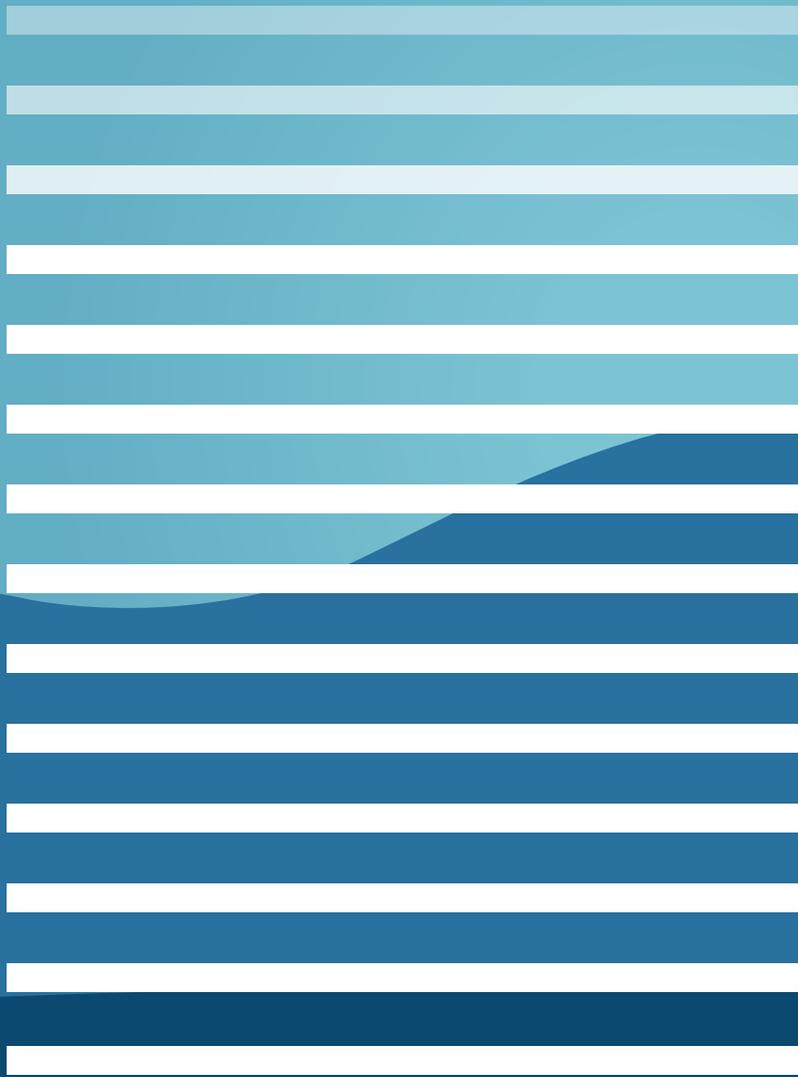


SMS IN *Rules*





SMS In[®] 2.3

Regole per la manipolazione dei messaggi *Guida alla scrittura di condizioni complesse*

Guida alle condizioni complesse v1.0-IT, 7 Dicembre 2012

Indice

Indice.....	2
Cronologia delle modifiche.....	3
Introduzione.....	4
Panoramica sulle regole.....	5
Struttura delle regole.....	5
<i>La condizione</i>	5
<i>L'azione</i>	5
<i>La priorità</i>	6
Scrittura di condizioni complesse.....	6
La sintassi generale.....	6
<i>I token</i>	6
<i>Gli operatori di confronto</i>	7
Condizioni sulle singole parole del messaggio.....	8
Condizioni sulle proprietà del messaggio.....	9
<i>Test del mittente</i>	10
<i>Test sull'orario di ricezione</i>	10
Condizioni su porzioni di testo.....	10
<i>Test sull'intero messaggio</i>	10
<i>Test sul messaggio privato della keyword</i>	10
<i>Test sulla parte finale del messaggio</i>	11
<i>Test sulla parte iniziale del messaggio</i>	12
<i>Test su porzioni qualsiasi del messaggio</i>	12
Uso di espressioni regolari.....	13
Uso dei token nel messaggio di risposta.....	14
Token che restituiscono porzioni del messaggio.....	14
Token che restituiscono informazioni sul messaggio.....	15
Token speciale {receiver}.....	15
Documenti collegati.....	17

Cronologia delle modifiche

Versione	Quando	Cosa	Chi
1.0	7 dic 2012	Prima versione ufficiale	Giuseppe Costa

Introduzione

Questa guida si prefigge lo scopo di descrivere dettagliatamente la sintassi per la definizione di regole con condizioni complesse, fornendo di conseguenza un ampio grado di flessibilità nell'analisi e nel trattamento del contenuto dei messaggi ricevuti.

Geodrop è un marchio registrato proprietario di A-Tono.

Panoramica sulle regole

In questo capitolo riassumiamo velocemente la struttura delle regole, per poi passare ad una analisi dettagliata della sintassi per la scrittura delle condizioni e di una particolare azione: l'invio del messaggio di risposta.

Struttura delle regole

Le regole descrivono il comportamento del sistema nel momento in cui un messaggio è ricevuto in una data Inbox. Una regola si compone di:

- **una condizione:** descrive la struttura dei messaggi su cui la regola deve essere applicata. Se un dato messaggio corrisponde alla struttura descritta dalla condizione, allora si dice che la condizione è *verificata* (o è *vera*) per tale messaggio.
- **un'azione:** descrive le operazioni che il sistema dovrà eseguire ogni volta che riceve un messaggio per cui la condizione è verificata.
- **una priorità:** è un valore numerico intero non negativo che stabilisce un ordinamento tra le regole. Le regole con priorità più bassa vengono per prime in tale ordinamento.

La condizione

Si tratta di un'espressione logica, ovvero una serie di operazioni il cui risultato finale può essere solo VERO o FALSO.

Il risultato delle operazioni dipende dalle caratteristiche del messaggio (mittente, testo, ora di ricezione, ecc) per il quale viene valutata l'espressione.

Se il valore finale della condizione è VERO, allora il sistema *applica* la regola, ovvero esegue l'azione. Altrimenti la regola non viene applicata.

Un esempio semplice di condizione, abbastanza intuitivo, è: *{sender} = 3933412345567*.

La condizione sopra è vera per tutti e soli i messaggi inviati dal numero di cellulare +393341234567 (si noti il fatto che il "+" del prefisso internazionale NON deve essere specificato), falsa per tutti gli altri.

L'azione

Le possibili azioni previste dal sistema sono:

- inoltra ad un numero di cellulare
- inoltra a una URL

- invio di un SMS di risposta
- inserimento in un gruppo di contatti in rubrica
- rimozione da un gruppo di contatti in rubrica

Non è scopo di questa guida analizzare nel dettaglio le possibili azioni. In un paragrafo dedicato sarà presa in considerazione solo l'azione di invio messaggio di risposta, in quanto all'interno del messaggio è possibile usare elementi comuni alle condizioni.

La priorità

Quando ci sono più regole, serve a stabilire quale sia l'ordine in cui il sistema proverà ad applicarle. A parità di priorità, l'ordine è casuale.

Scrittura di condizioni complesse

La sintassi generale

Le condizioni delle regole sono formate da condizioni elementari nella forma **[TOKEN - operatore di confronto – VALORE]** combinate tramite gli operatori logici **AND** e **OR**. I **token** sono *stringhe*, ovvero sequenze di caratteri, racchiuse tra parentesi graffe.

Un esempio di condizione complessa è la seguente:

```
{2} = 'mario' AND {timestamp} > '2012-05-16'
```

Senza scendere per ora in dettaglio sul significato generale o specifico delle varie parti, nel precedente esempio possiamo individuare:

- le condizioni elementari:
 - `{2} = 'mario'`
 - `{timestamp} > '2012-05-16'`
- gli operatori di confronto = e >;
- i token `{2}` e `{timestamp}`
- l'operatore logico *AND*.

I token

I token riconosciuti sono i seguenti:

Token che identificano singole parole del messaggio o specifiche caratteristiche:

1. **{x}** : Il token più tipico è quello che inserisce un numero (ovviamente intero) tra le parentesi graffe. Questo token indica la parola in posizione x (contando a partire da 0) del messaggio.

La parola in posizione 0 è la keyword che identifica la inbox.

2. **{sender}**: Indica il numero mittente del messaggio
3. **{timestamp}**: Indica l'istante di arrivo del messaggio nel formato: "aaaa-mm.gg HH:mi:ss", ovvero *anno*(4cifre)-*mese*(2 cifre)-*giorno*(2 cifre) seguito da *ora*(2cifre, in formato 24 ore):*minuti*(2 cifre):*secondi*(2cifre)

Token che identificano porzioni del messaggio:

4. **{>x}**, **{<x}**, **{>=x}**, **{<=x}** : Questi token indicano tutte le parole nel messaggio che siano, rispettivamente, in posizioni maggiori, minori, maggiori o uguali e minori o uguali a x.
5. **{x.y}**: Questo token indica tutte le parole comprese tra la x-esima e la y-esima. Nel caso in cui x sia maggiore di y, le parole saranno considerate in ordine inverso.
6. **{text}**: Indica l'intero testo del messaggio compresa la keyword iniziale (se presente)
7. **{message}**: Indica il testo del messaggio esclusa la keyword iniziale (se presente)

Token speciale per l'uso di espressioni regolari

8. **{r'<expr>'}**: È un token speciale utilizzabile solo all'interno delle condizioni. La componente **<expr>** all'interno della struttura del tag indica un'*espressione regolare* (o *regular expression* o ancora *regexp*): gli apici sono obbligatori e devono racchiuderla. Se l'espressione regolare risulta soddisfatta dal messaggio, l'intero token assume il valore true, altrimenti esso assume il valore di false.

Per maggiori informazioni sulle espressioni regolari, si vedano ad esempio i seguenti link:

- http://it.wikipedia.org/wiki/Espressione_regolare (introduzione su Wikipedia)
- http://it.wikibooks.org/wiki/Espressioni_regolari (guida su Wikibook)
- <http://www.regular-expressions.info/> (sito dedicato, in lingua inglese)

Gli operatori di confronto

Gli operatori di confronto supportati sono:

- = (uguale);
- != (diverso);
- > (maggiore);
- < (minore);
- >= (maggiore o uguale);
- <= (minore o uguale).

Condizioni sulle singole parole del messaggio

La condizione può effettuare dei test sulle singole parole del messaggio tramite il token **{x}**, che permette di identificarle in base alla posizione.

Esempio 1

Volendo controllare se la prima parola del messaggio è “SI”, scriveremmo la seguente condizione elementare:

`{1} = 'SI'`

Supponendo di definire la regola sulla inbox identificata dalla keyword “iscrizione”, la condizione sopra è vera per messaggi del tipo:

- iscrizione si
- iscrizione SI
- iscrizione si desidero iscrivermi
- iscrizione si

Non è vera per messaggi del tipo:

- iscrizione ho detto si
- iscrizione no
- iscrizione

Esempio 2

Volendo controllare se la prima parola del messaggio è “SI” oppure “NO”, scriveremmo la seguente condizione:

`{1} = 'SI' OR {1} = 'NO'`

Si noti che la condizione sopra è composta da due condizioni elementari unite dall'operatore logico “OR”, il cui significato è “oppure”. La regola si legge quindi nel seguente modo:

“la prima parola è 'SI' oppure la prima parola è 'NO'”

Supponendo di definire la regola sulla inbox identificata dalla keyword “iscrizione”, la condizione sopra è vera per messaggi del tipo:

- iscrizione si
- iscrizione SI
- iscrizione si desidero iscrivermi
- iscrizione si

- iscrizione no

Non è vera per messaggi del tipo:

- iscrizione ho detto si
- iscrizione

Esempio 3

Volendo controllare che la prima parola del messaggio sia “SI” e che la seconda sia “desidero”, scriveremmo la seguente condizione:

`{1} = 'SI' AND {2} = 'DESIDERO'`

Si noti che la condizione sopra è composta da due condizioni elementari unite dall'operatore logico “AND”, il cui significato è “e”. La regola si legge quindi nel seguente modo:

“la prima parola è 'SI' e la seconda parola è 'DESIDERO’”

Supponendo di definire la regola sulla inbox identificata dalla keyword “iscrizione”, la condizione sopra è vera per messaggi del tipo:

- iscrizione si desidero
- iscrizione si desidero farlo
- iscrizione si desidero iscrivermi

Non è vera per messaggi del tipo:

- iscrizione si
- iscrizione SI
- iscrizione si
- iscrizione no
- iscrizione ho detto si
- iscrizione

Condizioni sulle proprietà del messaggio

La condizione può effettuare dei test su alcune proprietà del messaggio. Le proprietà attualmente supportate sono:

1. mittente (tramite il token **{sender}**);
2. orario di ricezione (tramite il token **{timestamp}**).

Test del mittente

È possibile vincolare l'applicazione di una regola alla condizione che il mittente sia o no un numero specifico. Il numero deve essere preceduto dal prefisso internazionale SENZA il “+” o gli zeri iniziali. Ad esempio, per il numero italiano 3331234567, che in formato internazionale sarebbe “+393331234567” o “00393331234567”, dovremo scrivere “393331234567”.

Ecco due esempi di test che verificano il mittente di un messaggio:

1. `{sender} = '393331234567'`, vera SOLO se il mittente è esattamente il numero '393331234567';
2. `{sender} != '393331234567'`, vera per qualunque mittente diverso dal numero di cellulare '393331234567'.

Test sull'orario di ricezione

È possibile eseguire test sull'orario di ricezione del messaggio da parte del sistema, vincolando l'applicazione di una regola ad una data o un'ora specifica. Un esempio di condizione di questo tipo è la seguente:

```
{timestamp} > '2012-12-07 14:00:00'
```

La condizione definita sopra è vera per tutti i messaggi ricevuti dopo le 14:00:00 del giorno 27 dicembre 2012 e falsa per tutti quelli ricevuti prima.

Condizioni su porzioni di testo

Oltre a controllare le singole parole, è possibile verificare se il messaggio contiene una particolare frase in posizione data. I tag da utilizzare in questo caso sono quelli descritti nei punti dal 4. al 7., i quali permettono di selezionare porzioni specifiche del messaggio ricevuto o l'intero messaggio.

Test sull'intero messaggio

Partiamo con un esempio di condizione che fa una verifica sull'intero messaggio, comprensivo della keyword iniziale:

```
{text} = "iscrizione si aderisco"
```

Tale condizione è vera solo su messaggi il cui testo comprensivo della keyword è esattamente quello indicato e falsa per tutti gli altri.

Test sul messaggio privato della keyword

Poiché la keyword è comune a tutti i messaggi gestiti da una data regola (la quel, ricordiamolo, è definita su una inbox che ha sua volta è identificata dalla keyword), si può ottenere una condizione equivalente alla precedente usando il token **{message}**, che individua il testo del messaggio depurato dalla keyword iniziale:

```
{message} = 'si aderisco'
```

Questa regola è esattamente analoga alla precedente, ma leggermente più compatta da scrivere.

Test sulla parte finale del messaggio

Per verificare se il messaggio termina con una specifica frase, si usano i tag:

- **{>x}**: tutte le parole in posizione successiva a x;
- **{>=x}**: tutte le parole a partire da quella in posizione x.

La differenza tra i due tag è che il secondo, in fase di controllo, considera anche la parola in posizione x come parte della frase da verificare, mentre il primo la esclude. Chiariamo meglio con un esempio. Consideriamo le seguenti condizioni elementari:

1. **{>1}** = 'si aderisco'
2. **{>=1}** = 'si aderisco'

Le due condizioni, se applicate al messaggio "iscrizione si aderisco", sono rispettivamente falsa e vera.

La condizione 1. infatti ESCLUDE la parola in posizione 1 (ovvero "si"). La porzione di messaggio su cui effettua il test risulta quindi essere la sola parola "aderisco", che è chiaramente diversa dalla frase indicata.

La condizione 2. INCLUDE invece la parola in posizione 1. La porzione di messaggio su cui viene effettuato il test è quindi "si aderisco" che corrisponde alla frase indicata.

Entrambe le condizioni sono false sui seguenti messaggi:

- "iscrizione si aderisco subito"
 - testo su cui esegue il controllo la condizione 1.: "aderisco subito"
 - testo su cui esegue il controllo la condizione 2.: "si aderisco subito"
- "iscrizione non aderisco"
 - testo su cui esegue il controllo la condizione 1.: "aderisco"
 - testo su cui esegue il controllo la condizione 2.: "non aderisco"

Sul seguente messaggio, invece, la condizione 1. è vera e la 2. è falsa:

- "iscrizione assolutamente si aderisco"
 - testo su cui esegue il controllo la condizione 1.: "si aderisco"
 - testo su cui esegue il controllo la condizione 2.: "assolutamente si aderisco"

Test sulla parte iniziale del messaggio

Per verificare se il messaggio inizia con una specifica frase, si usano i tag:

- **{<x}**: tutte le parole che precedono quella in posizione x;
- **{<=x}**: tutte le parole sino a quella in posizione x (compresa).

Il ragionamento è analogo a quello fatto per i tag sulla parte finale del messaggio e non stiamo a ripeterlo. Consideriamo le seguenti condizioni:

1. **{<2}** = 'si aderisco'
2. **{<=2}** = 'si aderisco'

Forniamo i seguenti esempi per mostrare concretamente come funzionano i due token:

- “iscrizione si aderisco subito”
 - testo su cui esegue il controllo la condizione 1.: “si” (falsa)
 - testo su cui esegue il controllo la condizione 2.: “si aderisco” (vera)
- “iscrizione non aderisco”
 - testo su cui esegue il controllo la condizione 1.: “non” (falsa)
 - testo su cui esegue il controllo la condizione 2.: “non aderisco” (falsa)
- “iscrizione assolutamente si aderisco”
 - testo su cui esegue il controllo la condizione 1.: “si” (falsa)
 - testo su cui esegue il controllo la condizione 2.: “assolutamente si” (falsa)

In particolare, la condizione 1. sarà sempre falsa. Le parole che precedono quella in posizione 2 sono in fatti sempre ... una sola (quella in posizione 1)! Una sola parola non potrà quindi mai essere uguale ad una frase composta da due.

Test su porzioni qualsiasi del messaggio

Il token **{x.y}** permette di effettuare controlli su porzioni qualsiasi del messaggio. La frase selezionate per il confronto **COMPRENDE** le parole in posizione x e y. Ecco un esempio di condizione elementare:

{1.3} = 'si aderisco adesso'

La condizione dell'esempio è vera per i seguenti messaggi:

- “iscrizione si aderisco adesso”
- “iscrizione si aderisco adesso al concorso”

mentre è falsa per i seguenti:

- “iscrizione si aderisco subito”
- “iscrizione non aderisco adesso”
- “iscrizione sicuramente si aderisco adesso”

Uso di espressioni regolari¹

I token che abbiamo analizzato sin'ora permettono di definire test su porzioni specifiche del messaggio, individuate da uno o due estremi fissi. Esiste un token che ci permette di effettuare controlli sulla struttura del messaggio o sulla presenza di un determinato testo in una posizione qualsiasi del messaggio.

L'uso di questo token richiede una buona conoscenza delle espressioni regolari (si vedano i link indicati nel paragrafo “I token” per approfondimenti sulle espressioni regolari).

Il token restituisce sempre un valore vero o falso e non prevede l'uso di operatori: tutto il lavoro è svolto dall'espressione regolare al suo interno.

Alcuni esempi di token per l'uso di espressioni regolari sono i seguenti:

1. `{r'[0..9]+}`, vero se il messaggio contiene un numero composto da 1 o più cifre in qualsiasi posizione. Alcuni messaggi con relativo esito del controllo sono i seguenti:
 - “iscrizione si al concorso 4” (vero: è presente il numero 4)
 - “iscrizione si 5 volte” (vero: è presente il numero 5)
 - “iscrizione no tre volte” (falso: non è presente alcun numero espresso in cifre)
 - “iscrizione 4 volte si al numero 78” (vero: è presente il numero 4 e anche il numero 78)
2. `{r'\bsi\b}`, vero se il messaggio contiene la parola “si” preceduta e seguita da uno spazio. Alcuni messaggi con relativo esito del controllo sono i seguenti:
 - “iscrizione si al concorso 4” (vero)
 - “iscrizione 5 volte si ” (vero)
 - “iscrizione no tre volte” (falso)
 - “iscrizione si” (falso: dopo la parola “si” non c'è lo spazio!)

¹ Al momento della scrittura della guida, il supporto alle espressioni regolari non è completo. Per poterle utilizzare correttamente è necessario aggiungere una condizione elementare sempre falsa in OR con il token regexp (es: `{r'\bsi\b} OR {0} = ""`). Alcuni tipi di espressioni regolari non sono inoltre gestite correttamente. Il supporto completo alle espressioni regolari è comunque previsto e in fase di implementazione. La presente guida sarà aggiornata costantemente per riflettere le evoluzioni di questa funzionalità.

Uso dei token nel messaggio di risposta

Quando la regola prevede come azione l'invio di un messaggio di risposta, è possibile utilizzare, nel testo di quest'ultimo, alcuni dei token visti per la scrittura delle condizioni.

I token utilizzabili sono tutti quelli che restituiscono una porzione del testo del messaggio ricevuto o che permettono di accedere alle caratteristiche del messaggio stesso.

Token che restituiscono porzioni del messaggio

I seguenti token permettono di riportare nel messaggio di risposta parti del messaggio originale:

1. **{x}** : inserisce la parola in posizione x nel messaggio di risposta
2. **{>x}**, **{>=x}** : inseriscono la parte finale del messaggio ricevuto a partire dalla parola in posizione x (esclusa o inclusa rispettivamente).
3. **{<x}**, **{<=x}** : inseriscono la parte iniziale del messaggio ricevuto sino alla parola in posizione x (esclusa o inclusa rispettivamente).
4. **{x.y}**: inserisce la porzione di messaggio compresa tra le parole in posizione x e y comprese tali parole.
5. **{text}**: inserisce l'intero testo del messaggio ricevuto, compresa la keyword.
6. **{message}**: inserisce il testo del messaggio ricevuto, ripulito dalla keyword iniziale.

Considerando il seguente messaggio:

“iscrizione si al concorso per paracadutisti”

Vediamo cosa succederebbe utilizzando i vari token in possibili messaggi di risposta inviati alla ricezione di tale messaggio (in grassetto la porzione del messaggio ricevuto inserito in quello di risposta):

1. singola parola:
 - esempio di messaggio configurato nella regola: “La prima parola è: {1}”.
 - messaggio di risposta inviato: “La prima parola è: **si**”
2. parte finale del messaggio:
 - esempio di messaggio configurato nella regola: “Hai concluso con: {>2}”.
 - messaggio di risposta inviato: “Hai concluso con: **concorso per paracadutisti**”
3. parte iniziale del messaggio:
 - esempio di messaggio configurato nella regola: “Hai iniziato con: {<=3}”.
 - messaggio di risposta inviato: “Hai iniziato con: **si al concorso**”

4. porzione generica di messaggio:

- esempio di messaggio configurato nella regola: “Il messaggio contiene: {2.4}”.
- messaggio di risposta inviato: “Il messaggio contiene: **al concorso per**”

5. testo completo di keyword:

- esempio di messaggio configurato nella regola: “Il messaggio che hai inviato è: {text}”.
- messaggio di risposta inviato: “Il messaggio che hai inviato è: **iscrizione si al concorso per paracadutisti**”

6. messaggio ripulito dalla keyword:

- esempio di messaggio configurato nella regola: “Il messaggio inviato è: {message}”.
- messaggio di risposta inviato: “Il messaggio inviato è: **si al concorso per paracadutisti**”.

Token che restituiscono informazioni sul messaggio

I seguenti token permettono di inserire nel messaggio di risposta informazioni sul messaggio ricevuto:

1. **{sender}**: inserisce il numero del mittente del messaggio ricevuto
2. **{timestamp}**: inserisce l'istante di arrivo del messaggio ricevuto

Supponendo di ricevere alle 12:30 del 3 dicembre 2012 un messaggio da parte del numero 3331234567, ecco come si comporterebbero i token di cui sopra, se inseriti in un messaggio di risposta (in grassetto le informazioni sul messaggio ricevuto inserite in quello di risposta):

- messaggio configurato nella regola: “Il messaggio inviato dal numero {sender} è stato ricevuto in data {timestamp}”
- messaggio di risposta inviato: “Il messaggio inviato dal numero **+393331234567** è stato ricevuto in data **2012-12-03 12:30:00**”

Token speciale {receiver}

Il token speciale {receiver} permette di recuperare l'informazione sul numero SMS In al quale è stato inviato il messaggio su cui è stata applicata la regola. Questo è utile ad esempio se si hanno più inbox individuate dalla stessa keyword su numerazioni differenti.

Supponiamo di avere ad esempio le seguenti inbox:

1. keyword “iscrizione” sul numero 3202041300
2. keyword “iscrizione” sul numero 3404324043

In entrambe le inbox definiamo una regola che invia sempre un messaggio di risposta con il seguente testo: “Il messaggio '{text}' è stato ricevuto sul numero {receiver}”. Supponiamo di:

- inviare il messaggio “iscrizione si” alla prima inbox. Otterremo come risposta “Il messaggio '**iscrizione si**' è stato ricevuto sul numero **3202041300**”;
- inviare il messaggio “iscrizione si” alla seconda inbox. Otterremo come risposta “Il messaggio '**iscrizione si**' è stato ricevuto sul numero **3404324043**”.

Documenti collegati

Codice	Descrizione
[SMSIN_GUIDE]	SMS In user guide