

Présentation du projet PYTHON SOCKET

- Python Socket permet de gérer les connexions par [socket](#). Un socket est un objet qui permet d'ouvrir une connexion avec une machine locale ou distante, et d'échanger avec elle.
- Sur ce projet nous sommes en local, nous utilisons [XAMP](#) avec l'interface [PHPMyAdmin](#).
- Le but est de communiquer avec un code [serveur](#) et [client](#) en langage [Python](#).
- Le code serveur ouvre la connexion avec le socket sur un port.
- Le client se présente sur le socket et un autre port.
- Mon objectif était de relier le serveur à une base de donnée [MYSQL](#) pour inscrire un client pour qu'il puisse s'authentifier au travers d'une table nommée authentication.
- Cette table récupère le nom d'utilisateur, le mot de passe et l'adresse IP.
- De plus, j'ai ajouté une table nommée [Connexion](#) qui permet de [sauvegarder](#) les [échanges](#) entre le [serveur](#) et le [client](#) : les [données](#), la [date](#) et l'[heure](#), le [port](#) de connexion et l'[IP](#).
- J'ai ensuite [encodé](#) le [mot de passe](#) du [client](#) dans la base de donnée. C'est le serveur qui encode à la réception et l'enregistre dans la base cela améliore la [sécurité](#) de la base de donnée, j'ai utilisé l'encodage en [Base64](#).

```

import socket
from datetime import datetime

def client_program():
    host = socket.gethostname() # as both code is running on same pc
    port = 25 # socket server port number

    client_socket = socket.socket() # instantiate
    client_socket.connect((host, port)) # connect to the server

    message_introduction = input("Voulez-vous vous inscrire ? Tapez 'oui' ou 'non' ")
    client_socket.send(message_introduction.encode())

    if message_introduction=="oui":
        print("Inscription compte client")
        message_inscri_id = input("écrivez votre identifiant : ")
        message_inscri_mdp = input("écrivez votre mot de passe : ")
        client_socket.send(message_inscri_id.encode()) # send message
        client_socket.send(message_inscri_mdp.encode())
    else:
        print("connectez-vous avec vos identifiants")

    message_id = input("Votre identifiant -> ") # again take input

    message_mdp= input("Votre mot de passe -> ")

    client_socket.send(message_id.encode()) # send message
    client_socket.send(message_mdp.encode())

    while message_id.lower().strip() != 'bye':
        now = datetime.now()
        dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
        print(dt_string)
        data = client_socket.recv(1024).decode() # receive response

        print('Le serveur: ' + data) # show in terminal

        message = input("-> ")# again take input
        client_socket.send(message.encode())

```

```
client_socket.close() # close the connection
```

```
if __name__ == '__main__':  
    client_program()
```

```

import socket
import MySQLdb #C:/Users/33679/AppData/Local/Programs/Python/Python39/python.exe -m pip install
mysqlclient
import sys
import base64
from datetime import datetime

# datetime object containing current date and time
now = datetime.now()

dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
print("date =", dt_string)

def server_program():
    db = MySQLdb.connect(host="localhost", # your host, usually localhost
                        user="pma",      # your username
                        passwd="",       # your password
                        db="serveur_client") # name of the data base

    # get the hostname
    host = socket.gethostname() # fetch the hostname
    port = 25 # protocole SMTP uses the TCP ports 25 and 587
    server_socket = socket.socket() # get a socket instance
    server_socket.bind((host, port)) # bind host address and port together
    print("le serveur est démarré....")

    server_socket.listen(2) # configure how many client the server can listen simultaneously
    conn, address = server_socket.accept() # accept new connection
    print("Connection from: " + str(address))

    message_introduction = conn.recv(1024).decode()
    if message_introduction=='oui':

        insertrec=db.cursor()
        message_inscri_id = conn.recv(1024).decode()
        message_inscri_mdp = conn.recv(1024).decode()

        data_base64_message = message_inscri_mdp
        base64_message = data_base64_message.encode('ascii')
        base64_bytes = base64.b64encode(base64_message)

```

```

message_bytes = base64_bytes.decode('ascii')

try:
    sqlquery_inscription=f'INSERT INTO authentication VALUES ("{message_inscri_id}", "{message_bytes}",
"{address[0]}")'
    insertrec.execute(sqlquery_inscription)
    db.commit()
except:
    print("Inscription deja faite") #levée une erreur, je pensais a l'erreur message_inscri_id==id car id clef
primaire
else:
    print("pas d'inscription requise")

insertrec=db.cursor()
sqlquery_authentication= 'SELECT * from authentication'
insertrec.execute(sqlquery_authentication)
list_authentication=insertrec.fetchall() #recupere les lignes de la db.

data_id = conn.recv(1024).decode()
data_mdp = conn.recv(1024).decode()

#decodage du mdp
message = data_mdp
base64_message = message.encode('ascii')
base64_bytes = base64.b64encode(base64_message)
base64_message = base64_bytes.decode('ascii')
#message = message_bytes.decode('ascii')

compteur=0
for ligne in range (len (list_authentication)):

    if data_id==list_authentication[ligne][0]:
        print("l'identifiant est correct")

    if base64_message==list_authentication[ligne][1]:
        print("le mot de passe est correcte")

    if address[0]==list_authentication[ligne][2]:
        compteur+=1

```

```

        print("l'adresse ip est correcte")
        break

if compteur==0:
    print("la connexion est refusée pour", data_id)
    sys.exit()

data = "la connexion est bien établie"
conn.send(data.encode())

id=-1
while True:
    id+=1
    now = datetime.now()
    dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
    print(dt_string)
    data = conn.recv(1024).decode() # receive data stream. it won't accept data packet greater than 1024 bytes
    if not data:
        break # if data is not received break
    print("Le client: " + str(data))
    insertrec = db.cursor()
    sqlquery=f'INSERT INTO connexion VALUES ("{dt_string}", "client_{data_id}", "{address[0]}", "{address[1]}",
"{data}")'
    insertrec.execute(sqlquery)
    #premier crochet choisit la ligne dans la table
    #le second crochet choisit l'élément de cette ligne
    db.commit() #commettre=faire/ écrire dans la db
    data = input(' -> ')

    insertrec = db.cursor()
    sqlquery=f'INSERT INTO connexion VALUES ("{dt_string}", "serveur", "{host}", "{port}", "{data}")'
    insertrec.execute(sqlquery)
    db.commit()

    conn.send(data.encode()) # send data to the client

conn.close() # close the connection
db.close()

```

```
if __name__ == '__main__':  
    server_program()
```

Visual Studio Code interface showing two Python files: `2021_11_11_Server_dm3.py` and `2021_11_11_Client_dm3.py`.

2021_11_11_Server_dm3.py (server_program):

```
62 message = data_md5
63 base64_message = message.encode('ascii')
64 base64_bytes = base64.b64encode(base64_message)
65 base64_message = base64_bytes.decode('ascii')
66 #message = message_bytes.decode('ascii')
67
68 compteur=0
69 for ligne in range (len (list_authentication)):
70
71     if data_id==list_authentication[ligne][0]:
72         print("l'identifiant est correct")
73
74         if base64_message==list_authentication[ligne][1]:
75             print("le mot de passe est correcte")
76
77             if address[0]==list_authentication[ligne][2]:
78                 compteur+=1
79                 print("l'adresse ip est correcte")
80                 break
81
82
83 if compteur==0:
84     print("la connexion est refusée pour", data_id)
85     sys.exit()
86
87 data = "la connexion est bien établie"
88 conn.send(data.encode())
89
90
91 id=-1
92 while True:
93     id+=1
94     now = datetime.now()
95     dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
96     print(dt_string)
```

2021_11_11_Client_dm3.py:

```
1 import socket
2 from datetime import datetime
3
4 # pseudo=input("Votre identifiant----->")
5 # motdepasse=input("Votre motdepasse----->")
6
7 def client_program():
8     host = socket.gethostname() # as both code is running on same pc
9     port = 25 # socket server port number
10
11     client_socket = socket.socket() # instantiate
12     client_socket.connect((host, port)) # connect to the server
13
14     message_introduction = input("Voulez-vous vous inscrire ? Tapez 'oui' ou 'non' ")
15     client_socket.send(message_introduction.encode())
16
17     if message_introduction=="oui":
18         print("Inscription compte client")
19         message_inscri_id = input("écrivez votre identifiant : ")
20         message_inscri_mdp = input("écrivez votre mot de passe : ")
21         client_socket.send(message_inscri_id.encode()) # send message
22         client_socket.send(message_inscri_mdp.encode())
23     else:
24         print("connectez-vous avec vos identifiants")
25
26     message_id = input("Votre identifiant -> ") # again take input
27
28     message_mdp= input("Votre mot de passe -> ")
29
30     client_socket.send(message_id.encode()) # send message
31     client_socket.send(message_mdp.encode())
32
33
34 while message_id.lower().strip() != 'bye':
35     now = datetime.now()
36     dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
```

TERMINAL:

```
PS C:\Users\33679\Documents\Code\Python\code> py 2021_11_11_Server_dm3.py
date = 18/09/2022 13:49:01
le serveur est démarré...
Connection from: ('192.168.56.1', 64917)
l'identifiant est correct
le mot de passe est correcte
l'adresse ip est correcte
18/09/2022 13:49:43
Le client: Serveur vous me recevez ?
-> Tout à fait, client
18/09/2022 13:50:54
```

Output:

```
Voulez-vous vous inscrire ? Tapez 'oui' ou 'non' oui
Inscription compte client
écrivez votre identifiant : Romain
écrivez votre mot de passe : Romain_1995
Votre identifiant -> Romain
Votre mot de passe -> Romain_1995
18/09/2022 13:49:43
Le serveur: la connexion est bien établie
-> Serveur vous me recevez ?
18/09/2022 13:50:44
Le serveur: Tout à fait, client
->
```


FichierEditionSélectionAffichageAtteindreExécuterTerminalAide

2021_11_11_Server_mdpcoder.py - code - Visual Studio Code

2021_11_11_Server_mdpcoder.py X

2021_11_11_Server_mdpcoder.py > ...

```
1 import socket
2 import MySQLdb #C:/Users/33679/AppData/Local/Programs/Python/Python39/python.exe -m pip install m
3 import sys
4 import base64
5 from datetime import datetime
6
7
8 # datetime object containing current date and time
9 now = datetime.now()
10 # dd/mm/YY H:M:S
11 dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
12 print("date =", dt_string)
13
14
15 def server_program():
16     db = MySQLdb.connect(host="localhost", # your host, usual
17                          user="pma", # your username
18                          passwd="", # your password
19                          db="serveur_client") # name of the
20
21     # get the hostname
22     host = socket.gethostname() # fetch the hostname
23     port = 25 # protocole SMTP uses the TCP ports 25 and 587
24     server_socket = socket.socket() # get a socket instance
25     server_socket.bind((host, port)) # bind host address and port
26     print("le serveur est démarré....")
27
28     server_socket.listen(2) # configure how many client the serv
29     conn, address = server_socket.accept() # accept new connect
30     print("Connection from: " + str(address))
31
32     message_introduction = conn.recv(1024).decode()
33     if message_introduction=="oui":
34         insertrec=db.cursor()
```

2021_11_11_Client_mdpcoder.py X

2021_11_11_Client_mdpcoder.py > ...

```
1 import socket
2 import base64
3 from datetime import datetime
4
5 # pseudo=input("Votre identifiant----->")
6 # motdepasse=input("Votre motdepasse----->")
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
Apache		13572 4684	80, 443	Stop Admin Config Logs
MySQL		15272	3306	Stop Admin Config Logs
FileZilla		17288	21, 14147	Stop Admin Config Logs
Mercury		22228	25, 79, 105, 106, 110, 143, 2224	Stop Admin Config Logs
Tomcat				Start Admin Config Logs

12:12:20 [mysql] Status change detected: running

12:12:20 [filezilla] Status change detected: running

12:12:20 [mercury] Attempting to start Mercury app...

12:12:21 [tomcat] Attempting to start Tomcat app...

12:12:21 [mercury] Status change detected: running

12:12:25 [tomcat] Tomcat Started/Stopped with errors, return code: 1

12:12:25 [tomcat] Make sure you have Java JDK or JRE installed and the required ports are free

12:12:25 [tomcat] Check the "/xampp/tomcat/logs" folder for more information

ConfigNetstatShellExplorerServicesHelpQuit

PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL JUPYTER

le serveur est démarré....
Connection from: ('192.168.56.1', 51434)
[

PS C:\Users\33679\Documents\Code\Python\code>
PS C:\Users\33679\Documents\Code\Python\code> py 2021_11_11_Client_dm2.py
Voulez-vous vous inscrire ? Tapez 'oui' ou 'non' [

L 1, col 1 Espaces : 4 UTF-8 LF Python 3.9.0 64-bit

localhost / 127.0.0.1 / serveur_client

localhost/phpmyadmin/index.php?route=/sql&server=1&db=serveur_client&table=authentification&pos=0

110%

localhost/phpmyadmin

phpMyAdmin

RécentesPréférées

Nouvelle base de données

information_schema

mysql

performance_schema

phpmyadmin

serveur_client

Nouvelle table

authentification

connexion

essaiconnect

test

Parcourir

Structure

SQL

Rechercher

Insérer

Exporter

Importer

Privilèges

Opérations

Suivi

Déclencheurs

Affichage des lignes 0 - 4 (total de 5, traitement en 0,0005 seconde(s).) [ip: [BLOB - 12 O]... - [BLOB - 12 O]...]

SELECT * FROM `authentification` ORDER BY `ip` DESC

Profilage

Éditer en ligne

Éditer

Expliquer SQL

Créer le code source PHP

Actualiser

Tout afficher

Nombre de lignes : 25

Filtrer les lignes: Chercher dans cette ta

Trier par clé : Aucun(e)

+ Options

←T→

id

mdp

ip

Éditer

Copier

Supprimer

mlk

bmjlbw==

192.168.56.1

Éditer

Copier

Supprimer

mimi

bWlyYWVsZQ==

192.168.56.1

Éditer

Copier

Supprimer

Romain

Um9tYWVlXzE5OTU=

192.168.56.1

Éditer

Copier

Supprimer

achille

YWNoaWxsZTI=

192.168.56.1

Éditer

Copier

Supprimer

nicolasok

amVyb3hkdQ==

192.168.56.1

Tout cocher

Avec la sélection :

Éditer

Copier

Supprimer

Exporter

Tout afficher

Nombre de lignes : 25

Filtrer les lignes: Chercher dans cette ta

Trier par clé : Aucun(e)

Opérations sur les résultats de la requête

Imprimer

Copier dans le presse-papiers

Exporter

Afficher le graphique

Créer une vue

Conserver cette requête SQL dans les signets

Intitulé :

Signet visible pour les autres utilisateurs

Conserver cette requête SQL dans les signets

Console de requêtes SQL

localhost / 127.0.0.1 / serveur_client X

phpMyAdmin

RécentesPréférées

Nouvelle base de données

information_schema

mysql

performance_schema

phpmyadmin

serveur_client

Nouvelle table

authentification

connexion

essaiconnect

test

localhost/phpmyadmin/index.php?route=/sql&server=1&db=serveur_client&table=connexion&pos=0

localhost / 127.0.0.1 » Base de données : serveur_client » Table : connexion

ParcourirStructureSQLRechercherInsérerExporterImporterPrivilègesOpérationsSuiviDéclencheurs

⚠ La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les cases à cocher ainsi que les liens Éditer, Copier et Supprimer ne sont pas disponibles.

✔ Affichage des lignes 0 - 24 (total de 26, traitement en 0,0006 seconde(s)) [port: [BLOB - 2 O]... - [BLOB - 5 O]...]

SELECT * FROM `connexion` ORDER BY `port` ASC

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >>

☐ Tout afficher

Nombre de lignes : 25

Filtrer les lignes: Chercher dans cette ta

+ Options

date_time	type	ip	port	donnée
10/11/2021 19:05:24	serveur	DESKTOP-HIS2V51	25	ou des rois
11/11/2021 15:02:49	serveur	DESKTOP-HIS2V51	25	test reponse serveur
02/01/2022 15:14:34	serveur	DESKTOP-HIS2V51	25	hey
02/01/2022 16:27:40	serveur	DESKTOP-HIS2V51	25	a
02/01/2022 16:34:59	serveur	DESKTOP-HIS2V51	25	hello je suis ton serveur
18/09/2022 12:34:08	serveur	DESKTOP-HIS2V51	25	Que voulez-vous ?
18/09/2022 12:34:34	serveur	DESKTOP-HIS2V51	25	Cela fonctionne
18/09/2022 13:39:20	serveur	DESKTOP-HIS2V51	25	on regarde
18/09/2022 13:42:48	serveur	DESKTOP-HIS2V51	25	a
18/09/2022 13:47:46	serveur	DESKTOP-HIS2V51	25	Oui mon client ?
18/09/2022 13:49:43	serveur	DESKTOP-HIS2V51	25	Tout à fait, client
02/01/2022 16:27:40	client_plk	192.168.56.1	50509	Y29ldXI=
02/01/2022 16:28:01	client_plk	192.168.56.1	50509	plk
02/01/2022 16:34:59	client_casio	192.168.56.1	50622	bonjour je suis le client
18/09/2022 12:34:08	client_nicolaserres	192.168.56.1	51434	bonjour Mr server
18/09/2022 12:34:34	client_nicolaserres	192.168.56.1	51434	Verifiez le bon fonctionnement
10/11/2021 19:05:24	client_nicolas	192.168.1.30	52449	Yop mec on est les meilleursd
11/11/2021 15:01:08	client_nicolas	192.168.1.30	53167	hello
11/11/2021 15:02:49	client_nicolas	192.168.1.30	53503	test envoie client
11/11/2021 15:11:29	client_nicolas	192.168.1.30	54771	a
02/01/2022 15:14:34	client_nico25	192.168.56.1	59618	hi
02/01/2022 15:14:41	client_nico25	192.168.56.1	59618	je suis le client
18/09/2022 13:39:20	client_mimi	192.168.56.1	64677	j'ai réussi ?
18/09/2022 13:42:48	client_nicolasok	192.168.56.1	64741	a

Console de requêtes SQL