my Accounting

Manuel Escriche Vicente

January 8, 2024

Abstract

Contents

1 Setup python environment

Get a list of python installation:

```
>> type -a python
python is /Users/MANEV/.pyenv/shims/python
python is /usr/bin/python
```

From this list, the first one is used:

```
>> which python
/Users/MANEV/.pyenv/shims/python
```

Check what version of python is in use:

```
>> python -V
Python 3.8.6
```

Check available versions

```
>> pyenv versions
  system
  3.6.1
* 3.8.6 (set by /Users/MANEV/.pyenv/version)
  3.9.16
```

Set proper version in working folder.

```
 >> pyenv local 3.9.16
 >> python -V
 Python 3.9.16
```

Install pipenv to setup our virtual python environment. Notice pipenv doesn't get the version established by pyenv local previously, but the global one. Therefore, you have to care to set it up again in next steps.

```
 >> pip install pipenv
 >> pipenv install --python 3.9.16
 >> pipenv shell
```

In case you have to remove your virtual environment

```
 >> pipenv --rm
```

Install sqlalchemy

```
>> pipenv install sqlalchemy
```

Now you're ready to configure the application for any user and, finally, execute the program

```
>> python myAccounting.py <user>
```

2 Configuration tools

There are two tools available to configure users for the application:

config_user.py used to create user's folder, copy the generic profile chosen, and generate specific operative files.

db_tool.py used to create user's data base, and populate it with the accounts.json file, created with previous tool.

2.1 config_user.py

This tool is used to create the user folder where to store all their specific data, included configuration data.

Execute `>>config_user.py <user> create <profile>` to create the user with the specied profile. This command creates users' folders, and stores its accounting profile at due place: `.\users\<user>\configfiles`

The accounting profile file is stored as `<user>_profile.json`, that you can edit to adapt it to your specifics needs. Once editing is finished, you have to execute `>> config_user.py <user> refresh` to regenerate consistently the application files:

1. accounts.json used to create database accounts
2. income.json used to create income reporting
3. balance.json used to create balance reporting

Commands

```
>> python tools/config_user.py <user> create <profile>
>> python tools/config_user.py <user> refresh
```

2.2 db_tool.py

Before starting you have to use this tool to create and prepare the user database. Execute the following sequence:

1. `db_tool.py <user> create` to create user database file
2. `db_tool.py <user> init` to create user database structure
3. `db_tool.py <user> setup` to populate user database with accounts
4. `db_tool.py <user> query` to verify accounts creation

Commands

```
>> python tools/db_tool.py <user> create
>> python tools/db_tool.py <user> init
>> python tools/db_tool.py <user> setup
>> python tools/db_tool.py <user> query
>> python tools/db_tool.py <user> remove
```

3 Edition tools

There are two additional tools available:

excel.py tool used to take bank accounts excel files to export their transactions into json format so that main program can use them.

remake.py tool used to modify first opening seat and reconstruct the user data base and their year closing and opening seats consistently

Commands

```
 >> python tools/excel.py <user>
 >> python tools/remake.py <user>
```

4 Main Program

Make sure `<user>` has been created before executing:

```
 >> python myAccounting.py <user>
```