot sein. Beispielweise sollte eine Datei mit Links für das Fach ST1 welche den Kanalnamen „ST1-Korrektur“ verwendet „ST1-Korrektur.md“ heißen.

Das Programm starten.

Funktionsweise des Bots:

Der Bot verwaltet zu korrigierende Studenten. Hierfür verwendet er eine Klasse Student, welche einen Studenten durch seine CampusID repräsentiert. Zusätzlich enthält die Klasse die Anzahl der Korrekturen und den Korrektur-Status (soll der Student momentan noch korrigiert werden). Diese Studentenobjekte werden in mehreren Listen eines Dictionaries verwaltet und in JSON-Dateien persistiert um die Daten bei Schließen des Programms (des Bots) nicht zu verlieren. Hieraus folgt auch, dass bei Start des Bots die Listen der Studenten aus den JSON-Dateien, falls vorhanden, geladen wird.

Der Bot nutzt die Commands Erweiterung der Discord Bibliothek, dies ermöglicht das einfache Implementieren neuer Befehle. Folgende Befehle sind in dem Bot implementiert: „!correct“, „!nickname“, „!next“, „!clearlist“.

Mit dem „!correct“ Befehl können Studenten sich in die Korrekturliste eintragen. Hierfür werden neue Studentenobjekte angelegt, wenn ein Student noch nicht in der Liste vorhanden ist. Sollte er schon einen Eintrag in der Liste haben, so wird sein Korrekturstatus auf „False“ gesetzt, um zu symbolisieren, dass er wieder korrigiert werden möchte. Hierbei wird auch die Reihenfolge der zu bearbeitenden Studenten beachtet. Die Priorisierung sieht wie folgt aus: 1. geringste Anzahl an Korrekturen 2. Reihenfolge der Einträge. Das heißt, dass zuerst alle Studenten mit 0 Korrekturen ausgegeben werden, bevor der erste Student eine Zweitkorrektur erhält, auch wenn sich dieser vor einem Studenten mit 0 Korrekturen angemeldet hat.

Mittels „!nickname“ kann man dem Kanal, in den man diesen Befehl schreibt, eine Abkürzung geben. Diese Abkürzung wird benötigt, um die Befehle „!next“ und „!clearlist“ zu verwenden.

Den „!next“ Befehl können nur Personen mit der „Staff“ Rolle ausführen. Dieser Befehl gibt den als nächstes zu korrigierenden Studenten mit seiner CampusID, den Links zu seinen Repos, der Anzahl der Korrekturen dieses Studenten und der Anzahl der noch zu korrigierenden Studenten aus. Außerdem wird dieser Student auf korrigiert gesetzt.

Der „!clearlist“ Befehl löscht die Liste aller Studentenobjekte und leert die JSON-Datei. Sollte eine neue Aufgabe zu korrigieren sein, so eignet sich dieser Befehl, um die Daten der letzten Korrektur zu löschen.

Bedienungsanleitung Bot:

Studenten:

„!correct“: Möchte ein Student sich zur Korrektur anmelden, so schreibt er in den passenden Kanal (für ST1 z.B. „ST1-Korrektur“) den Befehl „!correct“ gefolgt von seiner CampusID z.B. „!correct ghammers“.

Betreuer (nur Nutzer mit der Rolle „Staff“ können diese Befehle ausführen) :

„!nickname“: Dieser Befehl muss ausgeführt werden, bevor der „!next“ und „!clearlist“ Befehl genutzt werden kann. Es wird durch den Befehl dem Kanal, in dem dieser genutzt wird, eine Abkürzung zugeordnet, welche in dem folgenden beiden Befehlen verwendet werden. Möchte man z.B. den Kanal „ST1-Korrektur“ zur Korrektur von ST1 verwenden, so schreibt man in diesen Kanal den Befehl „!nickname st1“. Hierdurch kann man auf die in diesem Kanal erstellte Korrektur-Liste mit der Abkürzung „st1“ zugreifen.

„!next“: Um einen Studenten zu korrigieren gibt man diesen Befehl, gefolgt von der gewünschten Abkürzung ein. Hat man beispielsweise eine Abkürzung „st1“ wie im in der oben genannten Erklärung erstellt, so kann man einen Studenten des Kurses ST1 durch den Befehl „!next st1“ korrigieren.

„!clearlist“: Wenn alle Studenten korrigiert wurden, kann man alle Daten eines Kurses durch diesen Befehl löschen. Hierfür ist auch eine Abkürzung nötig, wodurch ein Befehl z.B. „!clearlist st1“ lauten könnte.