

Como Fazer um Bot no Discord utilizando Python

Tutorial Original:

[*Real Python*](#)

[*Raissa Azevedo*](#)

Em um mundo onde videogames ganham cada vez mais importância na vida das pessoas, a comunidade e a comunicação entre estes no mundo dos games é essencial. O Discord oferece ambas oportunidades e muito mais em um pacote muito bem desenvolvido. Nesse tutorial você irá aprender como fazer um Bot no Discord utilizando Python, assim poderá aproveitar o máximo dessa incrível plataforma.

No final deste tutorial, você aprenderá:

- A importância do Discord
- Como fazer um bot no Discord através do Portal de Desenvolvedor.
- Como criar conexões no Discord
- Como manejar eventos
- Como aceitar comandos e validar assuntos
- Como interagir entre várias APIs no Discord.

Mas, você irá começar aprendendo o porquê o Discord se tornou tão importante.

O que é o Discord?

O Discord é uma plataforma para gamers de comunicação de texto e voz.

Jogadores, streamers e desenvolvedores usam o Discord para discutir sobre games, responder questões, apontar erros e conversar enquanto jogam. O Discord até mesmo conta com uma loja de games, com reviews completas e serviços de inscrição. É praticamente a primeira parada para a comunidade gamer.

Enquanto isso, há também uma série de coisas que você pode desenvolver utilizando as APIs do Discord. Esse tutorial irá focar em: **“Como criar um bot no Discord utilizando Python”**.

O que é um Bot?

A popularidade do Discord vem crescendo, devido aos processos automatizados, como banir usuários inapropriados e reação a demandas da comunidade são vitais para o estabelecimento e crescimento da plataforma.

Programas automatizados que se parecem e agem como os usuários e automaticamente respondem a eventos e comandos no Discord são chamados de Bot Users ou Usuário Bot, ou simplesmente Bot.

Os Bots do Discord possuem aplicações praticamente ilimitadas.

Por exemplo, vamos supor que você está gerenciando uma nova guilda no Discord e um usuário se junte pela primeira vez. Empolgado com a situação, você pode entrar em contato pessoalmente com esse usuário e recebê-lo na sua comunidade. Você pode também contar a eles sobre seus canais ou pedir para que os mesmos se apresentem.

O usuário irá se sentir bem-vindo, irá se juntar as discussões sobre diversos assuntos na sua guilda e eventualmente irá convidar amigos.

Com o passar do tempo sua comunidade irá crescer tanto que já não será mais viável receber cada novo usuário pessoalmente, e todo aquele processo de “Welcome” e toda a explicação proveniente se tornará uma atividade maçante. É aí que entram os processos de automatização envolvendo os Bot Users.

Com um bot, é possível automatizar as reações a um novo usuário se juntando a sua guilda. Você pode mesmo customizar o comportamento do bot a seu gosto pessoal. Baseando as reações através do contexto e comando de resposta para cada interação com o novo usuário.

Este é apenas um pequeno exemplo do como um bot pode ser útil, há um gama de oportunidades para você utilizar sua criatividade com os bots. Você só precisa aprender como desenvolver um.

Nota: *Embora o Discord permita que você crie bots que lidam com comunicação por voz, este tutorial se limitará ao serviço de texto.*

Existem duas etapas principais quando você está criando um bot:

- Crie o usuário bot no Discord e registre-o em uma guilda.
- Escreva um código que use APIs do Discord e implemente os comportamentos do seu bot.

Como fazer um Discord Bot no Portal do Desenvolvedor

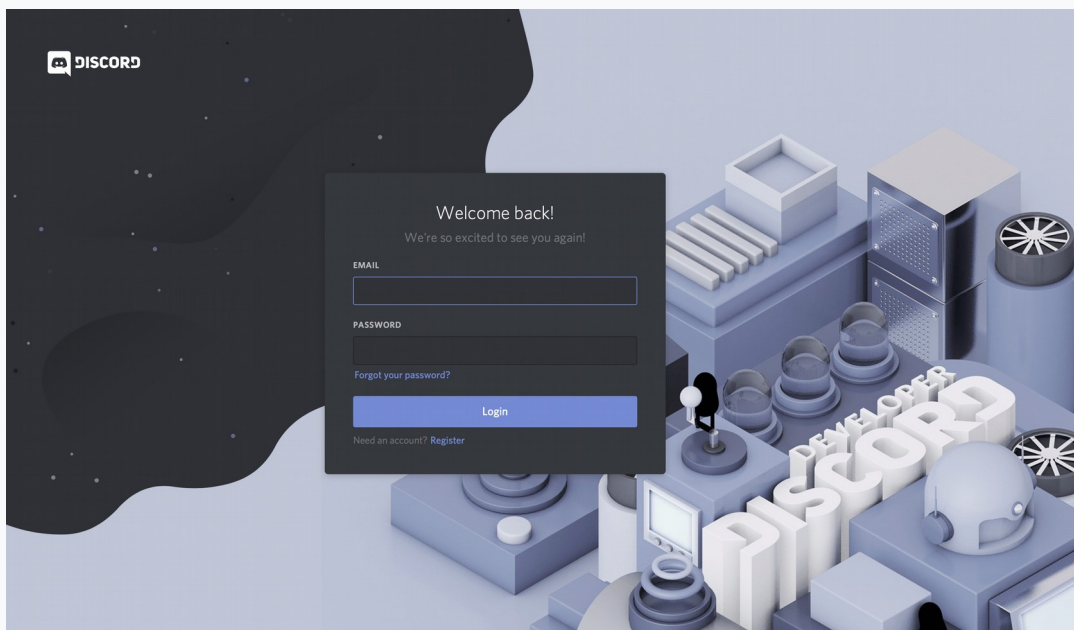
Antes de mergulhar em qualquer código Python para lidar com eventos e criar automações emocionantes, você precisa primeiro criar alguns componentes do Discord:

- Uma conta
- Uma aplicação
- Um bot
- Uma guilda

Você aprenderá mais sobre cada peça nas seções a seguir. Depois de criar todos esses componentes, você os unirá registrando seu bot com sua guilda. Você pode começar acessando o **Portal do desenvolvedor do Discord**.

Criando uma conta no Discord

A primeira coisa que você verá é uma página de destino onde você precisará fazer login, se tiver uma conta existente, ou criar uma nova conta.



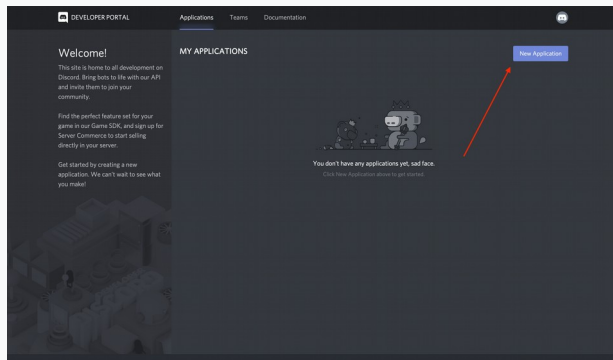
Se você precisar criar uma nova conta, clique no botão Registrar abaixo de Login e insira as informações da sua conta.

Importante: você precisará verificar seu e-mail antes de prosseguir.

Quando terminar, você será redirecionado para a página inicial do Portal do desenvolvedor, onde criará seu aplicativo.

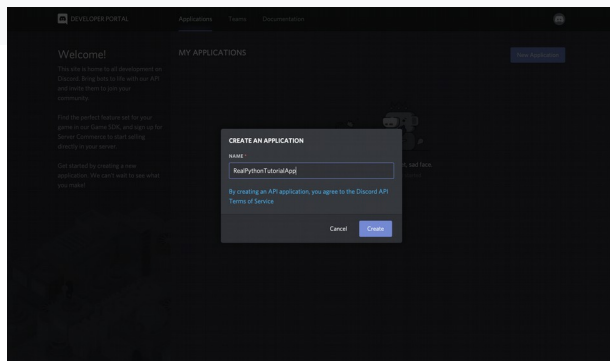
Criação de um aplicativo

Um aplicativo permite que você interaja com as APIs do Discord, fornecendo tokens de autenticação, designando permissões e assim por diante. Para criar um novo aplicativo, selecione Novo aplicativo:

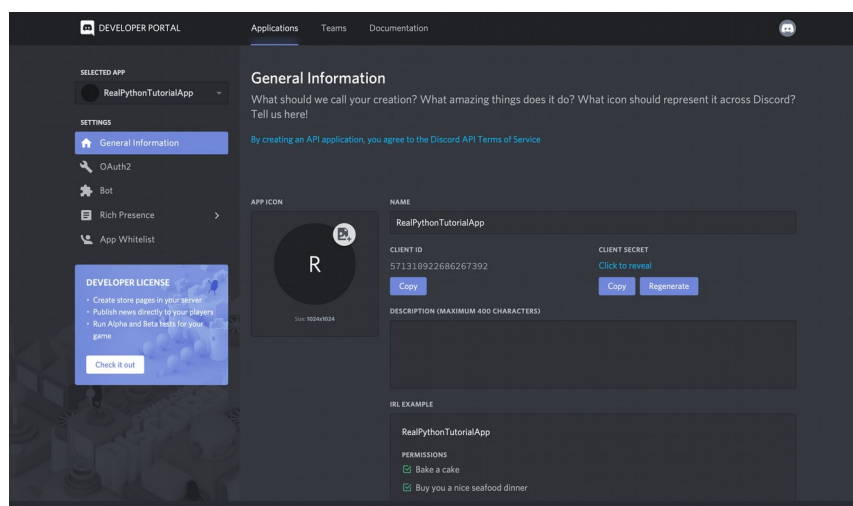


Em seguida, você será solicitado a nomear seu aplicativo.

Selecione um nome e clique em Criar: Ou “Create”



Parabéns! Você fez um aplicativo Discord. Na tela resultante, você pode ver informações sobre seu aplicativo:

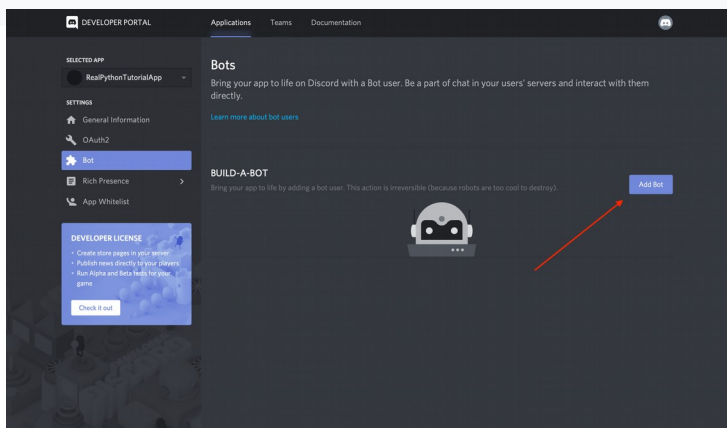


Lembre-se de que qualquer programa que interage com APIs Discord requer um aplicativo Discord, não apenas bots. APIs relacionadas a bots são apenas um subconjunto da interface total do Discord. No entanto, como este tutorial é sobre como fazer um bot Discord, navegue até a guia Bot na lista de navegação do lado esquerdo.

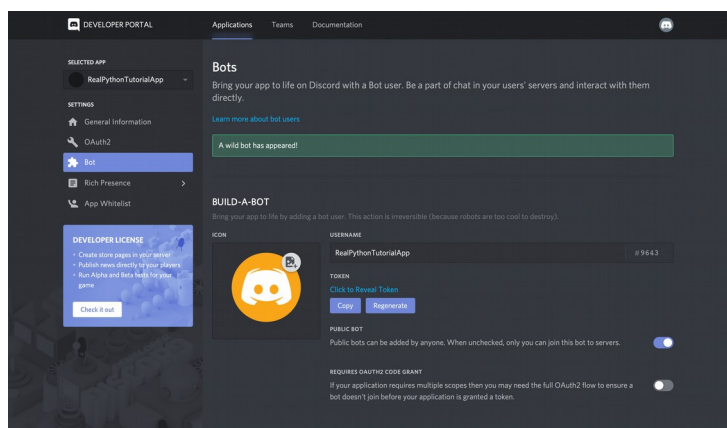
Criando um Bot

Como você aprendeu nas seções anteriores, um usuário de bot é aquele que escuta e reage automaticamente a certos eventos e comandos no Discord.

Para que seu código seja realmente manifestado no Discord, você precisará criar um usuário bot. Para fazer isso, selecione **Adicionar Bot**:



Depois de confirmar que deseja adicionar o bot ao seu aplicativo, você verá o novo usuário do bot no portal:



Observe que, por padrão, o usuário do bot herdará o nome do seu aplicativo. Em vez disso, atualize o nome de usuário para algo mais parecido com um bot, como RealPythonTutorialBot, e salve as alterações:

Agora, o bot está tudo configurado e pronto para ir, mas para onde?

Um usuário bot não é útil se não estiver interagindo com outros usuários. Em seguida, você criará uma guilda para que seu bot possa interagir com outros usuários.

Criando uma Guilda

Uma guilda (ou um servidor, como costuma ser chamada na interface do usuário do Discord) é um grupo específico de canais onde os usuários se reúnem para conversar.

Nota: Embora guilda e servidor sejam intercambiáveis, este artigo usará o termo guilda principalmente porque as APIs seguem o mesmo termo. O termo servidor só será usado quando se referir a uma guilda na IU gráfica.

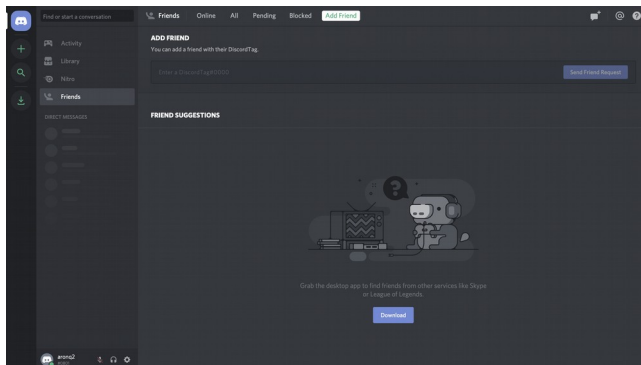
Por exemplo, digamos que você queira criar um espaço onde os usuários possam se reunir e falar sobre seu último jogo.

Você começaria criando uma guilda. Então, em sua guilda, você pode ter vários canais, como:

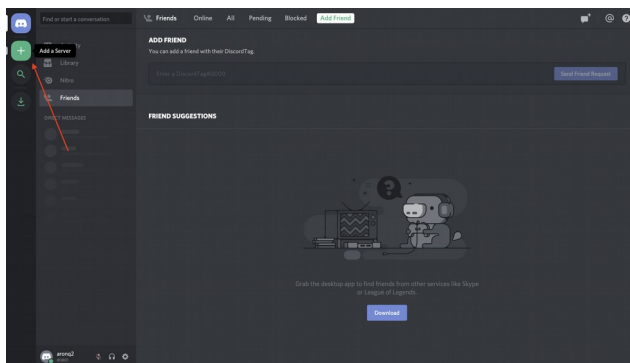
- Discussão geral: um canal para os usuários falarem sobre o que quiserem
- Spoilers, cuidado: um canal para usuários que terminaram seu jogo para falar sobre todas as revelações do jogo final
- Anúncios: um canal para você anunciar as atualizações do jogo e para os usuários discuti-las

Depois de criar sua guilda, você convidaria outros usuários para preenchê-la.

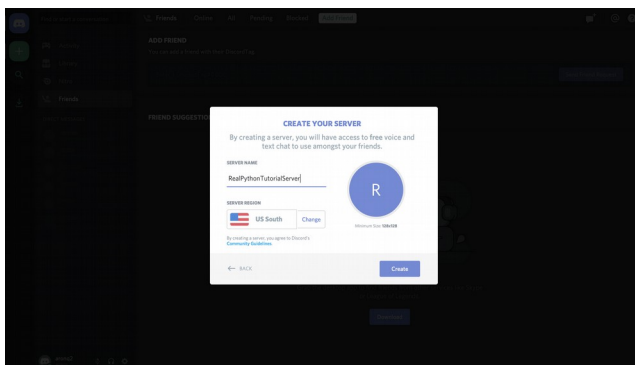
Então, para criar uma guilda, vá para a **página inicial do Discord**:



Nessa página inicial, você pode visualizar e adicionar amigos, mensagens diretas e guildas. A partir daqui, selecione o ícone + no lado esquerdo da página da web para adicionar um servidor:



Isso apresentará duas opções, Criar um servidor e Entrar em um servidor. Nesse caso, selecione Criar um servidor e digite um nome para sua guilda:



Depois de terminar de criar sua guilda, você poderá ver os usuários do lado direito e os canais do lado esquerdo:

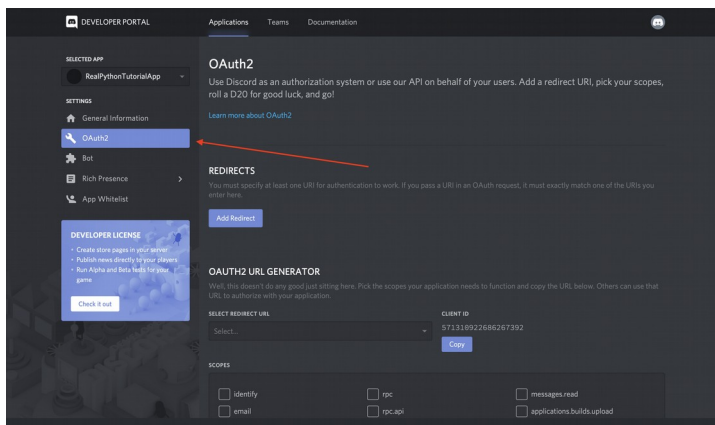
A etapa final no Discord é registrar seu bot na sua nova guilda.

Adicionando um Bot a uma Guilda

Um bot não pode aceitar convites como um usuário normal. Em vez disso, você adicionará seu bot usando o protocolo **OAuth2**.

Detalhe técnico: *OAuth2 é um protocolo para lidar com autorização, em que um serviço pode conceder a um aplicativo cliente acesso limitado com base nas credenciais do aplicativo e escopos permitidos.*

Para fazer isso, volte para o Portal do desenvolvedor e selecione a página OAuth2 na navegação à esquerda:

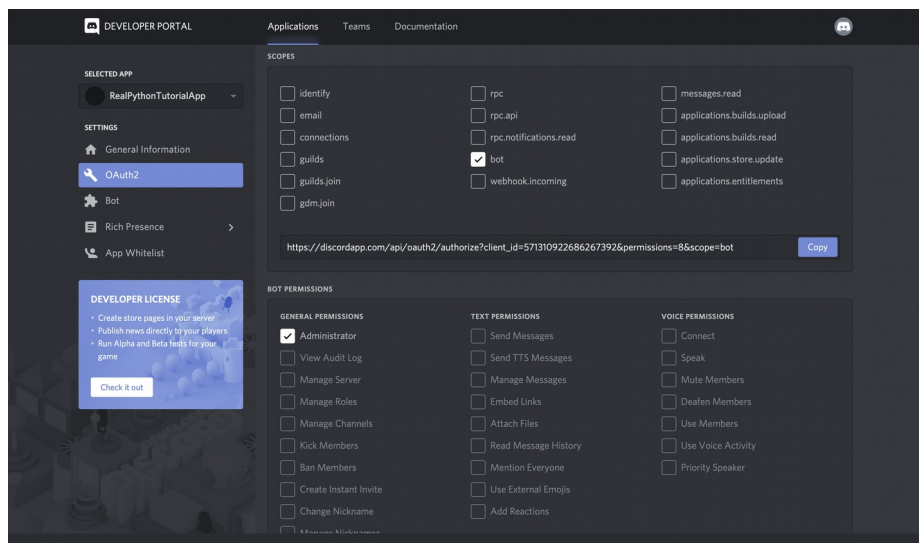


Nessa janela, você verá o Gerador de URL OAuth2.

Esta ferramenta gera um URL de autorização que atinge a API OAuth2 do Discord e autoriza o acesso à API usando as credenciais do seu aplicativo.

Nesse caso, você deseja conceder ao usuário bot do seu aplicativo acesso às APIs do Discord usando as credenciais OAuth2 do seu aplicativo.

Para fazer isso, role para baixo e selecione bot nas opções SCOPES e Administrador em PERMISSÕES DE BOT:

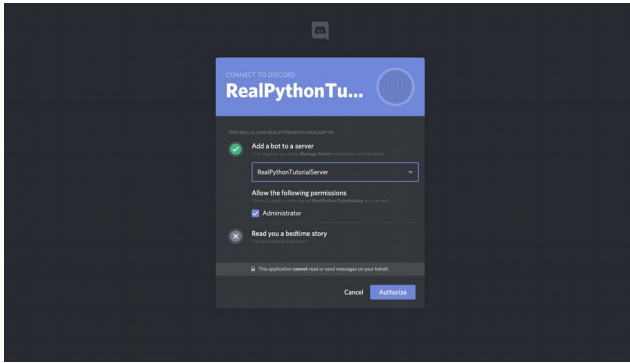


Agora, o Discord gerou o URL de autorização do seu aplicativo com o escopo e as permissões selecionadas.

Exoneração de responsabilidade: embora estejamos usando o Administrador para os fins deste tutorial, você deve ser o mais granular possível ao conceder permissões em um aplicativo do mundo real.

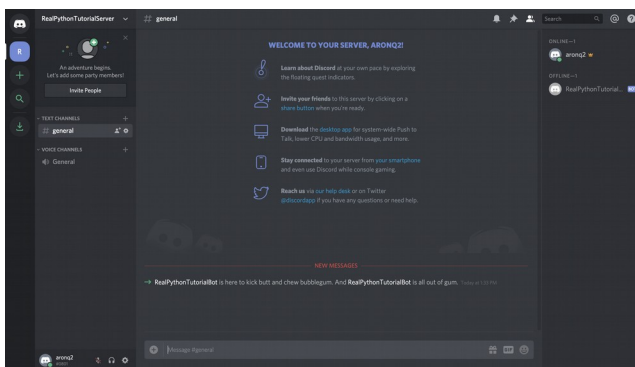
Selecione Copiar ao lado do URL que foi gerado para você, cole-o em seu navegador e selecione sua guilda nas opções suspensas:

Clique em Autorizar e pronto!



Observação: você pode obter um reCAPTCHA antes de prosseguir. Nesse caso, você precisará provar que é um humano.

Se você voltar para sua guilda, verá que o bot foi adicionado:



Em suma você criou:

- Um aplicativo que seu bot usará para autenticar com APIs do Discord
- Um usuário bot que você usará para interagir com outros usuários e eventos em sua guilda
- Uma guilda na qual sua conta de usuário e seu

usuário bot estarão ativos

- Uma conta Discord com a qual você criou todo o resto e que usará para interagir com seu bot

Agora, você sabe como fazer um bot Discord usando o Portal do Desenvolvedor. A seguir vem a parte divertida:

implementar seu bot em Python!

Como fazer um Discord Bot em Python

Já que você está aprendendo como fazer um bot Discord com Python, você usará **discord.py**.

discord.py é uma biblioteca Python que implementa exaustivamente as APIs do Discord de uma forma eficiente e Pythonica. Isso inclui a utilização da implementação de Python do Async IO.

Comece instalando discord.py com pip:

```
pip install -U discord.py
```

OBS: Algumas distribuições do Linux podem exigir o pip3 ao invés do pip.

```
sudo pip3 install discord.py
```

Agora que instalou o discord.py, você o usará para criar sua primeira conexão com o Discord!

Criando uma Conexão Discord

O primeiro passo para implementar o seu usuário bot é criar uma conexão com o Discord. Com discord.py, você faz isso criando uma instância de Client:

Agora na sua IDE para linguagem Python:

```
# bot.py
import os

import discord,asyncio,random

TOKEN = Secret.token()
msg_id = None
msg_user = None

client = discord.Client()
guild = "Bot de Raii_Azevedo"

@client.event
async def on_ready():
    print (f'{client.user} está conectado ao seguinte
servidor/guilda:\n'
          f'{guild}'
          )
client.run(SEU_TOKEN)
```

Agora que instalou o discord.py, você o usará para criar sua primeira conexão com o Discord!

Criando uma Conexão Discord

Um Cliente é um objeto que representa uma conexão com o Discord. Um cliente lida com eventos, controla o estado e geralmente interage com as APIs do Discord.

Aqui, você criou um cliente e implementou seu manipulador de eventos **on_ready ()**, que lida com o evento quando o cliente estabeleceu uma conexão com o Discord e terminou de preparar os dados que o Discord enviou, como estado de login, guilda e canal dados e muito mais.

Em outras palavras, **on_ready ()** será chamado (e sua mensagem será impressa) assim que o cliente estiver pronto para outras ações. Você aprenderá mais sobre manipuladores de eventos posteriormente neste artigo.

Quando você está trabalhando com segredos como seu token Discord, é uma boa prática lê-lo em seu programa a partir de uma variável de ambiente. Usar variáveis de ambiente ajuda você a:

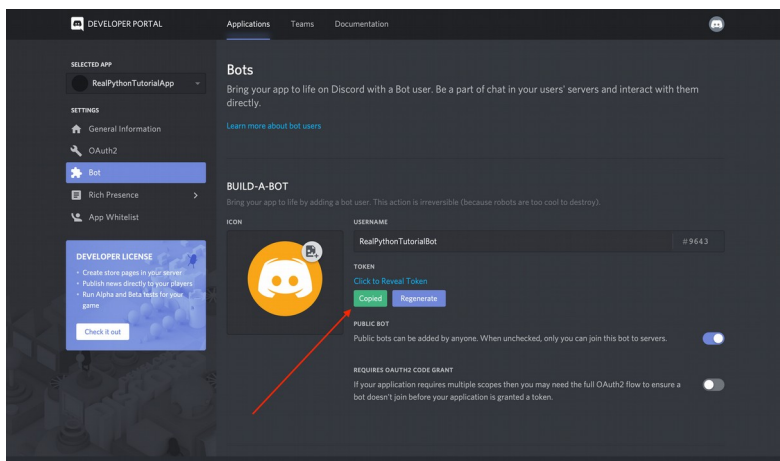
- Evitar colocar os segredos no controle de origem
- Usar variáveis diferentes para ambientes de desenvolvimento e produção sem alterar seu código

Embora você possa exportar `DISCORD_TOKEN = {your-bot-token}`, uma solução mais fácil é salvar um arquivo `.env` em todas as máquinas que estarão executando este código. Isso não é apenas mais fácil, já que você não terá que exportar seu token toda vez que limpar seu shell, mas também o protege de armazenar seus segredos no histórico de seu shell.

Crie um arquivo chamado `.env` no mesmo diretório que `bot.py`:

```
Text
# .env
DISCORD_TOKEN={your-bot-token}
```

Você precisará substituir `{your-bot-token}` pelo token do seu bot, que pode ser obtido voltando para a página do Bot no Portal do desenvolvedor e clicando em Copiar na seção TOKEN:



Olhando novamente para o código **bot.py**, você notará uma biblioteca chamada **dotenv**.

Esta biblioteca é útil para trabalhar com arquivos **.env**. `load_dotenv ()` carrega variáveis de ambiente de um arquivo **.env** nas variáveis de ambiente do seu shell para que você possa usá-las em seu código.

Instale o dotenv com pip:

```
pip install -U python-dotenv
```

OBS: Algumas distribuições do Linux exigirão o pip3 (sudo pip3 install -U python-dotenv).

Finalmente, **client.run ()** executa seu cliente usando o token do seu bot.

Agora que configurou bot.py e .env, você pode executar seu código:

```
$ python bot.py RealPythonTutorialBot # 9643 conectou-se ao Discord!
```

Ótimo! Seu cliente se conectou ao Discord usando o token do seu bot. Na próxima seção, você desenvolverá este Cliente interagindo com mais APIs Discord.

Interagindo com APIs Discord

Usando um cliente, você tem acesso a uma ampla variedade de APIs do Discord.

Por exemplo, digamos que você queira escrever o nome e o identificador da guilda com a qual você registrou seu usuário bot no console.

Primeiro, você precisará adicionar uma nova variável de ambiente:

```
Text
# .env
DISCORD_TOKEN={your-bot-token}
DISCORD_GUILD={your-guild-name}
```

`{nome-da-sua-guilda}`

Não se esqueça de que você precisará substituir os dois marcadores de posição por valores reais: `{seu-bot-token}`

Lembre-se de que Discord chama **on_ready ()**, que você usou antes, uma vez que o cliente tenha feito a conexão e preparado os dados. Portanto, você pode contar com os dados da guilda disponíveis em **on_ready ()**: