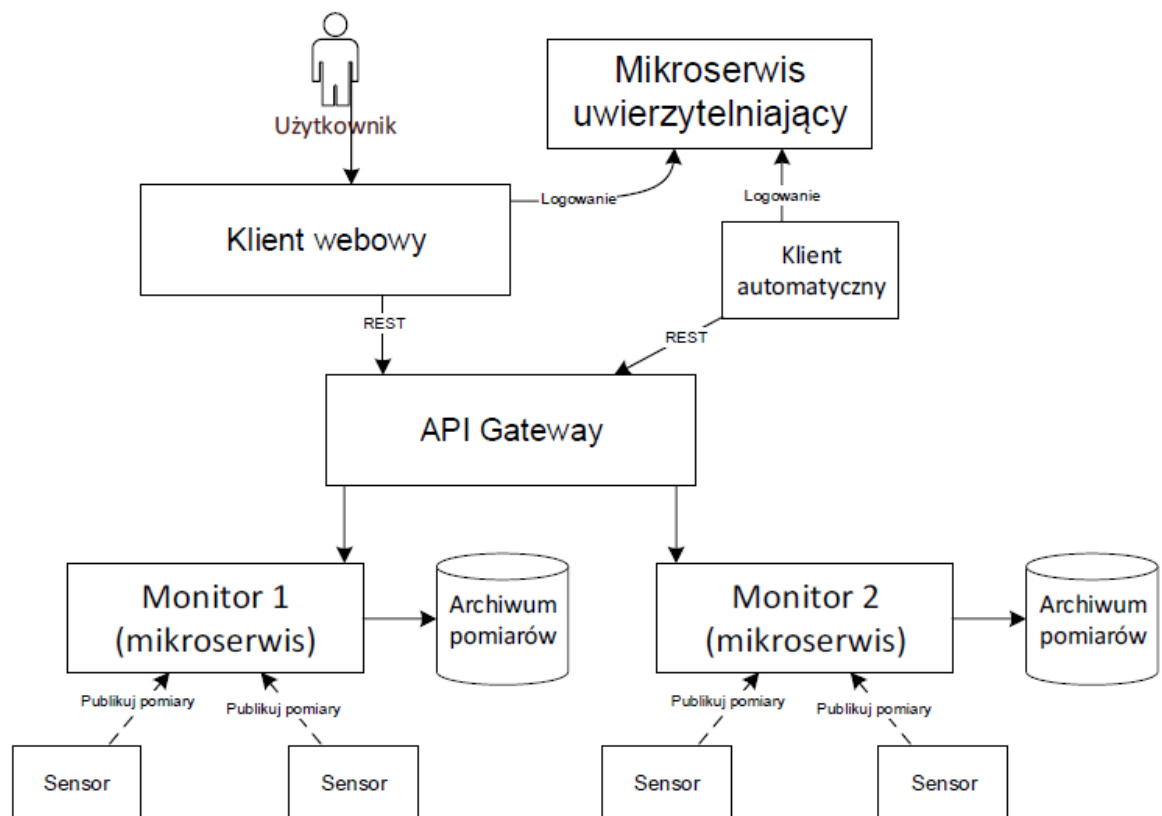


System monitorowania rozproszonych zasobów komputerowych

1. Wprowadzenie

Projekt przedstawia system do monitorowania rozproszonych zasobów komputerowych (np. Obciążenia CPU czy zużycia pamięci). Architekturę projektu przedstawiono poniżej:



Do tego celu wykorzystuje następujące komponenty:

2. Sensor

Aplikacja sensor służy do zbierania metryk z urządzenia zawierającego CPU oraz baterię i przekazywania zmierzonych danych do monitora.

- Dokonuje **pomiarów** wybranych **metryk** dla określonych **zasobów** i wysyła pomiary cyklicznie do Monitora.
 - o Przykład zasobu i metryki: *Host / CPU Utilization* (aktualne zużycie CPU dla hosta).
- Wiadomość przesyłana przez sensor do monitora powinna zawierać: nazwę (identyfikator) zasobu, nazwę metryki, dane pomiarowe (zależne od metryki).
 - o Sensor przesyła dwa rodzaje informacji do monitora: (i) metadane (opisują szczegółowo co jest monitorowane), (ii) właściwe dane pomiarowe. Dane pomiarowe wysyłane są regularnie, natomiast metadane też trzeba przesłać, ale nie ma sensu tego robić za każdym razem gdy przesyłane są dane pomiarowe! (Dla wielu sensorów znacznie zwiększyłyby to obciążenie sieci z powodu monitorowania).
- Wiadomości powinny mieć formę tekstową. Sugerowany format bazowy: JSON.

Mierzone są następujące parametry:

- aktualne zużycie procesora,
- poziom naładowania baterii

Aby uruchomić aplikację należy w konsoli:

- zainstalować potrzebne pakiety:
 - `pip install -r requirements.txt`
- uruchomić aplikację komendą:
 - `python source\sensor.py`

Więcej informacji zawarty jest w pliku README.md.

Dodatkowo dodano konsolowy help, aby ułatwić parametryzowanie sensora. Aby zobaczyć opcje należy w konsoli wpisać komendę: `python source\sensor.py --help`

```
(winenv) C:\Users\User\Desktop\PZ\PZ-gr.2\Sensor\source>python sensor.py --help
usage: sensor.py [-h] [--reg REG] [--name NAME] [--interval INTERVAL]
                 [--measurements_amount MEASUREMENTS_AMOUNT]
                 [--metrics METRICS] [--url URL] [--urlAS URLAS]

optional arguments:
  -h, --help            show this help message and exit
  --reg REG, -r REG     registration flag. Options:yes or no. Default: yes
  --name NAME, -n NAME  sensor name
  --interval INTERVAL, -i INTERVAL
                        measurements interval. Min: 1.0. Max: 30.0
  --measurements_amount MEASUREMENTS_AMOUNT, -ma MEASUREMENTS_AMOUNT
                        number of measurements sended in one package. Min: 5.
                        Max: 100
  --metrics METRICS, -m METRICS
                        metrics flag. Options: CPU, Battery or Both, Default:
                        Both
  --url URL, -u URL      url to monitor instance. Default:
                        http://localhost:8080/v1/
  --urlAS URLAS, -uAS URLAS
                        url to AS instance. Default: http://127.0.0.1:8000/v1/
```

Rys. Dostępne opcje dla użytkownika.

3. Monitor

Monitor jest aplikacją, która zbiera wysyłane przez sensory pomiary. Otrzymane wartości pomiarów są zapisywane w bazie danych **PostgreSQL**.

- Usługa, która zbiera dane od sensorów i **udostępnia pomiary dla klientów**.
- Również umożliwia tworzenie nowych **pomiarów złożonych**, które np. agregują pomiary proste.
 - Pomiary złożone są obliczane w Monitorze na podstawie innych pomiarów.
 - Przykład złożonego pomiaru: *średnie obciążenie CPU z ostatnich 5 minut obliczane co minutę*.

Użycie:

- Aby uruchomić program należy skorzystać z komendy (operacja jest dostępna po wcześniejszej kompilacji kodu (więcej informacji w pliku README załączonym do projektu))
 - **mvn spring-boot:run**
- Dzięki umieszczeniu wszystkich przydatnych zmiennych globalnych w pliku ***application.properties***, projekt można szybko dostosować do potrzeb oraz uruchomić.
- Wszelkie szczegóły dotyczące kodu zostały opisane w dokumentacji ***Javadoc***.

4. Klient webowy

Graficzny interfejs webowy sterowany przez użytkownika, który umożliwia wyszukiwanie dostępnych zasobów i pomiarów, wizualne przeglądanie wartości pomiarów i definiowanie nowych pomiarów złożonych.

a) Rejestracja / logowanie

Po przejściu do strony startowej użytkownika zostaje wyświetlony formularz logowania.

The image shows a login form with a dark background. At the top, the title 'Logowanie' is centered. Below it, there are two input fields. The first field is labeled 'Nazwa' and has a person icon to its left. The second field is labeled 'Hasło' and has a lock icon to its left. Below these fields, there are two buttons: 'Zarejestruj się' on the left and 'Zaloguj' on the right.

Rys. Widok formularza logowania.

Pole z hasłem w celu zwiększenia bezpieczeństwa ukrywa treść hasła.

Logowanie

Nazwa

test

Hasło

.....

Zarejestruj się
Zaloguj

Rys. Widok formularza logowania. Wypełniony.

Rejestracja

Nazwa

Nazwa

Hasło

hasło

Zarejestruj

Rys. Widok formularza rejestracji.

b) Lista

←

→

↺

127.0.0.1:8081/#/list

Wstrzymano

1

PZ-gr.2 (tom)

DOM

WYKRESY

LISTA

Wyloguj

Lista

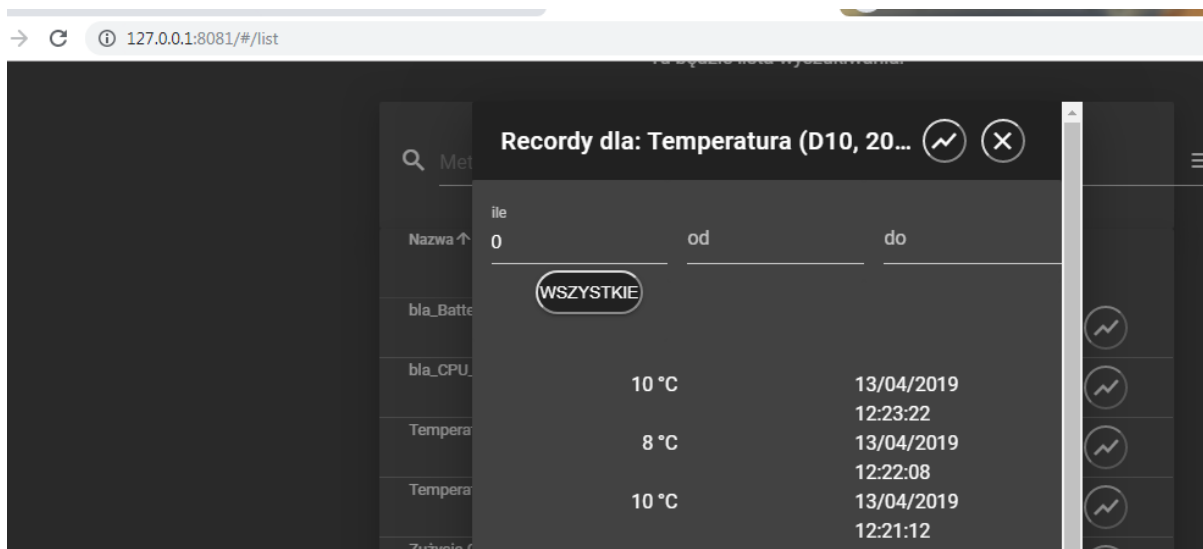
Tu będzie lista wyszukiwania.

Metryki

Szukaj...

Nazwa ↑	Akcje	Typ
bla_Battery_Host		(Battery)
bla_CPU_Host		(CPU)
Temperatura (D10, 205, stanowisko 1)		(temperatura)
Temperatura (D10, 205, stanowisko 2)		(temperatura)
Zużycie GPU (Cyfronet, 402, st 4)		(zużycie GPU)

Rys. Widok listy metryk



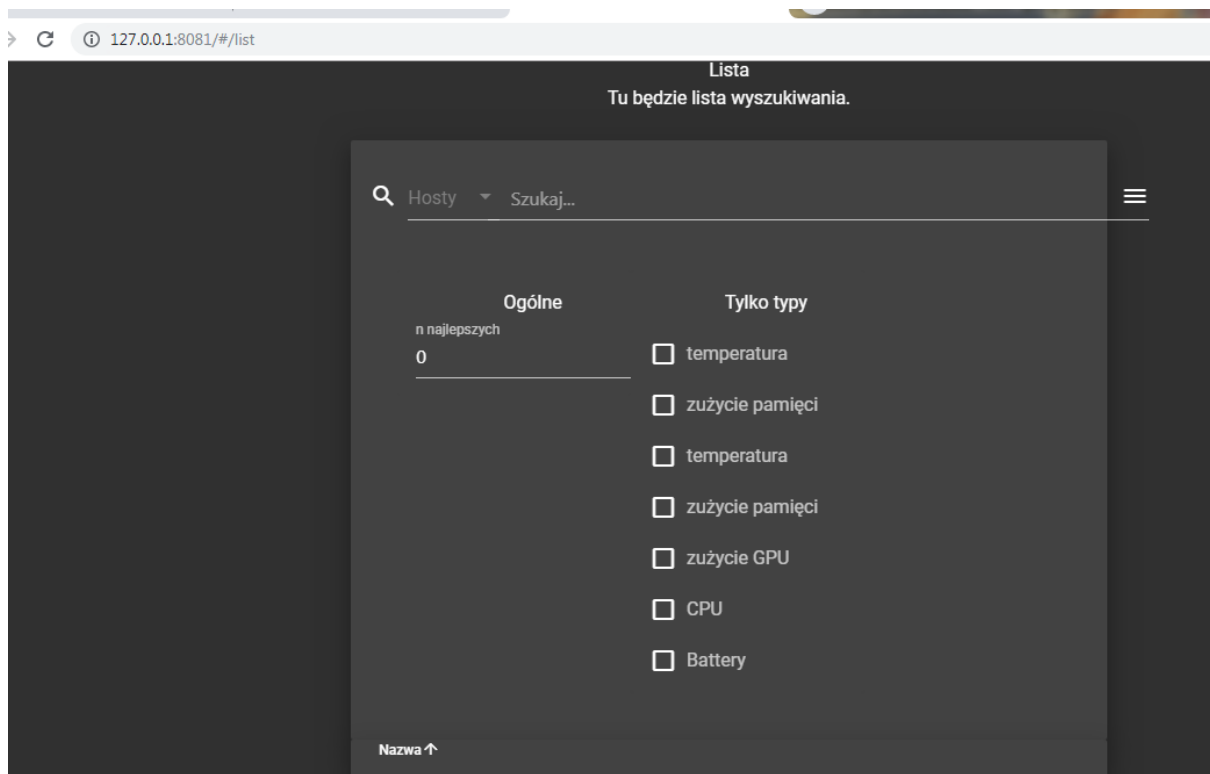
Rys. Widok szczegółów metryki

Lista

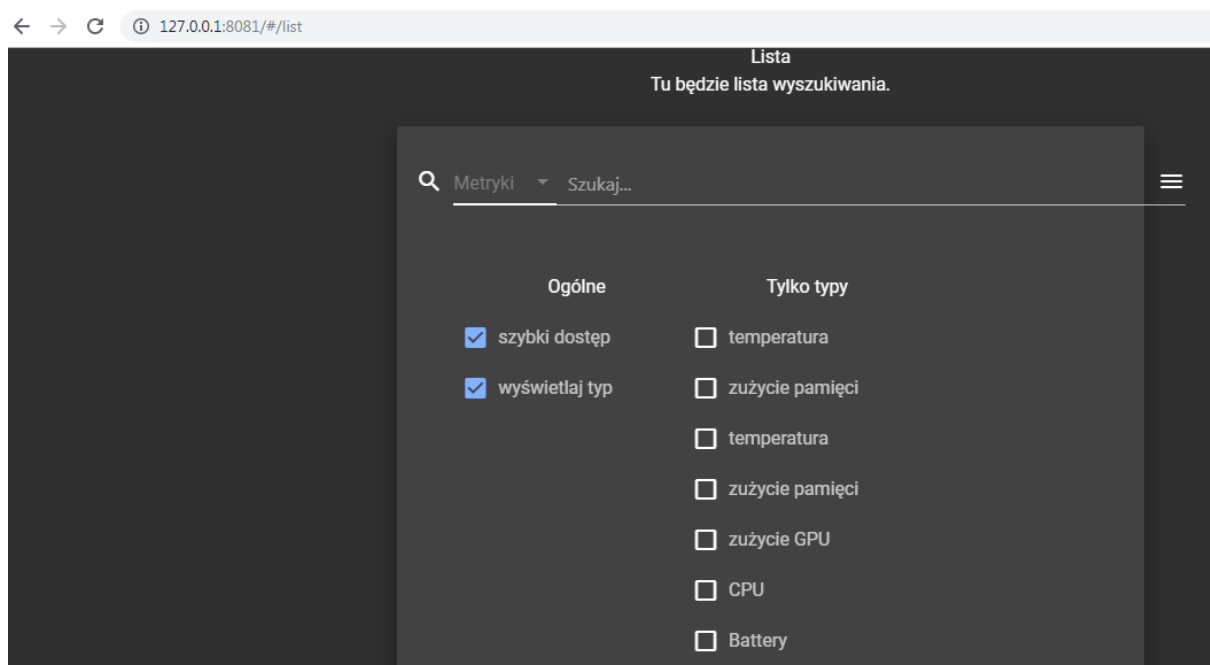
Tu będzie lista wyszukiwania.

Nazwa
bla
Cyfronet, 402, stanowisko 4
D10, 205, stanowisko 1
Temperatura (D10, 205, stanowisko 1)
temperatura [°C]
Zużycie pamięci (D10, 205, stanowisko 1)
zużycie pamięci [MB]
D10, 205, stanowisko 2

Rys. Widok listy hostów



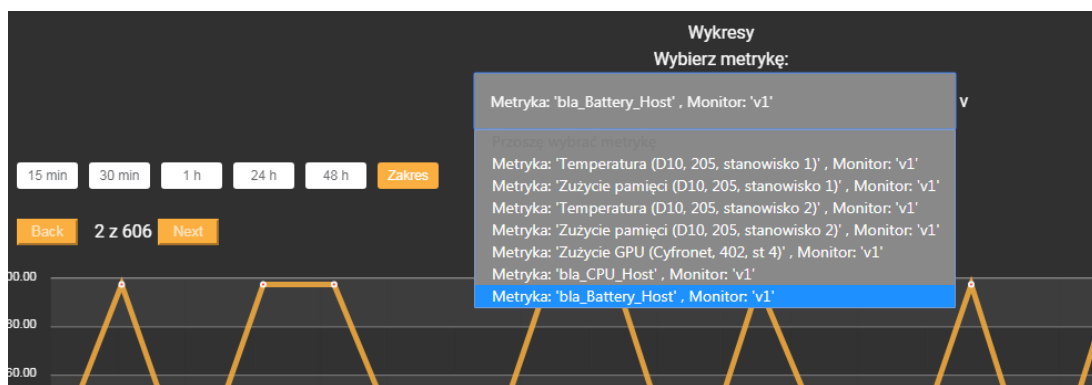
Rys. Widok konfiguracji wyświetlania hostów



Rys. Widok konfiguracji wyświetlania metryk

c) Wykresy

Zakładka 'Wykresy' zawiera przycisk typu 'select', który umożliwia użytkownikowi wybór dostępnej metryki.

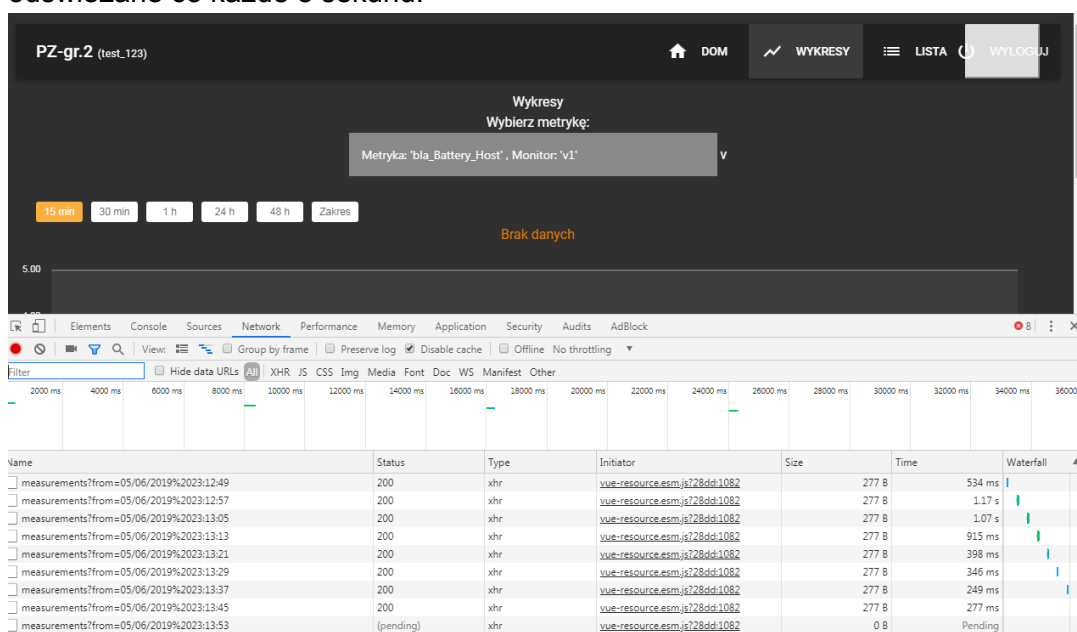


Rys. Menu przedstawiające dostępne metryki

Po dokonaniu wyboru metryki, dane zostaną automatycznie wyświetlone na wykresie dla zakresu ostatnich 15 minut.

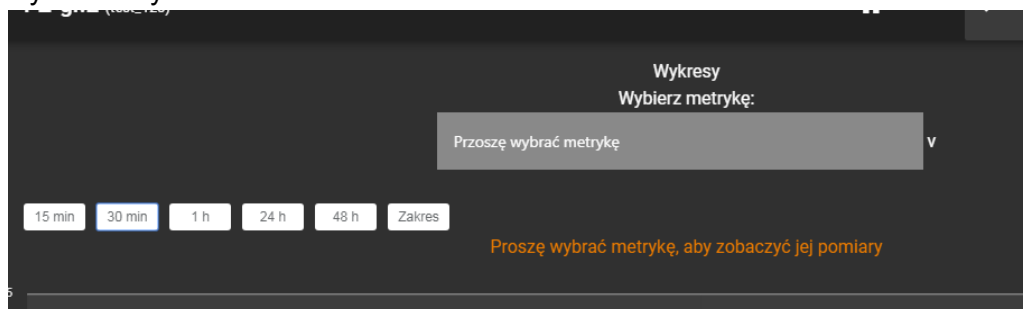
Informacje dodatkowe:

- Gdy user podgląda metrykę dla ostatnich 15min, to dane są automatycznie odświeżane co każde 8 sekund.



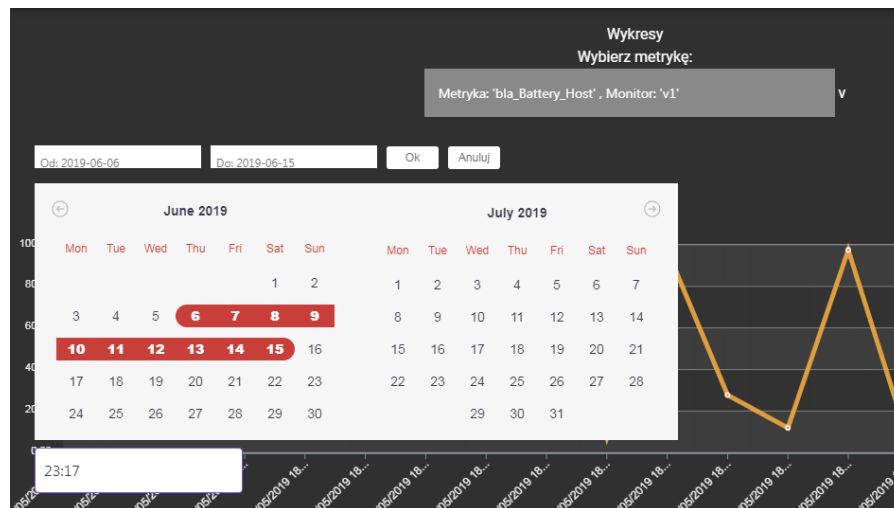
Rys. Odświeżanie danych dla zakresu 15minut. Widoczne ponawianie żądania.

- Gdy user próbuje podglądać dane bez ustawienia metryki to zostaje wyświetlony komunikat.

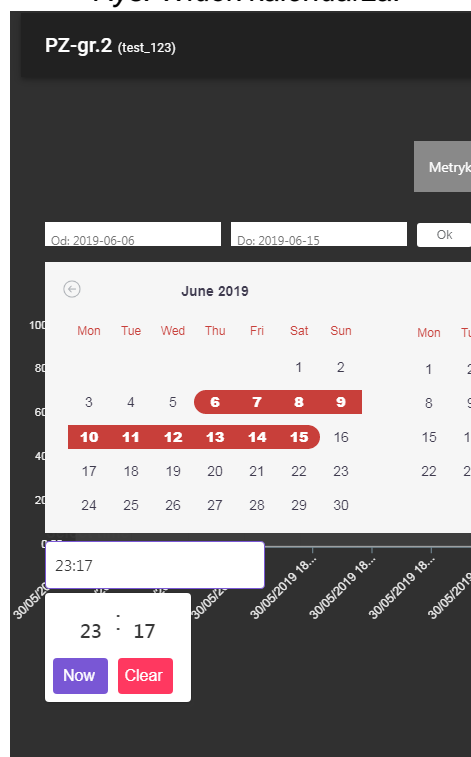


Rys. Komunikat braku ustawienia metryki.

- Istnieje kilka opcji wyboru zakresu. Wygodną opcją jest wybór kalendarza. Umożliwia on ustawienie dowolnego zakresu czasu. Zalecany jest zakres 1-2 dni. Dłuższe zakresy mogą wpłynąć niekorzystnie na czas odpowiedzi żądania (jego zwiększenie).

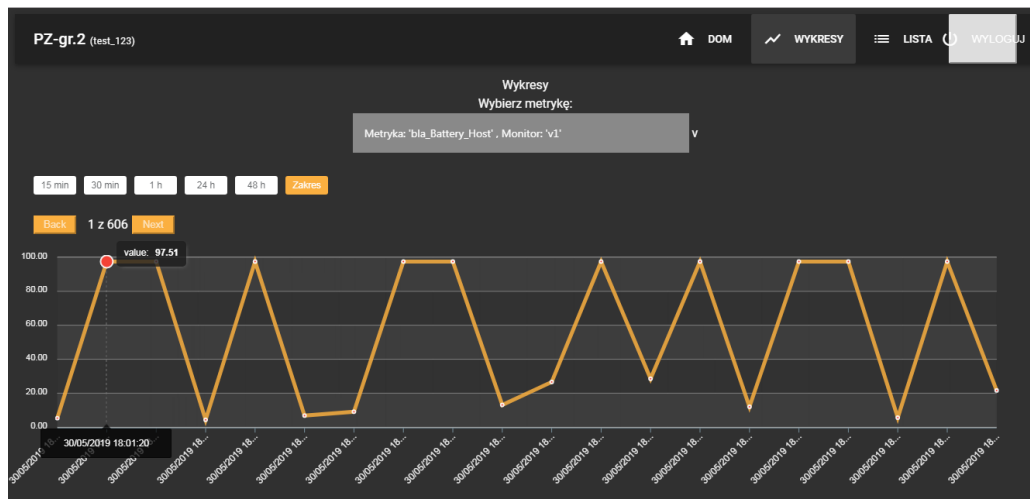


Rys. Widok kalendarza.



Rys. Widok okna ustawienia godzin i minut.

- dane na osi Y mają ustawioną precyzję do '2 miejsc po przecinku'.
- wyświetlane dane zawierają max 30 pomiarów. Jeżeli pomiarów dla zadanego okresu czasu jest więcej to pojawiają się specjalne przyciski do przewijania (tzw. paginacja)



Rys. Widok wyświetlania dużych ilości danych. Widoczne tooltip'y, paginacja oraz początkowe 30 pomiarów dla zadanego zakresu czasu.

5. Klient automatyczny

Klient automatyczny jest programem automatycznie przeszukującym monitorowane zasoby i pomiary w celu znalezienia 10 najbardziej obciążonych komputerów.

Użycie :

1. `./client.sh -h`
Wyświetlenie instrukcji użytkownika tzw. help'a.
2. `./client.sh -l`
Pobranie i wyświetlenie listy dostępnych typów metryk.
3. `./client.sh -m typ_metryki` (np. `./client.sh -m CPU`)
Uruchomienie właściwego programu pobierającego i wyświetlającego top10 najbardziej obciążonych hostów. Wyniki są odświeżane co 1s.

6. API Gateway

API Gateway umożliwia komunikację między klientem webowym i automatycznym a monitorami. Zapewnia interfejs dla różnych monitorów. Do komunikacji wykorzystuje następujące zasoby:

v1/metrics

GET - Returns an object containing a list of all metrics in the system.

POST - Creates a new compound metric in a monitor containing a simple metric(s) on which the compound metric is based on.

v1/metrics/:id

GET - Returns metric's info.

DELETE - Deletes the metric(only compound metrics). Metric can only be removed by the creator.

v1/metrics/:id/measurements

GET - Returns [n] latest measurements for the metric.

v1/hosts

GET - Returns a list of all hosts in the system.

v1/hosts/:id

GET - Returns host info.

v1/monitors

POST - (FOR INTERNAL USE) Registers a new instance of a monitor in API Gateway.

DELETE - (FOR INTERNAL USE) Removes an instance of a monitor from list of active monitors registered in the gateway.

7. Mikroservis uwierzytelniający

Mikroservis uwierzytelniający umożliwia bezpieczne korzystanie z produktu osobom do tego upoważnionym. Jest oparty na standardzie JWT (JSON Web Tokens), który definiuje sposób wymiany danych między stronami w bezpieczny sposób poprzez obiekt JSON. Przesyłane informacje mogą być weryfikowane dzięki cyfrowemu podpisowi, który jest elementem tokenu.

Rejestracja użytkownika w serwisie dokonuje się za pomocą następującego zasobu:

v1/users

POST

Create new user.

Request parameters

Body:

```
{
```

```
"username":"<username>",  
"password":"<password>"  
}
```

Response

Success: 201 Failure: 400

Kiedy użytkownik jest już zarejestrowany, może się zalogować do serwisu:

v1/login

POST

Returns JWT authentication token and refresh token in login process.

Request parameters

Body:

```
{  
  "username":"<username>",  
  "password":"<password>"  
}
```

Response:

Success: 200

```
{  
  "access_token":"<access_token>",  
  "refresh_token":"<refresh_token>"  
}
```

Failure: 400, 401

Prawidłowe logowanie zwraca użytkownikowi access_token oraz refresh_token. Access_token klient może wykorzystać, by uzyskać dostęp do określonych zasobów. Jest on przekazywany kolejnym komponentom, które zwracają dane tylko w przypadku ważnego tokena. Ważność tokena komponenty sprawdzają za pomocą następującego endpointa:

v1/protected

GET

Returns current user name only when valid access-token.

Body:

```
{  
  "access_token":"<access_token>"  
}
```

Response:

Success: 200

```
{  
  "logged_in_as": "<username>"  
}
```

```
}
```

Failure: 401, 422

Access_token ma jednak stosunkowo krótki okres ważności. Po jego upływie, klient może wykorzystać refresh_token, do odnowienia wygasającego access_tokena:

v1/token

POST

Returns new JWT access-token in token refresh process.

Body:

```
{  
  "refresh_token":"<refresh_token>"  
}
```

Response:

Success: 200

```
{  
  "access_token":"<access_token>"  
}
```

Failure: 400, 401, 422

Generowany jest wtedy nowy access_token. Refresh token posiada natomiast długi okres ważności, dlatego wymagane jest jego usunięcie w razie wylogowania użytkownika. Można to zrobić dodając refresh_token do czarnej listy, korzystając z następującego zasobu:

v1/token

DELETE

Invalidates refresh token.

Body:

```
{  
  "refresh_token":"<refresh_token>"  
}
```

Success: 200 Failure: 400, 401, 422