



Demo Helper

Reference Manual

for Demo Helper v2.0

<https://github.com/Amheus/DemoHelper>
support@samlewis.dev

Samuel Lewis
v1
last updated July 31, 2021

Contents

1	Introduction	2
2	Hosting Requirements	3
2.1	Prerequisites	3
2.1.1	Creating The Bot Instance	3
2.1.2	The .env File	5
2.1.3	The newrelic.ini File	5
2.2	Getting Demo Helper Onto Your Server	7
2.3	Running Demo Helper As A Service	8
2.3.1	Linux - Systemd	8
2.3.2	Linux - PM2	9
2.4	General Maintenance	10

1 Introduction

This documentation aims to provide a detailed understanding on how to use DemoHelper.

The version this documentation will detail is Amheus' fork of AberDiscordBotsTeam's demoHelperBot.

DemoHelper stores a queue of students waiting for demonstrators to aid them in their work.

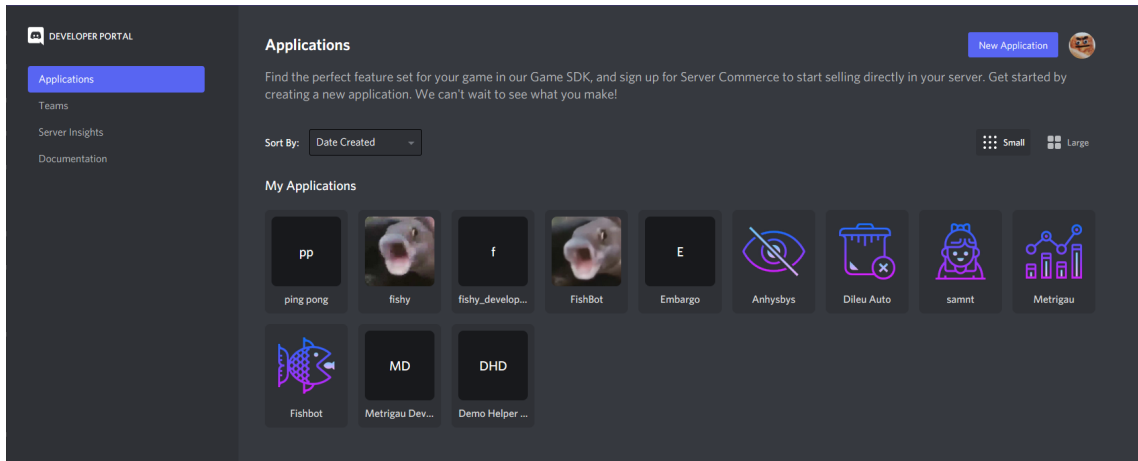
2 Hosting Requirements

This section is applicable if you want to host DemoBot on your own server.

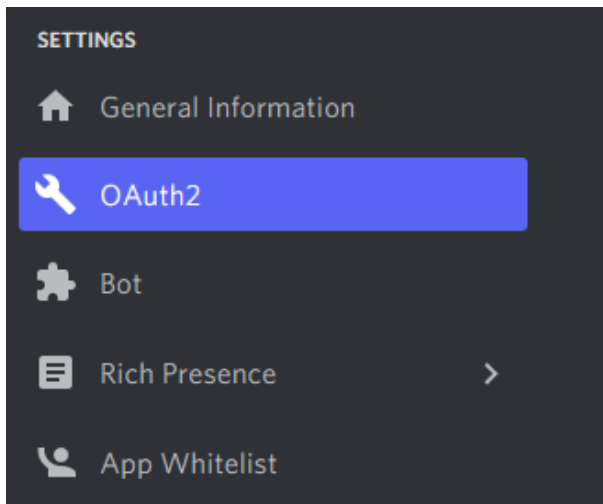
2.1 Prerequisites

2.1.1 Creating The Bot Instance

1. Head over to <https://discord.com/developers/applications>
2. Click on the button labelled ‘New Application’ which is located at the top right.



3. Give it a name. For example, *DemoHelper*.
4. Head over to the menu titled *OAuth2* as seen below:



5. Select the checkboxes as seen below:

OAuth2 URL Generator

Well, this doesn't do any good just sitting here. Pick the scopes your application needs to function and copy the URL below. Others can use that URL to authorize with your application.

SELECT REDIRECT URL

Select...

SCOPES

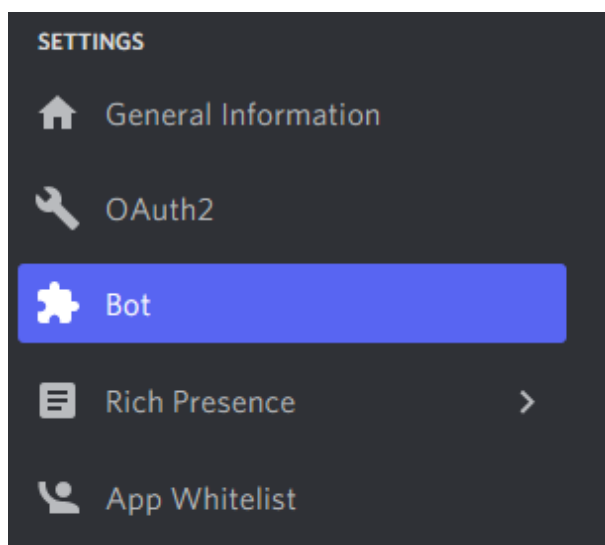
<input type="checkbox"/> identify	<input type="checkbox"/> rpc.voice.read	<input type="checkbox"/> applications.builds.read
<input type="checkbox"/> email	<input type="checkbox"/> rpc.voice.write	<input checked="" type="checkbox"/> applications.commands
<input type="checkbox"/> connections	<input type="checkbox"/> rpc.activities.write	<input type="checkbox"/> applications.store.update
<input type="checkbox"/> guilds	<input checked="" type="checkbox"/> bot	<input type="checkbox"/> applications.entitlements
<input type="checkbox"/> guilds.join	<input type="checkbox"/> webhook.incoming	<input type="checkbox"/> activities.read
<input type="checkbox"/> gdm.join	<input type="checkbox"/> messages.read	<input type="checkbox"/> activities.write
<input type="checkbox"/> rpc	<input type="checkbox"/> applications.builds.upload	<input type="checkbox"/> relationships.read
<input type="checkbox"/> rpc.notifications.read		

https://discord.com/api/oauth2/authorize?client_id=870284459419783208&permissions=2164272128&scope=bot%20applications.commands [Copy](#)

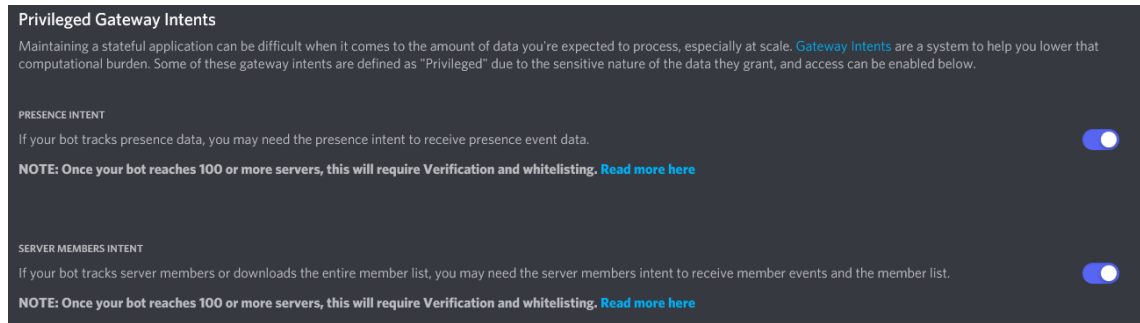
BOT PERMISSIONS

GENERAL PERMISSIONS <ul style="list-style-type: none"> <input type="checkbox"/> Administrator <input type="checkbox"/> View Audit Log <input type="checkbox"/> View Server Insights <input type="checkbox"/> Manage Server <input type="checkbox"/> Manage Roles <input type="checkbox"/> Manage Channels <input type="checkbox"/> Kick Members <input type="checkbox"/> Ban Members <input type="checkbox"/> Create Instant Invite <input type="checkbox"/> Change Nickname <input type="checkbox"/> Manage Nicknames <input type="checkbox"/> Manage Emojis and Stickers <input type="checkbox"/> Manage Webhooks <input checked="" type="checkbox"/> View Channels 	TEXT PERMISSIONS <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Send Messages <input type="checkbox"/> Use Public Threads <input type="checkbox"/> Use Private Threads <input type="checkbox"/> Send TTS Messages <input checked="" type="checkbox"/> Manage Messages <input type="checkbox"/> Manage Threads <input type="checkbox"/> Embed Links <input type="checkbox"/> Attach Files <input type="checkbox"/> Read Message History <input type="checkbox"/> Mention Everyone <input type="checkbox"/> Use External Emojis <input type="checkbox"/> Use External Stickers <input type="checkbox"/> Add Reactions <input checked="" type="checkbox"/> Use Slash Commands 	VOICE PERMISSIONS <ul style="list-style-type: none"> <input type="checkbox"/> Connect <input type="checkbox"/> Speak <input type="checkbox"/> Video <input type="checkbox"/> Mute Members <input type="checkbox"/> Deafen Members <input checked="" type="checkbox"/> Move Members <input type="checkbox"/> Use Voice Activity <input type="checkbox"/> Priority Speaker
--	---	---

- Copy the generated URL and paste it into the .env file (see section 2.1.2) as the value for key *INVITE_URL*.
- Head over to the menu titled *Bot* as seen below:



8. Click the button labelled *AddBot*, then click *Yes, doit!*.
9. Click the button labelled *Copy* which is located under *ClicktoRevealToken*. Paste the contents into the *.env* file (see section 2.1.2) as the value for the key *DISCORD_TOKEN*.
10. Finally, navigate to the section titled *PrivilegedGatewayIntents* and check both intents as seen below:



2.1.2 The .env File

This is a critical file that doesn't come with the git repo. Its purpose is to contain environment variables such as the Discord Token which are sensitive therefore MUST NOT BE SHARED. A template can be seen below:

```
DISCORD_TOKEN="<<your-token-here>>"
INVITE_URL="<<your-invite-url-here>>"
```

2.1.3 The newrelic.ini File

Demo Helper utilises New Relic to monitor the status of Demo Helper. This isn't a requirement however and if you wanted to, you could remove the New Relic code and it won't impede Demo Helper performance. However, if you want New Relic monitoring, then you will need a configuration file in the same place as the *.env* file. A template can be seen below:

```
[newrelic]
license_key = <<your-license-key-here>>
app_name = Demo Helper
distributed_tracing.enabled = true
monitor_mode = true
log_level = info
ssl = true
high_security = false
transaction_tracer.enabled = true
transaction_tracer.transaction_threshold = apdex_f
transaction_tracer.record_sql = obfuscated
transaction_tracer.stack_trace_threshold = 0.5
transaction_tracer.explain_enabled = true
transaction_tracer.explain_threshold = 0.5
error_collector.enabled = true
```

```
browser_monitoring.auto_instrument = true
thread_profiler.enabled = true
```

```
# -----
```

```
[newrelic:development]
monitor_mode = false
```

```
[newrelic:test]
monitor_mode = false
```

```
[newrelic:staging]
app_name = Demo Helper (Staging)
monitor_mode = true
```

```
[newrelic:production]
monitor_mode = true
```

```
# -----
```

2.2 Getting Demo Helper Onto Your Server

The easiest way to go about doing this is to pull the git repository. If you have git already installed then go ahead and run the following commands:

1. `sudo apt update`
2. `sudo apt install -y git python3 python3-pip python3-dotenv`
3. `git clone https://github.com/Amheus/DemoHelper`
4. `pipenv install`

This will download almost everything you require to get Demo Helper running. You will need to manually add two files as detailed in section 2.1.2 and section 2.1.3.

2.3 Running Demo Helper As A Service

There are many ways you can go about doing this.

2.3.1 Linux - Systemd

1. Create the following file */etc/systemd/system/DemoHelper.service*

```
[Unit]
Description=DemoHelper
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=<username>
WorkingDirectory=/home/<username>/DemoHelper
ExecStart=/usr/bin/pipenv run python3 -m DemoHelper

[Install]
WantedBy=multi-user.target
```

2. Then run the following command to load in the configuration file we've just created.

```
sudo systemctl daemon-reload
```

3. Then to start the service, run the following command:

```
sudo systemctl start DemoHelper
```

If you want Demo Helper to run on system startup, then run the following command:

```
sudo systemctl enable DemoHelper
```

The following commands are used to manage a Systemd process.

- General

- `sudo systemctl start DemoHelper`
- `sudo systemctl restart DemoHelper`
- `sudo systemctl stop DemoHelper`

- Monitoring

- `sudo systemctl status DemoHelper`
- `sudo journalctl -ru demohelper.service`
Returns logs for the service.

- System Startup

- `sudo systemctl enable DemoHelper`
Sets the service to automatically start on system startup.
- `sudo systemctl disable DemoHelper`
Stops the service from automatically starting on system startup.

2.3.2 Linux - PM2

To install Process Manager 2, run the following commands:

1. `sudo apt update`
2. `sudo apt install -y npm`
3. `sudo npm install pm2@latest -g`
4. `cd /DemoHelper`
5. `pm2 start main.py --name DemoHelper --interpreter python3`
6. `pm2 save`
7. `pm2 startup`

After a server reboot, run `pm2 resurrect` to bring back all process instances that were present when you ran `pm2 save`.

The following commands are used to manage a PM2 process:

- General

- `pm2 start <pm2-id>`
- `pm2 restart <pm2-id>`
- `pm2 stop <pm2-id>`

- Monitoring

- `pm2 list`
Returns a list of all processes managed by PM2.
- `pm2 monit`
Similar to `pm2 list` but gives a more detailed overview
- `pm2 status <pm2-id>`

- System Startup

- `pm2 save`
- `pm2 startup`
- `pm2 resurrect`

2.4 General Maintenance

1. Update dependencies using `pipenv update`.
2. Pull updates from the repository using `git pull`.
3. Then restart the service.