

SAE 302: Developing communicating applications

Server Documentation

Language: English

Autor: Emeline CRISTEL Promotion: RT221

First Version

Sommaire:

- I. Introduction
- II. Installation of tools
- III. Server use
- IV. Code structure

I. Introduction

This documentation is the server documentation. Its purpose is to help a programmer to exploit the server file which is called "serv1.py"

II. Installation

In order to use the server without problems, you must first install the libraries used in this program.

To install these packages we will use pip. If you don't have pip you can install it with this command:

```
# python3 -m pip install
```

Once done we will install the packages used

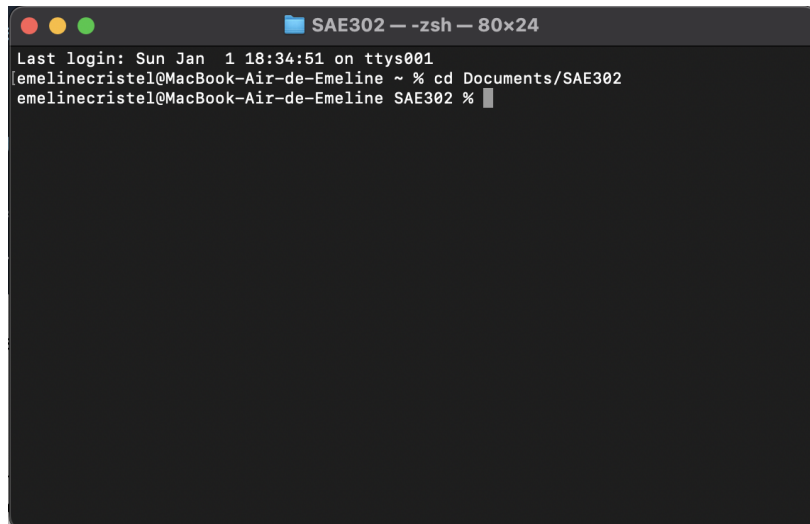
```
import socket
import platform
import threading
import psutil
```

Here you have to install socket and threading. For this we will use the command:

```
# pip install socket
```

III. Server use

To use the server, you must first launch it. To do this, open a terminal, access the directory where there is the server file.

A terminal window titled "SAE302 -- -zsh -- 80x24" with a dark background. The text inside shows a login session: "Last login: Sun Jan 1 18:34:51 on ttys001", followed by the prompt "emelinecristel@MacBook-Air-de-Emeline ~ % cd Documents/SAE302", and then "emelinecristel@MacBook-Air-de-Emeline SAE302 %".

```
SAE302 -- -zsh -- 80x24
Last login: Sun Jan 1 18:34:51 on ttys001
emelinecristel@MacBook-Air-de-Emeline ~ % cd Documents/SAE302
emelinecristel@MacBook-Air-de-Emeline SAE302 %
```

Then you can run the file with the command:

python3 serv1.py

Once the server is launched, you can connect a client to it.

IV. Code structure

Client part:

The client consists of two classes:

- **MainWindow class:** we find there the code of the graphical interface pyqt6 and all the commands to use the different buttons of the interfaces
- **Threadreception class:** there is the code to start the thread with the run function, and the code to stop the thread.

Server part:

the server contains three functions:

- **clients:** allows to start the server and accept the connection to the client. A thread starts
- **thread_reception:** there is a while loop that responds to all commands sent by the client
- **main**

What remains to be done:

We can then add a function to take the csv file into account. Then one could do asynchronous threading which would allow communication with multiple servers and clients.