

# API servizi di messaggistica

## Soap Gateway

v. 0.95

Importante! Gli esempi XML forniti nella presente documentazione servono solo a dare un'idea dell'uso dei parametri e non sono intesi come base per la costruzione di codice funzionante, poiché fortemente dipendenti dall'ambiente di sviluppo. Per essere più chiari: non usate le porzioni di XML qui presentate per interagire con il sistema.

## Definizioni

Piattaforma: l'insieme di apparecchiature e servizi di KQI che erogano i servizi di trasmissione fax, sms e email broadcast.

Documento: il documento da trasmettere via fax, email ecc. e che costituisce il corpo del messaggio da trasmettere

Lista Destinatari: un elenco di uno o più numeri telefonici o di indirizzi email, che costituiscono i destinatari di un messaggio

Trasmissione: la richiesta alla Piattaforma di trasmettere a destinazione uno o più Documenti a una Lista Destinatari.

Stato Trasmissione: informazione codificata fornita dalla Piattaforma in merito allo stato di una Trasmissione, ad esempio "COMPLETED", "AWC", ecc.

Identificativo Trasmissione: numero univoco identificativo di una Trasmissione, associato automaticamente dalla Piattaforma in seguito alla acquisizione corretta di una Trasmissione.

## Introduzione

Il Soap Gateway KQI (KQUMSG1) rende accessibili le funzioni di trasmissione della piattaforma di messaggistica tramite protocollo SOAP.

In questo documento è descritto il comportamento del Gateway SOAP dal punto di vista della programmazione di applicazioni. Non è riportata la descrizione e le caratteristiche dei servizi per le quali si rimanda al contratto o ad altra documentazione disponibile.

Ricordiamo che alcuni parametri relativi alla trasmissione di messaggi sono definiti nel Pannello di Controllo di gestione dei servizi di messaggistica, ad esempio:

- La scelta della qualità di trasmissione dei messaggi fax e la specifica del CSID
- L'impostazione del mittente e la modalità "smart" per gli SMS

## Modalità di accesso

---

Il servizio è accessibile sia in HTTP sia in HTTP+SSL/TLS.

## Servizi supportati

---

Sono supportati i seguenti servizi:

- Invio di fax singoli e circolari (destinazioni multiple)
- Invio di sms singoli e circolari (destinazioni multiple) ad alta qualità
- Invio di sms "bulk" circolari (destinazioni multiple)
- email broadcast
- invio di circolari fax personalizzate "mail merge" (foglio Excel per la lista Destinazioni+ documento Word)

## Funzionamento in generale

---

In generale, per richiedere alla Piattaforma di trasmettere messaggi a destinazione debbono essere effettuate tre macro operazioni, supportate da tre metodi offerti dal Gateway

Operazione	Descrizione	Metodo
Autenticazione	Necessaria a farsi riconoscere dal Gateway, che apre una sessione restituisce un "token" da usare in tutte le chiamate successive all'interno della sessione.	<code>start</code>
Caricamento Documenti da trasmettere	I Documenti da trasmettere via fax o email devono essere caricati sulla Piattaforma, oppure resi disponibili a un URI del Cliente.  Per i messaggi SMS singoli questa operazione non è necessaria, poiché il messaggio è passato direttamente come parametro di <code>sendMessage</code> .	<code>storeDocuments</code> o <code>storeDocument</code>
Richiesta di trasmissione	Richiesta di trasmettere Documenti a una Lista Destinatari, specificata nella chiamata o pre-caricata	<code>sendMessage</code>

## Note

- Come descritto nella documentazione successiva, appare che i documenti da trasmettere (tranne i messaggi SMS), possono essere:
  - pre-caricati tramite storeDocuments o storeDocument
  - resi disponibili al Gateway specificando un URI remoto
  - caricati direttamente tramite la chiamata sendMessage

Ricordiamo ai clienti che l'uso del solo metodo sendMessage, anche se apparentemente semplifica la scrittura di codice, rende più difficoltosa la diagnosi in caso di problemi. E' perciò fortemente consigliato l'uso di storeDocument/s per pre-caricare i documenti, in modo da separare il caricamento dall'effettiva trasmissione.

- I Documenti e la Lista Destinatari possono essere caricati separatamente e riutilizzati in messaggi distinti all'interno della stessa sessione (ad esempio per riutilizzare la stessa lista per spedire un documento differente)
- Per quanto riguarda i Documenti, è possibile istruire il Gateway affinché li recuperi remotamente tramite HTTP.
- Tutte le operazioni di caricamento o richiesta di trasmissione richiedono che una sessione sia stata attivata con successo, altrimenti restituiscono un errore.

## Stato Trasmissione

---

Lo Stato Trasmissione può essere recuperato chiamando opportuni metodi, specificando l'Identificativo della Trasmissione, oppure un intervallo temporale (ad esempio è possibile recuperare tutte le trasmissioni effettuate dal... al...), con un limite di tre giorni e di 1000 record per chiamata

## Call back

Endpoint del servizio SOAP implementato dal cliente e conforme al WSDL "Servizi di callback lato client" (vedi definizione WSDL), utilizzato dalla piattaforma per la trasmissione asincrona delle notifiche.

## WSDL

---

La descrizione completa delle operazioni e delle strutture dati è disponibile ai seguenti indirizzi:

Servizi esposti da KQUMSG1

<http://soap.kqumg.it/chaka/current/0.9.5/Chaka.wsdl>

Servizi di callback lato client

[http://soap.kqumg.it/chaka/current/0.9.5/Chaka\\_CB.wsdl](http://soap.kqumg.it/chaka/current/0.9.5/Chaka_CB.wsdl)

Il Gateway è disponibile anche via HTTP+SSL/TLS all'indirizzo:

<https://ssl.kqumg.it/chaka/current/0.9.5/Chaka.wsdl>

*Nota: "chaka" significa "ponte" in lingua Quechua*

## Autenticazione e gestione sessione

Per creare una nuova sessione deve essere usato il metodo `start` con i seguenti parametri:

- *userid*: la user-id generata dal Pannello di Controllo
- *password*: la password associata alla user-id di cui sopra
- *opzioni*: lista di opzioni (vedi di seguito)

## Opzioni

---

<i>contact-person</i>	un indirizzo email valido (obbligatorio). Questo indirizzo deve essere specificato in modo da avere sempre un responsabile dello sviluppo che KQI possa contattare per comunicazioni urgenti.
<i>Soap call-back</i>	indirizzo http per la notifica asincrona dello Stato Trasmissione; il servizio deve implementare sia ping sia processStatusNotifications come descritto nel Chaka_CB.wsdl. Attenzione: se specificato, il Gateway cercherà di chiamare l'operazione "ping" come parte del processo di attivazione della sessione.
<i>max-browse-block100-count</i> (non ancora implementato)	numero massimo di blocchi da 100 record restituiti dalle chiamate a <i>browseMessagesByDate</i> o <i>browseDestinationsBySeqId</i> .
<i>max-remote-operation-time</i> (non ancora implementato)	tempo massimo di completamento per recupero informazioni dal server remoto
<i>max-method-completion-time</i> (non ancora implementato)	tempo massimo di completamento per le operazioni Soap

## Comportamento

---

Se la chiamata del metodo `start` ha successo, è restituito un “token” che dovrà essere usato come parametro in tutte le operazioni successive.

Se, successivamente, la chiamata di altri metodi fallisce (ad es. perché la sessione è scaduta) ed è restituito un codice numerico (403 o 500), dovrà essere richiamato il metodo `start` ed usare il nuovo “token” nelle operazioni successive.

Per controllare la validità del “token” di sessione o determinarne la sua scadenza è disponibile il metodo `check` che restituisce un singolo elemento “expires” di tipo `xsd:dateTime` (esempio: 2001-10-26T21:32:52+02:00)

## XML: metodo start e possibile risposta

---

### Richiesta

```
<SOAP-ENV:Body xmlns:m1="urn:chaka-0.9.5" >
<m1:start>
<m1:userid>user@example.com</m1:userid>
<m1:password>password</m1:password>
<m1:options>
<m1:contact-person>mailto:develop-staff@example.com</m1:contact-person>
<m1:soap-callback>http://soap.example.com/soap-kq-callback</m1:soap-callback>
</m1:options>
</m1:start>
```

### Risposta

```
<SOAP-ENV:Body xmlns="urn:chaka-0.9.5">
<m1:startResponse>
<m1:auth>4uY29tbTIwMDAuaXQvY2dpLWJpbi9zb2FwY2Jsb2cucGwKCSA=</m1:auth>
</m1:startResponse>
</SOAP-ENV:Body>
```

## Caricare Documenti

I Documenti da trasmettere possono essere caricati usando i metodi `storeDocument` (per singoli Documenti) o `storeDocuments` (per più Documenti). Le funzioni sono altrimenti identiche, per cui descriviamo solo la seconda.

`storeDocuments` richiede di specificare due parametri:

<i>auth</i>	il token di sessione restituito dal metodo <code>start</code>
<i>sources</i>	sequenza di elementi di tipo <i>DocumentSource</i>

e restituisce una sequenza di strutture *storeDocumentResult*

- Ciascun elemento (item) della sequenza specificata in *sources* deve essere un singolo elemento di tipo *Document* o di tipo *RemoteDocument*.
- Se l'elemento è di tipo *Document* deve essere incluso (embedded) nel sotto elemento usando una codifica "base64". Il fatto che il contenuto sia codificato in base64 è dichiarato nel WSDL e normalmente gestito implicitamente dal framework SOAP. Solitamente il client non deve applicare la codifica esplicitamente
- Se invece, l'elemento è di tipo *RemoteDocument* il sotto elemento *location* specifica l'indirizzo remoto (URI) dove il Gateway recupererà il Documento. Al momento è disponibile solo tramite http.
- Attenzione: ciascun Documento deve avere un nome univoco che lo identifica durante la durata della sessione. Il nome del documento (attributo *name*) deve essere nel formato `string.extension` dove *extension* è una tra le estensioni supportate dal servizio (doc, pdf, txt, ecc.)
- Ciascun Documento caricato può essere identificato successivamente durante la sessione (di durata di almeno un'ora) tramite l'elemento *DocumentRef*. Usando un elemento di tipo *DocumentRef*, il client può specificare il nome di un documento precedentemente caricato. Come nome, occorre usare il valore usato come unique-name in fase di caricamento

Il metodo restituisce elementi di tipo *storeDocumentResult* che descrivono il risultato della chiamata e includono le seguenti informazioni per ciascun documento caricato correttamente:

<i>name</i>	il nome univoco del documento
<i>size</i>	la dimensione in bytes
<i>md5</i>	MD5 digest in formato esadecimale (hash del documento)

## XML: metodo `storeDocuments` e possibile risposta

Nell'esempio si procede al caricamento di due documenti: un pdf e un tif, di cui il primo disponibile in "remoto" e il secondo caricato tramite il metodo.

Ricordiamo che il metodo non deve essere usato per la trasmissione di SMS, perché in questo caso il testo del messaggio viene passato come parametro direttamente dal metodo `sendMessage` descritto successivamente.

### Richiesta

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <storeDocuments xmlns="urn:chaka-0.9.5" >
    <auth> [...] </auth>
    <sources>
      <source>
        <remote-document>
          <location>http://www.example.com/documents/myfax.pdf</location>
          <unique-name>myfax.pdf</unique-name>
        </remote-document>
      </source>
      <source>
        <document><unique-name>myfax.tif</unique-name>
        <data>
          SUKqABQSAQAAKJsVAGWzAqBsVgCUzQqAslkBUDYrAMpmBUDZrAAomxUAZbMCoGxWAJTNCy
          WQFQNIaYmYFQNmsACibFQBlswKgbFYAlMOKgIJZAVA2KwDKZgVA2awAKJsVAGWzAqBsVgCU
          ICAwNjIzMzI3Nzc3ICBGcmkgTm92IDE5IDE2OjU0OjE3IDlwMDQKAAAAGBkaACAAAACAGAAA
          IAA=
        </data>
      </document>
    </source>
  </sources>
</storeDocuments>
</SOAP-ENV:Body>
```

### Risposta

```
<SOAP-ENV:Body xmlns:m1="urn:chaka-0.9.5">
  <m1:storeDocumentsResponse>
    <m1:stored-list>

      <m1:stored>
        <m1:name>myfax.pdf</m1:name>
        <m1:md5>4cf791b5c9d0b1e4627b0130a05a9a69</m1:md5>
        <m1:size>98891</m1:size>
      </m1:stored>
      <m1:stored>
        <m1:name>myfax.tif</m1:name>
        <m1:md5>375992b51f242f46ae284c72298bd928</m1:md5>
        <m1:size>70472</m1:size>
      </m1:stored>
    </m1:stored-list>
  </m1:storeDocumentsResponse>
</SOAP-ENV:Body>
```

## Richiedere una trasmissione

Una volta aperta una sessione e caricati Documenti è possibile richiedere al Gateway di iniziare una Trasmissione tramite il metodo `sendMessage`.

Il metodo richiede i seguenti parametri, descritti di seguito.

<i>auth</i>	token di sessione, ottenuto tramite il metodo <code>start</code> (vedi sezione Autenticazione)
<i>request-uid</i>	codice specificato dal client per gestire interruzioni di connessione con il gateway
<i>message</i>	la richiesta di trasmissione

Il metodo restituisce una struttura che include un identificativo univoco della trasmissione, generato dalla Piattaforma, insieme ad altre informazioni quali il numero di destinazioni e il numero di destinazioni rifiutate. Inoltre, il campo booleano `cached-result` dice se il messaggio è stato effettivamente elaborato oppure è stato restituito il risultato di una richiesta precedente associata allo stesso `request-uid` (gestione ritrasmissioni accidentali)]

### request-uid

Questo attributo è utile per gestire situazioni nelle quali la connessione tra il client e il gateway s'interrompe prima che il client possa ricevere la risposta del server.

In questi casi, quando una chiamata fallisce a causa di problemi di rete il client può (e dovrebbe) ri-sottoporre la richiesta usando lo stesso identificativo.

In questi casi, se la chiamata precedente è stata presa in carico con successo, il gateway restituirà un messaggio di "cached" e non genererà un duplicato della richiesta di trasmissione.

### message

Le componenti principali sono

<i>subject</i>	l'oggetto (subject), del messaggio. (Utilizzato per riconoscere il messaggio nel sito di Status)
<i>options</i>	Le opzioni di trasmissione (flags, scheduling e cost-center)
<i>fax / sms / umail / faxsmmerge / bulksms</i>	elementi che rappresentano la trasmissione da effettuare, come descritti di seguito



### Trasmissione sms o bulksms (per bulksms s'intendono gli sms a bassa qualità)

<i>destinations</i>	Sequenza di elementi di tipo <i>Destination</i> .
(In alternativa a destinations) <i>destination-list-name</i>	Nome di una Lista Destinazioni pre-caricata (vedere Caricamento di Lista Destinazioni)
<i>short-msg</i>	testo del messaggio SMS Attenzione: le caratteristiche dell'alfabeto usato e il loro impatto sulla tariffazione del messaggio sono descritte nella documentazione contrattuale. In breve, il set di caratteri usato (standard della trasmissione sms), utilizza due caratteri per rappresentare alcuni simboli, per cui un messaggio apparentemente più corto di 160 caratteri può risultare più lungo, con il conseguente impatto sulla tariffazione.

### Trasmissione fax

<i>destinations</i>	Sequenza di elementi di tipo <i>Destination</i> .
(In alternativa a destinations) <i>destination-list-name</i>	Nome di una Lista Destinazioni pre-caricata (vedere Caricamento di Lista Destinazioni)
<i>documents</i>	Sequenza di elementi di tipo <i>Document</i> o <i>DocumentRef</i>

### Email broadcast

<i>destinations</i>	Sequenza di elementi di tipo <i>Destination</i> .										
(In alternativa a destinations) <i>destination-list-name</i>	Nome di una Lista Destinazioni pre-caricata (vedere Caricamento di Lista Destinazioni)										
<i>documents</i>	Sequenza di elementi di tipo <i>Document</i> o <i>DocumentRef</i>										
<i>options</i>	Opzioni <table border="1" data-bbox="486 1529 1469 1731"> <tr> <td><i>embed-files</i></td> <td>se "true", i messaggi generati includono documenti e immagini, se "false", sono conservati sui server e resi disponibili come collegamenti ipertestuali.</td> </tr> <tr> <td><i>from-name</i></td> <td>Display name usato nel messaggio generato</td> </tr> <tr> <td><i>from-email</i></td> <td>Indirizzo email del campo "from" nel messaggio generato</td> </tr> <tr> <td><i>subject</i></td> <td>L'oggetto del messaggio</td> </tr> <tr> <td><i>no-receipt-request</i></td> <td>se "true", disabilita la richiesta di ricevuta nei messaggi generati</td> </tr> </table>	<i>embed-files</i>	se "true", i messaggi generati includono documenti e immagini, se "false", sono conservati sui server e resi disponibili come collegamenti ipertestuali.	<i>from-name</i>	Display name usato nel messaggio generato	<i>from-email</i>	Indirizzo email del campo "from" nel messaggio generato	<i>subject</i>	L'oggetto del messaggio	<i>no-receipt-request</i>	se "true", disabilita la richiesta di ricevuta nei messaggi generati
<i>embed-files</i>	se "true", i messaggi generati includono documenti e immagini, se "false", sono conservati sui server e resi disponibili come collegamenti ipertestuali.										
<i>from-name</i>	Display name usato nel messaggio generato										
<i>from-email</i>	Indirizzo email del campo "from" nel messaggio generato										
<i>subject</i>	L'oggetto del messaggio										
<i>no-receipt-request</i>	se "true", disabilita la richiesta di ricevuta nei messaggi generati										

### Trasmissione faxmsmerge (doc)

<i>destinations</i>	Sequenza di elementi di tipo <i>Destinations</i> . Attenzione: per trasmissioni dello stesso messaggio a più Destinazioni è necessario l'uso della Lista Destinazioni.
(In alternativa a <i>destinations</i> ) <i>destination-list-name</i>	Nome di una Lista Destinazioni pre-caricata (vedere Caricamento di Lista Destinazioni)
<i>doc</i>	<i>Document</i> o <i>DocumentRef</i> di un documento in formato .doc

## destinations

Il parametro *destinations* consiste in una sequenza di elementi in cui ciascuno include i seguenti campi:

<i>address</i>	La destinazione a cui inviare il Documento o nel caso SMS, il messaggio. Ad esempio, un numero telefonico o un indirizzo email.
<i>cc1</i>	Informazione addizionale, restituita nelle eventuali richieste di stato del messaggio. Ad es. un codice cliente o un'altra informazione utile a riconoscere il destinatario.
<i>cc2</i>	Come <i>cc1</i>
<i>extra</i>	Insieme di 10 attributi testo liberi da usare per personalizzare un messaggio di email broadcast. In pratica sono i valori da sostituire ai "placeholder" del documento HTML caricato.

A ciascuna destinazione nella lista è assegnato dal sistema un unico identificativo sequenziale che rappresenta la posizione della destinazione nella sequenza, a partire da 1. Tale valore sarà restituito come *seq-id* dalla chiamata *browseDestinationsBySeqId*

## Importante

La Piattaforma gestisce in modo differente i messaggi singoli dai messaggi circolari, poiché per i secondi implementa procedure che permettono di ottimizzare il carico dei sistemi e assicurare un buon tasso di smaltimento a tutti i clienti.

E' perciò deprecato e può essere causa di immediata sospensione del servizio, in quanto, come previsto da contratto danneggia le prestazioni anche per altri clienti, l'uso di chiamate ripetute *sendMessage* per trasmettere lo stesso documento a destinazioni singole differenti, invece della procedura corretta di usare una singola chiamata *sendMessage* specificando gli indirizzi multipli nei parametri *destinations* o *destinations-list-name*.

Tale pratica non assicura tempi di evasione più veloci, anzi, normalmente peggiora le prestazioni della Piattaforma, ad esempio poiché il rendering e i controlli sono effettuati su ogni messaggio trasmesso.

## XML: metodo sendMessage e possibile risposta

---

### Richiesta

```
<SOAP-ENV:Body xmlns="urn:chaka-0.9.5" >

  <sendMessage>
    <auth> [ auth string ] </auth>
    <request-uid>0123456789890123456789</request-uid>
    <message >
      <subject>Test message (via SOAP)</subject>
      <fax>
        <destinations>
          <item><address>0270030070</address><cc1>code1A</cc1><cc2>code1B</cc2>
          </item>
        </destinations>
        <documents>
          <item>
            <document-ref><unique-name>myfax.pdf</unique-name></document-ref>
          </item>
        </document>
      </fax>
    </message>
  </sendMessage>
```

### Risposta

```
<SOAP-ENV:Body xmlns:m1="urn:chaka-0.9.5">
  <m1:sendMessageResponse>
    <m1:result>
      <m1:request-uid>0123456789890123456789</m1:request-uid>
      <m1:cached-result>false</m1:cached-result>
      <m1:info> <m1:msg-id>13283256</m1:msg-id> </m1:info>
    </m1:result>
  </m1:sendMessageResponse>
</SOAP-ENV:Body>
```

## Caricare una lista Destinazioni

Per inviare messaggi circolari (lo stesso messaggio o documento per più destinazioni), dati i limiti del parametro *destinations* del metodo [sendMessage](#) , è necessario caricare una Lista Destinazioni tramite il metodo [storeDestinationsList](#), che in caso di successo restituisce una struttura contenente il nome univoco (*name*) e il numero di record importati correttamente.

Le liste caricate sono disponibili solo all'interno della stessa sessione e devono essere usate nel metodo [sendMessage](#) tramite il parametro *destination-list-name*.

Il metodo richiede i seguenti parametri

<i>Unique-name</i>	Nome univoco della lista
<i>Table list</i>	Un elemento di tipo <i>DestinationTable</i> oppure una sequenza di elementi di tipo <i>Destination</i>

## Struttura di DestinationTable

<i>address-column, cc1-column, cc2-column</i>	Nomi delle colonne che ospitano i campi indirizzo, cc1 e cc2 che sono utili per la trasmissione del messaggio. Il server deve sapere quali sono le colonne dove trovare ad es. il numero telefonico cui trasmettere il messaggio e i dati da riportare nelle risposte alle richieste di stato.
<i>columns</i>	Stringa di tipo <code>xsd:NMTOKENS</code> che rappresenta i nomi dei campi della tabella. Se non è specificato, si assume che la prima riga della tabella funga da intestazione.
<i>format</i>	Formato dei dati, attualmente uno dei seguenti: semicolon- csv, comma- csv, tab- csv, json- array
<i>data</i>	Righe della tabella (codifica base64)
<i>expected_count</i>	Opzionale. Numero di righe che il client si aspetta siano lette e acquisite dal gateway. Se il conteggio non corrisponde, viene restituito un errore.

Quando una tabella di questo tipo è importata, la tabella è trasformata in un elenco di record. A ciascun campo di ciascun record è assegnato un nome in base al valore contenuto nel parametro *columns* oppure, in sua assenza, in base al contenuto della prima riga.

Quindi, l'indirizzo e i campi cc1 e cc2 sono selezionati in base ai nomi specificati in *address-column, cc1-column, cc2-column*: se almeno uno di questi nomi di colonna non è specificato, è restituito un errore e la lista non è importata.

## XML: metodo storeDestinationsList

```
<SOAP-ENV:Body xmlns:m1="urn:chaka-0.9.5">
<m1:storeDestinationsList>
  <m1:auth> [ auth string ] </m1:auth>
  <m1:list>
    <m1:unique-name>phone-list-1</m1:unique-name>
    <m1:table>
      <m1:address-column>phone</m1:address-column>
      <m1:cc1-column>name</m1:cc1-column>
      <m1:cc2-column>company</m1:cc2-column>
      <m1:columns>
```

```

name company address1 address2 city zip state fax phone email
</m1:columns>
<m1:format>semicolon-csv</m1:format>
<m1:data>
<!-- base64 data -->
</m1:data>
</m1:table>
</m1:list>
</m1:storeDestinationList>

```

## Recuperare lo stato di una trasmissione

Lo stato di una trasmissione può essere recuperato tramite due metodi

<p><u><a href="#">browseDestinationsBySeqId</a></u></p>	<p>Permette di recuperare gli stati di tutti i messaggi aventi numero sequenziale compreso tra i parametri <i>first</i> e <i>last</i> della lista destinazioni di una trasmissione avente un determinato “<i>msg-id</i>”.</p> <p>Ciascuno stato è rappresentato da un elemento di tipo <i>DestinationDetail</i>, che include tra le altre informazioni anche la stringa di stato (<i>status</i>), un timestamp (<i>time</i>) e il numero sequenziale (<i>seq-id</i>)</p>										
<p><u><a href="#">browseMessagesByDate</a></u></p>	<p>Permette di recuperare gli stati di tutti i messaggi inviati in un certo periodo, definito dai parametri <i>since</i> e <i>until</i>. Restituisce una struttura che include una sequenza di elementi di tipo <i>MessageDetail</i>, ciascuno dei quali comprende:</p> <table border="1" data-bbox="541 1229 1481 1552"> <tr> <td><i>msg-id</i></td> <td>Identificativo unico del messaggio</td> </tr> <tr> <td><i>subject</i></td> <td>L’oggetto del messaggio</td> </tr> <tr> <td><i>creation time</i></td> <td>Data ora di creazione</td> </tr> <tr> <td><i>dest-count</i></td> <td>Numero di destinazioni</td> </tr> <tr> <td><i>completed-dest-count</i></td> <td>Numero di destinazioni che hanno raggiunto uno stato finale o pseudo-finale (chiarire)</td> </tr> </table> <p>In più, è incluso il dettaglio della prima destinazione (<i>first-dest</i>), in modo da risparmiare una chiamata a <i>browseDestinationsBySeqId</i> nel caso in cui la destinazione sia una sola.</p>	<i>msg-id</i>	Identificativo unico del messaggio	<i>subject</i>	L’oggetto del messaggio	<i>creation time</i>	Data ora di creazione	<i>dest-count</i>	Numero di destinazioni	<i>completed-dest-count</i>	Numero di destinazioni che hanno raggiunto uno stato finale o pseudo-finale (chiarire)
<i>msg-id</i>	Identificativo unico del messaggio										
<i>subject</i>	L’oggetto del messaggio										
<i>creation time</i>	Data ora di creazione										
<i>dest-count</i>	Numero di destinazioni										
<i>completed-dest-count</i>	Numero di destinazioni che hanno raggiunto uno stato finale o pseudo-finale (chiarire)										

## Notifiche asincrone

E’ possibile richiedere che tutte le notifiche di stato per un dato messaggio siano notificate in modo asincrono a un servizio http del cliente che esponga l’interfaccia appropriata e che implementi le

operazioni SOAP descritte nel [http://soap.kqumg.it/chaka/current/0.9.5/Chaka\\_CB.wsdl](http://soap.kqumg.it/chaka/current/0.9.5/Chaka_CB.wsdl) e che sia accessibile sulle porte 80, oppure 8080, oppure su una porta nell'intervallo 40000-50000.

Il servizio Chaka\_CB (CB = callback) include due operazioni:

- Ping: chiamato dal gateway durante l'attivazione della sessione
- processStatusNotifications, chiamato quando nuove informazioni di stato sono disponibili

L'URI del servizio è passato al Gateway tramite l'opzione *soap-callback* del metodo `start` (consultare "Autenticazione e gestione della sessione") e il server tenterà di chiamare l'operazione ping prima di terminare. Se il processo termina con successo, tutte le trasmissioni richieste successivamente saranno connesse all'URI di callback, altrimenti è restituito un codice di errore

Sia ping, sia processStatusNotifications sono metodi "one way". La loro implementazione deve restituire una risposta http con un "body" vuoto e uno status-code che indichi successo (200-299).

Qualora le chiamate a processStatusNotifications falliscano, le notifiche saranno conservate e la chiamata sarà ritentata successivamente

Ricordiamo che non è necessario realizzare un servizio SOAP completo per usare questa funzionalità: un semplice web server e un parser XML sono sufficienti, dato che non debbono essere generate risposte.

Esempio di processStatusNotifications (SOAP-ENV:Body part):

```
<SOAP-ENV:Body xmlns:m2="urn:chaka-callback-0.9.5" xmlns:m1="urn:chaka-0.9.5" >
  <m2:processStatusNotifications>
    <m2:notif-list>
      <m2:notif>
        <m2:msg-id>13283256</m2:msg-id>
        <m2:request-uid>0123456789890123456789</m2:request-uid>
        <m2:dest-count>992</m2:dest-count>
        <m2:completed-dest-count>992</m2:completed-dest-count>
        <m2:subject>Test message (via SOAP)</m2:subject>
        <m2:dest>
          <m1:seq-id>1</m1:seq-id>
          <m1:address>0270030070</m1:address>
          <m1:cc1>code1A</m1:cc1>
          <m1:cc2>code1B</m1:cc2>
          <m1:status>OK</m1:status>
          <m1:time>2006-11-20T10:24:40Z</m1:time>
          <m1:pages>1</m1:pages>
        </m1:dest>
      <m2:notif>
    </m2:notif-list>
  </m2:processStatusNotifications>
</SOAP-ENV:Body>
```

## Errori

A fronte di un errore è generata una risposta SOAP-ENV:Fault, come nell'esempio.

```
<SOAP-ENV:Fault>  
  <faultcode>SOAP-ENV:Client</faultcode>  
  <faultstring>403 AUTH_BAD_CREDENTIALS</faultstring>  
  <detail>Invalid or expired 'auth' token</detail>  
</SOAP-ENV:Fault>
```

## Assistenza

Richieste di assistenza e segnalazioni di problemi debbono essere inviati via email all'indirizzo:

<soap-support@kqumg.it>.

Attenzione, saranno prese in considerazione richieste includano un log completo di tutti i messaggi SOAP rilevanti inviati e ricevuti (inclusivi di HTTP headers e XML body)

Non è inclusa la consulenza sullo sviluppo di applicazioni.

## Integrazioni funzionanti

Sono state testate integrazioni con i seguenti ambienti:

- Perl + SOAP::Simple
- Python + NuSOAP/0.7.2
- PHP-SOAP/5.2.2
- 4D built-in SOAP Client
- Axis/1.4
- Java/1.5.0\_06
- C#/.NET