

Testowanie - opis

Przetestowano kolejno:

- Możliwość zalogowania się w aplikacji i sprawdzenia pomiarów
- Działanie monitora i sensora
- Konfigurację Sensora w szczególności ze zmiennymi parametrami takimi jak interwały
- Konfigurację Monitora
- Rejestrację Sensora w Monitorze
- Rejestrację danych otrzymanych z Sensora w Monitorze

Działanie monitora i sensora:

Monitor działa poprawnie zarówno lokalnie jak i na przygotowanej wcześniej maszynie.

Sensor został najpierw włączony lokalnie – działa i otrzymuje odpowiedzi w postaci:

```
Data response : OK
Data: {'sensor_id': 4, 'values': [{'measurement_id': 7,
'measurement_value': 5.4}, {'measurement_id': 8, 'measurement_value':
71.4}]}
```

Konfiguracja Sensora (interwały, monitor)

Konfiguracja została sprawdzona poprzez zmienianie w pliku konfiguracyjnym w formacie JSON odpowiednich parametrów. Sprawdzone zostały interwały od 0-10, następnie wartości losowe z przedziału 10 – 1000. Nie znaleziono żadnych błędów, sensor działa poprawnie, natomiast logi w odpowiednim pliku pojawiają się dopiero po zastopowaniu skryptu.

Konfiguracja Monitora (baza danych)

Po starcie aplikacji jeżeli nie ma bazy danych automatycznie powinna zostać utworzona – działanie to zostało sprawdzone i jest poprawne. Po zmianach w pliku konfiguracyjnym baza reaguje prawidłowo, tworzą się nowe instancje o nowych nazwach. Pomiary zapisywane są poprawnie, a logi nie zawierają żadnych błędów.

rejestrację Sensora w Monitorze

Sensor rejestruje się w monitorze – działa poprawnie

rejestrację danych z Sensora w Monitorze

Dane z mojego komputera zostały zarejestrowane w poprawny sposób.

Do przetestowania aplikacji użyto programu Postman. Testowane zapytania wyglądały następująco:

- 52.232.125.49:8080/login
- 52.232.125.49:8080/hosts
- 52.232.125.49:8080/measurements
- 52.232.125.49:8080/measurements/values
- 52.232.125.49:8080/measurements/{4}/values

Przykładowe zapytanie i otrzymany wynik:

