

# DATA2410 - Datanettverk og Skytjenester

## Obligatory Assignment 2

Group 17

Andreas Torres Hansen (s338851)

Uy Quoc Nguyen (s341864)

Anders Hagen Ottersland (s341883)

Deadline:

April 14th 2021

## Contents

<b>1</b>	<b>How to run the script</b>	<b>2</b>
1.1	The server . . . . .	2
1.2	The client as a user . . . . .	3
1.3	The client as a bot . . . . .	5

## 1 How to run the script

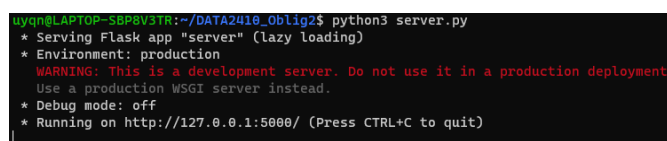
The chat has been implemented using Python. Furthermore, the modules that was used was Flask and Flask-RESTful. Thus, it is required to install these modules before the script can be successfully ran. Follow the instructions provided below to install these modules:

[Flask](#)

[Flask-RESTful](#)

### 1.1 The server

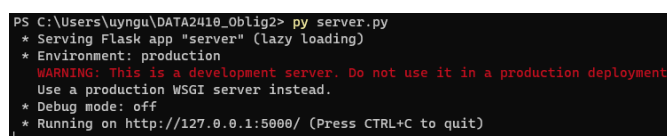
Assuming that the computer has Flask and Flask-RESTFUL successfully installed. One must first start the server script as shown below:



```
uyqn@LAPTOP-SBP8V3TR:~/DATA2410_Oblig2$ python3 server.py
* Serving Flask app "server" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 1: Running server.py from a Unix terminal

or using a Windows PowerShell:



```
PS C:\Users\uyngu\DATA2410_Oblig2> py server.py
* Serving Flask app "server" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 2: Running server.py from Windows PowerShell

The server can also run on a different host or/and port, and debug mode if desired. Check the help option by running the following command on a unix terminal:

```
$ python3 server.py --help
```

or in Windows PowerShell:

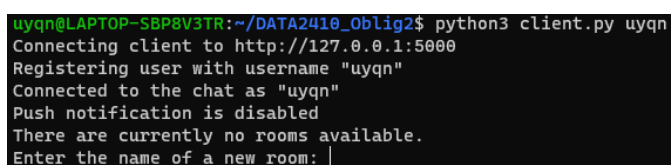
```
> py server.py -h
```

to learn more about the available optional arguments of the script.

After successfully starting the server script one can now continue to start the client script to connect to the API server.

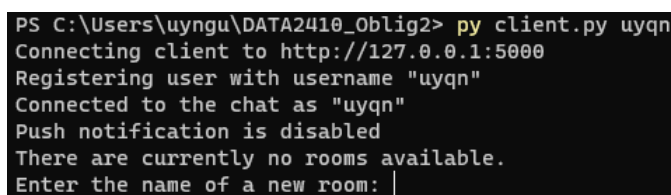
## 1.2 The client as a user

Given that the server is up and running. One can then run the client.py script to connect with a chosen username with



```
uyqn@LAPTOP-SBP8V3TR:~/DATA2410_Oblig2$ python3 client.py uyqn
Connecting client to http://127.0.0.1:5000
Registering user with username "uyqn"
Connected to the chat as "uyqn"
Push notification is disabled
There are currently no rooms available.
Enter the name of a new room: |
```

Figure 3: Running client.py from a Unix terminal



```
PS C:\Users\uyngu\DATA2410_Oblig2> py client.py uyqn
Connecting client to http://127.0.0.1:5000
Registering user with username "uyqn"
Connected to the chat as "uyqn"
Push notification is disabled
There are currently no rooms available.
Enter the name of a new room: |
```

Figure 4: Running client.py from Windows PowerShell

Note that a username must be provided in order to run the script. In the example screenshots above we ran the script connecting as user with username “uyqn”. Here we also have implemented several options for running the script. As previously mentioned one can use the following command in a unix terminal

```
$ python3 client.py —help
```

or in Windows PowerShell:

```
> py client.py —help
```

After the client has successfully started it will ask the user to specify the room the user want to join. It makes no sense to send messages while not in

a room. Thus, after specifying a room for instance a room called “chicken” as shown below:

```
uyqn@LAPTOP-SBP8V3TR:~/DATA2410_Oblig2$ python3 client.py uyqn
Connecting client to http://127.0.0.1:5000
Registering user with username "uyqn"
Connected to the chat as "uyqn"
Push notification is disabled
There are currently no rooms available.
Enter the name of a new room: chicken|
```

Figure 5: Entering a room called “chicken”

Upon entering a room a message will be displayed as shown below:

```
There are currently no rooms available.
Enter the name of a new room:
chicken
Creating a new room with room name "chicken"
Joining room with name "chicken"

Entering chicken in 2...
```

Figure 6: A little buffer time before joining a room

Then you should be successfully be joining a room in which you can send messages:

```
-----
-----
```

Figure 7: An empty chatroom

```
-----
-----
Success!|
```

Figure 8: Writing a message in the chatroom

The chatroom will automatically update the chatroom by calling the API and display any messages posted in that room:

```
-----  
Mon Apr 12 16:11:26 2021  
uyqn: Success!  
-----  
|
```

Figure 9: Chat history after sending a message in a room

### 1.3 The client as a bot

In order to make a bot join a chatroom simply run the script as before but include `-b` after the name. The bots and their responses has been saved as a json file and currently we have implemented 4 bots called Joe, Peter, Anna and Razor:

```
uyqn@LAPTOP-SBP8V3TR:~/DATA2410_Oblig2$ python3 client.py joe -b  
Connecting client to http://127.0.0.1:5000  
Registering user with username "Joe"  
Connected to the chat as "Joe"  
Push notification is disabled  
These rooms are currently available:  
['chicken']  
Enter the name of the room you want to join  
or enter a new name to create a new room:  
|
```

Figure 10: Joining the chat as a bot using Unix terminal

At this point the user uyqn has already created a room called chicken. Here one can join the room chicken by entering chicken as the input or just create a new name by providing another room name.

```
uyqn@LAPTOP-SBP8V3TR:~/DATA2410_Oblig2$ python3 client.py joe -b  
Connecting client to http://127.0.0.1:5000  
Registering user with username "Joe"  
Connected to the chat as "Joe"  
Push notification is disabled  
These rooms are currently available:  
['chicken']  
Enter the name of the room you want to join  
or enter a new name to create a new room:  
|
```

Figure 11: Choose the room you want to join

When the bot joins a room the bot will automatically send a greeting in the chat:

```
-----  
Mon Apr 12 16:11:26 2021  
uyqn: Success!  
  
Mon Apr 12 16:16:17 2021  
Joe: Hello, I am Joe  
  
-----  
|
```

Figure 12: Greeting sent by Joe the bot in the room chicken

The bots has been implemented to respond to a sentence if any of the trigger words has been mentioned. For instance

```
-----  
Mon Apr 12 16:11:26 2021  
uyqn: Success!  
  
Mon Apr 12 16:16:17 2021  
Joe: Hello, I am Joe  
  
Mon Apr 12 16:21:32 2021  
uyqn: joke  
  
Mon Apr 12 16:21:32 2021  
Joe: I ordered a chicken and an egg from Amazon. I'll let you know.  
  
-----  
|
```

Figure 13: Ask the bot to tell a joke!

The file 'bots.json' contains all the trigger words and responses of each bot. Check out the json file for more details.