# Overview

TeamImages project was created to support image file verification in terms of file names, file metadata and file content. One of the goals of this tool is also to create thumbnail of the image and perform upload process from local Image directory to Team Polska (<https://teampolska.eu/>) web application using set of restful web services.

# Requirements

Before the process is started, it is required to configure the machine where script is located. Server needs to have Python environment configured. System was tested on Python version 3.6 and 3.7 on x64 bit machine. This is suggested configuration. It can be run on Windows or Linux operating systems.

Number of environment variables must be present in the system for the script to operate correctly. The list is placed below:

* APP\_SETTINGS\_URL (URL of web applications to upload images to)
* APP\_SETTINGS\_TEAM\_SERVER (Name of Database server)
* APP\_SETTINGS\_TEAM\_DATABASE (Database name)
* APP\_SETTINGS\_TEAM\_USER (User used to connect to the database)
* APP\_SETTINGS\_TEAM\_PWD (Password for the user)
* APP\_IMAGE\_FOLDER (Folder where images to process are located)

While APP\_SETTINGS\_\* variables are also required for other processes in the solution, APP\_IMAGE\_FOLDER is dedicated one for this application and must be present in the system. Process must have write permission to the folder. Script needs to have SQL connection opened (port 1433) to SQL server box if this is located on another machine in the network. Remote web application will be most likely located outside of the LAN network and HTTPS and HTTPS connections must be opened to that remote location.

Inside project folder there are 6 logo template files that are used for logo matching and logo information verification. Currently process is able to recognize 3 logos (efbe, kalorik and kitchen-originals in horizontal or vertical orientation). Those files are also required for the process to run.

# Running the process

With Python environment correctly set up it should be very easy to start the script. In situation where current directory is the one with upload.py file it is required to start it with a command **python upload.py**. There are no parameters required for the script to work. From version 0.20, script supports optional parameters to extend logging and file name filtering. To display help on parameters user can add **-h** or **--help** option. It will display message like this one:

C:\TeamAssets\TeamImages> python .\upload.py –help  
usage: upload.py [-h] [-ll {debug,info,warning,error,critical}]  
 [-ff FILENAMEFILTER]  
optional arguments:  
 -h, --help show this help message and exit  
 -ll {debug,info,warning,error,critical}, --loglevel {debug,info,warning,error,critical}  
 Information type captured in log file  
 -ff FILENAMEFILTER, --filenamefilter FILENAMEFILTER  
 Only files containing this value will be processed  
 (case sensitive)

Default values are warning and no filter applied. More detailed logging can be specified adding for example **--ll info** parameter when running the script. When i.e. **info** logging is enabled process will log all cases with levels info, warning, error and critical.  
 If there is need to process only images for given product, user can add filter option i.e. **--ff <product\_code>** and with that option added, files containing phrase <product\_code> will be processes.

# Output

Process will modify folder structure inside APP\_IMAGE\_FOLDER directory. Number of subfolders will be created. Folders are briefly described below:

* **errors** (in this folder process will place all the files that are not accepted by the script or files where some errors were found and program was not able to fix them. Each process run will create dedicated subfolder with timestamp in the name)
* **logs** (this location will hold all log files from the application. Each run will create dedicated log file with timestamp in the name)
* **processed** (in this location all image files successfully process will be stored)
* **thumbnails** (in this location all thumbnail image files successfully created will be stored)
* **unprocessed** (this is starting directory where all images ready to processed should be stored. If there are new images to be processed, they should be added to this location)