

Do-Dots

Protocollo di comunicazione

Ultimo aggiornamento - 10 maggio 2011

rev3 – Spiegazioni – 10/05/2011

rev2 – Primo aggiornamento con attuali comandi – 03/05/2011

rev1 - Stesura iniziale – 14/05/2010

**DOCUMENTO PRELIMINARE – QUESTA VERSIONE DEL
DOCUMENTO NON È COMPLETA DI TUTTE LE INFORMAZIONI
NECESSARIE A RENDERLO EFFETTIVAMENTE UTILIZZABILE**



do-Dots – Protocollo di comunicazione

Introduzione

Il protocollo definito all'interno della piattaforma **do-Dots** è un protocollo text based a livello applicativo (Application Layer) che può essere usato per trasferire tra i componenti informazioni e comandi all'interno di sessioni di comunicazione.

Le comunicazioni tipiche possono comprendere il trasferimento di informazioni (variabili di stato), l'invio di comandi ed eventi, la sottoscrizione ad un set di informazioni e la trasmissione di stream di informazioni multimediali.

Il protocollo è teoricamente indipendente dal protocollo di trasporto utilizzato. Le implementazioni principali comprendono il trasferimento di pacchetti tramite socket UDP/TCP oppure su linea seriale in collegamento punto/punto. I pacchetti sono inoltre codificati in modo da essere facilmente trasportati attraverso chiamate HTTP.



Struttura e codifica

Ogni transazione nel protocollo **do-Dots** prevede una richiesta di un particolare 'comando' da parte di un Dot (il client) a un server (il Broker) seguita da una risposta.

Ogni pacchetto del protocollo do-Dots è codificato in formato JSON (Javascript Simple Object Notation) come da esempio:

```
{ "SET" : { "res" : "RoboCamera/AngleX", "val" : 35 } }
```

L'oggetto codificato in JSON è composto da una prima chiave ("GET") che identifica il *tipo di comando*, e da un sotto-oggetto contenente una serie di parametri (in questo caso "res" e "val") che caratterizzano il tipo di azioni che il comando andrà ad effettuare.

I parametri di un comando possono essere *obbligatori* o *opzionali*. I parametri obbligatori devono essere inseriti nella struttura del comando, pena il fallimento della richiesta con errore. I parametri opzionali possono essere inseriti per modificare il comportamento del comando o il tipo di informazioni ritornate.



Comandi del protocollo do-Dots

Ecco una lista dei comandi supportati nel protocollo **do-Dots** e il loro significato.

- GET** Richiede lo stato di una variabile o informazione al Broker. Se presente in cache, la richiesta non viene girata ad un altro Dot: il Broker restituisce immediatamente l'informazione.
- SET** Imposta il nuovo stato di una variabile. Il Broker modifica inizialmente la propria cache globale con il nuovo stato, e nello stesso momento mette in coda il comando per la successiva connessione del Dot interessato.
- CON** Comando di connessione del Dot al Broker. Oltre a scambiare informazioni utili all'avvio della comunicazione, il comando serve ad aggiungere dettagli circa la configurazione e il contenuto del Dot.
- SUB** Sottoscrizione all'aggiornamento di una o più variabili. Un Dot che si 'iscrive' ad una variabile ne riceve i valori aggiornati come risposta ad un comando SYN.
- SYN** Sincronizzazione nei valori delle variabili tra il Dot e il Broker. Il Dot invia le proprie variabili che sono state modificate, il broker risponde con i nuovi valori delle variabili a cui il Dot si è 'iscritto'.
- CLL** Call, esecuzione di un metodo di un Dot. Permette di lanciare comandi ed azioni personalizzate implementate dai singoli Dot.
- UPD** Update, comando per richiedere al Broker i messaggi diretti al Dot.



Comandi

Comando GET

Il comando GET permette di richiedere il valore di una o più variabili di sistema.

```
{ "GET" : {  
  "res" : "Weather/Temperature"  
}
```

Parametri obbligatori

res Indica la risorsa globale (variabile) che viene richiesta.

Il formato del parametro prevede di identificare una variabile indicando un testo del tipo 'Dot/Variabile' oppure 'Dot/Gruppo/Variabile' e così via.

Sono permesse wildcard come per esempio 'Box/S*' (ritorna il valore di tutte le variabili che cominciano per 'S' nel Dot 'Box').

Parametri opzionali

dot Se 'res' non identifica un Dot, è possibile specificarlo con questo parametro.

src Con questo parametro è possibile specificare esplicitamente il Dot da cui proviene la richiesta.

Risposta

```
{ "ret" : {  
  "res" : "Weather/Temperature",  
  "value" : 15 } }  
  
{ "ret" :  
  { "rst" : "err",
```



```
"err": "not found"}}
```



Comando SET

Il comando SET permette di modificare il valore di una variabile di sistema.

```
{ "SET": {  
  "res": "Weather/Temperature",  
  "value": 16  
}}
```

Parametri obbligatori

res	Indica la risorsa globale (variabile) che viene richiesta. Il formato del parametro prevede di identificare una variabile indicando un testo del tipo 'Dot/Variabile' oppure 'Dot/Gruppo/Variabile' e così via.
value	Il valore che si vuole fare assumere alla variabile.

Parametri opzionali

dot	Se 'res' non identifica un Dot, è possibile specificarlo con questo parametro.
src	Con questo parametro è possibile specificare esplicitamente il Dot da cui proviene la richiesta.

Risposta

```
{"ret": {  
  "rst": "ok" }  
  
{"ret": {  
  "rst": "err",  
  "err": "readonly" } }
```



Comando CON

Il comando CON (Connect) permette di indicare l'avvenuta connessione di un Dot alla rete, con conseguente inizializzazione di informazioni.

```
{ "CON" : {  
  "dot" : "Weather",  
  "category" : "protocol"  
}}
```

Parametri obbligatori

dot	Nome del Dot che effettua la segnalazione di connessione.
------------	---

Parametri opzionali

category	Tipo di Dot, utile ad identificarne le caratteristiche basilari (es. 'UI', 'protocol', 'remote', 'script', 'logger' ecc.)
productKey	Se il Dot appartiene a un prodotto in licenza, la chiave ProductKey può servire ad identificarlo e ad ottenere, tramite i server online, altre informazioni come loghi, setup predefiniti, aggiornamenti al firmware, ecc.

Risposta

```
{"ret" : {  
  "rst" : "ok" } }  
  
{"ret" :  
  {"rst" : "err",  
   "err" : "refused" } }
```




Comando SYN

Il comando SYN (Synchronize) permette di trasmettere al server lo stato aggiornato delle variabili del Dot.

```
{ "SYN": {  
  "src": "Weather",  
  "vars": {  
    "Temperature": 15,  
    "Humidity": 40,  
    "WindSpeed": 1.5,  
    "WindDirection": "NW",  
    "Weather": "Cloudy"  
  }  
}}
```

Parametri obbligatori

vars	Stato delle variabili da inviare al server. Per non appesantire la comunicazione, può contenere solo le variabili il cui stato è cambiato rispetto all'ultima sincronizzazione. E' formattato come oggetto contenente una serie di parametri 'variabile': 'valore'.
src	Dichiara esplicitamente il Dot da cui proviene la richiesta. Questo parametro è necessario per avere in risposta i valori delle variabili alle quali il Dot ha effettuato una subscription.

Parametri opzionali

Risposta

La risposta a questo comando può essere una delle seguenti nel caso in cui il Dot non abbia effettuato richieste di subscription.

```
{ "ret": {  
  "rst": "ok" }  
  
{ "ret": {  
  "rst": "err",
```



```
"err": "refused"}}
```

Nel caso in cui il Dot abbia effettuato richieste di subscription, la risposta al comando è un comando 'SYN' contenente le variabili di sistema alle quali il Dot si è 'abbonato'.

```
{"SYN": {  
  "vars": {"Camera/Angle": 321,  
  "SolarPanel/Active": true}}}
```



Comando SUB

Il comando SUB (Subscribe) permette di definire un set di variabili globali che il Dot intende 'seguire'. I valori, se cambiati, verranno ritornati automaticamente dal Broker in seguito alla richiesta di un successivo comando 'SYN' da parte del Dot.

```
{ "SUB": {  
  "src": "Weather",  
  "res": "Solar/Active"  
}}
```

Parametri obbligatori

src Nome del Dot che effettua la richiesta di Update.

Parametri opzionali

res Se la richiesta di 'abbonamento' è riferita ad un'unica variabile, è possibile usare il parametro 'res' con un formato del tipo 'Dot/Variabile' o 'Dot/Gruppo/Variabile'.

vars Se la richiesta di 'abbonamento' è riferita ad un insieme di variabili oppure se si intende cambiare lo stato di 'abbonamento', occorre utilizzare il parametro 'vars'.

Questo può contenere un insieme di parametri con formato del tipo 'Dot/Variabile':valore'. Il 'valore' può essere 1 per attivare la sottoscrizione, oppure 0 per annullarla.

Altri esempi

```
{ "SUB": {  
  "src": "Weather",  
  "vars": {  
    "Solar/Active": 1,  
    "Camera/Angle": 1,  
    "Garden/Gate/Status": 0  
  }  
}}
```



Risposta

```
{"ret":{  
  "rst":"ok"}}  
  
{"ret":  
  {"rst":"err",  
   "err":"refused"}}
```



Comando UPD

Il comando UPD (Update) permette di ricevere dal server uno o più comandi di aggiornamento dello stato relativi alle variabili ed eventi su cui ha effettuato una subscription il Dot.

```
{ "UPD" : {  
  "src" : "Weather"  
}}
```

Parametri obbligatori

src Nome del Dot che effettua la richiesta di Update.

Parametri opzionali

Risposta

La risposta ad un Update è una serie di comandi indirizzati al Dot, che verranno interpretati. Questi comandi possono essere richieste di scrittura di variabili (SET), richieste di esecuzione di metodi (CALL), o altri comandi esplicitamente diretti al Dot che ne fa richiesta.



Comando CLL

Il comando CLL (Call) permette di eseguire un metodo di un Dot.

```
{ "CLL": {  
  "dot": "Weather",  
  "method": "Stop",  
  "params": {  
    "clear": "true"  
  }  
}}
```

Parametri obbligatori

method	Nome del metodo da eseguire. Può essere un nome semplice ('Start', 'Stop', ecc.) oppure può contenere il nome del Dot di riferimento ('Weather/Start').
---------------	---

Parametri opzionali

dot	Se 'method' non identifica un Dot di riferimento, è possibile specificarlo con questo parametro.
params	Se il metodo da richiamare ha dei parametri, questi vengono specificati all'interno di questo sotto-oggetto.

Risposta

```
{"ret": {  
  "rst": "ok" }  
  
{"ret":  
  {"rst": "err",  
   "err": "refused" } }
```

Appendice

Sequenza di comunicazione

Lo schema seguente mostra un esempio di sequenza di comunicazione tra un Dot e un Broker.

