

Naismith Bot

Ben Alter, Sam Rea

This is the Naismith Bot. The bot ingests data from every player in the NBA from 1996-2022. That is, from every season it has data on every player. The bot can respond to help and hello commands. The main purpose of the Naismith Bot is to answer questions regarding NBA players from the years 1996-2022. It does so by linking to OpenAi. It is named the "Naismith Bot" after basketball's founder Dr. James Naismith.

Here are some example questions:

- 1) What college did Jae Crowder go to?
- 2) How many points did Chauncey Billups average in the year 1999?
- 3) What team did Ty Jerome get drafted to?
- 4) Who averaged the most points in 2005?

This bot is useful for sports nerds. It is kind of like Statmuse, but internalized to discord. The code to create the bot is similar to the Tony Bennett bot example, but it has been updated to answer questions about the NBA as James Naismith. There were also lots of local changes made for the Naismith bot to work.

Here is the command line of the Naismith Bot connecting to discord!

```
C:\Users\DS2002>python C:\Users\DS2002\Documents\bens_code.py
Bot Ready
[2023-12-11 16:26:56] [INFO      ] discord.client: logging in using static token
[2023-12-11 16:26:56] [INFO      ] discord.gateway: Shard ID None has connected to Gateway (Session ID: 474fec41d40c2559c7df349cdfd4954b).
Bot is ready
hi
Ben Alter#7865
Naismith Bot#7063
hi
[2023-12-11 16:27:07] [ERROR     ] discord.client: Ignoring exception in on_message
Traceback (most recent call last):
  File "C:\Miniconda\lib\site-packages\discord\client.py", line 441, in _run_event
    await coro(*args, **kwargs)
  File "C:\Users\DS2002\Documents\bens_code.py", line 112, in on_message
    answer = chat(message.content)
  File "C:\Users\DS2002\Documents\bens_code.py", line 70, in chat
    chatgpt_raw_output = chatcompletion(user_input, impersonated_role, explicit_input, chat_history).replace(f'{name}:',
    ..)
  File "C:\Users\DS2002\Documents\bens_code.py", line 47, in chatcompletion
    output = openai.ChatCompletion.create(
  File "C:\Miniconda\lib\site-packages\openai\api_resources\chat_completion.py", line 25, in create
    return super().create(*args, **kwargs)
  File "C:\Miniconda\lib\site-packages\openai\api_resources\abstract\engine_api_resource.py", line 153, in create
    response, _ = api_key = requestor.request(
```