Python script for downloading time recordings for all devices within a spefic time range

There are two main scripts :

- download_recordings.py
- download_recordings_tdms.py

Script 1: download_recordings.py

download_recordings.py is an example of an python script for automatated download and delete of time recordings. The time recording data is returned in json format one channel at a time

1. The script will logon to pchcloud or local server.
2. Query for devices
3. For each device it will query for any recordings within a time range (default is the last day). If any then download and (optionally) delete the recoding on the server

Script 2: download_recordings_tdms.py

download_recordings_tdms.py is an example of an python script for automatated download in *National Intstrument tdms* format, save, and delete of time recordings.

The downloaded files are saved on the local disk in the path specified by the "download_path" in the config.json

1. The script will logon to pchcloud or local server.
2. Query for devices
3. For each device it will query for any recordings within a time range (default is the last day). If any then download in *tdms* format, save the file and (optionally) delete the recoding on the server

Install

The installation steps may be different on different platforms. However the basic priciple is the same.

1. Create virtual environment

```
py -m venv env
```

2. Activate virtual environment

```
.\env\Scripts\activate
```

3. Install requirements listed in requirements.txt

```
pip3 install -r requirements.txt
```

or if using python3.8 use

```
pip3 install -r requirements3_8.txt
```

Setting up

In order for the script to run - 2 files are required. The files must be locatated in the same folder as the script

1. **hosts.json** for setting up urls to backend-services

```
{
    "pchcloud": {
        "backend": "https://pchcloud.pch-engineering.dk/backend",
        "usermanager": "https://pchcloud.pch-engineering.dk/usermanager",
        "devicemanager": "https://pchcloud.pch-engineering.dk/devicemanager"
    },

    "local": {
        "backend": "http://localhost:5000/api",
        "usermanager": "http://localhost:5020/api",
        "devicemanager": "http://localhost:5030/api"
    },

}
```

The example has two known hosts.

- **"pchcloud"** can be used if the data is download from https://pchcloud.pch-engineering.dk
- **"local"** can be used if the backend-services are running on the same pc as the script is executed.

More hosts can be added by inserting a new section, for example

```
 "mypc": {
        "backend": "http://192.168.1.105:5000/api",
        "usermanager": "http://192.168.1.105:5020/api",
        "devicemanager": "http://192.168.1.105:5030/api"
    },
```

To the **host.json**

```
{
   "pchcloud": {
       "backend": "https://pchcloud.pch-engineering.dk/backend",
       "usermanager": "https://pchcloud.pch-engineering.dk/usermanager",
       "devicemanager": "https://pchcloud.pch-engineering.dk/devicemanager"
   },
   "local": {
       "backend": "http://localhost:5000/api",
       "usermanager": "http://localhost:5020/api",
       "devicemanager": "http://localhost:5030/api"
   },
   "mypc": {
       "backend": "http://192.168.1.105:5000/api",
       "usermanager": "http://192.168.1.105:5020/api",
       "devicemanager": "http://192.168.1.105:5030/api"
   },
   "pchedge": {
       "backend": "https://172.19.1.181/backend",
       "usermanager": "https://172.19.1.181/usermanager",
       "devicemanager": "https://172.19.1.181/devicemanager"
   },

}
```

This can be usefull is the backend-services is running on another pc.

2. **config.json** is for configurtion settings

```
{
    "host":"local",
    "username": "local",
    "password": "pass",
    "delete_on_server": false,
    "query_passed_days": 1,
    "download_path":"downloads"
}
```

- "host" sets the server connections from the list of known hosts in the **hosts.json* file
- "username" and "password" is the account login. If connecting to local the default username is 'local' and the password is 'pass'
- "delete_on_server", if true recordings WILL BE DELETED on the server
- "query_passed_days", query time range number of days back from current time
- "query_passed_days", query time range number of days back from current time
- "download_path", local download path, currently used by the download_recordings_tdms.py

Execute python script

py ./download_recordings.py

or

py ./download_recordings_tdms.py