

# FileMonitor v0.1

Данный проект является демоном, который в бэкграунде мониторит появление новых файлов на некотором ресурсе, ведет учет обработанных файлов и является первым шагом для целевой обработки файлов, попадающих в S3.

## Структура проекта

Программа является CLI-based приложением, которое просто висит в памяти и мониторит некоторую веб-страницу на предмет появления на ней новых ссылок на скачивание файлов. Обновления по файлам программа складывает в базу данных.

## Переменные окружения

Name	Description
FM_WEBPAGE_URL	URL страницы, которую нужно парсить
FM_EMAIL_SENDER	Email отправителя
FM_EMAIL_RECEIVER	Email получателя
FM_EMAIL_SENDER_PWD	Пароль почты отправителя
FM_EMAIL_SERVER	Адрес почтового сервера
FM_EMAIL_PORT	Порт почтового сервера
FM_DEBUG	Флаг режима отладки (True / False)
FM_S3_*	Переменные для настройки доступа к S3 (добавить по мере необходимости)
FM_DB_NAME	Название БД. Для SQLite - путь к файлу БД
FL_LOGFILE	Путь к файлу логов

Для работы с переменными окружения рекомендуется библиотека **pyenv**:

<https://github.com/BEGOSoft/pyenv>

## Обработка исключений

Основной цикл программы должен перехватывать все исключения, возникающие в нем. Если получен `KeyboardInterrupt`, предложить юзеру сообщение "Quit? y/n", и для ответа "y", "yes", "Yes" - закончить работу, в противном случае - возобновить цикл. Все остальные исключения трекать в логах и просто "тушить".

## Структура базы данных

Таблица `File`:

- `id int PK autoincrement`
- `filename varchar(255) not null unique`
- `download_url varchar(255) not null`
- `created datetime default NOW` (разместить в базовой модели `BaseModel`)
- `changed datetime default NOW + on change row = NOW` (разместить в базовой модели `BaseModel`)
- `storage_url varchar(255) not null` (адрес файла в S3)

Использовать **peewee** в качестве ORM для БД.

В качестве движка БД использовать любую подходящую. Допускается **sqlite3**.

## Принцип работы

1. Приложение стартует в ручном режиме (либо с использованием `supervisor` и пр.)
2. В бесконечном цикле с таймером происходит:
  - a. Вычитка веб-страницы
  - b. Парсинг на предмет урлов файлов
  - c. Новые файлы добавляются в таблицу `File`
  - d. Новые файлы скачиваются и складываются в S3. Обновляются `storage_url` в `File`
  - e. Программа формирует Email со списком новых файлов и шлет на указанный в настройках email. Если новых файлов не было, письмо не должно отправляться - вместо этого просто пишем в логах, что новых файлов нет на такой-то момент времени.
3. Все действия программы логируются. Настройки логирования допускаются минимальные (задать файл для записи логов и формат вывода данных в лог с одним хендлером).