

При нажатии на реакцию бот проверяет id сообщения и нажатую реакцию:

```
if payload.member.bot:
    return
if payload.message_id == message_id:
    if payload.emoji.name == '👉':
        emoji = "👉"
    elif payload.emoji.name == '🌟':
        emoji = "🌟"
    elif payload.emoji.name == '📄':
        emoji = "📄"
    elif payload.emoji.name == '👉':
        emoji = "👉"
    elif payload.emoji.name == '🌈':
        emoji = "🌈"
    elif payload.emoji.name == '🌀':
        emoji = "🌀"
    elif payload.emoji.name == '🐙':
        emoji = "🐙"
    elif payload.emoji.name == '🐰':
        emoji = "🐰"
    elif payload.emoji.name == '🐱':
        emoji = "🐱"
```

Если id сообщения и реакция соответствует ожидаемым, он получает с сервера дискорд отображаемое на сервере имя и сохраняет его в переменной old_name:

```
for guild in bot.guilds:
    if guild.id == guild_id:
        for member in guild.members:
            if member.id == payload.member.id:
                old_name = member.display_name
```

Далее переименовывает пользователя, добавляет в начале и конец ника выбранную эмодзи, сохраняет первоначальный ник (old_name) в базе данных:

```
await payload.member.edit(nick=f"{emoji} {old_name} {emoji}")
sql_update_query = """INSERT INTO users VALUES (?, ?)"""
data = (payload.member.id, old_name)
cursor.execute(sql_update_query, data)
conn.commit()
```

По команде возврата ников бот загружает из базы данных все первоначальные ники и присваивает их пользователям, которые нажимали на эмодзи:



```
if message.content == "!rename":
    users = cursor.execute(f"SELECT * FROM users").fetchall()
    for user in users:
        user_id = user[0]
        user_old_name = user[1]
        for guild in bot.guilds:
            if guild.id == guild_id:
                for member in guild.members:
                    if member.id == user_id:
                        try:
                            await member.edit(nick=user_old_name)
                            print(f"Переименован {member.name} в {user_old_name}")
                        except Exception as ecx:
                            print(ecx)
```









Соответственно если пользователь не нажимал на эмодзи в нужном сообщении и его ник ботом не менялся, он не попадет в базу данных.

Если после изменения ника ботом он ещё раз будет изменён другим способом (например, его поменял админ или другой бот), бот про это ничего не узнает и не запишет в свою базу данных. Бот не получает данных о действиях администраторов и ботов. Он вернет ник в тот вид, который был при первоначальной его смене.

Например, если у пользователя был ник **Волчок**, бот при нажатии на эмодзи меняет ему ник на **🌟 Волчок 🌟** и сохраняет в своей базе данных первоначальный ник. Если далее кто-то поменяет ник другим способом, например на **🌟👁👁👁 пошлых лорну 🌟**, в базу данных бота эта запись не попадёт. Бот вернёт тот ник, о котором знал: **Волчок**

За 40 минут написания бота было невозможно запроектировать запись в БД учета всех изменений ников, всех действий администраторов, связи с другими ботами.

Всего сменой ника воспользовалось 38 человек. Как видно по логам во время нажатия на эмодзи у Freedom был ник вида « Freedom ».

Вероятнее всего это получилось из-за наложения работы двух ботов. Между первым отключением бота по переименованию и включением 1 апрельского бота прошло слишком мало времени и первоначальный ник у Freedom не успел вернуться. Потом первоапрельский бот сначала переименовал из « Freedom » в МАКСВЕЛЛ-22, потом обратно в « Freedom », потом запустился бот по переименованию, который при нажатии на эмодзи добавил к « Freedom » новые значки, а после отмены смена ника вернул соответственно к « Freedom ».