

## Sommario

1	Descrizione Servizio Azure IoT Hub .....	2
2	Messaggi di telemetria .....	4
2.1	Messaggio completo .....	4
2.2	Risparmio dati .....	4
3	Interrogazioni all'IoT Scada .....	6
3.1	Informazioni del sistema .....	7
3.1	Informazioni sull'IoT Scada .....	7
3.2	Configurazione dei devices collegati .....	8
4	Informazioni sulle variabili .....	9
4.1	Configurazione delle variabili .....	9
4.2	Dati attuali delle variabili .....	11
4.3	Dati di Log .....	12
4.4	Settaggio di una variabile .....	13
5	Informazioni sugli allarmi .....	14
5.1	Configurazione degli allarmi .....	14
5.2	Stato degli allarmi .....	15
6	Informazioni sugli eventi .....	16
6.1	Configurazione degli eventi .....	16
6.2	Dati storici sugli eventi .....	17
7	Altri messaggi automatici .....	19
7.1	Nuovo allarme sollevato .....	19
7.2	Nuovo evento sollevato .....	20

## 1 Descrizione Servizio Azure IoT Hub

L'IOT SCADA fornisce il servizio di inoltro messaggi al cloud mediante il servizio Microsoft Azure IoT Hub.

Se la licenza in possesso è abilitata e se il servizio è stato attivato, l'IoT Scada invierà periodicamente all'IoT Hub configurato dei messaggi di telemetria sotto forma di stringa in formato JSON.

L'IoT Scada fornisce un meccanismo di configurazione dei permessi (lettura/scrittura/inoltro allarmi) relativi alle variabili da inoltrare, pertanto verranno inoltrate tramite i messaggi di telemetria solamente le variabili per cui è stato preventivamente fornito tale permesso. Anche l'operazione di scrittura (se logicamente possibile) sarà effettuata solamente se è stato preventivamente abilitato tale permesso nella configurazione.

Esistono attualmente due tipologie di messaggi inoltrabili in maniera automatica dal dispositivo, che differiscono semplicemente per numero di informazioni contenenti, quindi dimensione finale del messaggio da inviare.

Nell'IoT Scada ogni singola variabile raccolta dal campo è identificata univocamente dalla coppia devId (identificativo del device di appartenenza della variabile) e varId (identificativo della variabile). E' possibile che alcune variabili abbiano un devId non valorizzato: queste sono le cosiddette variabili d'Impianto (chiamate anche Misure Personalizzate nell' IoT Scada) che non possiedono tale identificativo per la natura stessa per cui sono state create: esse non appartengono infatti ad alcun dispositivo, ma sono il risultato di un'elaborazione fatta su più variabili, tendenzialmente provenienti da dispositivi diversi (esempio: la somma dei kWh letti da N inverter collegati all'IoT Scada).

Il servizio Microsoft Azure IoT Hub prevede il pagamento di licenze con prezzo variabile a seconda del numero di messaggi giornalieri inviati verso l'hub. Attualmente il payload trasportato da ogni singolo messaggio corrisponde a 0,5Kb per la versione free trial e 4Kb per le versioni a pagamento. E' quindi importante cercare di trasferire la massima

quantità di dati nel minimo spazio possibile. Per far ciò è stato ideato un meccanismo che predeve tali operazioni:

L'utilizzatore finale dei dati effettua una prima ed unica interrogazione inoltrando messaggi con una sintassi ben definita (spiegata in seguito in questo manuale) verso l'Hub e quindi verso l'IoT Scada per ottenere in risposta la configurazione del dispositivo, device connessi, configurazione di variabili, allarmi, eventi.

Successivamente, elaborando tali informazioni, l'utilizzatore avrà a disposizione le chiavi per identificare univocamente i dati inoltrati automaticamente nei successivi messaggi di telemetria cosicché possa essere ridotta al minimo la quantità di dati da inoltrare per ogni singola variabile.

## 2 Messaggi di telemetria

I messaggi di telemetria inoltrati automaticamente a frequenza impostabile nell'interfaccia del dispositivo hanno la seguente sintassi:

### 2.1 Messaggio completo

```
{
  "telemetryDataList": [
    {
      "date": "Jan 27, 2017 10:04:11 AM", //java.util.Date
      "devId": 31, //int
      "varId": 39, //int
      "value": 0, //Object(String,Boolean,Numeric)
      "quality": false, //boolean
    }, //..... N variabili
  ],
  "devSn": "IOTSPIXXXXXXXXXX", //String
  "onTime": "Jan 27, 2017 10:04:11 AM" java.util.Date
}
```

### 2.2 Risparmio dati

```
{
  "telemetryDataList": [
    {
      "devId": 31,
      "varId": 39,
      "value": 0,
      "quality": false,
    }, //..... N variabili
  ],
  "devSn": "IOTSPIXXXXXXXXXX",
  "onTime": "Jan 27, 2017 10:04:11 AM"
}
```

La lista di telemetryData restituita dall' IoT Scada sono oggetti JSON composti da:

- devId – identificatore del device a cui appartiene la variabile (può essere null in caso di variabili d'impianto)
- varId – identificatore della variabile

- value – contiene il valore letto per la variabile
- quality – valore booleano che indica se l'ultimo tentativo di lettura ha avuto successo o meno, quindi se il valore contenuto in value è attuale o potenzialmente datato
- date – data a cui risale l'ultima lettura corretta della variabile

Ogni messaggio avrà poi due campi di intestazione:

- devSn – contiene il seriale univoco del dispositivo che sta inoltrando le variabili
- onTime – data di inoltro del messaggio

N.B. E' possibile discriminare tra queste due tipologie di messaggi semplicemente spuntando o meno nell'interfaccia dell'IoT Scada la voce *"Il messaggio inoltrato conterra' solamente le informazioni essenziali"*

### 3 Interrogazioni all'IoT Scada

Come precedentemente accennato, l'IoT Scada può ricevere messaggi inoltrati dall'hub, rispettando una certa sintassi, e rispondere puntualmente a tali richieste di informazioni. Tutti i messaggi riconosciuti dall'IoT Scada condividono una stessa struttura di base, da compilare diversamente a seconda del tipo di richiesta.

Il messaggio di base riconosciuto dall'IOT SCADA possiede la seguente struttura:

```
{
  "component": "",
  "operation": "",
  "devId": [],
  "varId": [],
  "startTime": "",
  "endTime": "",
  "value": ""
}
```

Ovviamente, come verrà spiegato in seguito, alcuni campi dovranno essere compilati per determinate chiamate, e lasciati vuoti in altre, pena il mancato riconoscimento di tale messaggio e quindi di risposta.

La struttura di base è composta da:

- component – (obbligatorio) - enum dei componenti coinvolti (INFO, DEVICES, ALARMS, EVENTS)
- operation – (obbligatorio tranne per component INFO) - enum delle operazioni possibili (LIST, DATA, CONFIG, LOGDATA, SET, ACTIVE, HISTORY)
- devId – (opzionale) – lista di devicelds
- varId – (opzionale) – lista delle variabili
- startTime - (opzionale) – estremo iniziale dell'intervallo temporale su cui estrarre i dati
- endTime - (opzionale) – estremo finale dell'intervallo temporale su cui estrarre i dati
- value - (opzionale) – valore da scrivere nella variabile

## 3.1 Informazioni del sistema

### 3.1 Informazioni sull'IoT Scada

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{  
  "component":"INFO"  
}
```

L'IOT SCADA risponderà con una serie di informazioni relative all'IoT Scada:

```
{  
  "uuid": "49ab91753-9b94-3d2e-8ff6-17d3d5113606",  
  "hwModel": "4.0.5-SNAPSHOT",  
  "name": "IoT Gateway",  
  "webAppVersion": "4.0.3",  
  "devSn": " IOTSPIXXXXXXXXX ",  
  "onTime": "Sep 20, 2017 5:23:12 PM"  
}
```

### 3.2 Configurazione dei devices collegati

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "DEVICES",
  "operation": "LIST",
  "devId": [ 2, 3, 4 ] //OPZIONALE
}
```

Il parametro lista devId è facoltativo e serve per filtrare le informazioni: omettendo tale parametro il dispositivo fornirà le informazioni di tutti i device. Impostandolo, verranno restituite solamente le informazioni di quelli selezionati.

L'IOT SCADA risponderà con una serie di informazioni relative al device:

```
{
  "devices": [
    {
      "devId": 2,
      "description": "Fanuc 750",
      "linked": true
    },
    {
      "devId": 3,
      "description": "Fanuc 545",
      "linked": true
    },
    {
      "devId": 4,
      "description": "MCM-cnc",
      "linked": false
    }
  ],
  "devSn": " IOTSPIXXXXXXXXX ",
  "onTime": "Sep 20, 2017 5:23:12 PM"
}
```



## 4 Informazioni sulle variabili

### 4.1 Configurazione delle variabili

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "DEVICES",
  "operation": "CONFIG",
  "devId": [ 63 ], //OPZIONALE
  "varId": [ 40, 41, 39 ] //OPZIONALE
}
```

Il parametro lista devId è facoltativo e serve per filtrare le informazioni: omettendo tale parametro il dispositivo fornirà le informazioni di tutti i device. Impostandolo, verranno restituite solamente le informazioni di quelli selezionati. Il parametro lista varId è facoltativo. Se specificato, dovrà esserci **un solo** devId specificato nel corrispettivo parametro. Il dispositivo fornirà solamente le informazioni relative ai varId selezionati per tale dispositivo.

L'IOT SCADA risponderà con una serie di informazioni relative alle variabili:

```
{
  "varConfigList": [
    {
      "devId": 63,
      "varId": 21,
      "description": "Actual executed program name",
      "dataType": "String",
      "minimum": "null",
      "maximum": "null",
      "category": [ "main" ],
      "alarmable": false,
      "writable": false
    },
    {
      "devId": 63,
      "varId": 23,
      "description": "Automatic mode selection",
      "dataType": "Numeric",
      "minimum": "null",
      "maximum": "null",
    }
  ]
}
```

```
"category": [☐ "main" ],  
"alarmable": false,  
"writable": false  
],  
"devSn": " IOTSPIXXXXXXXXX",  
"onTime": "Feb 20, 2017 4:11:13 PM"  
}
```

## 4.2 Dati attuali delle variabili

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "DEVICES",
  "operation": "DATA",
  "devId": [ 63 ], //OPZIONALE
  "varId": [ 23, 430 ] //OPZIONALE
}
```

Il parametro lista devId è facoltativo e serve per filtrare le informazioni: omettendo tale parametro il dispositivo fornirà le informazioni di tutti i device. Impostandolo, verranno restituite solamente le informazioni di quelli selezionati. Il parametro lista varId è facoltativo. Se specificato, dovrà esserci **un solo** devId specificato nel corrispettivo parametro. Il dispositivo fornirà solamente le informazioni relative ai varId selezionati per tale dispositivo.

L'IOT SCADA risponderà con una serie di informazioni relative alle variabili:

```
{
  "variablesList": [
    {
      "devId": 63,
      "varId": 23,
      "value": "MDI",
      "decodedValue": "MDI",
      "date": "Sep 20, 2017 5:48:57 PM",
      "quality": false
    },
    {
      "devId": 63,
      "varId": 430,
      "value": false,
      "date": "Sep 20, 2017 5:48:57 PM",
      "quality": false
    }
  ],
  "devSn": " IOTSPIXXXXXXXXX",
  "onTime": "Feb 20, 2017 4:29:46 PM"
}
```

### 4.3 Dati di Log

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "DEVICES",
  "operation": "LOGDATA",
  "devId": [ 63 ], //OBBLIGATORIO
  "varId": [ 23 ], //OBBLIGATORIO
  "startTime": 0, //OPZIONALE
  "endTime": 123456789 //OPZIONALE
}
```

I parametri lista devId e varId sono obbligatori. Dovrà esserci un solo valore in ogni lista. I parametri startTime e endTime sono facoltativi. Tali parametri applicano un filtro temporale sui dati da restituire. I valori da inserire sono espressi in millisecondi a partire dal 1/1/1070 0:00 GMT.

L'IOT SCADA risponderà con una serie di informazioni relative alle variabili:

```
{
  "devId": 63,
  "varId": 23,
  "value": "MDI",
  "date": "Sep 8, 2017 1:37:14 PM",
  "quality": true
},
{
  "devId": 63,
  "varId": 23,
  "value": "MDI",
  "date": "Sep 8, 2017 1:40:14 PM",
  "quality": true
}
],
"devSn": " IOTSPIXXXXXXXXX",
"onTime": "Feb 20, 2017 5:02:49 PM"
}
```

I campi devId e varId sono facoltativi. Se lasciati vuoti, l'IOT SCADA risponderà con la lista di tutte le variabili; se specificato uno o più devIds, l'output verrà filtrato restituendo solamente quelli richiesti. Se specificati anche una lista di varId, il devId dovrà essere univoco: verranno restituite solamente le variabili con tale devId e tali varId.

#### 4.4 *Settaggio di una variabile*

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "DEVICES",
  "operation": "SET",
  "devId": [ 31 ], //OBBLIGATORIO
  "varId": [ 47 ], //OBBLIGATORIO
  "value": 10 //OBBLIGATORIO
}
```

I parametri lista devId e varId sono obbligatori. Dovrà esserci un solo valore in ogni lista.

Il parametro value indica il valore a cui si vuole settare la variabile.

L'IOT SCADA risponderà con una serie di informazioni relative alle variabili:

```
{
  "accepted": true,
  "description": "Accepted",
  "devSn": " IOTSPIXXXXXXXXXX",
  "onTime": "Feb 20, 2017 5:02:49 PM"
}
```

## 5 Informazioni sugli allarmi

### 5.1 Configurazione degli allarmi

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component":"ALARMS",
  "operation":"CONFIG",
  "varId": [ 47, 48] //OPZIONALE
}
```

Il parametro lista varId è facoltativo e serve per filtrare le informazioni: omettendo tale parametro il dispositivo fornirà le informazioni di tutti gli allarmi. Impostandolo, verranno restituite solamente le informazioni di quelli selezionati.

L'IOT SCADA risponderà con una serie di informazioni relative agli allarmi:

```
{
  "alarmConfigList": [
    {
      "id": 48,
      "description": "ALLARME CONTROTESTA",
      "condition": "$G_63_86 eq true"
    },
    {
      "id": 43,
      "description": "ALL.UTENSILI MOTORIZZATI",
      "condition": "$G_63_81 eq true"
    },
    {
      "id": 47,
      "description": "ALLARME CARICATORE",
      "condition": "$G_63_85 eq true"
    }
  ],
  "devSn": " IOTSPIXXXXXXXXX",
  "onTime": "Feb 20, 2017 5:25:48 PM"
}
```

## 5.2 Stato degli allarmi

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "ALARMS",
  "operation": "DATA",
  "varId": [ 47, 48] //OPZIONALE
}
```

Il parametro lista varId è facoltativo e serve per filtrare le informazioni: omettendo tale parametro il dispositivo fornirà le informazioni di tutti gli allarmi. Impostandolo, verranno restituite solamente le informazioni di quelli selezionati.

L'IOT SCADA risponderà con una serie di informazioni relative agli allarmi:

```
{
  "alarmDataList": [
    {
      "id": 48,
      "quality": true,
      "alarmed": false
    },
    {
      "id": 43,
      "quality": true,
      "alarmed": true
    },
    {
      "id": 47,
      "quality": false,
      "alarmed": false
    }
  ],
  "devSn": " IOTSPIXXXXXXXXX",
  "onTime": "Feb 20, 2017 5:25:48 PM"
}
```

## 6 Informazioni sugli eventi

### 6.1 Configurazione degli eventi

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component": "EVENTS",
  "operation": "INFO",
  "varId": [ 1 ] //OPZIONALE
}
```

Il parametro lista varId è facoltativo e serve per filtrare le informazioni: omettendo tale parametro il dispositivo fornirà le informazioni di tutti gli eventi. Impostandolo, verranno restituite solamente le informazioni di quelli selezionati.

L'IOT SCADA risponderà con una serie di informazioni relative agli eventi:

```
{
  "eventsInfoList": [
    {
      "eventId": 1,
      "eventName": "Nuovo Evento",
      "type": "boolean",
      "condition": "$G_63_71",
      "snapshotGlobalIds": "G_63_21",
      "comparisonOperator": "eq",
      "numericCompareValue": 1
    }
  ],
  "devSn": " IOTSPIXXXXXXXXX",
  "onTime": "Feb 20, 2017 5:25:48 PM"
}
```



## 6.2 Dati storici sugli eventi

Per ottenere tali informazioni il messaggio JSON da inoltrare è:

```
{
  "component":"EVENTS",
  "operation":"HISTORY",
  "varId": [ 1 ],    //OBBLIGATORIO
  "startTime":0,    //OPZIONALE
  "endTime":123456789 //OPZIONALE
}
```

Il parametro lista varId è obbligatorio. Dovrà esserci un solo valore nella lista. I parametri startTime e endTime sono facoltativi. Tali parametri applicano un filtro temporale sui dati da restituire. I valori da inserire sono espressi in millisecondi a partire dal 1/1/1070 0:00 GMT.

L'IOT SCADA risponderà con una serie di informazioni relative agli eventi:

```
{
  "eventHistoryList": [
    {
      "eventId": 1,
      "eventName": "Nuovo Evento",
      "timestamp": "Sep 11, 2017 11:00:58 AM",
      "state": true,
      "variablesSnapshot": [
        {
          "devId": 63,
          "varId": 21,
          "value": "//CNC_MEM/USER/PATH1/TECNO",
          "quality": true
        }
      ]
    }
  ],
  {
    "eventId": 1,
    "eventName": "Nuovo Evento",
    "timestamp": "Sep 11, 2017 11:11:28 AM",
    "state": true,
    "variablesSnapshot": [
      {
        "devId": 63,
        "varId": 21,
        "value": "//CNC_MEM/USER/PATH1/O9110",

```

```
    "quality": true
  }
]
},
],
"devSn": " IOTSPIXXXXXXXXX",
"onTime": "Feb 20, 2017 5:25:48 PM"
}
```

## 7 Altri messaggi automatici

Una volta attivato il servizio Azure IoT Hub sull'IoT Scada, questo potrà inoltrare automaticamente verso l'hub, oltre ovviamente ai messaggi di telemetria, messaggi relativi a nuovi allarmi e/o eventi, nel momento in cui questi si verificano.

Ovviamente come per le variabili, per far sì che il servizio li possa notificare, anche eventuali allarmi personalizzati e/o eventi dovranno prima essere abilitati all'inoltro nell'interfaccia dell'IoT Scada.

Le sintassi per riconoscere tali messaggi sono le seguenti.

### 7.1 Nuovo allarme sollevato

In caso di un nuovo allarme, verrà inoltrato verso l'hub il seguente messaggio JSON:

```
{
  "activeAlarmsList": [
    {
      "id": 11,
      "eventId": 16153,
      "deviceName": "Fanuc 545",
      "measure": "RIPARO APERTO",
      "description": "RIPARO APERTO",
      "onDate": "Sep 22, 2017 10:08:09 AM",
      "offDate": "Sep 22, 2017 10:10:25 AM"
    }
  ],
  "devSn": " IOTSPIXXXXXXXX",
  "onTime": "Feb 20, 2017 5:25:48 PM"
}
```

Il messaggio verrà inoltrato due volte: nel momento dell'attivazione dell'allarme, e sarà quindi ovviamente assente il parametro offDate, e nel momento del suo rientro, e sarà compilato come esattamente come sopra.

## 7.2 Nuovo evento sollevato

In caso di un nuovo evento, verrà inoltrato verso l'hub il seguente messaggio JSON:

```
{
  "newEventsList": [
    {
      "eventId": 1,
      "eventName": "Nuovo Evento",
      "type": "boolean",
      "condition": "$G_63_71",
      "snapshotGlobalIds": "G_63_21,G_63_820",
      "comparisonOperator": "eq",
      "numericCompareValue": 1,
      "timestamp": "Sep 21, 2017 2:58:49 PM",
      "snapshotVarsDatas": [
        {
          "globalId": "G_63_21",
          "snapshotValue": "//CNC_MEM/USER/PATH1/TECNO"
        },
        {
          "globalId": "G_63_820",
          "snapshotValue": false
        }
      ],
      "eventValue": "true"
    }
  ],
  "devSn": " IOTSPIXXXXXXXXXX",
  "onTime": "Feb 20, 2017 5:25:48 PM"
}
```

Esistono due tipologie di eventi: *boolean*, che sono scatenati al verificarsi di una condizione, e *onChange*, scatenati al cambiamento di valore di una variabile.

Per la tipologia di evento booleano verranno inoltrati due messaggi verso l'hub: uno nel momento in cui la condizione si verifica, un altro nel momento in cui la condizione risulta non più vera.

Gli eventi di tipo *onChange*, per loro natura, inoltreranno un messaggio verso l'hub ogni qualvolta la variabile che monitorano cambia di valore.