Une image contenant texte

Description générée automatiquement

**ARCHIVING TOOL: USER GUIDE**

**LEANG Denis**



2021-2022

FISE 2

Telecom Saint-Etienne

Table des matières

[I – About the project 2](#_Toc92671747)

[II - Getting started 3](#_Toc92671748)

[ **Prerequisites** 3](#_Toc92671749)

[ **Installation** 3](#_Toc92671750)

[III - Usage 4](#_Toc92671751)

[ **Json configuration file** 4](#_Toc92671752)

[1. IP adress 6](#_Toc92671753)

[2. Gmail adress 6](#_Toc92671754)

[ **SMB share server** 7](#_Toc92671755)

[1. Network activation 7](#_Toc92671756)

[2. Folder creation 8](#_Toc92671757)

[ **Task scheduler** 10](#_Toc92671758)

[1. Batch file 10](#_Toc92671759)

[2. Windows Task Scheduler 11](#_Toc92671760)

[IV – Conclusion 12](#_Toc92671761)

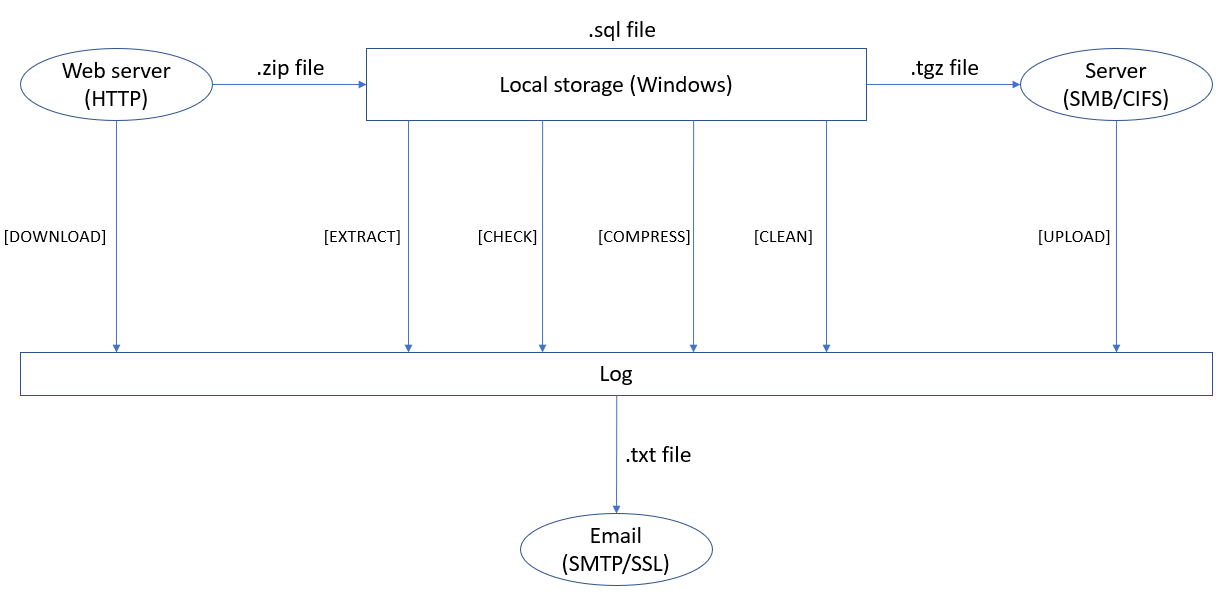
# **I – About the project**

This is a friendly user guide for my Python archiving tool. This script allows a user to do these successive actions:

1. Download a zip file from a web server (HTTP)
2. Extract that zip file
3. Check whether its content is up to date and valid
4. Compress it after validation into a tgz file
5. Send the tgz file to a server (SMB/CIFS)
6. (Optional) Send an email stating if operation succeeded or not

A logfile stores useful information throughout the whole operations to check if everything works properly or not. It can be attached to the email if the user chooses to do so.

Here is a simple drawing I have made at the beginning of the project to summarize the operations:



**Figure 1: Simple representation of the operations**

This script is aimed at Windows workstations for I have developed my script on Windows 10 using Visual Studio Code. Therefore, it may not work for MacOS or Linux users.

# **II - Getting started**

## **Prerequisites**

Firstly, you will need to have Python 3 installed in your computer. If you wish to check if you already have a version of Python on your device, type in your terminal:

* **py -V**

Une image contenant texte

Description générée automatiquement

**Figure 2: Check Python’s version**

If you do not have any, just install python [here](https://www.python.org/downloads/).

## **Installation**

I only used two external libraries which are *requests* and *pysmb*. To install them, just run these commands in your terminal:

* **py -m pip install pysmb**
* **py -m pip install requests**

Une image contenant texte

Description générée automatiquement

**Figure 3: pysmb library installation**

# **III - Usage**

## **Json configuration file**

The first thing that you will have to setup is the ***config.json*** file.

Une image contenant texte

Description générée automatiquement

**Figure 4: json.config file**

The names of the json objects are conveniently named after the classes in the code. But all you must worry about as a user are the following elements. Changing the values in which “NO CHANGE” is specified is not advised:

* **archive**:
  + **“url”**: NO CHANGE
  + **“zip\_file”**: NO CHANGE
  + **“file”**: NO CHANGE
  + **“log\_file”**: Name of the generated log file
  + **“expiring\_time”**: Duration (in seconds) after which a file is considered outdated
* **smb\_client**:
  + **“username”**: Current windows session username
  + **“password”**: Current windows session password
  + **“machine\_name”**: Whatever name you want to enter
  + **“server\_name”**: Name of the current PC
  + **“ip\_adress”**: IP address
  + **“smb\_share”**: Name of the shared samba server (see after)
  + **“save\_duration”**: Duration (in seconds) after which a file will be deleted in the smb server depending on its last modification date
* **archive**:
  + **“sender\_adress”**: An E-mail adress for the expeditor (Note: Gmail is advised)
  + **“password”**: Password for the adress above
  + **“port”**: NO CHANGE
  + **“smtp\_server”**: NO CHANGE
  + **“receiver\_adress”**: An E-mail adress for the receiver (Note: Gmail is advised)
  + **“send\_mode”**: Put on or off depending on if you want to receive mails
  + **“attach\_mode”**: Put on or off depending on if you want to have logs attached in the mails

### **IP adress**

If you have trouble finding your correct IP address, follow these steps:

1. Type *ipconfig* inside a terminal
2. Get the correct IP adress in front of IPv4:

Une image contenant texte

Description générée automatiquement

**Figure 5: ipconfig command to find IP adress**

### **Gmail adress**

Having a throwaway Gmail adress for the sender is advised for security reasons because you will have to enter login credentials inside the ***config.json*** to start the script. You must activate this mode: *“Allow less secure apps”* to on in Gmail. It is essential if you wish to receive the E-mails from the sender adress as a receiver. To find this mode, go to:

* Settings 🡪 Security 🡪 Access to less secure apps

## **SMB share server**

Now that the ***config.json*** is complete, you will now have to setup the SMB share server. You will have to follow these steps to ensure that everything will run smoothly later.

### **Network activation**

Ensure that you have Network activated on your Windows workstation. It is normally activated by default. If not, follow these steps:

1. Go to control panel and click on *Network and Internet*

Une image contenant texte

Description générée automatiquement

**Figure 6: Network configuration**

1. Go to *Network and Sharing Center*

Une image contenant texte

Description générée automatiquement

**Figure 6: Network configuration**

1. Finally in *Advanced sharing options*, check everything

Une image contenant texte

Description générée automatiquement

**Figure 6: Network configuration**

### **Folder creation**

You will then have to create a folder in the root of the code’s folder like this:

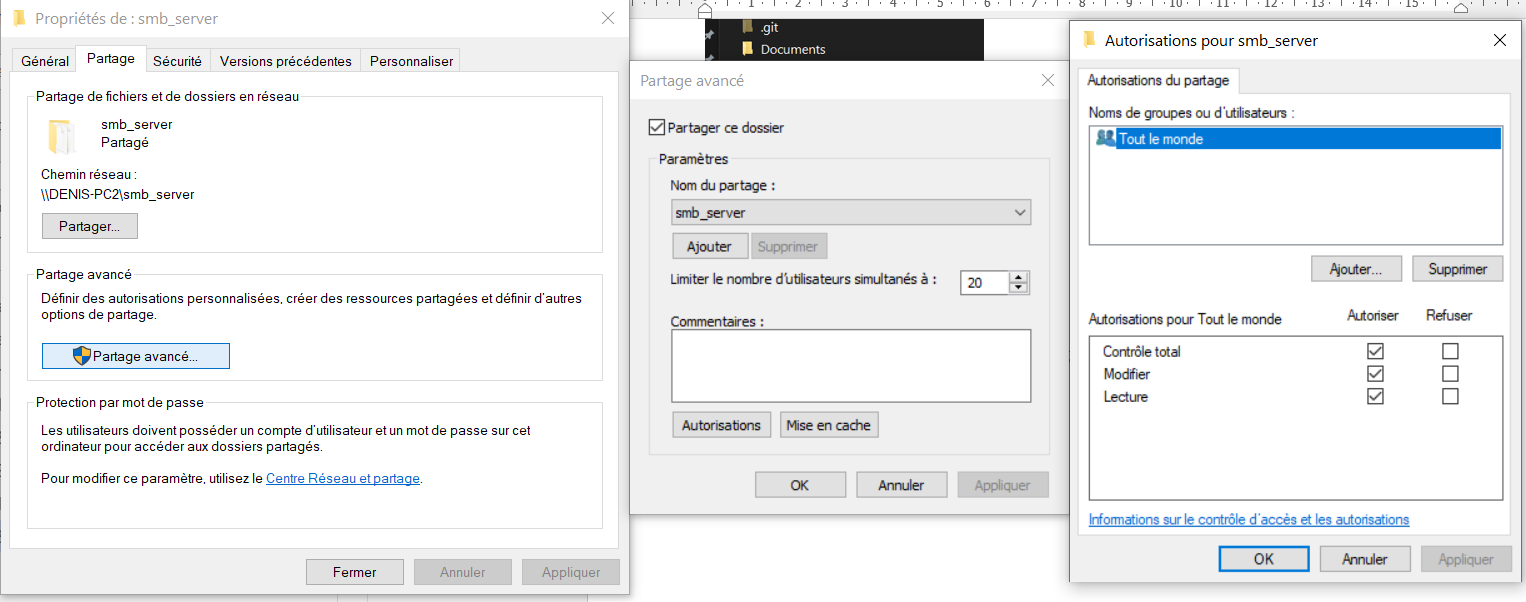
Une image contenant texte, capture d’écran, écran

Description générée automatiquement

**Figure 7: SMB folder creation**

It is mandatory to create that folder here for the code to work properly so do not place it anywhere else.

Then, activate those settings by right clicking the folder to setup the SMB server:



**Figure 8: SMB folder creation**

If everything went smoothly, you should now see this on your Windows explorer:

Une image contenant texte

Description générée automatiquement

**Figure 9: SMB folder in Network tab**

The SMB share is now ready to use.

## **Task scheduler**

Everything should now be ready for the code to work. If you run ***server.py*** and ***main.py*** (in that order) on any IDE, the script will work. In our case, we want it to be automatic. To do so you need to follow these steps.

### **Batch file**

You should see a start.bat file inside the folder. Inside it are these lines:



**Figure 10: start.bat file content**

* First line: Boots up the web server by launching ***server.py***
* Second line: Create a new task on Windows Task Scheduler which will launch ***main.py*** daily at 00:00

You must change the path to match with your folder’s path inside the batch file. It is also possible for you to change the frequency at which ***main.py*** is executed. For more information on how to do so, you can go [here](https://docs.microsoft.com/en-us/previous-versions/orphan-topics/ws.10/cc772785(v=ws.10)?redirectedfrom=MSDN).

When everything is changed, you can now run ***start.bat*** by double clicking it. One window will appear:

Une image contenant texte

Description générée automatiquement

**Figure 11: server.py launched**

### **Windows Task Scheduler**

You now must add path for the script to work in Task Scheduler.

1. Go to Task Scheduler, right click “Archiving Tool” and Properties:
2. Click on Actions and Modify

Une image contenant texte

Description générée automatiquement

**Figure 12: Path configuration**

1. Add the path to your project’s root folder inside this:

Une image contenant texte

Description générée automatiquement

**Figure 12: Path configuration**

Note that ***server.py*** will run indefinitely until the workstation is shut down. So, beware to click again on ***start.bat*** if you close your computer at some point else the script will run but fail with the operations. You do not need to modify the task which runs ***main.py*** whatsoever so just type “N” if prompted to.

## **Log Storage**

In the current state of the code, the logfile is deleted at the end of the operations. This is to ensure that it does not pollute the current folder. However, if you wish to keep it, you will need to delete this line in ***main.py*** (L47): archive.delete\_log\_file().

# **IV – Conclusion**

Everything is now ready and set to work properly. You can thus enjoy storing files with this Archiving Tool.

Une image contenant texte

Description générée automatiquement

**Figure 13: Successful transfer**

**Contact**: denis.leang@gmail.com