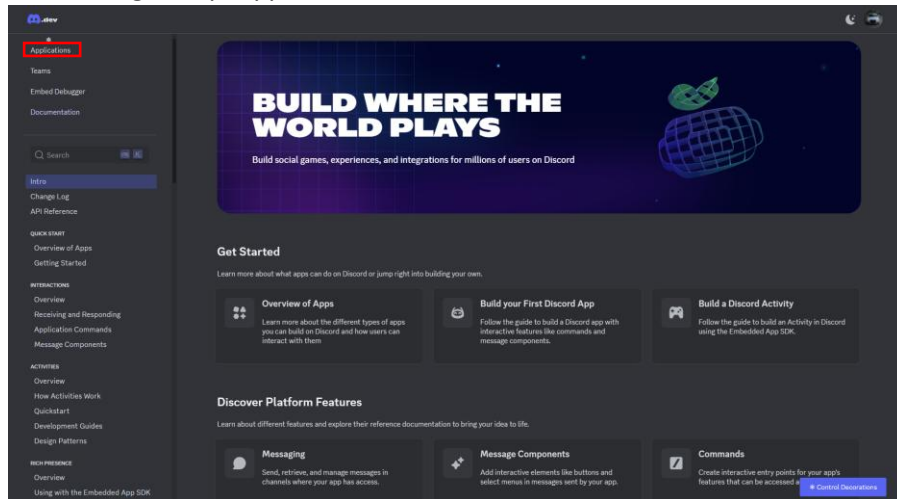


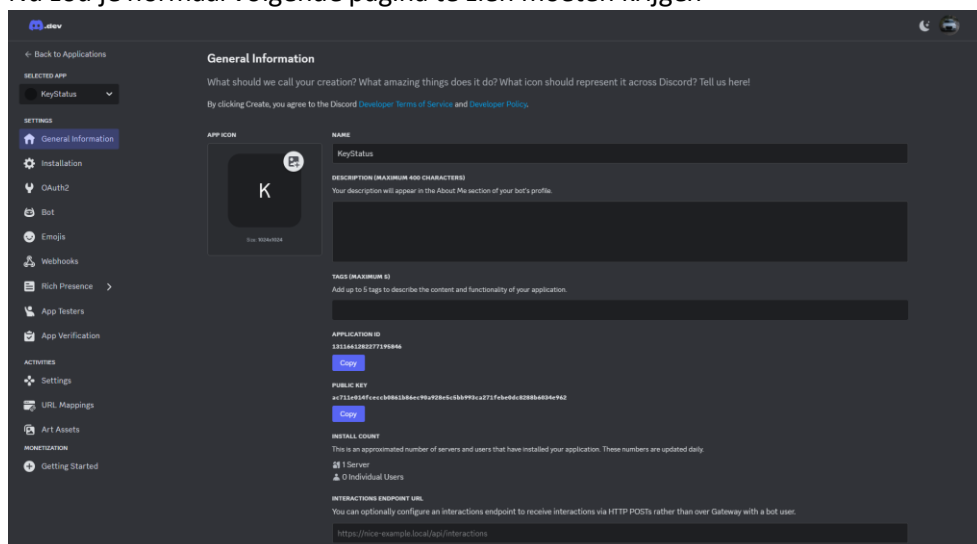
- **Discord bot setup**

- Ga naar het Discord Development Portal door in te geven in browser of onderstaande link te gebruiken
 - <https://discord.com/developers/docs/intro>
- Klik vervolgens op 'Application' in de linkerbovenhoek

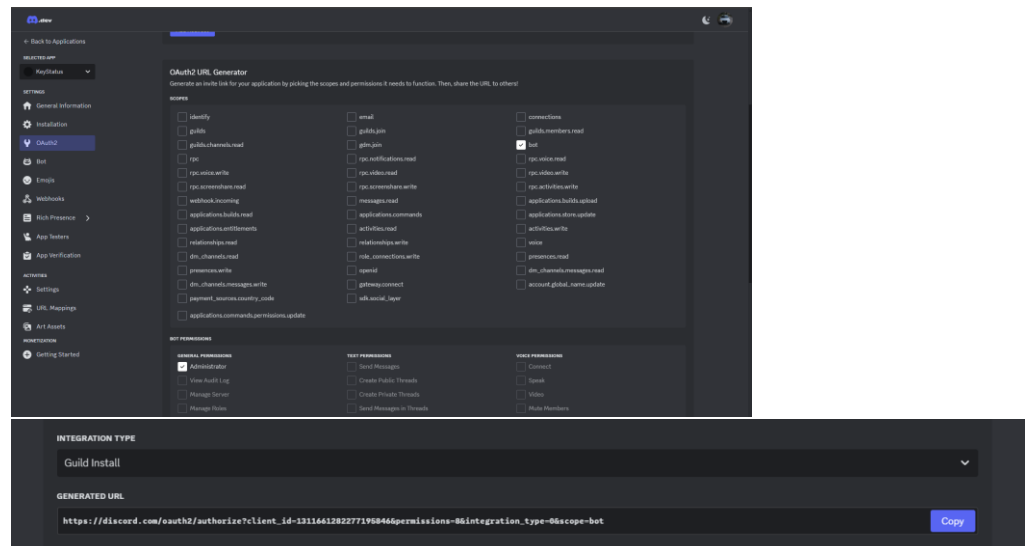


- Klik daarna op 'New Application' de knop is te vinden in de rechter bovenhoek van deze pagina
- Geef de nieuwe applicatie een naam en klik 'create'

- Nu zou je normaal volgende pagina te zien moeten krijgen



- Ga vervolgens naar de tab 'bot' en klik op de knop 'add bot'
- In de tab 'bot' kan je vervolgens een token genereren. Of op de knop 'Click to reveal token' te klikken
 - Belangrijk om deze token te bewaren wat je kan deze later niet meer terugkrijgen zonder deze opnieuw te moeten genereren.
- Onder 'OAuth2' kan je alle nodige instellingen van de bot aanpassen
 - In dit voorbeeld vinken we de checkbox van bot en administrator aan



- Kopier vervolgens bovenstaande link en plak deze in een nieuw tabblad
- Er zal gevraagd worden aan welke server de bot toegevoegd moet worden
- Klik op continue en op de volgende pop up vink administrator aan als dit nog niet gebeurd is, klik dan op authorize. Als je nu naar de discord server gaat zul je zien dat de bot in de server staat

- **Bot functionaliteiten geven met python**

- Voor volgende pip install commando uit om de nodige library's te installeren
 - `Pip install discord.py`
- Maak een .env file aan, hier gaan we alle credentials die nodig zijn in bewaren
 - In de .env file maak
 - `DISCORD_TOKEN=<jouw discord token van hierboven>`
- Importeer ook volgende library's
 - `import discord`
 - `from discord.ext import commands`
 - `from dotenv import load_dotenv`
- Maak een python bestand aan en plaats volgende lijnen om het bot object aan te maken
 - `intents = discord.Intents.default()`
 - `intents.message_content = True`
 - `client = commands.Bot(command_prefix = "/", intents=intents)`
- `intents` = welke permissies de bot mag gebruiken
- `command_prefix` = opgeven met welk teken het commando waarop gereageerd zal worden moet beginnen

```

@client.event
async def on_ready():
    print('The bot is now ready for use!')
    print('-----')

@client.group()
async def key(ctx):
    if ctx.invoked_subcommand is None and ctx.channel.name == "sleutel":
        await ctx.send("Please use '/key take' or '/key return'.")

@key.command(name="take")
async def key_take(ctx):
    if ctx.channel.name == "sleutel":
        create_payload(returned=True)
        await ctx.send("Key taken")

@key.command(name="return")
async def key_return(ctx):
    if ctx.channel.name == "sleutel":
        create_payload(returned=False)
        await ctx.send("Key returned")

# Run the Discord client
client.run(TOKEN)

```

-
- Uitleg bovenstaande code
 - @client.event (on_ready:)
 - Dit is een event dat word getriggerd zodra de bot klaar is om te werken
 - @client.group (key)
 - Dit definieert een commando-groep genaamd key
 - In dit geval bevat de groep de subcommando's key take en key return
 - @key.command(name="return") en @key.command(name="take")
 - Deze 2 subcommandos worden geactiveerd naargelang er /key return of /key take word gebruikt
 - Client.run(TOKEN)
 - Start de bot en verbindt deze met Discord
 - Token is de unieke sleutel voor authenticatie van de bot (staat in de .env file)