МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

Факультет информационных технологий

Кафедра «Инфокогнитивные технологии»

**КУРСОВОЙ ПРОЕКТ**

на тему: *«Создание бота на Python для Discord».*

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль «Корпоративные информационные системы»

**Выполнил:**

студент группы 211-362

Корнеев Никита Владимирович

|  |  |
| --- | --- |
| 26.06.2023 |  |
|  | (подпись) |

Москва 2023

**СОДЕРЖАНИЕ**

[1 Основные сведения 3](#_Toc138843413)

[1.1 Цель проекта 3](#_Toc138843414)

[1.2 Обоснование актуальности задачи 3](#_Toc138843415)

[1.3 Описание функциональности приложения 4](#_Toc138843416)

[1.4 Запуск проекта 5](#_Toc138843417)

[2 Общие принципы работы бота и функционал 6](#_Toc138843418)

[2.1 Общие принципы работы 6](#_Toc138843419)

[2.2 Функционал 7](#_Toc138843420)

[3 Работа с api 11](#_Toc138843421)

[4 Команды дискорд бота 13](#_Toc138843422)

[5 События в дискорд 15](#_Toc138843423)

[6 Ролевая модель дискорда 17](#_Toc138843424)

[Заключение 19](#_Toc138843425)

# Основные сведения

## Цель проекта

Целью данного проекта разработка Discord бота для взаимодействия с пользователем. При его разработки должны учитываться особенности внутренней и внешней ролевой системы, должны использоваться внешние библиотеки и учитываться работа с API.

## Обоснование актуальности задачи

Актуальность задачи разработки дискорд-бота является неоспоримой, поскольку платформа Discord стала одним из самых популярных инструментов для общения и сотрудничества в сообществах, игровых гильдиях, учебных заведениях и многих других областях. Распространенность Discord продолжает расти, и она предлагает различные возможности для создания и поддержки ботов.

Вот несколько основных аргументов, подтверждающих актуальность разработки дискорд-бота:

Улучшение управления сообществом: Дискорд-боты предоставляют разнообразные функции, которые помогают управлять сообществом. Они могут автоматизировать административные задачи, предоставлять информацию о правилах и полезные ссылки, управлять ролями и разрешениями пользователей, а также модерировать чаты. Боты облегчают работу администраторов, помогая им поддерживать порядок и обеспечивать гармоничное взаимодействие в сообществе.

Игровые возможности: Discord широко используется игровыми сообществами, и дискорд-боты могут значительно расширить игровой опыт пользователей. Они могут предоставлять информацию о игровых статистиках, помогать в организации мероприятий, предлагать игровые рекомендации, уведомлять о новостях и обновлениях, а также создавать уникальные игровые функции и мини-игры. Боты способны обогатить и усилить взаимодействие между игроками внутри сообществ.

Учебные и информационные ресурсы: Дискорд-боты могут быть полезными инструментами для образовательных учреждений и других организаций, предоставляющих информацию и ресурсы. Они могут отвечать на вопросы пользователей, предлагать образовательный контент, проводить тестирование и оценку знаний, а также предоставлять учебные материалы и ссылки.

## Описание функциональности приложения

Бот предоставляет следующую функциональность:

1. Автоматическая выдача роли новому пользователю.
2. Занесения нового префикса в JSON файл.
3. Удаление префикса из JSON файла.
4. Проверка на уникальность префикса.
5. Изменение префикса в JSON файле.
6. Команда для транслитерации текста.
7. Команда для поиска видео на ютуб.
8. Комана для поиска людей с определенной ролью в сети.
9. Команда для создания голосования.
10. Команда для назначения временной роли.
11. Команда для обращения к чату OpenIA.
12. Команда для вывода технических характеристик ПК.
13. Временные голосовые чаты.
14. Команда для системы тикетов.
15. Команда для вывода префикса сервера.
16. Вывод кнопки с переходом на ролевую систему.
17. Команда для вывода правил.
18. Команда для вызова учащихся.
19. Команда для выведения id роли.
20. Создания функционального ограничения для работы с командами.
21. Вывод всех пользователей на сервере.
22. Вывод информации про команды.
23. Авто-модератор по мату.
24. Вывод в консоль информации о состоянии бота.
25. Таймер.
    1. **Запуск проекта**

Для запуска проекта требуется запустить код и после чего дождаться пока в консоль не высветиться надпись «Pbot включен». Для запуска кода потребуется ПО «VisualStudioCode», Python последней версии, а также установить все библиотеки, использующиеся в проекте, а именно:

1. Discord.
2. Discord.ui.
3. Discord.ext.
4. Googletrans.
5. Pytils.
6. Datetime.
7. Googleapiclient.discovery.
8. Asyncio.
9. Openia.
10. JSON.

# Общие принципы работы бота и функционал

* 1. **Общие принципы работы**

Общие принципы работы дискорд-бота определяют основные принципы его функционирования и взаимодействия с пользователями. Вот несколько ключевых принципов:

Авторизация и доступ: для работы бота на платформе Discord необходимо получить авторизацию от пользователя или разрешение от администратора сервера. Бот должен иметь правильные разрешения и роли для выполнения требуемых функций, таких как отправка сообщений, управление каналами или модерация чата.

Обработка команд: бот должен быть способен обрабатывать команды, введенные пользователями. Это может быть символ или ключевое слово, сопровождаемое определенной командой или запросом. Бот анализирует ввод пользователя и выполняет соответствующие действия, например, отвечает на вопросы, выполняет поиск информации.

Реакции на события: бот может быть настроен на реагирование на определенные события, происходящие на сервере Discord. Это могут быть новые сообщения, присоединение или выход участника, изменение настроек сервера. Бот может отслеживать такие события и выполнять действия в соответствии с ними, например, приветствовать новых участников, отслеживать активность пользователей или автоматически реагировать на определенные ключевые слова.

Взаимодействие с API и сервисами: бот может использовать API (интерфейсы программирования приложений) для взаимодействия с другими сервисами и получения дополнительных данных. Интеграция с API расширяет функциональность бота и позволяет ему предоставлять более полезную информацию и услуги.

Хранение данных: бот может сохранять и использовать данные для выполнения различных задач.

Общие принципы работы дискорд-бота обеспечивают его функциональность, взаимодействие с пользователями и интеграцию с другими сервисами, что позволяет создать полезного и интерактивного помощника в сообществе Discord.

* 1. **Функционал**

К функционалу данного бота относятся ряд функций, которые помогают при работе с дискордом.

* + 1. **Автоматическая выдача роли новому пользователю**

Для автоматической выдачи роли требуется отслеживание того зашел ли новый человек на сервер это делается путем взаимодействия с пользователем. Для реализации используется «ивент» который отслеживает зашел ли кто-то на сервер. Если зашел новый пользователь, то этому пользователю присваивается ранее заданная роль.

* + 1. **Занесение префикса в JSON файл**

Для этого требуется отслеживать если бот на сервере если есть, то в ранее созданный JSON файл сохраняется id сервера и его префикс по стандарту это «!».

* + 1. **Удаление префикса из JSON файла**

Если бот покидает сервер, то он удаляет всю информацию по тому серверу, с которого он вышел, это происходит путем сравнения статического id.

* + 1. **Проверка на уникальность префикса**

Данная проверка требуется во избежание путаницы в префиксах на нескольких серверах. Проверка происходит по статическому id сервера.

* + 1. **Изменение префикса в JSON файле**

Для изменения префикса используется слеэш-команды. Замена префиска происходит следующим образом. Человек использует команду и его просят написать желаемый префикс. Далее введенная конструкция записывается в JSON при этом проверяет от какого сервера был создан запрос на смену префикса.

* + 1. **Транслитерация**

Для транслитерации текста используется специальная команда, которая на вход получает текст или слово, после чего начинает его переводить, после этого отправляет переведенный текст. Пример работы.

Исходный текст: ghbdtn rfr ltkf.

Результат: привет как дела.

* + 1. **Поиск видео**

Для поиска видео была реализована команда, которая путем подключения к API ютуба создает поисковой запрос после чего вам в личные сообщения отправляет ссылки на те видео, которые были найдены первые пять штук.

* + 1. **Поиск человека**

Данная команда позволяет найти людей с определенной ролью с статусом «в сети». После ввода команды вы должны указать людей с какой ролью вы ищите и после вам в личные сообщения отправят тэг данных людей.

* + 1. **Команда для создания голосования**

Для создания голосования требуется ввести команду, далее ввести заголовок и до 10 вариантов ответа. После чего бот отправит сообщение и добавит 10 «реакций» после нажатие на 1 из них будет добавлена строка ваш ответ «содержание ответа».

* + 1. **Временная роль**

Данная функция требуется для того, чтобы выдать определенному человеку определенную роль на определенное время. При вводе команды вас попросят написать 3 параметра кому выдать роль, какую выдать роль и на сколько выдать роль (время в секундах). После чего роль будет выдана человека на определенное время и будет оповещено об этом в чате. Далее о прохождение указанного времени роль с человека будет снята и также об этом будет оповещено в чате.

* + 1. **Чат GPT**

Для работы данной команды требуется API от OpenIA. При использовании команды введённый текст передается в чат GPT и после полученный ответ выдается в виде сообщения в чат от бота.

* + 1. **Проверка технического ПО**

Данная команда помогает проверить какое ПО находиться в том или ином ПК. Вывод команды содержит себе информацию о железе ПК.

* + 1. **Временные голосовые каналы**

Эта функция позволяет, зайдя в ранее созданный канал с определенным названием создать временный канал. После того как вы попадаете в такой канал создаётся временный канал и вас перемещает туда. После того как вы покинете этот канал он удалится.

* + 1. **Тикеты**

Данная команда создает текстовый чат, к которому есть доступ у вас и у администратора. После чего в данном чате появляется кнопка, которая по решению вас или администратора закрое его.

* + 1. **Вывод информации о префиксе**

Данная команда получает префикс из JSON файла после чего отправляет его в чат.

* + 1. **Ролевая кнопка**

Данная команда выводит в чат кнопку при нажатии на которую вас переносит на доску Miro c ролевой системой.

* + 1. **Правила**

Данная команда отправляет в чат текстовое сообщение с правилами далее после нажатия на кнопку согласен с правилами вам выдается советующая роль.

* + 1. **Вызов**

Данная команда предназначена для тэга студентов по определенной роли. При попытке тегнуть иную роль выдаст ошибку.

* + 1. **ID роли**

Данная команда нужна для вывода статического id роли.

* + 1. **Ролевое ограничение**

Это событие, которое указывает какой ролью должен обладать человек чтобы воспользоваться той или иной командой при попытки воспользоваться данной командой человеку без данной роли выдаст ошибку.

* + 1. **Пользователи**

Данная команда помогает показать всех пользователей на сервере написав их имена.

* + 1. **Команды**

Данная команда выводит текст с информацией про все команды.

* + 1. **Авто-мод**

Данное событие автоматически удаляет мат и предупреждает его.

* + 1. **Таймер**

Данная команда работает как обычный таймер. Бот вас тэгает через указанное вами время.

# Работа с api

Работа с API (интерфейсами программирования приложений) является важной составляющей функциональности дискорд-бота. Взаимодействие с API позволяет боту получать данные, отправлять запросы к внешним сервисам и интегрироваться с различными онлайн-ресурсами. Вот несколько ключевых аспектов работы с API для дискорд-бота.

Интеграция с внешними сервисами: Дискорд-бот может использовать API для взаимодействия с различными внешними сервисами и получения дополнительной информации. Например, бот может использовать API погоды для предоставления текущей погоды или API новостей для получения последних новостей. Интеграция с API расширяет возможности бота и позволяет ему предоставлять более разнообразные и полезные функции.

Получение данных: бот может отправлять запросы к API, чтобы получить определенные данные. Это может быть информация о пользователе, статистика игрового профиля, результаты поиска или любая другая информация, которая может быть полезной для пользователей бота. Бот анализирует ответы от API и может использовать эти данные для отображения информации или выполнения дополнительных действий.

Отправка данных: в некоторых случаях бот может отправлять данные к API для выполнения определенных действий или взаимодействия с другими сервисами. Например, бот может отправлять запросы для публикации сообщений в социальных сетях, отправки электронных писем или выполнения других действий на внешних платформах. Это позволяет боту интегрироваться и сотрудничать с другими онлайн-сервисами.

Обработка ответов и ошибок: бот должен быть способен обработать ответы от API и адекватно реагировать на ошибки или некорректные данные. Надежная обработка ответов и ошибок позволяет боту предоставлять точные и полезные данные пользователям, а также избегать возможных проблем, связанных с некорректными запросами или недоступностью API.

Аутентификация и безопасность: при работе с API боту может потребоваться аутентификация для доступа к определенным данным или функциям. Бот должен иметь соответствующие ключи API или токены, а также обеспечивать безопасность передаваемых данных. Реализация безопасности и правильная обработка авторизации позволяют боту работать с API безопасно и защищать конфиденциальность данных.

Работа с API является важной частью разработки дискорд-бота, поскольку она позволяет расширить его возможности, интегрироваться с другими сервисами и предоставлять более полезные функции для пользователей.

# Команды дискорд бота

Команды дискорд-бота представляют собой специальные команды, которые пользователи могут вводить для взаимодействия с ботом и использования его функциональности. Вот несколько ключевых аспектов команд дискорд-бота:

Синтаксис команд: каждая команда имеет свой уникальный синтаксис, который определяет, как пользователь должен ввести команду для ее выполнения. Синтаксис может включать символы префикса или ключевые слова, которые идентифицируют команду, а также аргументы, опции или флаги, которые могут быть добавлены для настройки или расширения функциональности команды.

Обработка команд: дискорд-бот должен быть способен обрабатывать введенные пользователем команды. Это включает анализ команды, извлечение аргументов, опций или флагов, а также выполнение соответствующих действий или вызов соответствующих функций. Бот может использовать различные алгоритмы обработки команд, включая регулярные выражения или разбор строки, чтобы правильно распознать и выполнить команду.

Отображение результата: после выполнения команды бот может отобразить результат пользователю. Это может быть текстовый ответ, изображение, вложение или любая другая форма информации, в зависимости от типа команды и ее функциональности. Бот должен предоставлять понятные и информативные ответы, чтобы пользователи могли правильно взаимодействовать с ним и получить требуемую информацию или результаты.

Помощь и справка: дискорд-бот может предоставлять функционал помощи и справки по командам. Это включает возможность отображения списка доступных команд, описания и примеров использования каждой команды, а также предоставление дополнительной информации и руководств по использованию бота. Пользователи могут использовать команды помощи для получения дополнительной информации и разъяснений по функциональности бота.

Настройка и расширение: дискорд-бот может предоставлять возможности настройки и расширения команд. Это включает возможность пользователей настраивать опции, параметры или поведение команды в соответствии со своими предпочтениями или потребностями. Бот может также поддерживать плагины или расширения, которые добавляют новые команды или функциональность в боте, расширяя его возможности и адаптируясь к различным потребностям сообщества.

Команды дискорд-бота позволяют пользователям взаимодействовать с ботом и использовать его функциональность для получения информации, выполнения задач или развлечения.

# События в дискорд

События в Discord для бота представляют собой определенные ситуации или действия, которые могут произойти в сообществе Discord и на которые бот может отреагировать. Понимание и обработка событий является важным аспектом разработки дискорд-бота. Вот несколько ключевых аспектов работы с событиями в дискорд-боте:

Регистрация событий: дискорд-бот должен зарегистрировать интересующие его события для их обработки. Это может быть событие, связанное с отправкой сообщения, присоединением или выходом пользователя, изменением состояния членства на сервере и другие. Бот использует API Discord для регистрации событий и установки обработчиков для каждого события.

Обработка событий: когда происходит зарегистрированное событие, дискорд-бот вызывает соответствующий обработчик для этого события. Обработчик выполняет необходимые действия в ответ на событие, такие как отправка сообщения, изменение ролей, запись в логи и другие операции. Обработка событий позволяет боту реагировать на действия пользователей или изменения в сообществе.

Извлечение данных: во время обработки событий бот может извлекать необходимые данные из события. Например, при обработке события отправки сообщения бот может получить текст сообщения, автора сообщения или данные о канале, в котором сообщение было отправлено. Эти данные могут быть использованы ботом для принятия решений или выполнения определенных действий.

Управление состоянием: события также могут использоваться для управления состоянием бота. Например, при событии присоединения нового пользователя бот может добавить его в список пользователей или приветствовать его в текстовом канале. События могут быть использованы для обновления данных, сохранения состояния и обеспечения согласованности бота с сообществом.

Обработка ошибок: бот должен быть способен обрабатывать возможные ошибки или исключительные ситуации при работе с событиями. Это может включать проверку прав доступа, обработку ошибок сети или обработку некорректных данных, полученных из события. Бот должен быть устойчивым к ошибкам и обеспечивать понятные сообщения об ошибках для пользователя, если что-то пошло не так.

Работа с событиями в Discord позволяет дискорд-боту активно взаимодействовать с сообществом и отвечать на действия пользователей или изменения в сервере. Это дает боту возможность быть более отзывчивым и функциональным в рамках сообщества Discord.

# Ролевая модель дискорда

Ролевая модель в Discord представляет собой систему, позволяющую управлять привилегиями, правами доступа и иерархией пользователей на сервере. Роли играют важную роль в организации и управлении сообществом. Вот несколько ключевых аспектов ролевой модели Discord.

Создание и назначение ролей: Администраторы сервера могут создавать различные роли и назначать их пользователям. Каждая роль имеет свое имя, цвет, права доступа и позицию в иерархии. Роли могут быть назначены отдельным пользователям или группам пользователей.

Иерархия ролей: роли могут быть упорядочены в иерархическом порядке, где одна роль может иметь приоритет над другой. Это позволяет определить и управлять уровнем доступа и привилегиями пользователей на основе их ролей. Например, роль «Администратор» может иметь больше прав, чем роль «Модератор».

Права доступа: роли могут иметь различные права доступа к функциям и возможностям сервера. Например, роль «Администратор» может иметь полный доступ ко всем функциям сервера, включая управление каналами, бан пользователей, в то время как роль «Участник» может иметь ограниченные права.

Наследование прав: Роли могут наследовать права доступа от других ролей. Например, если у пользователя есть несколько ролей, он будет иметь все права, назначенные каждой из этих ролей. Это позволяет гибко управлять правами пользователей, исключая необходимость повторного назначения прав для каждой роли.

Кастомизация ролей: Роли могут быть настроены с помощью различных параметров, таких как цвет, разрешения каналов, возможность отправлять сообщения, использовать определенные эмодзи и другие. Это позволяет администраторам сервера настроить роли в соответствии со специфическими потребностями сообщества.

Управление ролями: Администраторы сервера имеют возможность управлять ролями, включая создание, удаление, редактирование и назначение ролей пользователям. Это позволяет поддерживать актуальную и сбалансированную ролевую модель в сообществе.

Ролевая модель Discord обеспечивает эффективное управление пользователями и привилегиями на сервере, позволяя создавать иерархию, устанавливать права доступа и настраивать роли в соответствии с потребностями сообщества. Это способствует организации и сотрудничеству внутри сообщества Discord.

# Заключение

В заключение, создание и разработка дискорд-бота предоставляет уникальные возможности для автоматизации и улучшения опыта пользователей в сообществе Discord. Боты могут выполнять разнообразные задачи, начиная от предоставления информации и управления ролями, до взаимодействия с API и обработки событий.

Важно понимать, что разработка дискорд-бота требует хорошего понимания основных принципов работы Discord API, а также навыков программирования. Однако, благодаря обширной документации и ресурсам, доступным для разработчиков, процесс создания бота становится более доступным.

Использование различных функций и возможностей Discord API, таких как команды, события, работа с API и ролевая модель, позволяет создать мощного и гибкого бота, способного адаптироваться к потребностям сообщества и предоставлять удобный интерфейс взаимодействия для пользователей.

Наконец, хранение данных в JSON файле обеспечивает простой и структурированный способ сохранения информации, который может быть использован для хранения настроек, конфигураций, пользовательских данных и других важных данных, необходимых для работы бота.

В целом, разработка дискорд-бота с использованием Discord API и JSON файлов предоставляет возможность создания уникального и полезного инструмента для вашего сообщества Discord, расширяя его функциональность, автоматизируя рутинные задачи и создавая более интерактивное и удобное взаимодействие для пользователей.