



**PROCEDURA APERTA PER L’AFFIDAMENTO DELLA FORNITURA DI AUSILI PER
INCONTINENZA E ASSORBENZA A MINOR IMPATTO AMBIENTALE 3**

**ALLEGATO 5.1
SISTEMA INFORMATIVO – SPECIFICHE MESSAGGI BACKBONE SOLE**

SPA
SVILUPPO PERCORSI AZIENDALI

SPECIFICHE TECNICHE

P1.10.1A-13

SPECIFICHE MESSAGGI BACKBONE SOLE

Sostituisce o modifica

| Versione/stato | Data | Oggetto | Commento |
|----------------|------|---------|----------|
| | | | |

Storia delle versioni

| Versione/stato | Data | Autore | Sintesi |
|----------------|------------|-------------|---|
| 1.0/finale | 05/07/2013 | G.Cuccu; | Prima versione finale |
| 1.1/finale | 10/07/2013 | G.Cuccu | Correzione sui nome dei campi nel messaggio JSON |
| 1.2/finale | 10/10/2013 | D.Gubellini | Correzione nomi dei campi sulla tabella nel paragrafo 2.2.1 |
| 1.3 finale | 29/10/2013 | S.Micocci | Cap2.2.3: chiarimento valorizzazione campo priority |

Limiti di utilizzo del documento

| |
|---|
| <p>La circolazione di questo documento è autorizzata unicamente ai soggetti che partecipano attivamente ai progetti e Servizi ICT per l'area sanitaria e socio sanitaria nei limiti della realizzazione dello stesso.</p> <p>Ogni altro utilizzo in contrasto con il limite suddetto o comunque non autorizzato sarà perseguito a termini di legge.</p> |
|---|

Scheda Riassuntiva Documento

| | |
|-----------------------------|------------------------------------|
| Progetto | Sviluppo Percorsi Aziendali |
| Acronimo | SPA |
| Capo Progetto | Alberto Anelli |
| Referente regionale | Anna Darchini |
| Responsabile BU | Caterina Lena |
| Data inizio Piano Operativo | 01/01/2013 |
| Data fine Piano Operativo | 31/12/2013 |

| | |
|-------------------------|--|
| Tipologia | Specifiche tecniche |
| Titolo Documento | Specifiche messaggi backbone SOLE |
| Attività di riferimento | Sviluppo Percorsi Aziendali |
| Autore | |
| Versione Stato | 1.3 finale |
| Data | 29/10/2013 |
| File | c:\documents and settings\sm08\documenti\bozze di sharepoint\p1.10.1a-13_spa_mia_reingegnerizzazione sole allegato_specifiche_messaggi_backbone sole.doc |

Abstract: Obiettivo del documento è quello di fornire le specifiche tecniche dei messaggi per il backbone SOLE.

Keywords: Reingegnerizzazione SOLE SPA

INDICE

| | |
|--|----|
| Scheda Riassuntiva Documento | 4 |
| 1 Introduzione | 6 |
| 2 Tipi di invocazioni supportate | 6 |
| 2.1 Protocollo di trasporto | 6 |
| 2.2 Struttura messaggi del backbone | 6 |
| 2.2.1 Definizione del formato del messaggio di richiesta | 6 |
| 2.2.2 Definizione del messaggio di risposta | 8 |
| 2.2.3 Gestione delle priorità | 8 |
| 2.2.4 Messaggio asincrono: consegna del messaggio dal bb-SOLE al sistema esterno ... | 10 |
| 2.2.5 Messaggi Chiamata sincrona endpoint-bb-SOLE | 11 |

1 Introduzione

Obiettivo del documento è quello di fornire le specifiche tecniche per i fornitori che vogliono inviare/ricevere messaggi dal backbone SOLE. Il backbone è un applicativo di integrazione che consente ad applicativi installati sul territorio regionale di poter comunicare fra di loro in modalità trasparente, ovvero parlando solo con il backbone senza un collegamento diretto con l'applicativo con il quale si devono scambiare informazioni.

2 Tipi di invocazioni supportate

Il backbone supporta due tipi di invocazioni:

1. Invocazioni asincrone in cui l'inviante affida il messaggio al backbone senza attendere la risposta dal sistema ricevente
2. Invocazioni sincrone in cui l'inviante invia il messaggio al backbone e attende la risposta dal sistema ricevente

Un applicativo quando si integra con il backbone può svolgere due ruoli:

1. Inviante, ovvero invia messaggi al backbone che devono essere recapitati ad un altro applicativo
2. Ricevente, ovvero riceve messaggi dal backbone che sono stati inviati da un applicativo inviante

A prescindere dal tipo di applicativo il formato veicolato dal backbone è sempre uno solo che in un caso (inviante) deve essere sintentizzato, mentre nell'altro, ricevente, deve essere letto e deve scatenare le opportune procedure.

2.1 Protocollo di trasporto

Le chiamate al backbone avverranno tramite l'utilizzo del protocollo https e gli url del backbone saranno esposti tramite la metodologia REST. Gli endpoint del backbone saranno invocati tramite metodo POST per l'invio dei messaggi e tramite metodo GET per il recupero.

Le chiamate devono essere effettuate tramite certificato client; ovvero il protocollo di comunicazione dei messaggi è https con certificato client. Il certificato client verrà utilizzato come elemento di autenticazione e sarà uno dei fattori utilizzato nelle politiche di autorizzazione.

2.2 Struttura messaggi del backbone

2.2.1 Definizione del formato del messaggio di richiesta

I messaggi da inviare al backbone hanno i seguenti campi:

| Campo | Significato |
|---------------|---|
| id | Identificativo del messaggio fornito dall'applicativo inviante |
| message | Dato che si vuole trasmettere |
| messageType | Specifica se il messaggio è di tipo stringa o binario |
| priority | Priorità del messaggio |
| customHeaders | Lista di elementi chiave->valore da utilizzare per specificare dei valori aggiuntivi da inviare al destinatario |

Id

Il campo Id è fornito dal mittente e serve per identificare il messaggio in caso di problemi con l'inviante. Il suo scopo è, in caso di problemi di fornire un riferimento che sia noto all'inviante e che questo possa utilizzare per effettuare operazioni di varia natura all'interno dei propri sistemi.

Il campo è obbligatorio, di tipo stringa e la sua lunghezza massima è 60 caratteri

Message

È il contenuto informativo da trasmettere, che viene recapitato al destinatario. La lunghezza massima è di 500MB. Il campo è obbligatorio.

MessageType

È il campo che definisce il formato del messaggio che può essere di due tipi: binario e stringa. Il campo è obbligatorio, di tipo stringa e può assumere due valori "string" o "binary"

Priority

Stabilisce la priorità del messaggio rispetto ad altri presenti nella stessa coda, le priorità previste sono 3: 1,2,3 ad un valore più alto è associata una priorità maggiore, ovvero un messaggio con priorità 3 verrà consegnato prima di un messaggio con priorità 1 o 2. Il campo è obbligatorio. Per i dettagli sulla corretta valorizzazione del campo priorità si faccia riferimento al paragrafo dedicato.

CustomHeaders

I custom header rappresentano la possibilità per l'inviante di spedire informazioni aggiuntive rispetto al messaggio. Essi sono modellati dal punto di vista logico come una sequenza di coppie

chiave/valore. Il campo chiave è di tipo stringa e ha lunghezza massima 60 caratteri, il campo valore è di tipo stringa e ha lunghezza massima di 2048 caratteri. Il numero massimo di header che si possono trasmettere per singolo messaggio è di 1024. Un valore di chiave non può essere ripetuto nel singolo invio, ovvero i valori presenti nei campi chiave devono essere distinti fra di loro nel singolo messaggio. Il campo è facoltativo.

2.2.2 Definizione del messaggio di risposta

Il messaggio di risposta del backbone risulta estremamente semplice in quanto è costituito, in caso di esito positivo da un codice che identifica il messaggio all'interno del backbone. Il codice restituito è di tipo stringa e ha lunghezza massima 128 caratteri.

In caso di errore, oltre al codice http sarà restituita all'interno del body HTTP anche una descrizione dell'errore.

2.2.2.1 Codici di risposta

Anche le risposte del backbone saranno fornite seguendo l'approccio REST e quindi l'esito della chiamata sarà comunicando tramite i codici di risposta HTTP.

I codici di risposta HTTP utilizzati saranno:

- 200 invocazione corretta
- 400 il messaggio non è corretto, con questo codice di errore si segnala sia la mancanza di campi obbligatori, valori dei campi non ammessi (es priorità pari a 4 o priorità non consentita per lo specifico endpoint)
- 401 l'inviante non ha inviato il certificato client
- 403 l'inviante ha inviato il certificato client, ma non è autorizzato ad usare l'endpoint
- 415 l'inviante ha inviato un messaggio usando un content-type non supportato
- 500 errore interno del backbone

2.2.3 Gestione delle priorità

I messaggi che vengono trasportati dal backbone devono obbligatoriamente indicare un grado di priorità, secondo il seguente schema:

| Tipo di coda | Valore del campo priorità (<i>priority</i>) |
|--|---|
| Sincrona | Valorizzare obbligatoriamente con il valore 1 |
| Asincrona, con priorità gestita dall'applicativo invocante | Valorizzare obbligatoriamente con uno dei valori previsti (1,2,3) |
| Asincrona, con priorità NON gestita dall'applicativo invocante | Valorizzare obbligatoriamente con il valore 1 |

2.2.3.1 Realizzazione del messaggio di ingresso tramite application/json

I paragrafi precedenti hanno definito il messaggio dal punto di vista logico, ma non hanno indicato come questo possa essere concretamente inviato al backbone. Seguendo la filosofia REST la proposta è di realizzare una infrastruttura che consente l'invio di messaggi secondo modalità di rappresentazione diverse. Il modo in cui il messaggio viene inviato è specificato tramite l'header http Content-Type. Il valore di tale header determina la rappresentazione fisica del messaggio. In questa versione del backbone verrà realizzata la parte di infrastruttura che consente di gestire solo il tipo application/json ovvero nella prima versione del backbone i messaggi sono spediti e gestiti sono con un formalismo json.

La versione attuale del backbone supporta application/json come unico Content-type, di seguito sono elencate le specifiche di composizione del messaggio. Il charset da utilizzare è solo uno e più precisamente charset=utf-8; tale valore deve essere specificato nel campo content-type subito dopo il tipo.

I nomi dei campi sono:

- id
- message
- messageType
- priority
- customHeaders

Di seguito vengono mostrati alcuni esempi di messaggi json

Messaggio senza custom header

```
{ "id": "ABCD", "message": "messaggio di
testo", "messageType": "string", "priority": 1, "customHeaders": {} }
```

Messaggio con custom header

```
{"id":"ABCD","message":"messaggio di  
testo","messageType":"string","priority":1,"customHeaders":{"chiaveCustom":"val  
oreCustom"}}
```

2.2.3.2 Realizzazione del messaggio di risposta tramite application/json

La rappresentazione della risposta del backbone ad un invio di messaggio è molto semplice in quanto l'unica informazione da veicolare è l'id del messaggio di risposta o la descrizione dell'errore. Di seguito due esempi, uno di id del messaggio di richiesta ed uno di messaggio di errore.

```
"id123456"  
"valore priorità non valido"
```

2.2.3.3 Consegna di messaggi multipli

Il backbone, all'atto della consegna dei messaggi consente di inviare all'applicativo di destinazione n messaggi con una sola invocazione. La messaggistica JSON utilizzata è quella dei paragrafi precedenti in cui si compone un array json di messaggi.

Si riporta un esempio di contenuto di una invocazione (http body) che contiene 2 messaggi

```
[{"id":"ABCD","message":"messaggio di  
testo","messageType":"string","priority":1,"customHeaders":{}},  
{"id":"ABCD1","message":"messaggio di  
testo1","messageType":"string","priority":1,"customHeaders":{}}]
```

2.2.4 Messaggio asincrono: consegna del messaggio dal bb-SOLE al sistema esterno

Il backbone ha il compito di ricevere un messaggio e di consegnarlo al mittente. Se la ricezione del messaggio da parte del backbone avviene in un unico modo ci sono diverse alternative per la sua consegna. L'attuale versione del backbone supporta tre modalità di consegna che sono descritte di seguito:

1. **Modalità push event based:** il backbone invoca un endpoint dell'applicativo a cui deve consegnare i messaggi: l'endpoint deve essere realizzando le specifiche di invio messaggi del backbone (ovvero deve apparire come se fosse un altro backbone che accetta messaggi). La modalità immediata significa che ogni volta che arriva un messaggio questo viene consegnato immediatamente. In questa modalità non è possibile bilanciare il carico

verso il sistema di consegna ed è da utilizzare nel caso in cui si voglia la minima latenza fra i due sistemi

2. **Modalità push polling based:** il backbone invoca un endpoint dell'applicativo a cui deve consegnare i messaggi: l'endpoint deve essere realizzando le specifiche di invio messaggi del backbone (ovvero deve apparire come se fosse un altro backbone che accetta messaggi). La modalità a polling significa che il backbone ogni tot secondi verifica se ci sono messaggi ed in caso positivo ne invia al massimo n verso la destinazione finale. Questa implementazione garantisce un carico uniforme sul sistema di destinazione a scapito della latenza dei messaggi.
3. **Modalità pull:** l'applicativo esterno invoca un url (in stile REST) del backbone chiedendo di ricevere fino a n messaggi.

La regola di consegna sarà configurabile da interfaccia di configurazione. Ogni applicativo che dovrà ricevere i messaggi dal backbone dovrà concordare qualche modalità di consegna intende sviluppare. I messaggi ricevuti sono quelli descritti nei paragrafi precedenti.

L'applicativo che deve ricevere dei messaggi dal backbone e ha scelto la modalità Push dovrà implementare un endpoint http o https che sia in grado di ricevere i messaggi descritti in precedenza. L'applicativo dovrà garantire di essere in grado di poter ricevere messaggi multipli in maniera concorrente; qualora questo non fosse possibile tale impossibilità dovrà essere dichiarata prima di cominciare l'integrazione.

Nel caso di integrazioni in modalità push polling o in modalità pull dove con un singola invocazione possono essere ricevuti n messaggi la struttura del messaggio sarà quella di un array di messaggi.

2.2.5 Messaggi Chiamata sincrona endpoint-bb-SOLE

Le invocazioni di tipo sincrone differiscono da quelle sincrone nel messaggio di risposta, in quanto la risposta non può essere un semplice messaggio ma deve essere un messaggio con del contenuto informativo. Nel caso di chiamata sincrona il messaggio di risposta sarà un messaggio che ha il formato descritto in precedenza sia a livello di campi che di codifica (json con encoding UTF-8).