

## Documentation for user monitoring interface



# Claudel Louis RT212

# TABLE OF CONTENTS

Installation and launch.....	1
Presentation of the client interface .....	2
Use of application .....	3

In this documentation, I will show you how to install, start and use the client interface and the server.

## **INSTALLATION AND LAUNCH**

The interface was built using python, the PyQt5 library, psutil library.

The first step is to have python installed on your computer.

The second step is to install the PyQt5 library and psutil library.

Once, these two steps accomplished.

You have to get the file python client.py and server.py which is on github.

link from github :

<https://github.com/Grievous400/R309/tree/master/Sae>

The third step is to launch the python program:

Now you have to go with your command prompt to the folder where the client.py file is located. You can also open the file with a python interpreter.

if you make with command prompt to launch the python file, you must write in your command prompt: "python3 client.py". Repeat the same thing for the server.

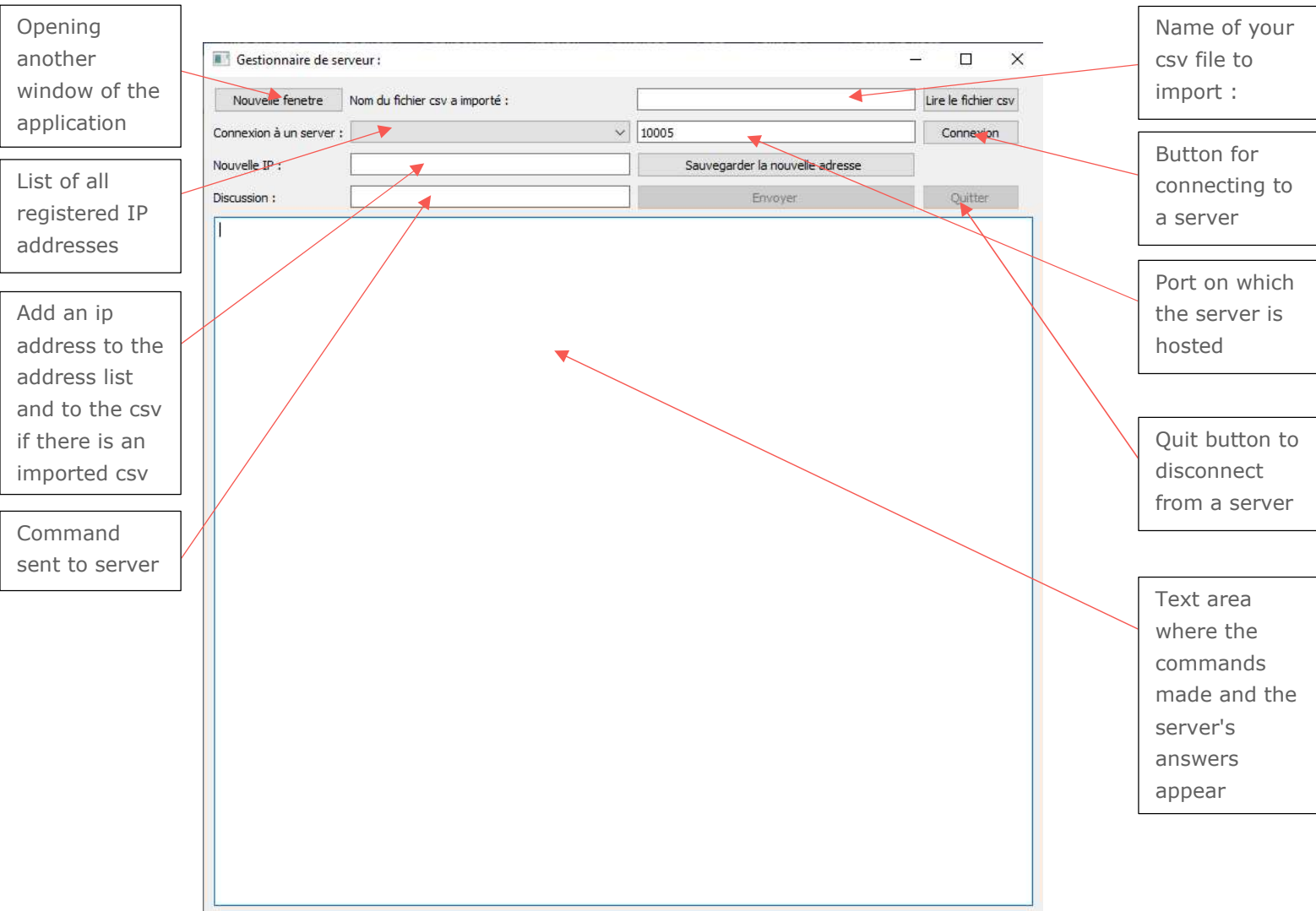
If this does not work, try the following command: "python client.py".

Normally if all the steps have been done correctly, you should see a graphical interface appear.

## PRESENTATION OF THE CLIENT INTERFACE

In this part, I will explain how to use the client interface:

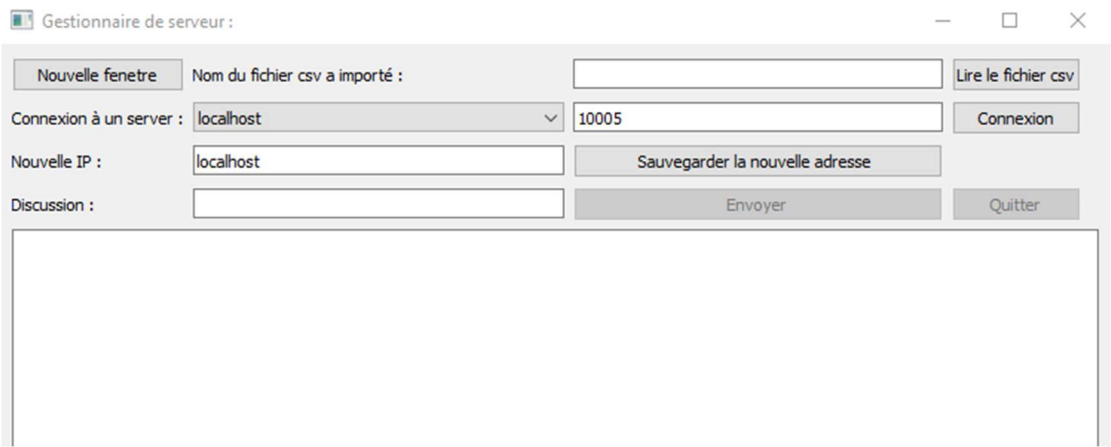
I will start by presenting you the different parts of the client interface.



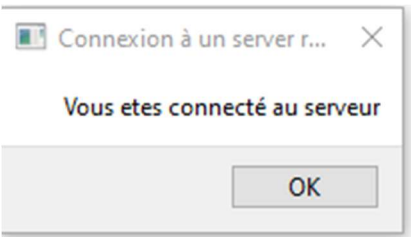
# USE OF APPLICATION

Once the server is launched, you do not need to touch it anymore.

By default, the client and server port are 10005. I will show you an example of connecting to a server. In this example, the server is on the same machine as my client so we will use the localhost address. You can import it via your csv file or save it directly via the interface. The format of the csv must be one address per line. I registered it via the interface and the address goes directly to the first address in the address list.



Once this is done, you can press the connected button. If you are connected, you will have a pop-up like this:



Once connected, you can send commands to the servers. You can write the commands in upper or lower case it doesn't matter. I will list the possible commands:

Disconnect	Disconnects the client from the server
Kill	Kills the server
Reset	Reset the server
OS	Return the server os

Ram	Return the used ram, the free ram and the total ram
Cpu	Return the processor usage over 5 seconds
Connexion information	Return the server name and its ip address
Ip	Return the ip address
Name	Return the server name
Python -version	Return the version of python
Ping address ip	Return the ping result of any address
Dos:command Linux:command Powershell:command Mac:command	Return the result of any command, as long as you specify which OS you are on. For windows, you can do in dos or in PowerShell.
Clear or cls	deletes the history of orders

To send commands, you must write in the chat line. Once the command is written, you must press the "envoyer" button to send it to the server.

To continue our example after the explanation of the different existing commands. We will test them:

OS:

Discussion :

Commande utilisé: OS  
OS = : Windows nt 10

RAM:

Discussion :

Commande utilisé: Ram  
La mémoire totale est 15 GB. Il y a 8 GB de ram utilisé et 7 GB de libre

CPU:

Discussion :

Commande utilisé: CPU  
Le cpu est utilisé à 1.0 % dans les 5 dernières secondes

## Connexion information:

Discussion :	<input type="text" value="connexion information"/>	<input type="button" value="Envoyer"/>	<input type="button" value="Quitter"/>
Commande utilisé: connexion information L ip de la machine est 192.168.56.1 et son nom est IUTC466			

## IP:

Discussion :	<input type="text" value="ip"/>	<input type="button" value="Envoyer"/>	<input type="button" value="Quitter"/>
Commande utilisé: ip 192.168.56.1			

## Name:

Discussion :	<input type="text" value="name"/>	<input type="button" value="Envoyer"/>	<input type="button" value="Quitter"/>
Commande utilisé: name IUTC466			

## Python --version:

Discussion :	<input type="text" value="python --version"/>	<input type="button" value="Envoyer"/>	<input type="button" value="Quitter"/>
Commande utilisé: python --version Python 3.10.5			

## Ping:

For example, I will ping google so 8.8.8.8

Discussion :	<input type="text" value="ping 8.8.8.8"/>	<input type="button" value="Envoyer"/>	<input type="button" value="Quitter"/>
Commande utilisé: ping 8.8.8.8 Envoi d'une requête 'Ping' 8.8.8.8 avec 32 octets de données : Réponse de 8.8.8.8 : octets=32 temps=14 ms TTL=113 Réponse de 8.8.8.8 : octets=32 temps=13 ms TTL=113 Réponse de 8.8.8.8 : octets=32 temps=14 ms TTL=113 Réponse de 8.8.8.8 : octets=32 temps=14 ms TTL=113  Statistiques Ping pour 8.8.8.8: Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%), Durée approximative des boucles en millisecondes : Minimum = 13ms, Maximum = 14ms, Moyenne = 13ms			

## Dos:command

For example, I will test with the dir command

Discussion :

dos:dir

Envoyer

Quitter

Commande utilisé: dos:dir  
Le volume dans le lecteur U s'appelle Data  
Le numéro de série du volume est 7CCE-8CEB  
  
Répertoire de U:\Documents\but2\pyhton\309\R309\Sae  
  
15/12/2022 10:28 <DIR> .  
15/12/2022 10:28 <DIR> ..  
15/12/2022 10:22 7 100 client.py  
15/12/2022 10:20 77 ip.csv  
15/12/2022 10:28 5 783 server.py  
3 fichier(s) 12 960 octets  
2 Rép(s) 3 291 959 296 octets libres

## Powershell:command

For example, I will test with the get-process command

Discussion :

powershell:get-process

Envoyer

Quitter

Commande utilisé: powershell:get-process  

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
217	15	7252	10308		3652	0	3CXMediaServer
310	19	7364	19212		3724	0	AdskAccessServiceHost
109	7	1172	5752		3640	0	alg
272	16	5656	25924	0,14	12500	1	ApplicationFrameHost
127	8	1564	6100		3632	0	armsvc
208	15	10416	19404	5,03	8200	0	audiodg
393	29	13464	17548		3520	0	client32
306	25	5292	17564		7848	1	client32
361	33	15896	33476		3604	0	CloudServicesWatcher
74	5	4268	4576	0,00	8	1	cmd
76	5	2204	3936	0,02	5720	1	cmd

## Linux:commad

For example, I will test with the ip a command

Discussion :

linux:ip a

Envoyer

Quitter

Commande utilisé: linux:ip a  
1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
inet 127.0.0.1/8 scope host lo  
valid\_lft forever preferred\_lft forever  
inet6 ::1/128 scope host  
valid\_lft forever preferred\_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000  
link/ether 08:00:27:61:e2:66 brd ff:ff:ff:ff:ff:ff  
inet 192.168.1.84/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3  
valid\_lft 86232sec preferred\_lft 86232sec  
inet6 2a02:842b:114:7901:1523:779a:73ed:2d21/64 scope global temporary dynamic  
valid\_lft 604633sec preferred\_lft 85882sec  
inet6 2a02:842b:114:7901:a00:27ff:fe61:e266/64 scope global dynamic mngtmpaddr noprefixroute  
valid\_lft 604634sec preferred\_lft 604634sec  
inet6 fe80::a00:27ff:fe61:e266/64 scope link noprefixroute  
valid\_lft forever preferred\_lft forever

To disconnect from the server, you have two options. The first is to press the "quit" button and the second is to send "disconnect" as a command.

Now you are ready to use the client and server interface and you know all the possible commands.