

PrusaLink DC bot käyttöohjeet ja yleinen dokumentaatio

Tässä ohjeistuksessa opetan miten asentaa ja ottaa käyttöön tämä ohjelma

Asennus:

Aloitetaan lataamalla botti [githubista](#).

Seuraavaksi asennamme [Node.js](#) ja ffmpeg (ei pakollinen mutta jos haluaa streamata webcamin nettiin niin pitää ladata)

Node.js asennus:



The screenshot shows the Node.js website with the headline "Run JavaScript Everywhere". Below the headline, it states: "Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts." There are two buttons: "Get Node.js®" and "Get security support for EOL Node.js versions". Below these are logos for various partners. To the right, there is a code editor showing a simple HTTP server in JavaScript:

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

Below the code editor, it says "JavaScript" and "Copy to clipboard". At the bottom, it says "Learn more what Node.js is able to offer with our Learning materials."

Paina "Get Node.js" ja avaa asennusohjelma ja seuraa sen ohjeita.

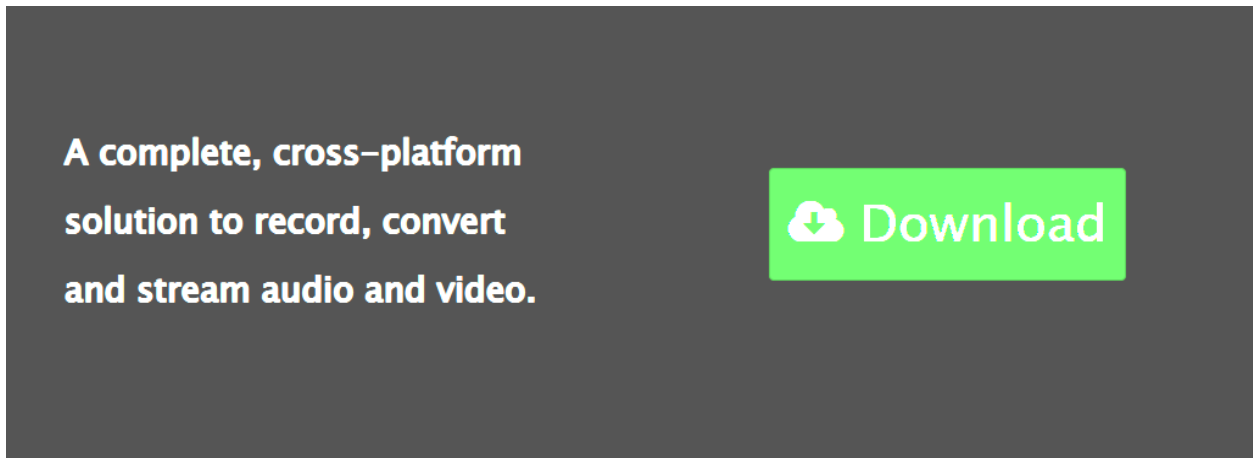
Seuraavaksi tarkistamme että se on asennettu oikein, avaa komentokehote ja kirjoita "node --version" ja sinun pitäisi nähdä seuraava:

```
C:\Users\krist>node --version
v25.2.1
```

Nyt node.js on asennettu ja on aika joko asentaa ffmpeg TAI aloittaa ohjelman käyttö

(Vaihtoehtoinen) FFmpeg asennus:

Mene ffmpeg.org ja pain "Download" nappia



Se vie sinut seuraavalle sivulle, siellä paina windowsin kuvaketta (tai mitä käyttöjärjestelmää käytät) ja paina "Windows builds from gyan.dev"

— Jatkuu Seuraavalla sivulla —

[Gyan.dev](#) sivu on aika epäselvä mutta painat vain “ffmpeg-git-full.7z”

FFmpeg Builds

binaries for Windows

Gyan Doshi
2025-10-27 v56

This page hosts packages containing binaries of **ffmpeg**, **ffprobe** and **ffplay**. Essentials build compatible with Windows 7 or above. Full build requires at least Windows 10.

If you're downloading a package to support features in a program like Krita or Blender, the **release essentials** build is sufficient. Read more in the section [about these builds](#).

The following builds are also available through package managers:

release essentials: `choco install ffmpeg` `winget install "FFmpeg (Essentials Build)"`

release full: `choco install ffmpeg-full` `scoop install ffmpeg` `winget install ffmpeg`

release full shared: `scoop install ffmpeg-shared` `winget install "FFmpeg (Shared)"`

git master: `scoop install ffmpeg-gyan-nightly`

Last build update: 2025-12-08 Next build update: 2025-12-11

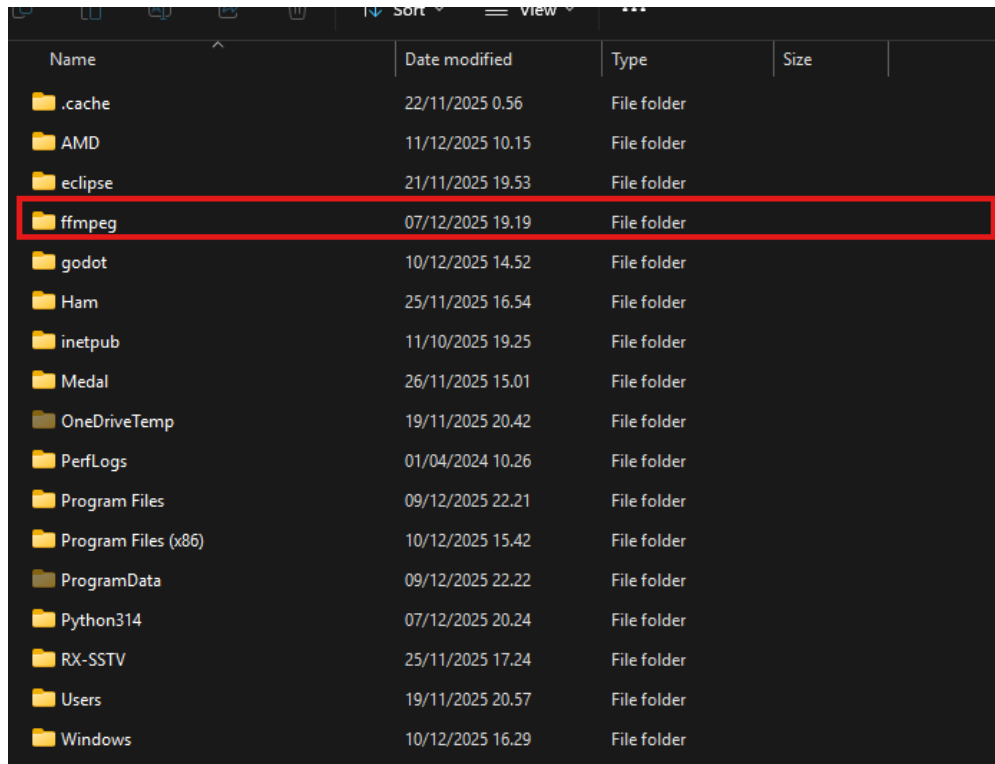
git master builds

latest git master branch build version: 2025-12-07-git-c4d22f2d2c

ffmpeg-git-essentials.7z	.ver	.sha256
ffmpeg-git-full.7z	.ver	.sha256

— jatkuu seuraavalla sivulla —

Seuraavaksi teemme kansion jonnekin mistä sitä ei helposti poisteta koneellasi, itse tein C: levyn juureen kansion nimeltä “ffmpeg”



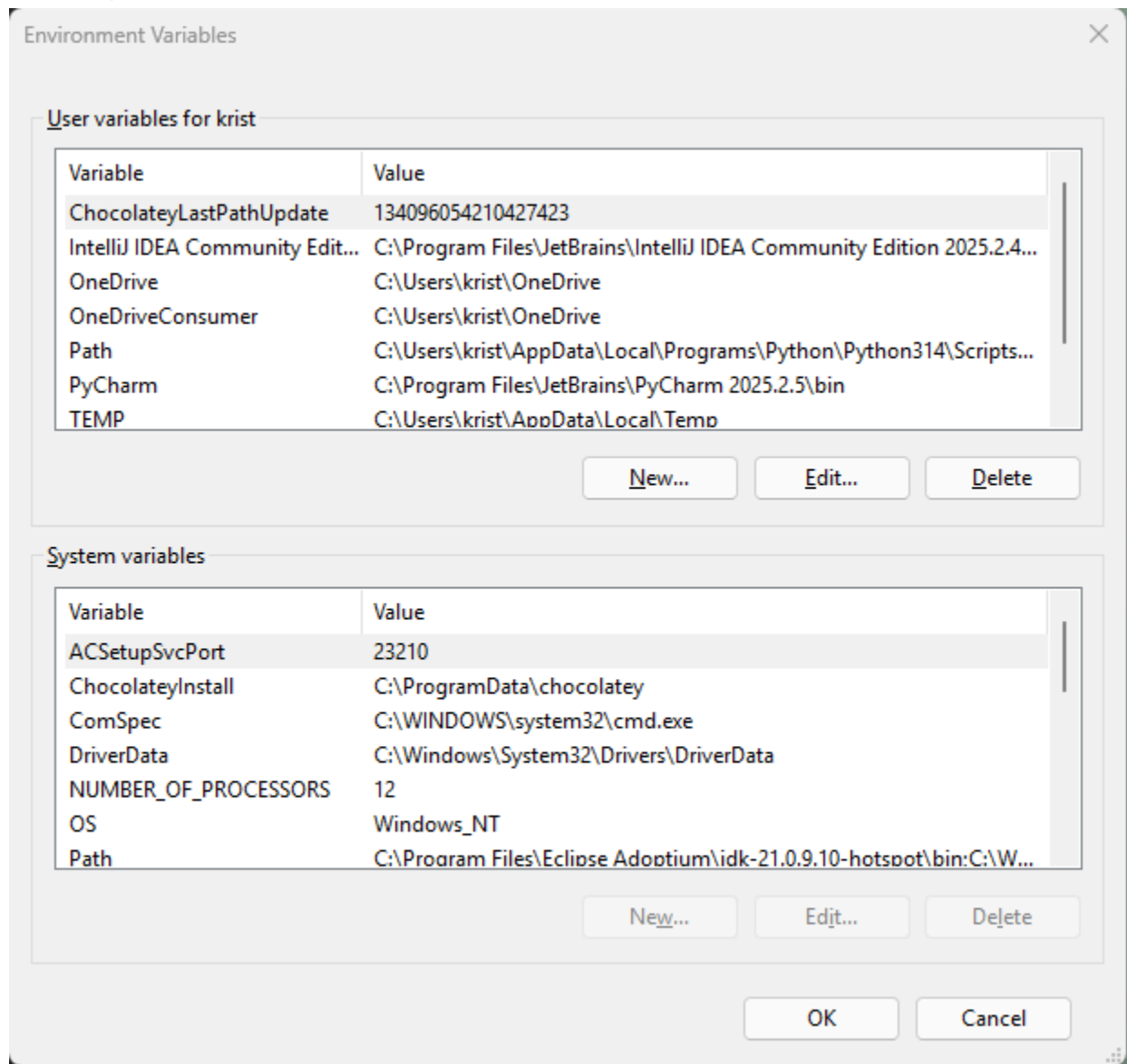
Name	Date modified	Type	Size
.cache	22/11/2025 0.56	File folder	
AMD	11/12/2025 10.15	File folder	
eclipse	21/11/2025 19.53	File folder	
ffmpeg	07/12/2025 19.19	File folder	
godot	10/12/2025 14.52	File folder	
Ham	25/11/2025 16.54	File folder	
inetpub	11/10/2025 19.25	File folder	
Medal	26/11/2025 15.01	File folder	
OneDriveTemp	19/11/2025 20.42	File folder	
PerfLogs	01/04/2024 10.26	File folder	
Program Files	09/12/2025 22.21	File folder	
Program Files (x86)	10/12/2025 15.42	File folder	
ProgramData	09/12/2025 22.22	File folder	
Python314	07/12/2025 20.24	File folder	
RX-SSTV	25/11/2025 17.24	File folder	
Users	19/11/2025 20.57	File folder	
Windows	10/12/2025 16.29	File folder	

Pura ladattu ffmpeg paketti ja heitä seuraavat tiedostot juuri tehtyyn ffmpeg kansioon.

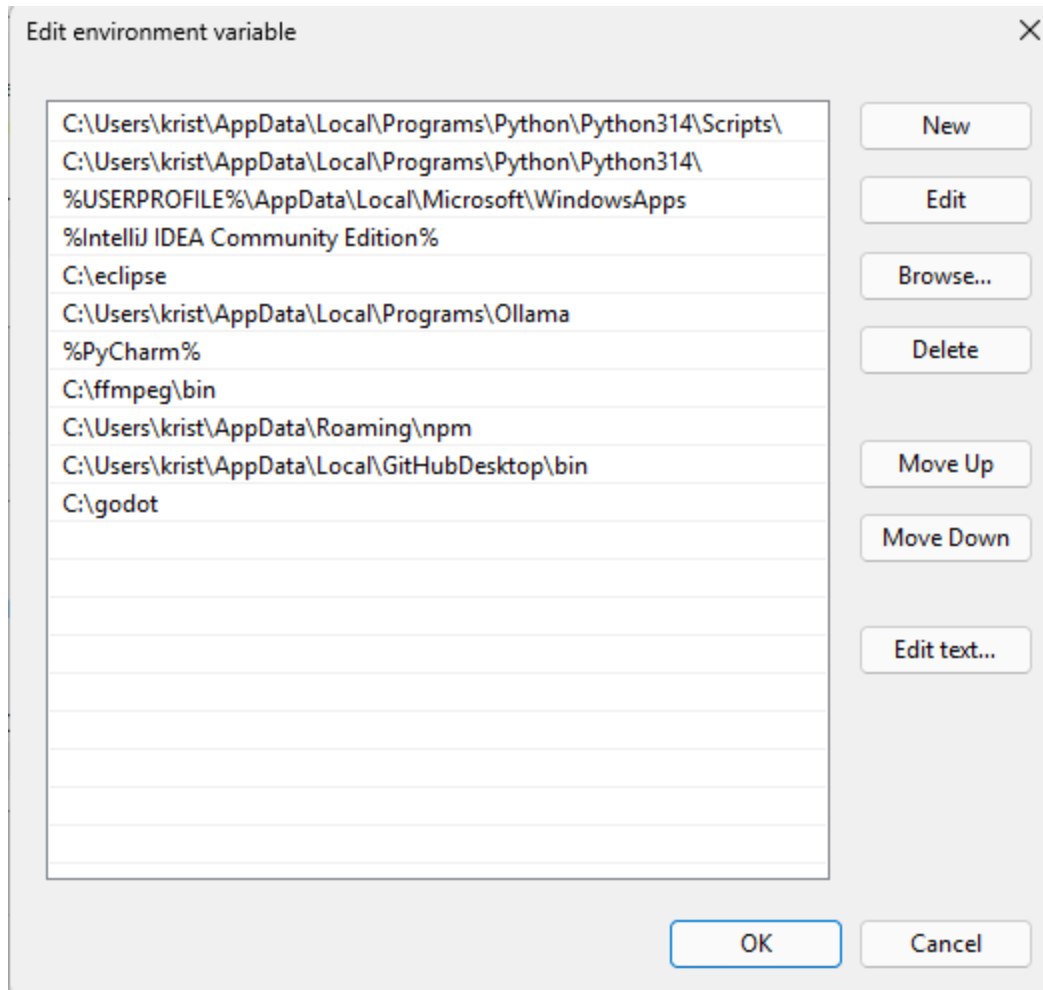
bin	644 103 168	?	File folder	08/12/2025 12....	
doc	11 467 411	?	File folder	08/12/2025 12....	
presets	1 127	?	File folder	08/12/2025 12....	
LICENSE	35 147	?	File	08/12/2025 12....	6677F57C
README.txt	45 037	?	Text Document	08/12/2025 12....	C1CD2422

Kopioi seuraavaksi kansion osoite ja avaa “muuta järjestelmä muuttujia”

Sinulle pitäisi avautua seuraavallinen ikkuna



Paina "Path" jonka näet ylemässä "User variables for [username]" kohdassa



Seuraavaksi paina "New" (tai "uusi" jos suomeksi) ja laita kansion osoite ja lisää "\bin" loppuun, paina ok ja avaa komentokehote ja kirjoita ffmpeg, jos komentokehote tulostaa paljon tekstiä niin olet asentanut oikein.

Botin Käyttöönotto:

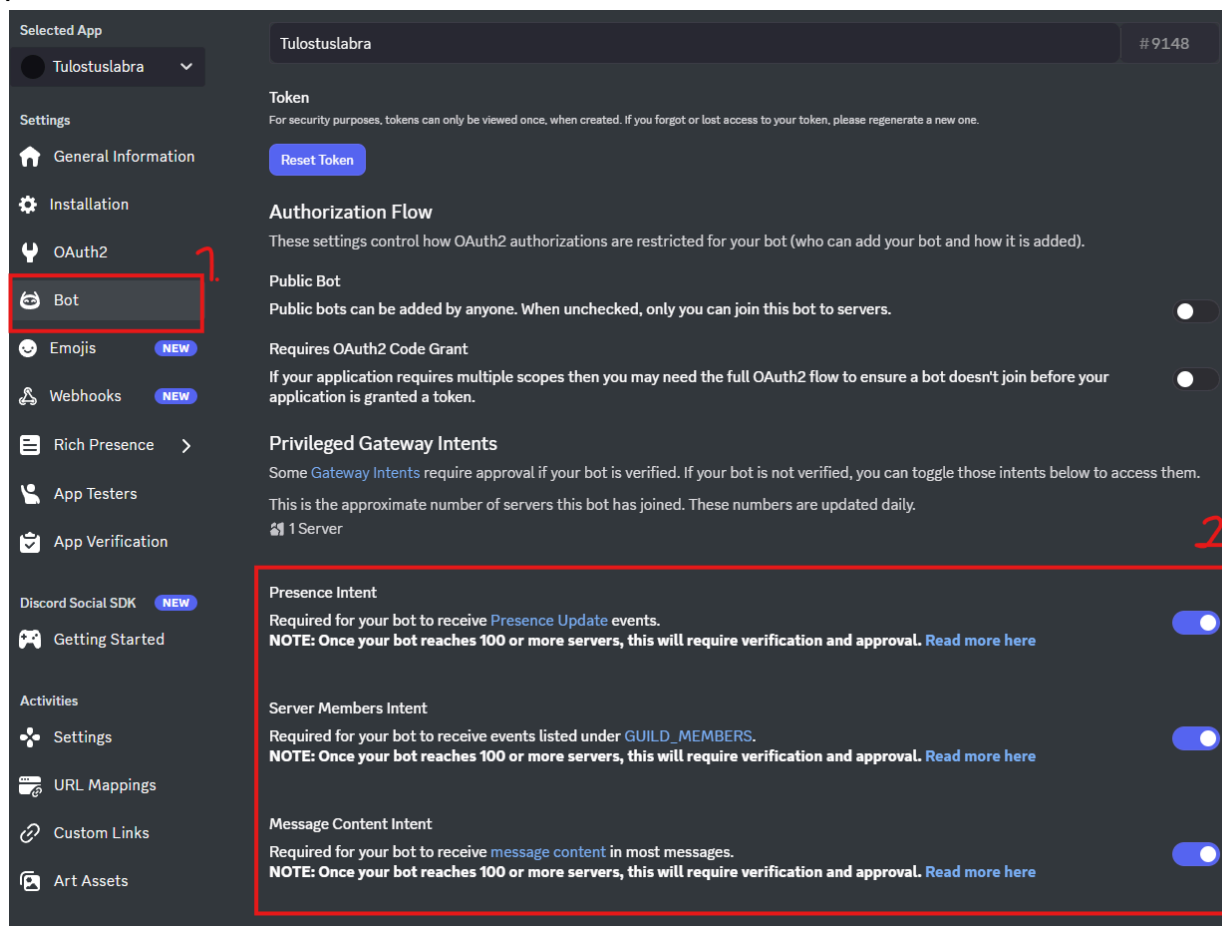
Aloitetaan avaamalla “.env” tiedosto (jos sellaista ei ole luo sellainen), siellä näet kolme kohtaa “TOKEN”, “PRINTER_IP” ja “PL_API”

Aloitetaan TOKEN kohdalla, tarvitset discord käyttäjän ja mene seuraavaan osoitteeseen: <https://discord.com/developers/applications>

Paina seuraavaa nappia, anna botille nimi ja paina create



Seuraavaksi annamme botille tarvittavat ominaisuudet, mene “bot” sivulle ja anna luvat “Presence Intent”, “Server Members Intent”, “Message Content Intent” ja paina talenna



Seuraavaksi menemme OAuth2 sivulla ja valitse “OAuth2 URL generator” kohdasta “bot” ja “bot permissions” kohdasta “Administrator” (voit valita yksittäisiä lupia mutta admin on vain helpompi), seuraavaksi kopioi linkki, heitä se hakukenttään ja invaa botti jollekin servulle

OAuth2 URL Generator

Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!

Scopes

<input type="checkbox"/> identify	<input type="checkbox"/> email	<input type="checkbox"/> connections
<input type="checkbox"/> guilds	<input type="checkbox"/> guilds.join	<input type="checkbox"/> guilds.members.read
<input type="checkbox"/> guilds.channels.read	<input type="checkbox"/> gdm.join	<input checked="" type="checkbox"/> bot
<input type="checkbox"/> rpc	<input type="checkbox"/> rpc.notifications.read	<input type="checkbox"/> rpc.voice.read
<input type="checkbox"/> rpc.voice.write	<input type="checkbox"/> rpc.video.read	<input type="checkbox"/> rpc.video.write
<input type="checkbox"/> rpc.screenshare.read	<input type="checkbox"/> rpc.screenshare.write	<input type="checkbox"/> rpc.activities.write
<input type="checkbox"/> webhook.incoming	<input type="checkbox"/> messages.read	<input type="checkbox"/> applications.builds.upload
<input type="checkbox"/> applications.builds.read	<input type="checkbox"/> applications.commands	<input type="checkbox"/> applications.store.update
<input type="checkbox"/> applications.entitlements	<input type="checkbox"/> activities.read	<input type="checkbox"/> activities.write
<input type="checkbox"/> activities.invites.write	<input type="checkbox"/> relationships.read	<input type="checkbox"/> relationships.write
<input type="checkbox"/> voice	<input type="checkbox"/> dm_channels.read	<input type="checkbox"/> role_connections.write
<input type="checkbox"/> presences.read	<input type="checkbox"/> presences.write	<input type="checkbox"/> openid
<input type="checkbox"/> dm_channels.messages.read	<input type="checkbox"/> dm_channels.messages.write	<input type="checkbox"/> gateway.connect
<input type="checkbox"/> account.global_name.update	<input type="checkbox"/> payment_sources.country_code	<input type="checkbox"/> sdk.social_layer_presence
<input type="checkbox"/> sdk.social_layer	<input type="checkbox"/> lobbies.write	<input type="checkbox"/> application_identities.write
<input type="checkbox"/> applications.commands.permissions.update		

Bot Permissions

General Permissions	Text Permissions	Voice Permissions
<input checked="" type="checkbox"/> Administrator	<input type="checkbox"/> Send Messages	<input type="checkbox"/> Connect
<input type="checkbox"/> View Audit Log	<input type="checkbox"/> Create Public Threads	<input type="checkbox"/> Speak
<input type="checkbox"/> Manage Server	<input type="checkbox"/> Create Private Threads	<input type="checkbox"/> Video
<input type="checkbox"/> Manage Roles	<input type="checkbox"/> Send Messages in Threads	<input type="checkbox"/> Mute Members
<input type="checkbox"/> Manage Channels	<input type="checkbox"/> Send TTS Messages	<input type="checkbox"/> Deafen Members
<input type="checkbox"/> Kick Members	<input type="checkbox"/> Manage Messages	<input type="checkbox"/> Move Members
<input type="checkbox"/> Ban Members	<input type="checkbox"/> Pin Messages	<input type="checkbox"/> Use Voice Activity
<input type="checkbox"/> Create Instant Invite	<input type="checkbox"/> Manage Threads	<input type="checkbox"/> Priority Speaker
<input type="checkbox"/> Change Nickname	<input type="checkbox"/> Embed Links	<input type="checkbox"/> Request To Speak
<input type="checkbox"/> Manage Nicknames	<input type="checkbox"/> Attach Files	<input type="checkbox"/> Use Embedded Activities
<input type="checkbox"/> Manage Expressions	<input type="checkbox"/> Read Message History	<input type="checkbox"/> Use Soundboard
<input type="checkbox"/> Create Expressions	<input type="checkbox"/> Mention Everyone	<input type="checkbox"/> Use External Sounds
<input type="checkbox"/> Manage Webhooks	<input type="checkbox"/> Use External Emojis	<input type="checkbox"/> Set Voice Channel Status
<input type="checkbox"/> View Channels	<input type="checkbox"/> Use External Stickers	
<input type="checkbox"/> Manage Events	<input type="checkbox"/> Add Reactions	
<input type="checkbox"/> Create Events	<input type="checkbox"/> Use Slash Commands	
<input type="checkbox"/> Moderate Members	<input type="checkbox"/> Use Embedded Activities	
<input type="checkbox"/> View Server Insights	<input type="checkbox"/> Use External Apps	
<input type="checkbox"/> View Server Subscription Insights	<input type="checkbox"/> Create Polls	

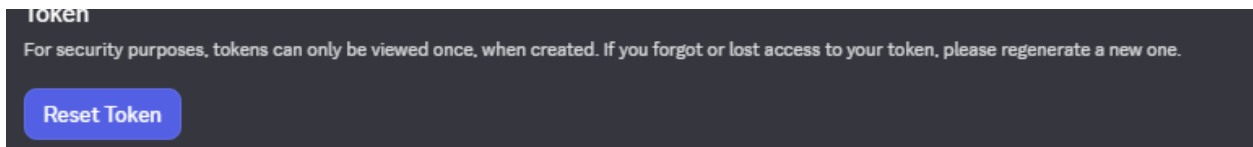
Integration Type

Guild Install

Generated URI

https://discord.com/oauth2/authorize?client_id=1446484685990531142&permissions=8&integration_type=0&scope=bot

Sitten takaisin “bot” sivulle ja paina “Reset Token” ja kopioi se uusi token, muista, ÄLÄ JAA TOKENIA KENELLEKKÄÄN



Mene takaisin .env tiedostoon ja lisää token “TOKEN” kohtaan

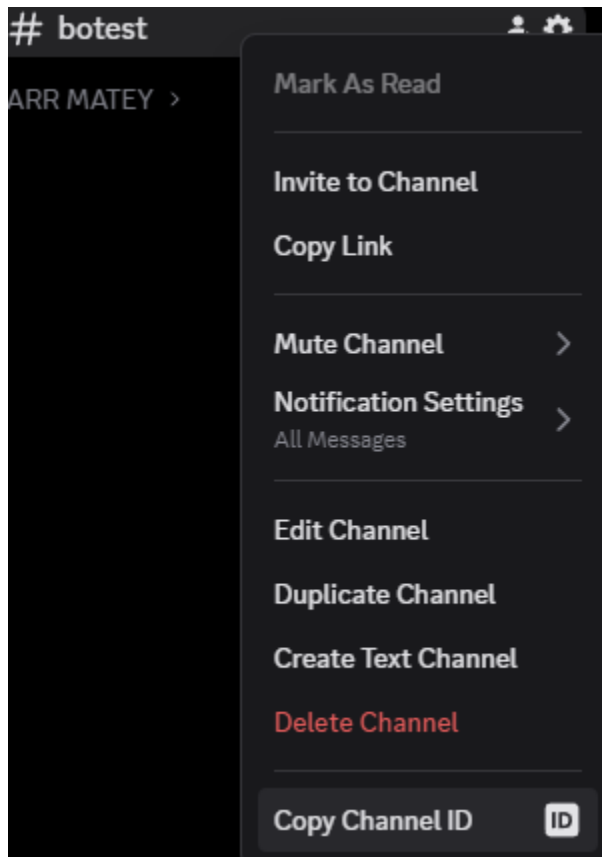
```
TOKEN = "MTQ[REDACTED]II"
```

Seuraavaksi hankimme printer IP ja PL (Prusalink) API avaimen, molemmat voit löytää PrusaConnect sivulta, joten kopioi ne ja heitä .env tiedostoon

Seuraavaksi laitamme default channelin botille, eli kanavan mihin botti heittää viestit, mene “index.js” tiedostoon ja löydä “DEFAULT_CHANNEL” muuttuja.

— Jatkuu Seuraavalla sivulla —

Mene discordiin ja löydä haluamasi kanava, klikkaa kanavalle oikealla hiiren näppäimellä ja paina "Copy Channel ID"



Heitä tämä ID "DEFAULT_CHANNEL" muuttuunaan ja olet valmis.

Nyt kaikki on valmista ja voit käynnistää sovelluksen kirjoittamalla komentokehotteeseen botti kansion sisällä "node [index.js](#)"

Teknistä hommaa (ei tärkeää botin toimivuuteen)

Botin kaikki toiminnot on jaettu eri tiedostoihin

“app.js” vastaa webserverin toimivuudesta ja ei ole pakollinen botin käyttöön

“index.js” vastaa itse botista

“Job_status.js” kysyy tulostimelta tämän hetkisestä tulosteesta jos sellaista on

“Printer_status.js” kysyy tulostimelta sen tämän hetkisestä tilanteesta

“quickjsonsafethingy.js” vastaa JSON tietojen korjaamisesta jos tulostin antaa väärää JSON koodia

“Time_calc.js” muuttaa tulostimen antaman tulostus ajan sekunneista tunneiksi, minuuteiksi, jne

Kaikki Komennot:

app.js

Start_webapp käynnistää webservering

start_webapp(app = express(), PORT = 8080, clients=[])

App = express() on webserver kirjasto... siihen ei tarvi koskea

PORT = 8080 on default portti mitä webserver käyttää, sen voi muuttaa

Clients = [] lista yhdistetyistä clientesita

Job_status.js

getJobStatus palauttaa tällä hetkellä käynnissä olevan tulosteen tiedot

getJobStatus(timeoutMS = 5000)

timeoutMS = 5000 jos tulostin ei vastaa 5 sekunnin sisällä palauttaa virheen

Cancel_job (timeoutMS = 5000, job_id)

Peruuttaa käynnissä olevan tulosteen

timeoutMS = 5000 jos tulostin ei vastaa 5 sekunnin sisällä palauttaa virheen

Job_id on tuloste työn id joka halutaan peruttaa

Printer_status.js

Palauttaa tulostimen tämän hetkisen tilanteen

getPrinterStatus(timeoutMs=5000)

timeoutMS = 5000 jos tulostin ei vastaa 5 sekunnin sisällä palauttaa virheen

quickjsonsafethingy.js

Annetaan JSON koodi ja se katsoo että sen on kelpaava

safeFetch(res)

res = JSON koodi joka halutaan tarkistaa

time_calc.js

formatduration(seconds)

Annetaan sanotaan 100 sekuntia ja muuttaa sen 1min 40sec muotoon.

