Internet Technology
Assignment 1

Priyanshu Tirkey
05
BCSE IV (Final Year)

Notes

- The assignment consists of two parts:
  - In **first part** no authentication is implemented and it's input and output is as defined in the assignment.

    i.e. ./client.py <ip> <port> <… get/put requests...> from terminal

    <response of each get request> in output
  - In **second part** authentication is implemented and code used in first part is modifed to to handle multiple logged in users/ managers/guests.
  - The slides contain output for only second part because output for first part is self explainatory.
- Output generated in auth_server.py are just logs.
- For each output of client.py note what command which generates that output is also mentioned.
- More details about the code and implementation is given in "auth_server.py" and "Assignment1_execution_flow.pdf"

Running auth_server.py

- **Starting the server**

```
[hiro@awesomebox assignment1]$ python auth_server.py
```

- **Output when a client connects**

```
[hiro@awesomebox assignment1]$ python auth_server.py
Receiving from ('127.0.0.1', 43588)
```

```
[hiro@awesomebox assignment1]$ python client.py
Enter the username ("guest" for temp. session):
```

- **Output when server receive a put request from a user (in this case “admin”) :**

<sender username>: ‘put <attribute> <value>

```
admin: 'put name admin'
```

- **Output when server receive a put request from a user (in this case “admin”) :**

<sender username>: ‘get <attribute>

```
admin: 'get name'
```

Running client.py

- One can run client form terminal by using **$./python client.py.** In case ip and port are provided the default would be overwritten.

```
HOST, PORT = "localhost", 9999
if len(args) == 3:
    HOST, PORT = args[1], args[2]
```

- Once program is started answer the queries server asks

```
[hiro@awesomebox assignment1]$ python client.py
Enter the username ("guest" for temp. session):
admin
Enter the password :
admin
Unknown username. Do you wish to signup(y/n)
y
Make it a manager account(y/n)
y
manager authenticated.
You may start making GET and PUT requests
```

- If the user enter wrong password or does not wishes to sign up the process jumps to question of asking the username and subsequent questions are asked again.
- Enter “EXIT” in order to teminate the client program.

Outputs client.py

- Output when PUT request is made

  server: <attribute> stored.

```
put name admin
server: name stored.
```

- Output when GET request is made

  Type 1: manager asks for all data in server.

  server: __all__ = <a dictionary where key is a username and value is all the data stored by him/her

```
get __all__
server: __all__ = {'admin': {'name': 'admin', 'place': 'kolkata'}, 'hiro': {'place': 'siliguri'}}
```

  Type 2: manager asks for a paricular user's database.

  Server: [<username>] = <dictionary of all data stored by the user>

```
get [hiro]
server: [hiro] = {'place': 'siliguri'}
```

  Type 3: manager/user asks for a paricular data stored by him/her.

  server: <attribute name> = <value>

```
get place
server: place = kolkata
```

- Output when unauthorized access

```
get __all__
server: Unauthorised access
```