Aleksandra Maczel, Dariusz Duda, Dawid Komorowski, Krzysztof Halwa, Łukasz Jedliński, Anna Konieczna, Paweł Jóźwik, Michał Jagieła, Mateusz Nawrot, Mateusz Smęt, Marcin Nycz

INTERFEJS REST

Jest to pierwsza wersja specyfikacji i może ona ulec drobnym modyfikacjom. Parametry obowiązkowe oznaczono gwiazdką.

1. Encje

Pełna lista encji wykorzystywanych w aplikacji znajduje się poniżej :

- **User** – użytkownik systemu; może dodawać oraz usuwać pomiary złożone
- **Resource** – obiekt, na którym dokonywane są pomiary wybranych metryk
- **Sensor** – dokonuje pomiarów wybranych metryk dla konkretnych zasobów
- **Measurement** – konkretna wartość pomiaru z sensora
- **Complex Measurement** – pomiar złożony, który może np. agregować pomiary proste
- **Metric** – cecha/typ opisująca dany pomiar

2. Wymagania odnośnie API

1. Klient ma możliwość pobrania z Monitora listy wszystkich monitorowanych zasobów i dostarczanych dla nich pomiarów.

Metoda	URI	Parametry
GET	/resources	
GET	/resources/{*resourceId*}/metrics	
GET	/resources/{*resourceId*}/measurements	metric startTime endTime

2. Klient może przeszukiwać Monitor przy pomocy zapytań, np. znaleźć zasób, który jest hostem i ma w nazwie ‘zeus’.

Metoda	URI	Parametry
GET	/resources	resourceType name metric

3. Pomiary proste (dostarczane bezpośrednio przez sensory) powinny być udostępniane przez monitor jako lista wartości. Domyślnie zwracana jest lista kilku ostatnich pomiarów.

Metoda	URI	Parametry
GET	/resources/{*resourceId*}/sensors	
GET	/sensors/{*sensorId*}/measurements	metric startTime endTime

4. W przeglądarce można:

    a. Przeglądać listy dostępnych zasobów i pomiarów.
    b. Podglądać ostatnie wartości pomiarów.
    c. Wyszukiwać dostępne zasoby i pomiary (np. po nazwie).
    d. Wyświetlać wybrane pomiary na wykresie, uaktualnianym automatycznie co pewien czas (np. 5 sekund).

    Uwzględnione w poprzednich punktach.

5. Klient ma możliwość stworzenia nowego pomiaru (złożonego) w Monitorze.
6. Pomiar złożony można usuwać

    a. Ograniczamy się do jednego typu pomiaru złożonego: **agregacji przez obliczanie średniej ruchomej**
    b. Przykład: średnie obciążenie CPU z ostatnich 5 minut, obliczane co 1 minutę. (Długość ruchomego okna czasowego = 5 minut, częstotliwość obliczania = 1 minuta).
    c. Pomiar złożony powinien być dostarczany przez Monitor w ten sam sposób jak prosty (jako lista ostatnich wartości).

Metoda	URI	Parametry
POST	/complex-measurement	*sensorId *metric *durationTime *intervalTime
GET	/complex-measurements	startTime endTime
DELETE	/complex-measurement/{*id*}	

7. Usunąć pomiar złożony może tylko ten użytkownik, który go stworzył. Oznacza to, że w systemie musi istnieć mechanizm uwierzytelniania i autoryzacji użytkowników.

Każdy użytkownik ma te same prawa, użytkownicy nie mają możliwości usuwania własnych kont.

Metoda	URI	Parametry
POST	/user	*username *password
POST	/users/authorisation	*userId *password

8. Monitor ma zapisywać historię wszystkich pomiarów do bazy danych.

Nie wymaga udostępniania dodatkowych metod.

9. W przeglądarce dodatkowo można:
    a. Podłączać się do wielu Monitorów.
    b. Definiować nowy pomiar złożony jako średnią ruchomą istniejącego pomiaru.
    c. Usuwać wcześniej zdefiniowany pomiar złożony.
    d. Uzyskiwać dostęp do historycznych pomiarów zgromadzonych w Archiwum
        i. Można tworzyć wykresy historycznych pomiarów z zadanego okresu czasu. (Wykresy te są statyczne, tj. nie odświeżają się tak jak wykresy pomiarów bieżących).
        ii. Na jednym wykresie można umieścić wiele pomiarów (oś X jest wspólna, osi Y jest kilka)
        iii. Z gotowego wykresu można importować dane na nim widoczne do formatu CSV.

    Uwzględnione w poprzednich punktach. Część wymagania dotycząca wykresów nie wymaga implementacji dodatkowych metod.

10. Ma być zaimplementowany klient automatyczny, tj. program, który działa podobnie jak komenda ‘top’: wypisuje i co pewien czas odświeża top 10 najbardziej obciążonych maszyn.

    a. Klient automatyczny może podłączyć się do kilku Monitorów jednocześnie.
    b. Klient automatyczny musi uwzględniać zmiany, np. musi wykrywać dodanie nowych lub usunięcie istniejących maszyn z listy monitorowanych zasobów.

    Funkcjonalność a) może zostać zrealizowana w oparciu o metody udostępnione w punkcie pierwszym. Natomiast część b) będzie obsługiwana po stronie serwerowej, klient będzie jedynie warstwą prezentacji otrzymanych danych.