



## Laboratorium 3

# Wprowadzenie do IoT Analytics

**Zakres:** Laboratorium obrazuje podstawy logowania danych z wykorzystaniem technologii chmury.

**Zasady zaliczenia:**

- zadanie 1 – 10%; Przygotowanie IoT Analytics
- zadanie 2 – 20%; Rejestracja komponentu.
- zadanie 3 – 20%; Podłączanie się do IoT Analytics.
- zadanie 4 – 50%; Zadanie do samodzielnej realizacji.

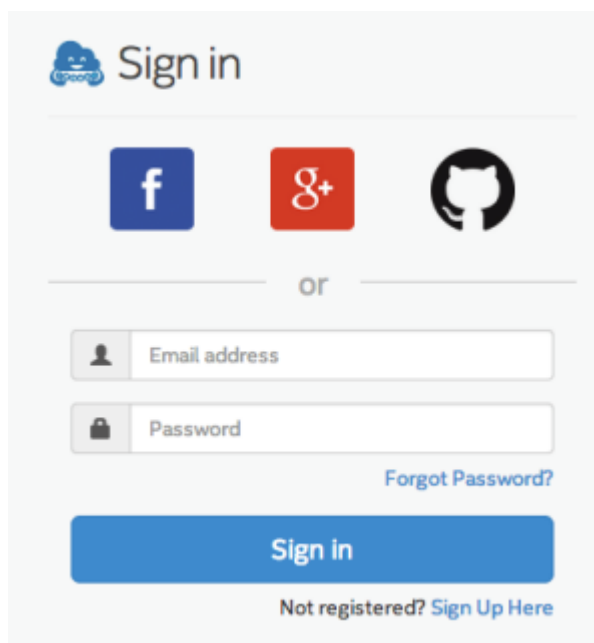
### 1. Przygotowanie IoT Analytics

Przygotuj stanowisko według instrukcji:

<https://software.intel.com/en-us/intel-iot-platforms-getting-started-cloud-analytics>

Założ konto na:

<https://dashboard.us.enableiot.com/ui/auth#/login>



Rysunek 1. Logowanie w IoT Analytics



### My Dashboard

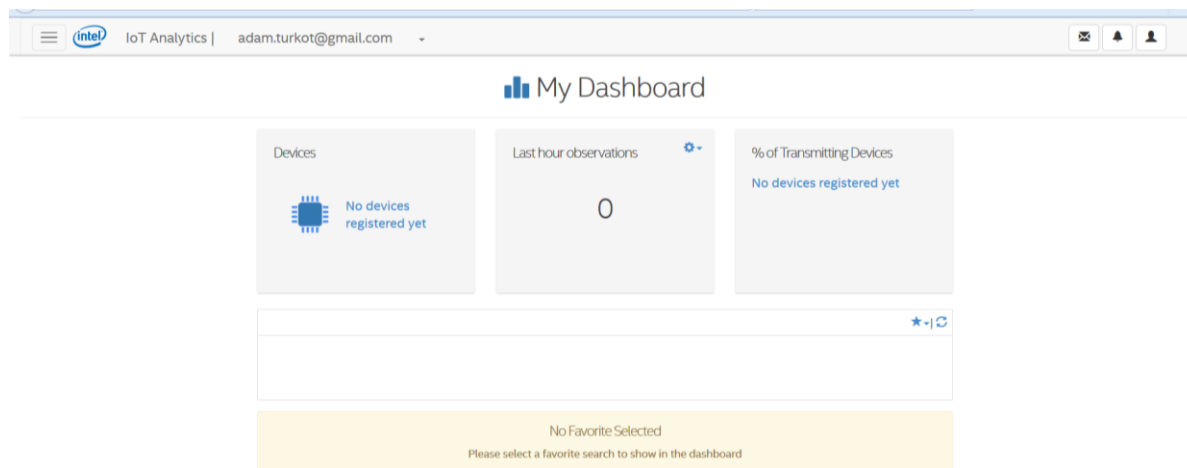


### Create new Account

Hi  
ANewUser@domain.com  
Please enter a name for your  
account.

Rysunek 2. Tworzenie konta w IoT Analytics

W tym momencie dostępna jest usługa My Dashboard.



Rysunek 3. Strona My Dashboard

The My Dashboard page provides access to all system functions, including the number of devices, the number of observations received in the past hour, and the percentage of devices that are “healthy,” meaning they have transmitted data recently. You can customize the lower section to show a “favorite” chart, which contains a chart of specific time series from specific devices.

Do prawidłowej pracy wymagane jest ustawienie daty systemowej w systemie Galileo – (pamiętaj że po wyłączeniu zasilanie nie jest mierzony czas). W systemie linux służy do tego polecenia date.



```
date -s "2015-01-29 16:54:00"
```

```
COM17 - PuTTY
root@galileo:~#
root@galileo:~# date -s "2015-01-29 16:54:00"
Thu Jan 29 16:54:00 UTC 2015
root@galileo:~# █
```

Sprawdzenie czy iotkit jest zainstalowany w systemie – jeżeli jest polecenie `iotkit-admin` oznacza to że jest zainstalowany, jeżeli nie ma to trzeba zainstalować. W przypadku zainstalowanego wcześniej obrazu systemu jest iotkit jest zainstalowany w systemie:



Sponsorspecjalności

```
COM17 - PuTTY
root@galileo:~# iotkit-admin

Usage: iotkit-admin [options] [command]

Commands:

  test                Tries to reach the server (using the current protocol
).
  activate <activation_code> Activates the device.
  register <comp_name> <catalogid> Registers a component in the device.
  reset-components    Clears the component list.
  observation <comp_name> <value> Sends an observation for the device, for the
specific component.
  catalog             Displays the Catalog from the device's account.
  components          Displays components registered for this device.
  initialize          Resets both the token and the component's list.
  update             Send update device request to dashboard
  pull-actuations    Fetches actuations from last time action was executed
or 24h if never and executes them one by one
  protocol <protocol> Set the protocol to 'mqtt' or 'rest' or 'rest+ws'
  host <host> [<port>] Sets the cloud hostname for the current protocol.
  device-id          Displays the device id.
  set-device-id <id> Overrides the device id.
  clear-device-id     Reverts to using the default device id.
  save-code <activation_code> Adds the activation code to the device.
  reset-code         Clears the activation code of the device.
  proxy <host> <port> Sets proxy For REST protocol.
  reset-proxy        Clears proxy For REST protocol.
  set-logger-level <level> Set the logger level to 'debug', 'info', 'warn', 'e
rror'
  set-data-directory <path> Sets path of directory that contains sensor data.
  reset-data-directory Resets to default the path of directory that contains
sensor data.
  move-data-directory <path> Change directory where data will be stored
  gateway-id         Displays the gateway id.
  set-gateway-id <id> Overrides the gateway id.
  set-device-name <name> Change device name
  reset-device-name  Resets to default device name.
  set-udp-port <udp_port> Overrides the port UDP listener binds to
*
                    Error message for non valid command

Options:

  -h, --help          output usage information
  -V, --version       output the version number
```

Kolejnym krokiem jest sprawdzenie połączenia internetowego:

```
iotkit-admin test.
```



Najczęstszym przypadkiem błędu połączenia jest błędna konfiguracja sieci:

```
COM17 - PuTTY
root@galileo:~# iotkit-admin test
2015-01-29T17:00:51.416Z - info: Trying to connect to host ...
2015-01-29T17:00:52.999Z - error: Data Submission Error code=ESRCH, errno=ESRCH
, syscall=getaddrinfo
2015-01-29T17:00:53.029Z - error: Connection failed to dashboard.us.enableiot.co
m
root@galileo:~#
```

W przypadku pozytywnego połączenia zostają wyświetlone komunikaty:

```
COM17 - PuTTY
ionice      iostat      iotkit-admin  iotkit-agent  iotkit-comm
root@galileo:~# iotkit-admin test
2016-07-03T13:57:27.472Z - info: Trying to connect to host ...
2016-07-03T13:57:29.962Z - info: Connected to dashboard.us.enableiot.com
2016-07-03T13:57:29.980Z - info: Environment: prod
2016-07-03T13:57:29.986Z - info: Build: 0.14.5
root@galileo:~#
```

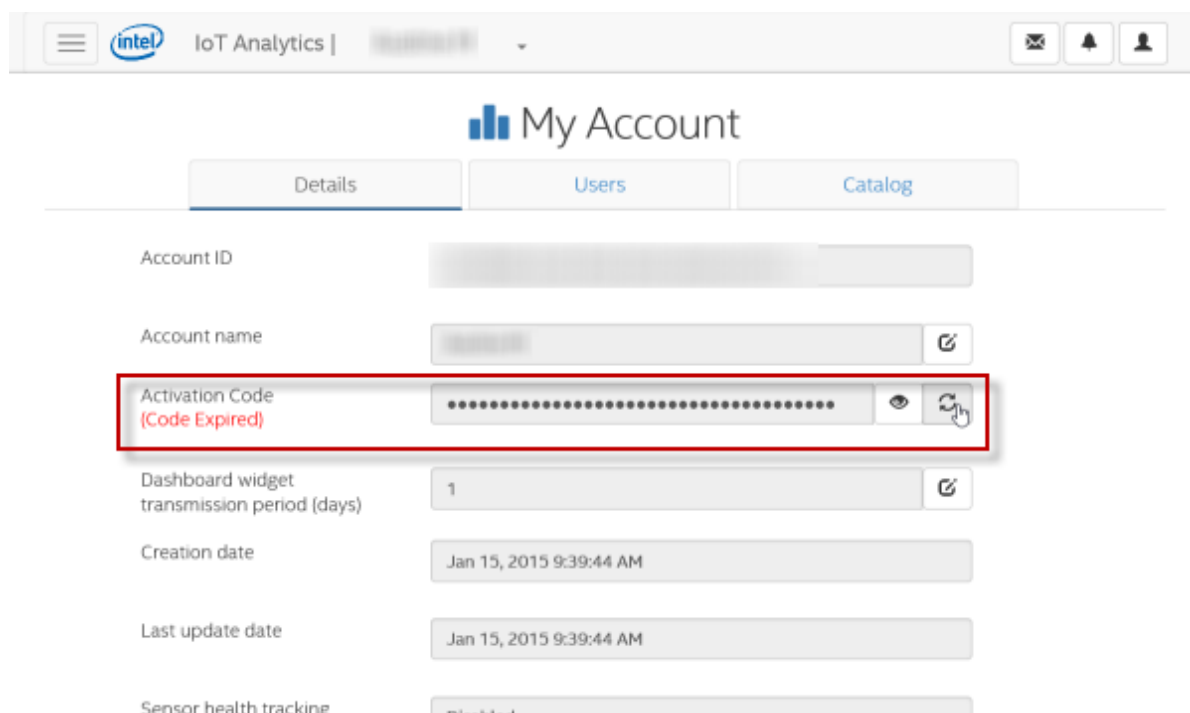
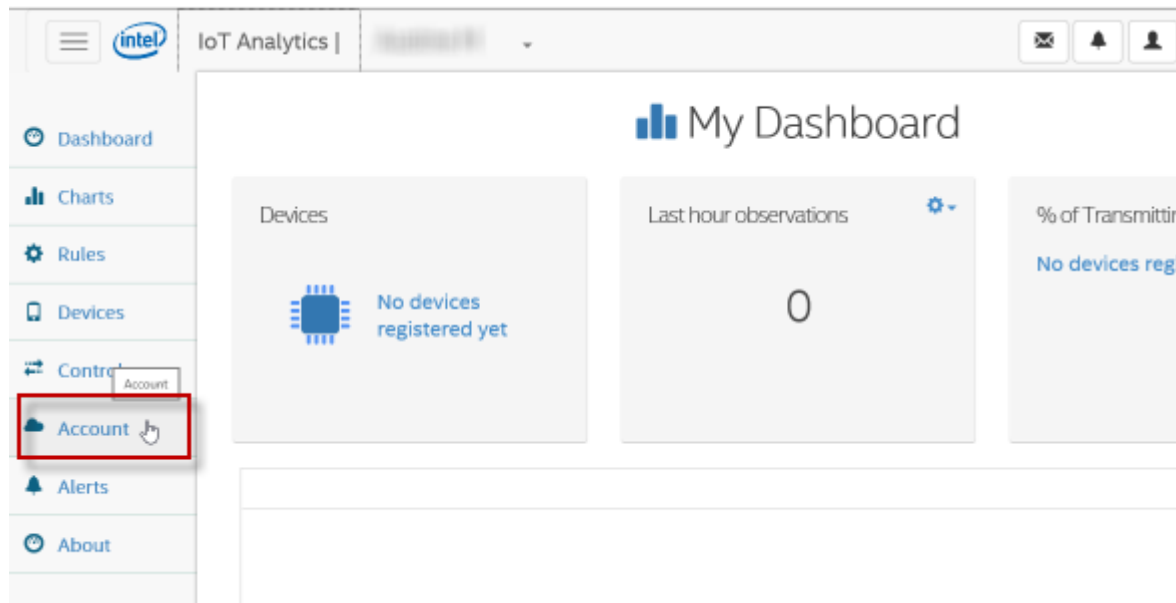
Następnym krokiem jest pobranie id urządzenia:

Kolejnym krokiem jest sprawdzenie połączenia internetowego:

**iotkit-admin device-id.**

```
COM17 - PuTTY
2016-07-03T13:57:27.472Z - info: Trying to connect to host ...
2016-07-03T13:57:29.962Z - info: Connected to dashboard.us.enableiot.com
2016-07-03T13:57:29.980Z - info: Environment: prod
2016-07-03T13:57:29.986Z - info: Build: 0.14.5
root@galileo:~# iotkit-admin device-id
2016-07-03T14:00:36.128Z - info: Device ID: 98-4F-EE-05-5E-C6
root@galileo:~#
```

Kolejny krok to aktywacja urządzenia:



Kod aktywacji jest aktywny tylko przez 1 minutę. Po wygaśnięciu można go odświeżyć. Otrzymany kod należy wykorzystać w poleceniu:

```
iotkit-admin activate activation_code
```

## 2. Rejestracja komponentu

Zgodnie z instrukcją ze strony:

<https://software.intel.com/en-us/intel-iot-platforms-getting-started-cloud-analytics>



zarejestruj komponent do rejestracji temperatury oraz przetestuj jego działanie, a także przesyłanie pakietów UDP.

### 3. Podłączanie się do IoT Analytics

Dla zarejestrowanego komponentu wykorzystaj dwie metody łączenia:

Z wykorzystaniem: IoT Kit Arduino library.

Z wykorzystaniem: REST API w JS.

### 4. Zadanie do samodzielnej realizacji

Napisz program z wykorzystaniem Arduino IDE rejestrujący w IoT Analytics bieżącą temperaturę, natężenie światła oraz pozycję potencjometru. Dane odczytaj za pomocą sensorów z Grove Starter Kit Plus co podane przez prowadzącego przedział czasu. Przygotuj ich wizualizację za pomocą panelu Charts strony IoT Analytics Dashboard.