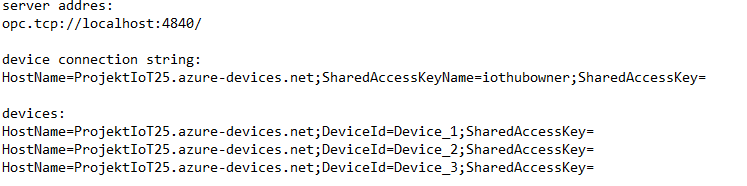
Przed uruchomieniem

Przed uruchomieniem AzureAgent należy stworzyć dowolną ilość urządzeń na IoT Hubie oraz wkleić ich podstawowe parametry połączenia w Settings.txt pod devices oraz podstawowe parametry połączenia iothubowner pod device connection string.

Przykład settings.txt:



W tym przykładzie będziemy używać 3 urządzeń. Następnie uruchamiamy udostępnioną aplikację IIoTSim oraz tworzymy tam naszą liczbę urządzeń.

Działanie AzureAgent

Pierwszym krokiem po uruchomieniu AzureAgent będzie wybranie maszyny na której Agent ma pracować.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatyczniePo zatwierdzeniu wyboru maszyny w konsoli będą wyświetlane aktualne dane telemetryczne maszyny.

Dane te są co 10 sekund odświeżane oraz zostaje wysłana wiadomość D2C zawierająca dane telemetryczne.

Przykładowa wiadomość zawierająca telemetrie:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

W przypadku pojawienia się błędu na maszynie będzie wysyłana jednorazowo wiadomość D2C zawierająca nazwę błędu.

Przykładowa wiadomość zawierająca informację o błędzie:

Obraz zawierający tekst, zrzut ekranu, linia, Czcionka

Opis wygenerowany automatycznie

Device twin oraz metody bezpośrednie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatyczniePo uruchomieniu AzureAgent device twin będzie wyglądał następująco:

Zawiera on pola desiredProduction, reportedProduction oraz deviceErrors.

deviceErrors (błędy urządzenia) oraz reportedProduction (zgłoszona produkcja) jest zmieniane na bieżąco. deviceErrors zostanie zmienione w przypadku wystąpienia błędu na maszynie, reportedProduction to aktualne production rate na maszynie.

desiredProduction (pożądana produkcja) to pole którego wartość zmieniamy używając metod bezpośrednich w IoTHubie. W celu ustawienia desiredProduction używamy metody SetDesiredProductionRatei jako ładunek ustawiamy docelową wartość.

Obraz zawierający zrzut ekranu, diagram, linia, design

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Po wywołaniu metody Production Rate na maszynie ulegnie zmianie.





Inne metody bezpośrednie:

EmergencyStop {}\* – wywołuję EmergencyStop na maszynie.

ResetErrorStatus{} – resetuje wszystkie errory na maszynie.

DecreaseDesiredProductionRate [value]\*\* – zmniejsza desired production rate o value (używane w business logic).

*\* {} – oznacza brak ładunku.*

*\*\* [value] – oznacza wartość podanego ładunku.*

Data calculations

W celu wykonania obliczeń w chmurze oraz ich przechowywania użyte zostały zapytania w Azure Stream Analytics oraz Blob storage

1. Zapytanie monitoruje wydajności produkcji. To zapytanie oblicza procent dobrych produktów dla każdego urządzenia w pięcio minutowych przedziałach i zapisuje wyniki w zestawie danych [iot-production].

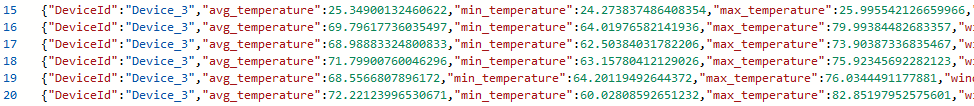
Przykładowa zawartość kontenera [iot-production]:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

2. Zapytanie monitoruje statystyki temperatury. To zapytanie oblicza średnią, minimalną i maksymalną temperaturę dla urządzeń w 1-minutowym oknie czasowym i zapisuje wyniki w [iot-temperature].

Przykładowa zawartość kontenera [iot-temperature]:



3. Zapytanie monitoruje błędy. To zapytanie sumuje liczbę błędów dla urządzeń w 1-minutowym oknie czasowym i zapisuje wyniki w [iot-errors]. Uwzględnia jedynie urządzenia, które przekroczyły 3 błędy.

Przykładowa zawartość kontenera [iot-errors]

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

Logika biznesowa

W celu wykonania logiki biznesowej użyte zostały Azure Logic app, Azure Stream Analytics oraz Blob storage.

1. Jeśli w urządzeniu wystąpią więcej niż 3 błędy w czasie krótszym niż 1 uruchom EmergencyStop.

Aby wykonać to zadanie najpierw przy pomocy zapytania w Azure Stream Analytics które monitoruje błędy (opisane w poprzednim punkcie) zapisujemy dane kiedy spełniają warunki. Używając Azure Logic app został ustawiony przełącznik który uruchamia się w momencie nadpisania [iot-errors]. Przełącznik wtedy uruchamia akcje http która wysyła do maszyny prośbę o wykonanie metody EmergencyStop.

Wygląd logiki w projektancie:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

1. Jeżeli w urządzeniu nastąpi spadek dobrej wydajności produkcyjnej poniżej 90% zmniejsz pożądane tempo produkcji o 10 punktów.

Aby wykonać to zadanie najpierw przy pomocy zapytania w Azure Stream Analytics które monitoruje spadek jakości produkcji (opisane poniżej) zapisujemy dane kiedy spełniają warunki. Używając Azure Logic app został ustawiony przełącznik który uruchamia się w momencie nadpisania [iot-decrease]. Przełącznik wtedy uruchamia akcje http która wysyła do maszyny prośbę o wykonanie metody DecreaseDesiredProductionRate.

Zapytanie monitoruje spadek jakości produkcji. To zapytanie oblicza procent dobrych produktów dla urządzeń w 5-minutowym oknie czasowym. Wyniki są zapisywane w [iot-decrease], ale uwzględniane są tylko przypadki, gdzie procent dobrych produktów spadł poniżej 90%.

Wygląd logiki w projektancie:

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Jeśli wystąpi błąd urządzenia (dowolnego typu) wyślij e-mail na wcześniej zdefiniowany adres.

Aby wykonać to zadanie najpierw przy pomocy zapytania w Azure Stream Analytics które wykrywa dowolne błędy (opisane poniżej) zapisujemy dane kiedy spełniają warunki. Używając Azure Logic app został ustawiony przełącznik który uruchamia się w momencie nadpisania [iot-anyerror]. Przełącznik wtedy uruchamia akcje która wysyła email na wcześniej zdefiniowany adres email.

Zapytanie wykrywa dowolne błędy. To zapytanie sumuje liczbę błędów dla urządzeń w 1-minutowym oknie czasowym i zapisuje wyniki w [iot-anyerror]. Uwzględnia tylko urządzenia, dla których wystąpił co najmniej jeden błąd w danym oknie czasowym.

Wygląd logiki w projektancie:

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Przykład otrzymanego maila:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie