Patryk Wawrzonek nr. Albumu: 402663

Dokumentacja projektowa IoT

Link do githuba: <https://github.com/Far-ki/IoT>

**Uruchomienie aplikacji**

Utworzyć:

* Resource group
* IoT Hub i 2 Device’y (u mnie są o nazwach: Device1, Device2)
* Storage account z kontenerami typu Blob (u mnie są o nazwach: kpicont,tempcont, deviceerrorcont)
* Stream Analytics job z 1 Inputem o nazwie ULZajeciaIoT

Należy wejść od folderu pobranego z repozytorium githuba i kliknąć IoTHubDevice.sln

Po uruchomieniu projektu zostaniemy poproszeni o 3 rzeczy a mianowicie:

* Device name znajduje się w: IoT Hub: Device management > Devices > [nazwa urządzenia]
* ConnectionString znajduje się w: IoT Hub: Device management > Devices > [nazwa urządzenia] > Primary connection string.
* UAstring w naszym przypadku to opc.tcp://localhost:4840/

Dla każdego urządzenia przysługuje jedna instancja programu, jeśli chcemy połączyć więcej urządzeń wystarczy uruchomić plik jeszcze raz.

Obraz zawierający tekst, zrzut ekranu, Czcionka, czarne

Opis wygenerowany automatycznie

Komunikacja z Azure

Widomość D2C

Aplikacja wysyła dane za pomocą D2C message, którego sekcja body wygląda tak :

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Ale także wiadomości o błędach które przechowywane są w blobe storage i wyglądają tak:

Obraz zawierający tekst, zrzut ekranu, Czcionka, czarne

Opis wygenerowany automatycznie

Device Twin

Za pomocą device twin trzymane jest aktualne tempo produkcji oraz błędy na urządzeniach.

Akcje te wywołane są na zmianę tych parametrów.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Wywołanie funkcji direct method EmergencyStop:

Wywołujemy funkcje EmergencyStop w Azure IoT Explorer.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Pole w symulatorze powinno się zaznaczyć po wywołaniu tej funkcji.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Wywołanie funkcji direct method ResetErrorStatus:

Wywołujemy funkcje ResetErrorStatus w Azure IoT Explorer.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

ResetErrorStatus powoduje usunięcie wszystkich błędów poniżej widzimy Symulator przed wywołaniem funkcji.

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

A po wywołaniu funkcji wszystkie błędy znikają.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

Kalkulacje

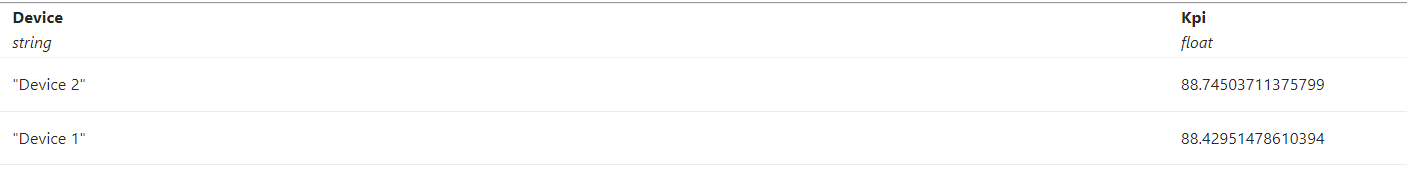
Zostały wykonane za pomocą Stream Analitics job.

KPI ( wskaźnik produktywności) z podziałem na 5 min , wyniki trzymane są w Blobie o nazwie kpicont.

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Wyniki:

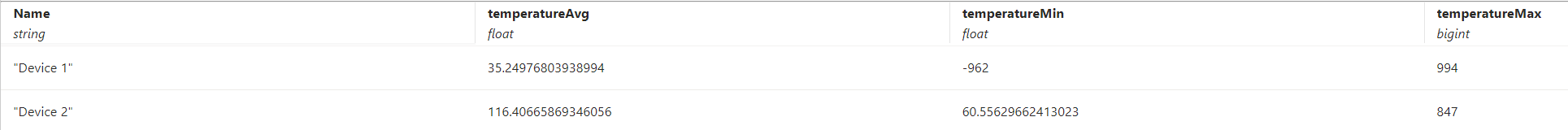


Temperatury przechowujące średnia, minimalną i maksymalną temperaturę z podziałem na 5 min z przeskokiem co min. Wynbiki przechowywane są w Blobie o nazie tempcont.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Wyniki:



Licznik błędów zlicza on liczbę wiadomości o błędzie w 1 minutowych oknach wyniki są przetrzymywane w Blobie o nazwie deviceerrorcont.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Wyniki

