



MANUALE D'USO ISC API REST

IoT Server REST API

1	Revision History	3
2	Introduzione.....	4
3	API informazioni di sistema	7
3.1	GET /api/v2/info.json	7
4	API configurazione dispositivi	8
4.1	GET /api/v2/devices.json	8
4.2	GET /api/v2/devices/{deviceId}/config.json	10
5	API configurazione variabili	11
5.1	GET /api/v2/devices/{deviceId}/variables/config.json	12
5.2	GET /api/v2/devices/{deviceId}/variables/{variableId}/config.json	13
5.3	GET /api/v2/devices/custom/variables/config.json	14
5.4	GET /api/v2/devices/custom/variables/{variableId}/config.json	15
6	API dati	16
6.1	GET /api/v2/devices/{deviceId}/variables/data.json.....	17
6.2	GET /api/v2/devices/{deviceId}/variables/{variableId}/data.json.....	18
6.3	GET /api/v2/devices/{deviceId}/variables/{variableId}/set.json	19
6.4	GET /api/v2/devices/{deviceId}/variables/{variableId}/logdata.json.....	20
6.5	GET /api/v2/devices/custom/variables/data.json.....	21
6.6	GET /api/v2/devices/custom/variables/{variableId}/data.json.....	22
6.7	GET /api/v2/devices/custom/variables/{variableId}/logdata.json.....	23
7	Gestione file e cartelle dei dispositivi collegati	24
7.1	GET /api/v2/devices/documents/list	24
7.2	GET /api/v2/devices/{deviceId}/documents/list	25
7.3	POST /api/v2/devices/{deviceId}/documents/upload.....	27
7.4	GET /api/v2/devices/{deviceId}/documents/download.....	28

7.5	DELETE /api/v2/devices/{deviceId}/documents/delete	29
7.6	POST /api/v2/devices/{deviceId}/documents/createSubfolder	30
7.7	DELETE /api/v2/devices/{deviceId}/documents/deleteSubfolder	31
8	API configurazione allarmi	32
8.1	GET /api/v2/alarms.json	32
8.2	GET /api/v2/alarms/{alarmId}/config.json	33
8.3	GET /api/v2/alarms/active.json	34
8.4	GET /api/v2/alarms/history.json.....	36
9	API eventi	38
9.1	GET /api/v2/events/info.json.....	38
9.2	GET /api/v2/events/{eventId}/info.json	40
9.3	GET /api/v2/events/{eventId}/history.json	41

1 Revision History

Rev	Data	Descrizione
1.0	2015-07-13	Prima bozza delle API v1 con API configurazione dispositivi, configurazione variabili di dispositivo, dati istantanei variabili di dispositivo.
1.1	2015-12-21	Estensione delle API v1 con API informazioni di sistema, configurazione variabili di impianto, dati istantanei variabili di impianto, storico dati variabili di dispositivo ed impianto, allarmi istantanei e storico allarmi.
1.2	2015-02-15	Correzione JSON di esempio nella GET /api/v1/devices/{deviceId}/variables/{variableId}/config.json
1.3	2017-03-29	DRAFT API list/upload/download file
2.0	2018-05-07	Riordino API pubblicate sull'endpoint v2

2 Introduzione

IoT Server fornisce accesso real time alle sue informazioni attraverso delle API REST di differenti tipologie:

- **API informazioni** – informazioni generali di sistema
- **API configurazione dispositivi** – recupero della configurazione dei dispositivi presenti in campo
- **API configurazione variabili** – recupero della configurazione dei dati ottenibili dai dispositivi presenti in campo, della configurazione dei dati di impianto e dei dati definiti dall'utente
- **API dati** – recupero del valore di ogni variabile ottenibile dai dispositivi e dall'impianto e recupero del loro storico. Scrittura delle variabili scrivibili.
- **API documents** – recupero informazioni sui file in una determinata directory di un device, upload e download di file da e verso un device, cancellazione, creazione di cartelle
- **API allarmi** – recupero dello stato degli allarmi configurati nell'IoT Server e del loro storico
- **API eventi** – recupero dello stato degli eventi configurati nell'IoT Server e del loro storico

Per tutte le API la rappresentazione dei dati ritornati è in formato JSON.

N.B. Il suffisso *.json* di ogni endpoint è opzionale e nel caso venga omissso il formato dei dati restituiti rimane comunque JSON.

API INFORMAZIONI HW

Endpoint	Richiesta HTTP
/api/v2/info.json	GET

API CONFIGURAZIONE DEVICES

Endpoint	Richiesta HTTP
/api/v2/devices.json	GET
/api/v2/devices/{deviceId}/config.json	GET

API CONFIGURAZIONE VARIABILI

Endpoint	Richiesta HTTP
/api/v2/devices/{deviceId}/variables/config.json	GET
/api/v2/devices/{deviceId}/variables/{variableId}/config.json	GET
/api/v2/devices/custom/variables/config.json	GET
/api/v2/devices/custom/variables/{variableId}/config.json	GET

API DATI

Endpoint	Richiesta HTTP
----------	----------------

/api/v2/devices/{deviceId}/variables/data.json	GET
/api/v2/devices/{deviceId}/variables/{variableId}/data.json	GET
/api/v2/devices/{deviceId}/variables/{variableId}/logdata.json	GET
/api/v2/devices/{deviceId}/variables/{variableId}/set.json	GET
/api/v2/devices/custom/variables/data.json	GET
/api/v2/devices/custom/variables/{variableId}/data.json	GET
/api/v2/devices/custom/variables/{variableId}/logdata.json	GET

API PART PROGRAMS

Endpoint	Richiesta HTTP
/api/v2/devices/{deviceId}/documents/list	GET
/api/v2/devices/{deviceId}/documents/upload	POST
/api/v2/devices/{deviceId}/documents/download	GET
/api/v2/devices/{deviceId}/documents/delete	DELETE
/api/v2/devices/{deviceId}/documents/createSubfolder	POST
/api/v2/devices/{deviceId}/documents/deleteSubfolder	DELETE

API ALLARMI

Endpoint	Richiesta HTTP
/api/v2/alarms	GET
/api/v2/alarms/{alarmId}/config.json	GET
/api/v2/alarms/{alarmId}/data.json	GET
/api/v2/alarms/active.json	GET
/api/v2/alarms/history.json	GET

API EVENTI

Endpoint	Richiesta HTTP
/api/v2/events/info.json	GET
/api/v2/events/{eventId}/info.json	GET
/api/v2/events/{eventId}/history.json	GET

HYPERMEDIA LINK

Alcune risposte contengono degli hypermedia link per permettere il recupero di altre informazioni associate alla richiesta appena effettuata, di seguito la struttura di ogni hypermedia link:

- *rel* – tipo di relazione, come definito nella RFC 5988, par. 4 “Link relation types”.
- *href* – indirizzo al quale accedere per recuperare l’informazione associata al link.
- *type* (opzionale) – MIME type del contenuto.
- *title* (opzionale) – nome descrittivo del contenuto del link.

3 API informazioni di sistema

3.1 GET /api/v2/info.json

Ritorna informazioni di sistema tramite un oggetto JSON avente i campi:

- *sn* – numero seriale univoco del dispositivo
- *uuid* – identificatore alfanumerico univoco del dispositivo
- *producer* – produttore
- *name* – nome che l'utente ha dato all'impianto
- *localDate* – data corrente locale del dispositivo
- *webAppVersion* – versione del software del dispositivo

3.1.1 Parametri URL

nessuno

3.1.2 Richiesta di esempio

URL di esempio `http://192.168.1.29/api/v2/info.json`

Body *vuoto*

3.1.3 Risposta di esempio

Status code 200 OK

Body {
 "producer": "Alleantia",
 "sn": "IOTSPI215060600",
 "uuid": "414da140b-1ead-32f8-ae78-adf800fc96",
 "name": "Monitoraggio impianto",
 "localDate": "2015-11-12T17:56:34.831+0100",
 "webAppVersion": "4.1.7"
}

4 API configurazione dispositivi

4.1 GET /api/v2/devices.json

Ritorna una lista con i dati necessari per recuperare le informazioni sui dispositivi configurati nell'IoT Server. La lista contiene 0 o più oggetti JSON aventi i campi:

- *id* – identificatore numerico del dispositivo. E' univoco all'interno di ogni IoT Server ed è valido solo per l'IoT Server che lo ha restituito. Nel caso in cui l'utente successivamente modifichi la configurazione dell'IoT Server disabilitando o eliminando il dispositivo in questione, le chiamate a questa API non forniranno più il suo id.
- *links* – lista di hypermedia link per il recupero delle informazioni di configurazione dispositivi e variabili configurati nell'IoT Server, i link contengono già i parametri *deviceId* necessari alle chiamate successive. Il parametro "rel" identifica il tipo di informazione ritornata dal link:
 - *config-device* - configurazione dispositivo
 - *config-variables* - configurazione variabili del dispositivo

Nel caso in cui non sia configurato nessun dispositivo viene ritornata una lista vuota.

4.1.1 Parametri URL

nessuno

4.1.2 Richiesta di esempio

URL di esempio `http://192.168.1.29/api/v2/devices.json`

Body *vuoto*

4.1.3 Risposta di esempio

Status code 200 Ok
204 No Content – Niente da restituire

Body [

```

{
  "id": 1,
  "links": [
    {
      "rel": "config-device",
      "href": "http://192.168.1.29/api/v2/devices/1/config",
      "title": "Device configuration"
    },
    {
      "rel": "config-variables",
      "href": "http://192.168.1.29/api/v2/devices/1/variables/config",

```

```
    "title": "Device variables configuration"
  }
]
},
{
  "id": 7,
  "links": [
    {
      "rel": "config-device",
      "href": "http://192.168.1.29/api/v2/devices/7/config",
      "title": "Device configuration"
    },
    {
      "rel": "config-variables",
      "href": "http://192.168.1.29/api/v2/devices/7/variables/config",
      "title": "Device variables configuration"
    }
  ]
}
]
```

4.2 GET /api/v2/devices/{deviceId}/config.json

Ritorna le informazioni di configurazione di un dispositivo configurato nell'IoT Server, il dato ritornato è un oggetto JSON contenente i campi:

- *id* – identificatore numerico del dispositivo. Si vedano le considerazioni fatte per la richiesta precedente.
- *manufacturer* – nome del produttore del dispositivo.
- *model* – modello del dispositivo.
- *version* (opzionale) – versione del dispositivo.
- *description* – descrizione del dispositivo assegnata dall'utente dell'IoT Server.
- *category* – categoria del dispositivo.
- *links* (opzionale) – lista di hypermedia link per il recupero dei documenti associati al dispositivo.

Nel caso il dispositivo richiesto non esista viene ritornato lo status code "204 - No content" e un body vuoto.

4.2.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
-----------------	-----------------------------------------

4.2.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/1/config.json</code>
-----------------------	---------------------------------------------------------------

Body	<i>vuoto</i>
-------------	--------------

4.2.3 Risposta di esempio

Status code	200 Ok
	404 Not Found – Device inesistente

Body	<pre>{ "id": 1, "manufacturer": "Fronius", "model": "IG Plus 120 V-3", "description": "INVERTER 1", "category": "Inverter", "links": [{ "rel": "edit-media", "href": "http://192.168.1.29/documents/34.pdf", "type": "application/pdf", "title": "Service manual" }] }</pre>
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5 API configurazione variabili

Permettono di ottenere la configurazione delle variabili di un device (specificando un deviceId) o delle variabili di impianto.

Per tutte queste API il tipo base ritornato è un oggetto JSON (o una lista di oggetti JSON) contenente i campi:

- *id* – identificatore numerico della variabile del dispositivo o dell'impianto. Per le variabili di dispositivo è univoco all'interno del deviceId a cui si riferisce ed è valido solo per l'IoT Server che lo ha ritornato. Per le variabili di impianto è univoco per l'IoT Server che lo ha ritornato. Nel caso in cui l'utente successivamente modifichi la configurazione dell'IoT Server disabilitando la variabile in questione, le chiamate a questa API non forniranno più il suo id.
- *description* – descrizione della variabile assegnata dal produttore o dall'utente dell'IoT Server.
- *dataType* – tipizzazione del valore della variabile, può assumere i valori "String", "Boolean" e "Numeric".
- *engUnit* (opzionale) – unità di misura della variabile.
- *minimum* (opzionale) – valore minimo che la variabile può assumere, viene assegnato dal produttore o dall'utente dell'IoT Server.
- *maximum* (opzionale) – valore massimo che la variabile può assumere, viene assegnato dal produttore o dall'utente dell'IoT Server.
- *link* – hypermedia link per il recupero del valore attuale della variabile. Il parametro "rel" ha valore "data".
- *alarmable* – booleano indicante l'esistenza di un allarme associato a tale variabile
- *writable* – booleano indicante se tale variabile è scrivibile o meno

5.1 GET /api/v2/devices/{deviceId}/variables/config.json

Ritorna una lista con le informazioni sulla configurazione delle variabili di un dispositivo configurato nell'IoT Server.

5.1.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
-----------------	-----------------------------------------

5.1.2 Parametri di query

id (opzionale)	Id variabile da utilizzare per filtrare il risultato
-----------------------	------------------------------------------------------

5.1.3 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/1/variables/config.json</code>
-----------------------	-------------------------------------------------------------------------

URL di esempio	<code>http://192.168.1.29/api/v2/devices/1/variables/config.json?id=1&id=2</code>
-----------------------	---------------------------------------------------------------------------------------

Body	<i>vuoto</i>
-------------	--------------

5.1.4 Risposta di esempio

Status code	200 Ok
	204 No Content – Niente da restituire
	404 Not Found – Device inesistente

Body	<pre>[{ "id": 1, "description": "Power - NOW", "dataType": "Numeric", "engUnit": "kW", "minimum": 0, "maximum": 10, "link": { "rel": "data", "href": "http://192.168.1.29/api/v2/devices/1/variables/1/data" }, "alarmable": false, "writable": false }, { "id": 2, "description": "Energy - TOTAL", "dataType": "Numeric", "engUnit": "kWh", "link": { "rel": "data", "href": "http://192.168.1.29/api/v2/devices/1/variables/2/data" }, "alarmable": false, "writable": false }]</pre>
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.2 GET /api/v2/devices/{deviceId}/variables/{variableId}/config.json

Ritorna le informazioni sulla configurazione di una variabile di un dispositivo configurato nell'IoT Server.

Nel caso il dispositivo o la variabile richiesta non esistano viene ritornato lo status code "204 - No content" e un body vuoto.

5.2.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
variableId	Identificativo numerico della variabile del dispositivo

5.2.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/1/variables/9/config.json</code>
Body	<i>vuoto</i>

5.2.3 Risposta di esempio

Status code	200 Ok 204 No Content – Niente da restituire 404 Not Found – Device inesistente
Body	<pre>{ "id": 9, "description": "DC voltage - NOW", "dataType": "Numeric", "engUnit": "V", "minimum": 230, "maximum": 600, "link": { "rel": "data", "href": "http://192.168.1.29/api/v2/devices/1/variables/9/data" }, "alarmable": false, "writable": false }</pre>

5.3 GET /api/v2/devices/custom/variables/config.json

Ritorna una lista con le informazioni sulla configurazione delle variabili di impianto dell'IoT Server.

5.3.1 Parametri URL

nessuno

5.3.2 Parametri di query

id (opzionale) Id variabile da utilizzare per filtrare il risultato

5.3.3 Richieste di esempio

URL di esempio <http://192.168.1.29/api/v2/devices/custom/variables/config.json>

URL di esempio <http://192.168.1.29/api/v2/devices/custom/variables/config.json?id=5&id=500>

Body *vuoto*

5.3.4 Risposta di esempio

Status code 200 Ok

204 No Content – Niente da restituire

Body [

```
{
  "id": 5,
  "description": "Potenza istantanea",
  "dataType": "Numeric",
  "engUnit": "kW",
  "link": {
    "rel": "data",
    "href":
"http://192.168.1.29/api/v2/devices/custom/variables/5/data"
  },
  "alarmable": false,
  "writable": false
},
{
  "id": 500,
  "description": "Integrale Energia",
  "dataType": "Numeric",
  "engUnit": "kWh",
  "link": {
    "rel": "data",
    "href":
"http://192.168.1.29/api/v2/devices/custom/variables/50000/data"
  },
  "alarmable": false,
  "writable": false
}
]
```

5.4 GET /api/v2/devices/custom/variables/{variableId}/config.json

Ritorna le informazioni sulla configurazione di una variabile di impianto dell'IoT Server.

Nel caso la variabile richiesta non esista viene ritornato lo status code "204 - No content" e un body vuoto.

5.4.1 Parametri URL

variableId	Identificativo numerico della variabile dell'impianto
-------------------	-------------------------------------------------------

5.4.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/devices/custom/variables/500/config.json
-----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

Body	<i>vuoto</i>
-------------	--------------

5.4.3 Risposta di esempio

Status code	200 Ok
--------------------	--------

	404 Not Found – Variabile inesistente
--	---------------------------------------

Body	<pre>{ "id": 500, "description": "Integrale Energia", "dataType": "Numeric", "engUnit": "kWh", "link": { "rel": "data", "href": "http://192.168.1.29/api/v2/devices/custom/variables/50000/data" }, "alarmable": false, "writable": false }</pre>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6 API dati

Per tutte queste API il tipo base di ritorno è un oggetto JSON contenente i campi:

- *id* – identificatore numerico della variabile del dispositivo o di impianto.
- *value* - valore della variabile, che può essere di tipo numerico, booleano oppure stringa.
- *decodedValue* (opzionale) – nel caso in cui i valori assumibili da una variabile siano dei codici per cui è prevista una decodifica volta a migliorarne la “leggibilità” all’utente finale, ad esempio 0=OK, 1=FAULT, questo campo contiene la decodifica del valore presente nel campo *value*.
- *timestamp* – data e ora in formato ISO-8601 alla quale il valore è stato rilevato.
- *quality* – valore booleano che indica se l’ultimo tentativo di lettura della variabile è andato a buon fine, quindi se il campo *value* contiene un valore attuale oppure vecchio e potenzialmente non corrispondente alla realtà. Se l’ultima comunicazione è stata OK il campo *quality*=true e *value* contiene il valore letto, se invece l’ultima comunicazione è stata KO *quality*=false e *value* contiene l’ultimo valore letto dal dispositivo.

In caso di risposta priva di errori il body contiene i dati JSON e lo status code è 200 OK, mentre se la variabile oggetto della richiesta non esiste nella configurazione dell’IoT Server possono verificarsi 2 possibili risposte dipendentemente dal tipo di ritorno dell’endpoint:

- *Endpoint con tipo di ritorno dato singolo* - viene ritornato lo status code “204 - No content” e un body vuoto
- *Endpoint con tipo di ritorno lista* – viene ritornata lo status code 200 OK ed una lista vuota

N.B. Nel caso in cui per la variabile richiesta non sia ancora disponibile un valore in quanto la comunicazione fra IoT Server ed il dispositivo non è ancora avvenuta, il campo *timestamp* ha valore null ed il campo *value* ha un valore di default che l’applicazione richiedente DEVE ignorare in quanto non realmente recuperato dal dispositivo.

6.1 GET /api/v2/devices/{deviceId}/variables/data.json

Ritorna una lista con i valori attuali delle variabili di un dispositivo configurato nell'IoT Server.

6.1.1 Parametri URL

deviceId Identificativo numerico del dispositivo

6.1.2 Parametri di query

id (opzionale) Id variabile da utilizzare per filtrare il risultato

6.1.3 Richiesta di esempio

URL di esempio <http://192.168.1.29/api/v2/devices/5/variables/data.json>

URL di esempio <http://192.168.1.29/api/v2/devices/5/variables/data.json?id=24&id=27>

Body *vuoto*

6.1.4 Risposta di esempio

Status code 200 Ok

 204 No Content – Niente da restituire

 404 Not Found – Device inesistente

Body [

```

{
  "id": 24,
  "value": true,
  "timestamp": "2015-02-13T16:17:42.831+0100",
  "quality": true
},
{
  "id": 27,
  "value": 12.76,
  "timestamp": "2015-02-13T16:17:41.299+0100",
  "quality": true
}
]

```

6.2 GET /api/v2/devices/{deviceId}/variables/{variableId}/data.json

Ritorna il valore attuale di una variabile di un dispositivo configurato nell'IoT Server.

6.2.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
variableId	Identificativo numerico della variabile del dispositivo

6.2.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/devices/5/variables/3/data.json
Body	<i>vuoto</i>

6.2.3 Risposta di esempio

Status code	200 OK
Body	<pre>{ "id": 3, "value": 2561, "timestamp": "2015-02-11T15:31:02.404+0100", "decodedValue": "FAN FAULT", "quality": false }</pre>

6.3 GET /api/v2/devices/{deviceId}/variables/{variableId}/set.json

Imposta il valore di una variabile scrivibile di un dispositivo nell'IoT Server. La risposta ritorna se la richiesta è stata accettata (anche se non ancora eseguita!).

6.3.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
variableId	Identificativo numerico della variabile del dispositivo
value	Nuovo valore della variabile.

6.3.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/11/variables/7/set?value=-54.34</code>
Body	<i>vuoto</i>

6.3.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>204 No Content – Niente da restituire</p> <p>400 Bad Request – Dati obbligatori mancanti</p> <p>401 Unauthorized – Licenza non abilitata all'operazione</p> <p>403 Forbidden – Variabile non abilitata/non scrivibile</p> <p>404 Not Found – Device inesistente</p>
Body	<pre>{ "accepted": true, "description": "Accepted" }</pre>

6.4 GET /api/v2/devices/{deviceId}/variables/{variableId}/logdata.json

Ritorna i valori storicizzati di una variabile di un dispositivo configurato nell'IoT Server in un intervallo temporale specificato. Ritorna un body vuoto se non ci sono dati nell'intervallo temporale specificato. Il numero massimo di valori ritornati è limitato ai primi 1000.

6.4.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
variableId	Identificativo numerico della variabile del dispositivo
startTime	Estremo iniziale dell'intervallo temporale dei dati da ritornare (numero intero, millisecondi a partire dal 1.1.1970 00:00:00.000)
endTime (opzionale)	Estremo finale dell'intervallo temporale dei dati da ritornare (numero intero, millisecondi a partire dal 1.1.1970 00:00:00.000). Se non specificato l'intervallo è non limitato superiormente.

6.4.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/3/variables/3/logdata?startTime=0</code>
Body	<i>vuoto</i>

6.4.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>204 No Content – Niente da restituire</p> <p>400 Bad Request – Dati obbligatori mancanti</p> <p>404 Not Found – Device inesistente</p>
Body	<pre>[{ "value": 1, "timestamp": "2015-11-12T16:14:01.713+0100", "quality": true }, { "value": 3, "timestamp": "2015-11-12T16:15:07.162+0100", "quality": true }, ...]</pre>

6.5 GET /api/v2/devices/custom/variables/data.json

Ritorna una lista con i valori attuali delle variabili di impianto nell'IoT Server.

6.5.1 Parametri URL

nessuno

6.5.2 Parametri di query

id (opzionale) Id variabile da utilizzare per filtrare il risultato

6.5.3 Richiesta di esempio

URL di esempio <http://192.168.1.29/api/v2/devices/custom/variables/data.json>

URL di esempio <http://192.168.1.29/api/v2/devices/custom/variables/data.json?id=5&id=500>

Body *vuoto*

6.5.4 Risposta di esempio

Status code 200 Ok

204 No Content – Niente da restituire

Body [

```
{
  "id": 5,
  "value": 0,
  "timestamp": "2015-11-16T10:29:35.182+0100",
  "quality": false
},
{
  "id": 500,
  "value": 2.2757686111111113,
  "timestamp": "2015-11-16T10:29:55.959+0100",
  "quality": false
}
]
```

6.6 GET /api/v2/devices/custom/variables/{variableId}/data.json

Ritorna il valore attuale di una variabile di impianto nell'IoT Server.

6.6.1 Parametri URL

variableId	Identificativo numerico della variabile di impianto
-------------------	-----------------------------------------------------

6.6.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/devices/custom/variables/50000/data.json
-----------------------	---------------------------------------------------------------------

Body	<i>vuoto</i>
-------------	--------------

6.6.3 Risposta di esempio

Status code	200 Ok
	404 Not Found – Variabile inesistente

Body	<pre>{ "id": 50000, "value": 2.2757686111111113, "timestamp": "2015-11-16T10:31:47.288+0100", "quality": false }</pre>
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------

6.7 GET /api/v2/devices/custom/variables/{variableId}/logdata.json

Ritorna i valori storicizzati di una variabile di impianto nell'IoT Server in un intervallo temporale specificato. Ritorna un body vuoto se non ci sono dati nell'intervallo temporale specificato. Il numero massimo di valori ritornati è limitato ai primi 1000.

6.7.1 Parametri URL

variableId	Identificativo numerico della variabile di impianto
startTime	Estremo iniziale dell'intervallo temporale dei dati da ritornare (numero intero, millisecondi a partire dal 1.1.1970 00:00:00.000)
endTime (opzionale)	Estremo finale dell'intervallo temporale dei dati da ritornare (numero intero, millisecondi a partire dal 1.1.1970 00:00:00.000). Se non specificato l'intervallo è non limitato superiormente.

6.7.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/custom/variables/500/logdata?startTime=0</code>
Body	<i>vuoto</i>

6.7.3 Risposta di esempio

Status code	200 Ok 400 Bad Request – Dati obbligatori mancanti 404 Not Found – Variabile inesistente
Body	<pre>[{ "value": 0.056256666666666667, "timestamp": "2015-11-12T16:14:02.272+0100", "quality": true }, { "value": 0.104333055555555555, "timestamp": "2015-11-12T16:15:07.367+0100", "quality": true }, ...]</pre>

7 Gestione file e cartelle dei dispositivi collegati

7.1 GET /api/v2/devices/documents/list

Restituisce per ogni device collegato all'IOT, l'elenco dei file e delle directory (aliases e path reali) su cui è possibile effettuare operazioni tramite le api documents.

Per convenzione, i paths utilizzano come carattere separatore "/". Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere "/".

7.1.1 Parametri URL

nessuno

7.1.2 Richiesta di esempio

URL di esempio http://192.168.1.29/api/v2/devices/documents/list

Body Vuoto

7.1.3 Risposta di esempio

Status code 200 Ok

204 No content – se non ci sono cartelle su cui operare

```

Body [
  {
    "devId": 179,
    "description": "Fanuc32i",
    "list": [
      {
        "name": "PATH1",
        "path": "//CNC_MEM/USER/PATH1/",
        "type": "FOLDER",
        "read": true,
        "write": true,
        "create": true,
        "delete": true
      }
    ]
  },
  {
    "devId": 180,
    "description": "Heidenhain640",
    "list": [
      {
        "name": "ROOT",
        "path": "TNC:\\",
        "type": "FOLDER",
        "read": true,
        "write": true,
        "create": true,
        "delete": true
      }
    ]
  }
]

```

7.2 GET /api/v2/devices/{deviceId}/documents/list

Restituisce l'elenco dei file e delle directory contenuti nella folder indicata, se i permessi lo permettono.

Per convenzione, i paths utilizzano come carattere separatore "/". Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere "/".

N.B: Il parametro "path" è facoltativo. Se omissso, l'api restituirà la lista di tutte le folder abilitate per la lettura.

7.2.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
path	Percorso della cartella di cui leggere il contenuto.

7.2.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/devices/1/documents/list?path=ROOT/nc_prog/
Body	Vuoto

7.2.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>204 No content – se non ci sono cartelle su cui operare</p> <p>400 Bad Request – Path mancante o errato</p> <p>403 Forbidden – Permessi insufficienti per l'operazione</p> <p>404 Not Found – Device non trovato</p> <p>406 Not acceptable – Device non abilitato allo scambio di part programs</p> <p>500 Internal Server Error – Errore generico</p>
Body	<pre>[{ "name": "nc_prog", "type": "folder", "size": 0, "lastModified": "2018-03-29T12:19:43.000+0200", "links": [{ "title": "Folder List", "rel": "documents", "href": "http://localhost:8080/api/v2/devices/162/documents/list?path=ROOT/nc_prog/" }, { "title": "File Upload", "rel": "documents", "href": "http://localhost:8080/api/v2/devices/162/documents/upload?destinationFolder=ROO</pre>

```
T/nc_prog/{FILENAME}"
    },
    {
        "title": "Folder Create",
        "rel": "documents",
        "href":
        "http://localhost:8080/api/v2/devices/162/documents/createSubfolder?folderPath=R
OOT/nc_prog/{FOLDERNAME}"
    },
    {
        "title": "Folder Delete",
        "rel": "documents",
        "href":
        "http://localhost:8080/api/v2/devices/162/documents/deleteSubfolder?folderPath=R
OOT/nc_prog/"
    }
    ]
    }, ...
]
```

7.3 POST /api/v2/devices/{deviceId}/documents/upload

Effettua l'upload di un file nella folder indicata.

Il Content-Type utilizzato dal servizio è multipart/form-data.

Per convenzione, i paths utilizzano come carattere separatore "/". Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere "/".

Specificare nel body della richiesta una entry con key=uploadFile e valore il file da inoltrare

7.3.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
destinationFolder	Percorso della cartella in cui caricare il file

7.3.2 Richiesta di esempio

URL di esempio	<pre>http://192.168.1.29/api/v2/devices/11/documents/upload? destinationFolder=ROOT/nc_prog/</pre>
Body	<i>Key=uploadFile – Value(File)=File da uploadare</i>

7.3.3 Risposta di esempio

Status code	200 Ok 400 Bad Request – Path mancante o errato 403 Forbidden – Permessi insufficienti per l'operazione 404 Not Found – Device non trovato 406 Not acceptable – Device non abilitato allo scambio di part programs 500 Internal Server Error – Errore generico
--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Body	<pre>{ "accepted": true, "description": "File uploaded successfully" }</pre>
-------------	----------------------------------------------------------------------------------

7.4 GET /api/v2/devices/{deviceId}/documents/download

Effettua il download del file indicato.

Per convenzione, i paths utilizzano come carattere separatore “/”. Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere “/”.

7.4.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
filePath	Percorso del file da scaricare

7.4.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/devices/11/documents/download?filePath=ROOT/nc_prog/file.h
Body	<i>Vuoto</i>

7.4.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>400 Bad Request – Path mancante o errato</p> <p>403 Forbidden – Permessi insufficienti per l’operazione</p> <p>404 Not Found – Device non trovato</p> <p>406 Not acceptable – Device non abilitato allo scambio di part programs</p> <p>500 Internal Server Error – Errore generico</p>
Body	<pre>{ "accepted": true, "description": "File downloaded successfully" }</pre>

7.5 DELETE /api/v2/devices/{deviceId}/documents/delete

Effettua la cancellazione del file indicato.

Per convenzione, i paths utilizzano come carattere separatore “/”. Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere “/”.

7.5.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
filePath	Percorso del file da eliminare

7.5.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/devices/11/documents/delete?filePath=ROOT/nc_prog/file.h
Body	<i>Vuoto</i>

7.5.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>400 Bad Request – Path mancante o errato</p> <p>403 Forbidden – Permessi insufficienti per l’operazione</p> <p>404 Not Found – Device non trovato</p> <p>406 Not acceptable – Device non abilitato allo scambio di part programs</p> <p>500 Internal Server Error – Errore generico</p>
Body	<pre>{ "accepted": true, "description": "File deleted successfully" }</pre>

7.6 POST /api/v2/devices/{deviceId}/documents/createSubfolder

Effettua la creazione di una sottocartella nel path indicato.

Per convenzione, i paths utilizzano come carattere separatore “/”. Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere “/”.

7.6.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
folderPath	Percorso della folder in cui creare la nuova cartella
subfolderName	Nome della cartella da creare

7.6.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/11/documents/createSubfolder?folderPath=ROOT/nc_prog/&subfolderName={FOLDERNAME}</code>
Body	<i>Vuoto</i>

7.6.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>400 Bad Request – Path mancante o errato</p> <p>403 Forbidden – Permessi insufficienti per l’operazione</p> <p>404 Not Found – Device non trovato</p> <p>406 Not acceptable – Device non abilitato allo scambio di part programs</p> <p>500 Internal Server Error – Errore generico</p>
Body	<pre>{ "accepted": true, "description": "Folder created successfully" }</pre>

7.7 DELETE /api/v2/devices/{deviceId}/documents/deleteSubfolder

Effettua la cancellazione della cartella al path indicato.

Per convenzione, i paths utilizzano come carattere separatore “/”. Nel caso in cui un path faccia riferimento ad una directory, questo dovrà terminare con il carattere “/”.

7.7.1 Parametri URL

deviceId	Identificativo numerico del dispositivo
folderPath	Percorso della folder da eliminare

7.7.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/devices/11/documents/createSubfolder?folderPath=ROOT/nc_prog/</code>
Body	<i>Vuoto</i>

7.7.3 Risposta di esempio

Status code	<p>200 Ok</p> <p>400 Bad Request – Path mancante o errato</p> <p>403 Forbidden – Permessi insufficienti per l’operazione</p> <p>404 Not Found – Device non trovato</p> <p>406 Not acceptable – Device non abilitato allo scambio di part programs</p> <p>500 Internal Server Error – Errore generico</p>
Body	<pre>{ "accepted": true, "description": "Folder deleted successfully" }</pre>

8 API configurazione allarmi

8.1 GET /api/v2/alarms.json

Ritorna una lista di tutti gli allarmi attivi nell'IoT Server.

Nel caso non ci siano allarmi attivi ritorna una lista vuota.

Il tipo base ritornato è una lista di oggetti JSON contenenti i campi:

- *id* – identificatore numerico dell'allarme.
- *links* – hypermedia link per il recupero della configurazione del singolo allarme. Il parametro "rel" ha valore "config-alarm".

8.1.1 Richiesta di esempio

URL di esempio `http://192.168.1.29/api/v2/alarms.json`

Body *vuoto*

8.1.2 Risposta di esempio

Status code 200 Ok

204 No Content – Niente da restituire

Body [

```
{
  "id": 5,
  "links": [
    {
      "rel": "config-alarm",
      "href": "http://192.168.1.29/api/v2/alarms/5/config",
      "title": "Alarm configuration"
    }
  ]
},
{
  "id": 1,
  "links": [
    {
      "rel": "config-alarm",
      "href": "http://192.168.1.29/api/v2/alarms/1/config",
      "title": "Alarm configuration"
    }
  ]
}
]
```

8.2 GET /api/v2/alarms/{alarmId}/config.json

Ritorna le informazioni sulla configurazione di un allarme configurato nell'IoT Server.

Nel caso in cui l'allarme richiesto non esista viene ritornato lo status code "204 - No content" e un body vuoto.

Il tipo base ritornato è un oggetto JSON contenenti i campi:

- *id* – identificatore numerico dell'allarme.
- *name* – nome dell'allarme
- *description* – descrizione dell'allarme
- *link* – hypermedia link per il recupero del valore attuale della variabile. Il parametro "rel" ha valore "data".
- *snapshotGlobalIds(opzionale)* – lista di globalId a cui fa riferimento l'allarme (no allarmi personalizzati)

8.2.1 Parametri URL

alarmId	Identificativo numerico dell'allarme
----------------	--------------------------------------

8.2.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/alarms/4/config.json</code>
-----------------------	--------------------------------------------------------------

Body	<i>vuoto</i>
-------------	--------------

8.2.3 Risposta di esempio

Status code	200 Ok
	204 No Content – Niente da restituire
	404 Not found – Allarme non computabile

Body	<pre>{ "id": 4, "name": "Allarme Sensore 3", "description": "Input channel 3 sensor error", "link": { "rel": "data", "href": "http://192.168.1.29/api/v2/alarms/4/data", "title": "Alarm value" } }</pre>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8.3 GET /api/v2/alarms/active.json

Ritorna l'elenco degli allarmi attivi nell'IoT Server. Se non ci sono allarmi attivi viene ritornata una lista vuota.

Il tipo base ritornato è una lista di oggetti JSON contenenti i campi:

- *id* – identificatore numerico dell'allarme.
- *eventId* (opzionale)– identificatore dell'evento (numero intero, identifica una coppia di istanti di attivazione e di relativa disattivazione di un allarme)
- *deviceName* (opzionale) – dispositivo al quale è associato un allarme
- *deviceSection* (opzionale) – sezione del dispositivo al quale è associato un allarme
- *measure* (opzionale) – misura alla quale è associato un allarme
- *description* – descrizione dell'allarme
- *onDate* – istante di attivazione dell'allarme
- *quality* – valore booleano che indica se l'ultimo tentativo di lettura è andato a buon fine
- *alarmed* – stato dell'allarme

8.3.1 Parametri di Query

alarmId	Id variabile da <i>ness</i>
(opzionale)	utilizzare per filtrare il risultato

8.3.2 Richiesta di esempio

URL di esempio	<code>http://192.168.1.29/api/v2/alarms/active?alarmId=5&alarmId=7&alarmId=9</code>
Body	<i>vuoto</i>

8.3.3 Risposta di esempio

Status code	200 Ok
	204 No Content – Niente da restituire

Body [

```
{
  "id": 5,
  "eventId": 166050,
  "deviceName": "Curr",
  "measure": "Curr 1",
  "description": "Test Curr",
  "onDate": "2015-11-16T11:01:42.883+0100",
  "quality": false,
  "alarmed": false
},
{
  "id": 7,
```

```
"eventId": 166051,  
"measure": "Potenza istantanea",  
"description": "Test VV",  
"onDate": "2015-11-16T11:01:57.462+0100",  
"quality": false,  
"alarmed": false  
},  
{  
  "id": 9,  
  "eventId": 166049,  
  "description": "Test multi",  
  "onDate": "2015-11-16T11:01:42.769+0100",  
  "quality": false,  
  "alarmed": false  
},  
]
```

8.4 GET /api/v2/alarms/history.json

Ritorna l'elenco degli allarmi nello storico dell'IoT Server ordinati per data crescente. Se non ci sono allarmi viene ritornata una lista vuota. Ogni elemento della lista è identificato da un eventId e raggruppa in sé un istante di attivazione ed un eventuale istante di disattivazione dell'allarme. È possibile specificare un intervallo di eventId per selezionare solo gli allarmi compresi al suo interno. Il numero di allarmi ritornati è limitato ai primi 1000 risultati della query.

Il tipo base ritornato è una lista di oggetti JSON contenenti i campi:

- *alarmType* – Tipo di allarme, può essere "DEVICE_ALARM" per gli allarmi associati ad un dispositivo, "SYS_ALARM" per gli allarmi di sistema e "ALARM" per gli altri allarmi.
- *alarmId* (opzionale) – Identificatore dell'allarme (solo se è un allarme "ALARM")
- *eventId* – identificatore dell'evento (numero intero, identifica una coppia di istanti di attivazione e di relativa disattivazione di un allarme)
- *onDate* – istante di attivazione dell'allarme
- *offDate* (opzionale) – istante di disattivazione dell'allarme
- *deviceName* (opzionale) – dispositivo al quale è associato un allarme
- *deviceSection* (opzionale) – sezione del dispositivo al quale è associato un allarme
- *description* – descrizione dell'allarme

8.4.1 Parametri URL

startEventId (opzionale)	Identificativo del primo evento dal quale iniziare la restituzione degli allarmi (compreso). Se non specificato comincia dal primo allarme.
endEventId (opzionale)	Identificativo dell'ultimo evento da ritornare (compreso). Se non specificato finisce con l'ultimo allarme.
includeActive (opzionale)	Se "true" ritorna anche gli allarmi correntemente attivi e non solo quelli storici (default: "false")

8.4.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/alarms/history.json
URL di esempio	http://192.168.1.29/api/v2/alarms/history?startEventId=166134&endEventId=166226&includeActive=true
Body	vuoto

8.4.3 Risposta di esempio

Status code 200 Ok

204 No Content – Niente da restituire

Body [

```
{
  "alarmType": "ALARM",
  "alarmId": 9,
  "eventId": 166134,
  "onDate": "2015-11-27T17:10:29.451+0100",
  "deviceName": "Tool",
  "deviceSection": "B1",
  "description": "Tool 3 Alarm"
},
{
  "alarmType": "DEVICE_ALARM",
  "eventId": 166135,
  "onDate": "2015-11-30T09:15:03.210+0100",
  "deviceName": "Tool",
  "description": "1"
},
{
  "alarmType": "DEVICE_ALARM",
  "eventId": 166226,
  "onDate": "2015-12-17T10:59:15.642+0100",
  "offDate": "2015-12-17T11:30:29.857+0100",
  "deviceName": "MB Write Test",
  "description": "1"
},
]
```

9 API eventi

9.1 GET /api/v2/events/info.json

Ritorna informazioni relative agli eventi generabili dal sistema tramite un oggetto JSON avente i campi:

- *eventId* – identificativo univoco dell'evento
- *eventName* – nome dell'evento
- *condition* – condizione scatenante l'evento
- *delayOn* – ritardo da applicare prima di marcare l'evento come attivo
- *delayOff* – ritardo da applicare prima di marcare l'evento come disattivo
- *type* – tipo di evento
- *snapshotGlobalIds* – lista di id delle variabili monitorate al triggering dell'evento
- *comparisonOperator* – (opzionale) operatore per la creazione di una condizione booleana
- *numericCompareValue* – (opzionale) valore con cui effettuare il confronto booleano

9.1.1 Parametri URL

nessuno

9.1.2 Richiesta di esempio

URL di esempio `http://192.168.1.29/api/v2/events/info.json`

Body *vuoto*

9.1.3 Risposta di esempio

Status code 200 Ok

204 No Content – Niente da restituire

Body `[`

```
{
  "eventId": 2,
  "eventName": "Evento2",
  "condition": "$G_21_47",
  "delayOn": 0,
  "delayOff": 0,
  "type": "onChange",
  "snapshotGlobalIds": "G_21_803,G_21_20",
  "comparisonOperator": "eq",
  "numericCompareValue": 0
},
{
  "eventId": 1,
  "eventName": "Evento1",
  "condition": "($G_21_47) eq true",
```

```
"delayOn": 0,  
"delayOff": 0,  
"type": "boolean",  
"snapshotGlobalIds": ["G_21_803","G_21_20"],  
"comparisonOperator": "eq",  
"numericCompareValue": 0  
}  
]
```

9.2 GET /api/v2/events/{eventId}/info.json

Ritorna informazioni relative al singolo evento generabili dal sistema tramite un oggetto JSON avente i campi:

- *eventId* – identificativo univoco dell'evento
- *eventName* – nome dell'evento
- *condition* – condizione scatenante l'evento
- *delayOn* – ritardo da applicare prima di marcare l'evento come attivo
- *delayOff* – ritardo da applicare prima di marcare l'evento come disattivo
- *type* – tipo di evento
- *snapshotGlobalIds* – lista di id delle variabili monitorate al triggering dell'evento
- *comparisonOperator* – operatore per la creazione di una condizione booleana /*utilizzabile in futuro
- *numericCompareValue* – valore con cui effettuare il confronto booleano /*utilizzabile in futuro

9.2.1 Parametri URL

<u>eventId</u>	Identificativo numerico dell'evento
-----------------------	-------------------------------------

9.2.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/info.json
-----------------------	--------------------------------------

Body	<i>vuoto</i>
-------------	--------------

9.2.3 Risposta di esempio

Status code	200 Ok
	204 No Content – Niente da restituire
	404 Not Found – Evento non trovato

Body	<pre>{ "eventId": 1, "eventName": " Evento1", "condition": "\$G_21_47", "delayOn": 0, "delayOff": 0, "type": "onChange", "snapshotGlobalIds": ["G_21_803","G_21_20"], "comparisonOperator": "eq", "numericCompareValue": 0 }</pre>
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9.3 GET /api/v2/events/{eventId}/history.json

Ritorna tutti gli eventi storicizzati nell'IoT Server in un intervallo temporale specificato. Ritorna un body vuoto se non ci sono dati nell'intervallo temporale specificato. Il numero massimo di valori ritornati è limitato ai primi 1000.

9.3.1 Parametri URL

eventId	Identificativo numerico dell'evento
startTime	Estremo iniziale dell'intervallo temporale dei dati da ritornare (numero intero, millisecondi a partire dal 1.1.1970 00:00:00.000)
endTime (opzionale)	Estremo finale dell'intervallo temporale dei dati da ritornare (numero intero, millisecondi a partire dal 1.1.1970 00:00:00.000). Se non specificato l'intervallo è non limitato superiormente.

9.3.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/events/1/history?startTime=0
Body	vuoto

9.3.3 Risposta di esempio

Status code	200 Ok 204 No Content – Niente da restituire 400 Bad Request – Dati obbligatori mancanti 404 Not Found – Evento non trovato
--------------------	--------------------------------------------------------------------------------------------------------------------------------------

Body []

```
[
  {
    "eventId": 3,
    "eventName": "Evento1",
    "timestamp": "2017-07-31T15:37:53.941+0200",
    "state": true,
    "variablesSnapshot": [
      {
        "devId": 21,
        "varId": 803,
        "quality": true,
        "value": true
      },
      {
        "devId": 21,
        "varId": 20,
        "quality": true,
        "value": 514000
      }
    ]
  }
]
```

```
    },  
    {  
      "eventId": 3,  
      "eventName": "Evento1",  
      "timestamp": "2017-07-31T15:38:36.301+0200",  
      "state": false,  
      "variablesSnapshot": [  
        {  
          "devId": 21,  
          "varId": 803,  
          "quality": true,  
          "value": true  
        },  
        {  
          "devId": 21,  
          "varId": 20,  
          "quality": true,  
          "value": 514000  
        }  
      ]  
    },  
    ...  
  ]  
}
```

10 Licenza

10.1 GET /api/v2/license/info.json

Ritorna il seriale del dispositivo, utile per l'attivazione della licenza e lo stato attuale di attivazione di quest'ultima. Se il software è licenziato, verrà restituito anche un resoconto delle funzionalità abilitate.

10.1.1 Parametri URL

Nessuno

10.1.2 Richiesta di esempio

URL di esempio <http://192.168.1.29/api/v2/license/info>

Body *vuoto*

10.1.3 Risposta di esempio

Status code 200 Ok

```
Body {  
  "devSn": "49ab91753-9b94-3d2e-8ff6-17d3d5113606",  
  "licensed": true,  
  "deviceCount": 15,  
  "variableCount": 5000,  
  "synopticCount": 6,  
  "logDaysCount": 90,  
  "iothubServicesCount": 2,  
}
```

```

"sqlServicesCount": 2,
"dropboxServicesCount": 2,
"onedriveServicesCount": 2,
"modbusGWEnabled": true,
"modbusWriteEnabled": true,
"restAPIEnabled": true,
"restAPIWriteEnabled": true,
"energyPackEnabled": true,
"machinePackEnabled": false,
"navyPackEnabled": true,
"yammerEnabled": true
}

```

10.2 POST /api/v2/license/activate.json

Attiva, se possibile, la licenza fornita passando come parametro il file license.lic nel corpo del messaggio. È possibile attivare la licenza una sola volta; attualmente non è permesso un upgrade di quest'ultima attraverso la chiamata alle api rest.

Specificare nel body della richiesta una entry con key=license e valore il file da inoltrare

10.2.1 Parametri URL

reboot (opzionale)	Boolean true/false per determinare se riavviare il software dopo l'attivazione della licenza (necessario per abilitare completamente le funzionalità licenziate)
------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------

10.2.2 Richiesta di esempio

URL di esempio	http://192.168.1.29/api/v2/license/activate?reboot=true
-----------------------	---------------------------------------------------------

Body	<i>Key=license – Value(File)=File da uploadare</i>
-------------	----------------------------------------------------

10.2.3 Risposta di esempio

Status code	200 Ok
	403 Forbidden – Se si tenta di riattivare/upgradare la licenza
	500 Internal Error – Se si presenta un errore generico durante l'attivazione

Body	<pre> { "accepted": true, "description": "License loaded. Rebooting..." } </pre>
-------------	--------------------------------------------------------------------------------------



Alleantia s.r.l.

www.alleantia.com

Sede legale: Via Tosco Romagnola, 136 - 56025 Pontedera (PI)

Sede operativa: Via Umberto Forti, 24/14 56121 Pisa

Partita IVA/Cod. fiscale: IT 02011550502

Tel: (+39) 050 9911933

Fax: (+39) 050 9655139

@: sales@alleantia.com

In ragione dell'evoluzione delle Norme, dei materiali e delle tecnologie, le caratteristiche riportate nei testi e nelle illustrazioni del presente documento si potranno ritenere impegnative solo dopo conferma da parte di Alleantia s.r.l.

M-API-0518-IT