

Resmon – auth

Podręcznik użytkownika

Spis treści

1. Opis.....	1
1.1. Wymagania.....	1
1.1.1. Wymagania sprzętowe	1
1.1.2. Wymagania systemowe.....	1
2. Instalacja.....	2
2.1. Opcje instalatora	2
3. Instrukcja użytkowania.....	2
3.1. Uruchomienie	2
4. Informacje dla programistów	3
4.1. Używane moduły Pythona.....	3
4.2. Skrypty dla programistów	3
4.3. Wydawanie aplikacji w kontenerze Docker	3
5. Autorzy	4

1. Opis

Program ResMon-auth jest częścią systemu ResMon. Służy do pokazywania określonej ilości hostów o największej wartości pomiaru dla każdej z metryk.

Bardzo ważne! Podany plik klucza (`./data/jwt.key`) jest używany tylko w środowisku testowym. Proszę absolutnie nie używać go w wersji produkcyjnej.

Uwaga! Wszystkie ścieżki wymienione w tym dokumencie są traktowane, jakby były używane w głównym katalogu projektu/instalacji.

1.1. Wymagania

1.1.1. Wymagania sprzętowe

Program nie posiada specjalnych wymagań sprzętowych.

1.1.2. Wymagania systemowe

Program został stworzony pod system operacyjny Linux z rodziny Ubuntu, Debian, Mint, itd.

Do poprawnego działania instalatora oraz samej aplikacji konieczne jest posiadanie zainstalowanego:

- interpretera języka **Python** w wersji **co najmniej 3.6**¹;
- odpowiedniej dla niej wersji menedżera pakietów **PIP**².

¹ <https://docs.python.org/3.6/>

² <https://pypi.org/project/pip/>

2. Instalacja

Program dla zwykłych użytkowników jest dystrybuowany jako pojedynczy plik instalatora *install-auth.sh*. Jest to skrypt powłoki Bash. W celu instalacji aplikacji ResMon-auth należy:

- 1) pobrać najnowszą wersję instalatora z repozytorium: <https://github.com/MAD-FiS/resmon-auth/releases> ;
- 2) przenieść go do lokalizacji, w której ma zostać zainstalowany klient automatyczny;
- 3) uruchomić instalator komendą:

```
./install-auth.sh
```

- 4) udzielić zgody instalatorowi na rozpakowanie plików;
- 5) poczekać do końca instalacji.

Uwaga! Ewentualne ostrzeżenia lub błędy wyświetlone podczas instalacji mogą spowodować nieuruchomienie się aplikacji lub jej niepoprawne działanie.

2.1. Opcje instalatora

Instalator posiada następujące opcje:

--quiet – włącza tzw. tryb cichy instalacji; w tym trybie automatycznie następuje akceptacja wszystkich zgód, o które normalnie instalator pyta użytkownika i czeka na jego odpowiedź; może być zastosowane w przypadku, gdy nie jest możliwa interakcja programu z użytkownikiem podczas procesu instalacji.

3. Instrukcja użytkowania

3.1. Uruchomienie

Aby rozpocząć pracę z programem, należy go uruchomić, wywołując następującą komendę w miejscu, w którym został on zainstalowany:

```
./resmon-auth [-h|--help] [-k|--key_path KEY_PATH] [-p|--port PORT]
```

W tabeli 1 zostały opisane możliwe opcje, jakie może przyjmować program.

Opcja	Wartość domyślna	Opis
-h --help	---	pokazuje komunikat pomocy oraz kończy pracę programu
-k --key_path KEY_PATH	./data/jwt.key	lokalizacja, w której znajduje się plik z kluczem
-p --port PORT	5000	Numer portu, na którym nasłuchuje serwer uwierzytelniający

Tabela 1 Opis możliwych parametrów uruchomieniowych programu

Przerwanie wykonywania programu następuje przez wciśnięcie kombinacji klawiszy Ctrl + C.

Uwaga! Plik z kluczem musi być taki sam jak na monitorach, które używają tego serwera uwierzytelniającego do potwierdzenia tożsamości użytkowników. Po wygenerowaniu klucza należy sprawdzić czy zarówno serwer autoryzacyjny, ja i monitory używają tej samej wersji tego pliku.

Uwaga! W przypadku równoczesnego wysłania do serwera więcej niż jednego zapytania, może wystąpić problem. Najlepiej unikać takiej sytuacji.

4. Informacje dla programistów

Uwaga! Ta instrukcja jest napisana tylko dla programistów, którzy używają systemu operacyjnego Linux.

Projekt jest utrzymywany na serwerze Github pod następującym adresem: <https://github.com/MAD-FiS/resmon-auth>. Przed rozpoczęciem pracy należy sklonować powyższe repozytorium.

W przypadku pracy nad repozytorium nie jest wymagana instalacja, opisana w sekcji [Instalacja](#).

Więcej szczegółów znajduje się w pliku *README.md* na [repozytorium projektu](#).

4.1. Używane moduły Pythona

Te moduły są wymagane przez aplikację. Jeśli potrzebne jest na przykład uruchomienie testów, należy się upewnić, że wszystkie one zostały zainstalowane na komputerze przy pomocy programu *PIP3*. Można użyć do tego pliku *./data/requirements*.

Lista modułów znajduje się w tabeli 3.

Nazwa modułu	Wersja
flask	0.12.2
flask_jwt_extended	3.7.2
Flask-RESTful	0.3.6
FlaskSQLAlchemy	2.3.2
passlib	1.7.1

Tabela 2 Moduły Pythona używane przez program

4.2. Skrypty dla programistów

Dla programistów, którzy chcieliby rozwijać aplikację, przygotowano kilka skryptów, pozwalających usprawnić pracę. Uruchamia się je, wywołując w głównym katalogu programu komendę:

```
./scripts.sh SCRIPT_NAME
```

gdzie *SCRIPT_NAME* może być jedną z następujących opcji:

- **build** – przygotowuje plik instalatora *install-autoclient.sh*;
- **docgen** – generuje dokumentację kodu i umieszcza ją w katalogu *./docs/*;
- **runtest** – uruchamia wszystkie testy, dostępne dla tego projektu.

Uwaga! Wszystkie elementy aplikacji automatycznie wczytują potrzebne środowisko. Jeśli trzeba zrobić to ręcznie, plik z tą konfiguracją znajduje się w katalogu *./data/*.

4.3. Wydawanie aplikacji w kontenerze Docker

Aplikacja może być uruchamiana w ramach kontenera Docker'owego. Może to być wykorzystane do testowania jej w czystym, izolowanym środowisku.

Na komputerze musi być zainstalowane oprogramowanie Docker³. Przed rozpoczęciem tworzenia obrazu kontenera należy się upewnić, iż w katalogu głównym projektu znajduje się

³ <https://docs.docker.com/>

plik *install-auth.sh*. Jeśli nie, należy wcześniej uruchomić skrypt, budujący instalator, opisany w sekcji [Instalacja](#).

Następnie trzeba zbudować obraz kontenera komendą:

```
docker build -t resmon-auth .
```

oraz uruchomić go:

```
docker run -it resmon-auth
```

Kontener zostanie uruchomiony razem z aplikacją na nim działającą.

5. Autorzy

Autorami projektu ResMon jest grupa MAD-Team⁴ w składzie:

1. Bajorek Tomasz – kierownik projektu, programista;
2. Bartocha Łukasz – specjalista z zakresu Dev Ops;
3. Chronowski Tomasz – programista;
4. Drzazga Kamil – programista;
5. Kwaśnicki Marcin – programista;
6. Rucki Grzegorz – programista;
7. Ryniak Grzegorz – kierownik techniczny, programista;
8. Szęszół Rafał – programista;
9. Słoczyński Tomasz – programista.

⁴ <https://github.com/MAD-FiS>