

1 Czujniki

Mamy zdefiniowaną listę czujników zamontowanych w różnych silnikach. Każdy silnik posiada dokładnie jeden czujnik ciśnienia oraz co najmniej jeden czujnik temperatury. Twoim zadaniem jest napisanie usługi z interfejsem REST/HTTP, która umożliwi aktualizowanie wartości czujników oraz na żądanie użytkownika zwróci listę wszystkich nieprawidłowo działających silników.

Serwer powinien pobrać dane wejściowe zawierające listę czujników z pliku w formacie YAML, zlokalizowanego w repozytorium na serwerze GitHub-a. Można założyć, że zawartość pliku z listą czujników nie może ulec zmianie w trakcie działania serwera. Dane wejściowe są w formacie listy YAML, przykładowo:

```
- id: "3142"
  engine: "123"
  type: "pressure"
  name: "Engine 123"
  value: 27
  min_value: 0
  max_value: 120
- id: "32234"
  master-sensor-id: "3142"
  type: "temperature"
  value: 82
  min_value: -50
  max_value: 150
- id: "57230"
  master-sensor-id: "3142"
  type: "temperature"
  value: 55
  min_value: 0
  max_value: 100
- id: "5703"
  engine: "156"
  location: ""
  type: "pressure"
  name: "Engine 156"
  value: 73
  min_value: 0
  max_value: 150
- id: "89145"
  master-sensor-id: "5703"
  type: "temperature"
  value: 65
  min_value: 0
  max_value: 100
```

- "id" – identyfikator czujnika
- "type" – zawiera informację o typie czujnika (dozwolone wartości to *pressure* i *temperature*)
- "engine" – identyfikator silnika, przypisany do czujnika ciśnienia
- "master-sensor-id" – odnośnik do nadrzędnego czujnika ciśnienia, tylko czujniki temperaturowe zawierają odnośniki do nadrzędnego czujnika
- "value" – początkowa wartość danego czujnika, liczba całkowita

Serwer HTTP powinien umożliwić podanie URL-a do pliku wejściowego jako jeden z argumentów przy uruchomieniu w linii poleceń, przykładowo:

```
./server --config=https://github.com/relayr/pdm-test/blob/master/sensors.yml
```

Serwer powinien umożliwić wykonanie następujących zapytań HTTP:

1. Pobranie listy niepoprawnie działających silników.

```
curl -XGET "http://localhost:8080/engines?pressure_threshold=40&temp_threshold=50"
```

gdzie

- *pressure_threshold* - minimalna wartość ciśnienia, poniżej której uznajemy że silnik działa nieprawidłowo
- *temp_threshold* - maksymalna wartość temperatury, powyżej której uznajemy że silnik działa nieprawidłowo

Wynik zwrócony w odpowiedzi w body HTTP powinien zawierać listę silników, w których ciśnienie spadło poniżej podanego poziomu i jednocześnie temperatura na co najmniej jednym czujniku wzrosła powyżej podanej wartości. Wynik powinien być w postaci listy w formacie JSON, np.:

```
["123"]
```

1. Zaktualizowanie wartości dla czujnika.

```
curl -XPOST "http://localhost:8080/sensors/89145" -H "Content-Type: application/json" -d '{"operation": "increment", "value": "5"}'
```

Możliwe wartości dla pola *operation* to *set* (ustawienie zadanej wartości na *value*), *increment* (zwiększenie wartości czujnika o *value*) i *decrement* (zmniejszenie wartości czujnika o *value*).

Należy zapewnić, aby ustawione wartości nie wykroczyły poza wartości zdefiniowane przez atrybuty *min_value* i *max_value* zawarte w definicji w pliku *.yaml*.

Rozwiązanie składa się z następujących elementów:

- kod źródłowy w języku Java lub Scala
- dokumentacja uruchomieniowa: plik README zawierający informacje jak skompilować kod i uruchomić program oraz jak uruchomić testy jednostkowe

Całość powinna być spakowana do pliku w formacie ZIP i wysłana jako załącznik do e-maila.

Kryteria oceny:

- poprawność działania
- jakość dokumentacji uruchomieniowej
- oprócz tego na ocenę zadania wpływają następujące elementy:
 - przejrzystość i elegancja kodu
 - jakość testów jednostkowych

Dodatkowe punkty można zdobyć za możliwość uruchomienia usługi jako kontenera Docker.