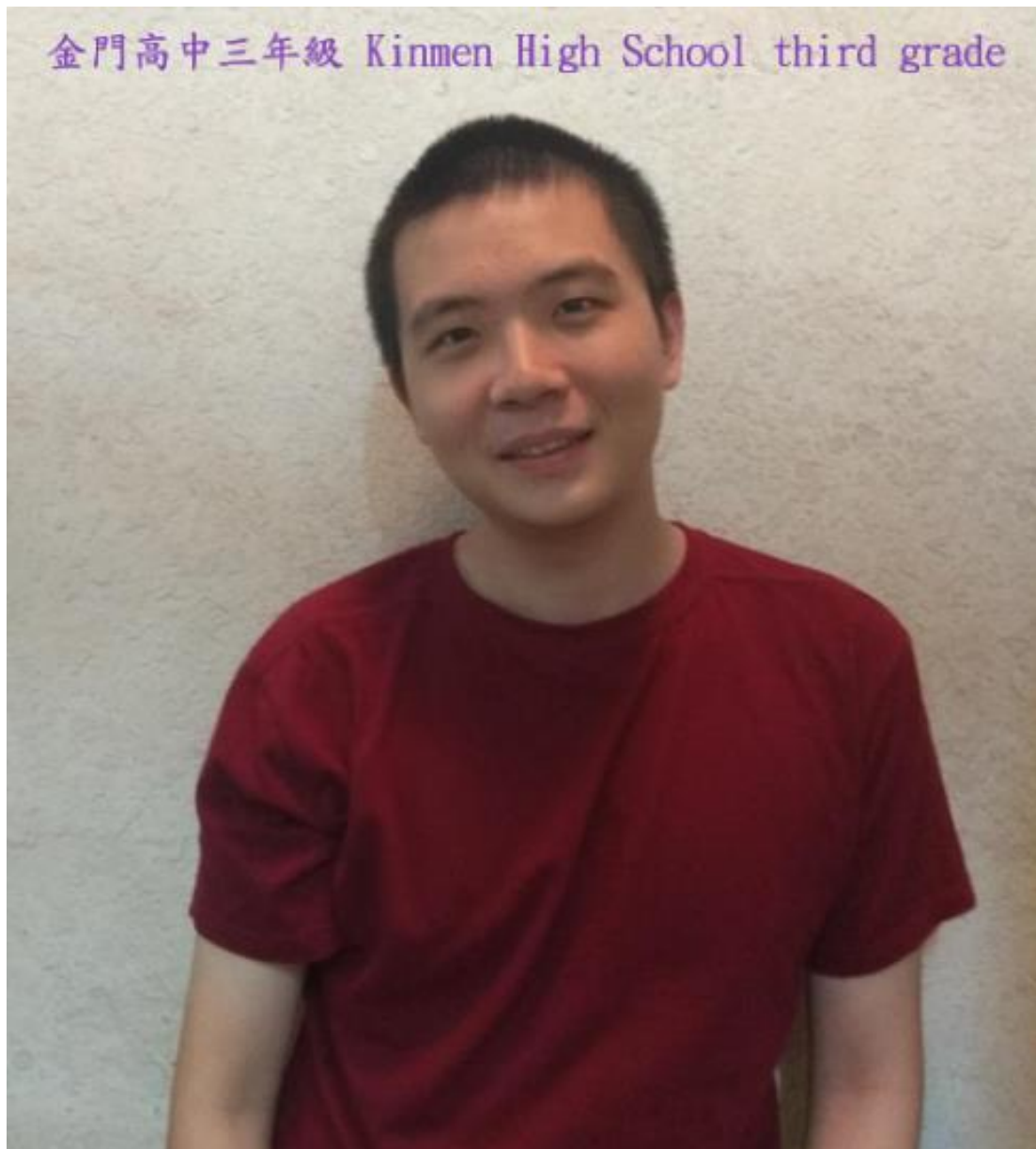


李易的應用程式
LeeYi's application program

著作人 李易 2021 年
The author LeeYi in 2021



李易的伺服器 · LeeYi's server

| LeeYi-user fix upload again | |
|-----------------------------|---------------------|
| 📁 .vscode | add chat |
| 📁 components | add upload/download |
| 📁 islands | fix upload again |
| 📁 routes | fix upload again |
| 📁 static | Initial commit |
| 📄 .env.example | Initial commit |
| 📄 .gitattributes | Initial commit |
| 📄 .gitignore | Initial commit |
| 📄 LICENSE | Initial commit |
| 📄 README.md | Initial commit |
| 📄 deno.json | add chat |
| 📄 dev.ts | Initial commit |
| 📄 fresh.gen.ts | add upload/download |
| 📄 import_map.json | Initial commit |
| 📄 main.ts | Initial commit |
| 📄 twind.config.ts | Initial commit |

李易的程式文章

LeeYi's programming article

李易的伺服器，程式文章又多又長……

LeeYi's server has many and long articles ...

所以僅編輯截取文章的片段，謹供學弟妹們免費參考。

So only edit the clip of the article,
and I would like to give the students and siblings to
reference for free.

```
import discord
from discord.ext import commands
from dislash import slash_commands

client = commands.Bot(command_prefix = commands.when_mentioned, intent
slash = slash_commands.SlashClient(client)
client.remove_command("help")

if __name__ == "__main__":
    client.load_extension("react.cmd")
    client.load_extension("react.event")
    client.load_extension("core")
    client.run("")

class Core(Cog):
    async def help(command):
        if command == None:
            return discord.Embed(title = "Commands", description = "`/ge
elif command == "get":
            return discord.Embed(title = "Example", description = "`/get
elif command == "help":
            return discord.Embed(title = "Example", description = "`/hel
elif command == "remove":
            return discord.Embed(title = "Example", description = "`/rem

        raise LookupError("can't find command")
```

```

async def get(intext, name, color):
    await Core.check(intext)
    role = await Core.find(intext.guild, intext.author)

    if len(intext.guild.roles) >= 250:
        raise MemoryError("can't create role")
    elif len(name) > 100:
        raise NameError("can't set name")

    with open("color_names.json") as f:
        color_names = json.loads(f.read())

    try:
        color = int(color_names[color.lower()].replace("#", "0x"), 16)
    except:
        try:
            if color[0] != "#":
                raise

            color = int(color.replace("#", "0x"), 16)

            if (color > 16777215) or (color < 0):
                raise
        except:
            raise LookupError("can't find color")

    if role != None:
        await role.delete()

    role = await intext.guild.create_role(name = name, color = discord
    for i in range(intext.me.top_role.position, 0, -1):
        try:
            await intext.guild.edit_role_positions(positions = {role:
            break
        except:
            pass

    await intext.author.add_roles(role)
    return discord.Embed(title = f"You Have Got the \"{role.name}\" Ro

@commands.command()
@commands.cooldown(1, 60, commands.BucketType.user)
async def help(self, ctx, *, command = None):
    permission = ctx.me.permissions_in(ctx.channel)

    if (permission.embed_links == True) and (permission.read_message_hist

```

```

@commands.command()
@commands.cooldown(1, 60, commands.BucketType.user)
async def get(self, ctx, *, msg):
    permission = ctx.me.permissions_in(ctx.channel)

    if (permission.embed_links == True) and (permission.read_message_history):
        await ctx.trigger_typing()
        strings = msg.split(" ")

        with open("color_names.json") as f:
            color_names = json.loads(f.read())

        if (len(strings) > 1) and ((strings[-1].lower() in color_names)):
            name = " ".join(strings[:-1])
            color = strings[-1]
        else:
            name = msg
            color = "black"

        await ctx.send(embed = await Core.get(ctx, name, color))

    ctx.command.reset_cooldown(ctx)

@get.error
async def get_error(self, ctx, error):
    permission = ctx.me.permissions_in(ctx.channel)

    if isinstance(error, commands.CommandOnCooldown) == True:

if isinstance(error, commands.MissingRequiredArgument) == True:
    pass
elif str(error.original) == "can't use here":
    await ctx.send(embed = discord.Embed(title = "You Cannot Use It Here"))
elif str(error.original) == "can't manage roles":
    await ctx.send(embed = discord.Embed(title = "I Need Manage Roles Permission"))
elif str(error.original) == "can't view audit log":
    await ctx.send(embed = discord.Embed(title = "I Need View Audit Log Permission"))
elif str(error.original) == "can't manage user":
    await ctx.send(embed = discord.Embed(title = "I Need Higher Role to Manage User"))
elif str(error.original) == "can't create role":
    await ctx.send(embed = discord.Embed(title = "Role Amount Must Be Less Than 10"))
elif str(error.original) == "can't set name":
    await ctx.send(embed = discord.Embed(title = "Name Length Must Be Less Than 32"))
elif str(error.original) == "can't find color":

```

```

@commands.command()
@commands.cooldown(1, 60, commands.BucketType.user)
async def remove(self, ctx):
    permission = ctx.me.permissions_in(ctx.channel)

    if (permission.embed_links == True) and (permission.read_message_hi
        await ctx.trigger_typing()
        await ctx.send(embed = await Core.remove(ctx))

    ctx.command.reset_cooldown(ctx)

@remove.error
async def remove_error(self, ctx, error):
    if isinstance(error, commands.CommandOnCooldown) == True:
        if ctx.me.permissions_in(ctx.channel).manage_messages == True:
            await ctx.message.delete()
        else:
            if str(error.original) == "can't use here":
                await ctx.send(embed = discord.Embed(title = "You Cannot Us
            elif str(error.original) == "can't manage roles":
                await ctx.send(embed = discord.Embed(title = "I Need Manage
            elif str(error.original) == "can't view audit log":
                await ctx.send(embed = discord.Embed(title = "I Need View A
            elif str(error.original) == "can't manage user":
                await ctx.send(embed = discord.Embed(title = "I Need Higher
            elif str(error.original) == "can't find role":
                await ctx.send(embed = discord.Embed(title = "Role is Not F
            else:
                await ctx.send(embed = discord.Embed(title = "Unknown Error
                print(error)

    ctx.command.reset_cooldown(ctx)

@commands.command()
@commands.is_owner()
async def server(self, ctx):
    permission = ctx.me.permissions_in(ctx.channel)

    if (permission.embed_links == True) and (permission.read_message_hi
        await ctx.send(embed = discord.Embed(title = f"There Are {len(s

if str(error) == "can't find command":
    await inter.reply(embed = discord.Embed(title = "Command is Not Four
else:
    await inter.reply(embed = discord.Embed(title = "Unknown Error Occur
    print(error)

```

```

@commands.Cog.listener()
async def on_ready(self):
    print(f"{self.client.user} has online")
    await self.client.change_presence(activity = discord.Activity(type

@commands.Cog.listener()
async def on_command_error(self, ctx, error):
    pass

@commands.Cog.listener()
async def on_slash_command_error(self, inter, error):
    pass

@commands.Cog.listener()
async def on_member_remove(self, member):
    try:
        role = await Core.find(member.guild, member)
        await role.delete()
    except:
        pass

@get.error
async def get_error(self, inter, error):
    if isinstance(error, slash_commands.CommandOnCooldown) == True:
        pass
    else:
        if str(error) == "can't use here":
            await inter.edit(embed = discord.Embed(title = "You Cannot
        elif str(error) == "can't manage roles":
            await inter.edit(embed = discord.Embed(title = "I Need Mana
        elif str(error) == "can't view audit log":
            await inter.edit(embed = discord.Embed(title = "I Need View
        elif str(error) == "can't manage user":
            await inter.edit(embed = discord.Embed(title = "I Need High
        elif str(error) == "can't create role":
            await inter.edit(embed = discord.Embed(title = "Role Amount
        elif str(error) == "can't set name":
            await inter.edit(embed = discord.Embed(title = "Name Length
        elif str(error) == "can't find color":
            await inter.edit(embed = discord.Embed(title = "Color is No
        else:
            await inter.edit(embed = discord.Embed(title = "Unknown Err
            print(error)

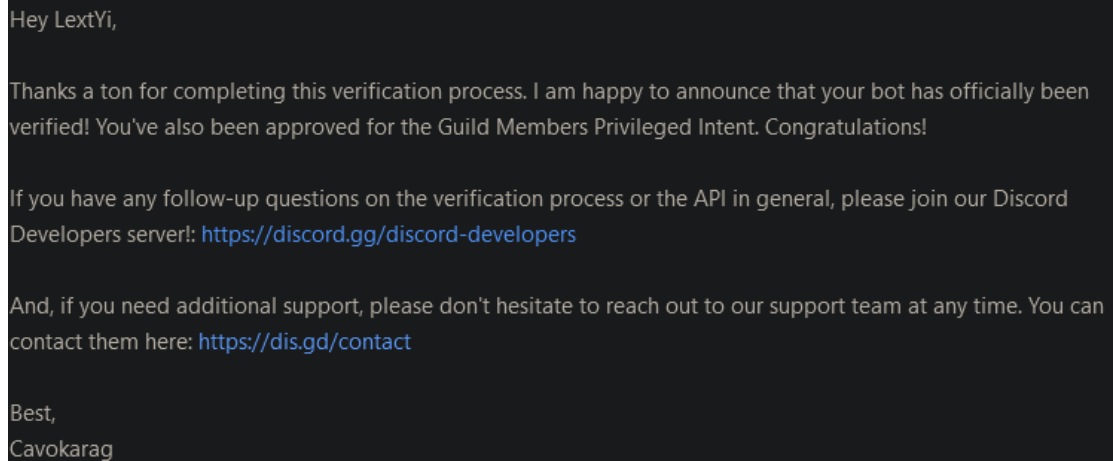
    inter.slash_command.reset_cooldown(inter)

```

李易的伺服器 · LeeYi's server

著作人 李易 2021 年
The author LeeYi in 2021

2021 年，李易得到了 discord 的認證回應。
信函的重點意思說：謝謝你…很高興的可以宣布，你的 discord 機器人通過驗證了。

A screenshot of a Discord message on a dark background. The text is white and reads: "Hey LextYi, Thanks a ton for completing this verification process. I am happy to announce that your bot has officially been verified! You've also been approved for the Guild Members Privileged Intent. Congratulations! If you have any follow-up questions on the verification process or the API in general, please join our Discord Developers server: <https://discord.gg/discord-developers> And, if you need additional support, please don't hesitate to reach out to our support team at any time. You can contact them here: <https://dis.gd/contact> Best, Cavokarag".

Hey LextYi,
Thanks a ton for completing this verification process. I am happy to announce that your bot has officially been verified! You've also been approved for the Guild Members Privileged Intent. Congratulations!
If you have any follow-up questions on the verification process or the API in general, please join our Discord Developers server: <https://discord.gg/discord-developers>
And, if you need additional support, please don't hesitate to reach out to our support team at any time. You can contact them here: <https://dis.gd/contact>
Best,
Cavokarag