Augmentor

The main objective of the script is to augment reports with existing asset data.

Getting Started

These instructions will help you get the script running on your local machines or virtual servers. This was developed using Python >= 3.x. However, Python 2.x can still be used to run the script. Minor changes might be required for using Python 2.x.

Prerequisites

The required libraries/packages are listed in requirements.txt file. Using Pip, it can be installed with this command: **pip install -r requirements.txt**

It is recommended to create a Python virtual environment using virtualenv & virtualenvwrapper (optional).

Preparing the files/reports

The script requires the following dataset:

1. **Assets** - these will be the main source of data points for augmenting. Asset files should have the ff. columns:
   a. **Asset ID**
   b. **Custom ID**
   c. **Asset Label**
   d. **Asset Title**

   Other columns will be ignored by the script. For best performance, it is recommended to remove other columns that are not in the list of data points.

2. **Reports** - these files will be filled up by the augmentor. Only the Asset ID is required.

Running the script

Once the files are ready, the script can be run with this command:

python run.py -a assets/ -i in/ -o out/

Args:

    -a, --assets: Directory for asset files, Default: ***assets/***
    -i, --indir: Input directory for augmentor, Default: ***in/***
    -o, --outdir: Output directory, Default: **out/**

Process Flow

Here's a breakdown of what the script does:
1. Load all assets into one Dataframe from the assets directory
2. Iterate over the input directory
3. Each file will be matched against the Asset dataframe using the Asset ID
4. The Custom IDs will be merged and delimited by a pipe '|' character
5. Asset Label and Asset Title are filled in using the data from the Asset dataframe
6. The output will be saved as a .csv file in the output directory.