



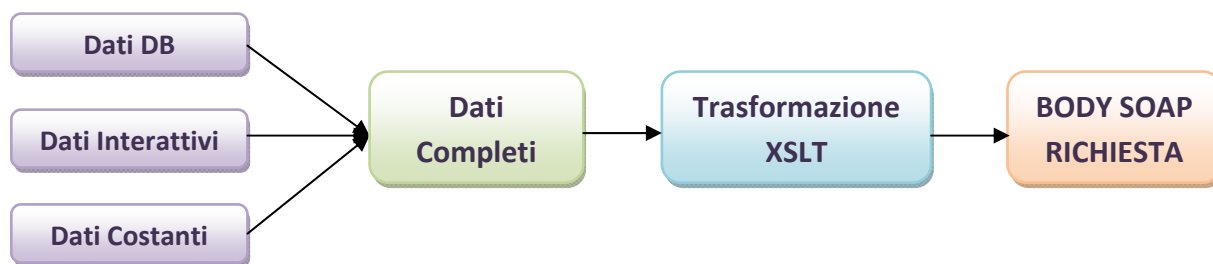
Guida tecnica di sil-vio 1.0

Sommario

1.	Creazione Nuova Istanza	2
1.1.	Configurazione Dati Richiesta Fruitore.....	2
1.1.1.	Scenario Payload Costante	2
1.1.2.	Scenario acquisizione dati	2
1.1.3.	Definizione trasformazione richiesta.....	3

1. Creazione Nuova Istanza

1.1. Configurazione Dati Richiesta Fruitore



1.1.1. Scenario Payload Costante

Per lo scenario in questione è necessario fornire a SIL-VIO il *body* del messaggio SOAP della richiesta. Questo messaggio sarà utilizzato dall'istanza del fruitore generata automaticamente da SIL-VIO per inviare la richiesta.

Per facilitare la generazione del messaggio, l'applicativo fornisce lo schema XSD a cui il messaggio dovrà essere conforme, elaborando i WSDL (e il file dei tipi ad esso associato) forniti in fase di configurazione dell'accordo di collaborazione. La generazione di un messaggio di esempio a partire da un XSD può essere realizzata automaticamente attraverso strumenti opensource (ex. soapUI¹) o commerciali (ex. Altova XMLSpy®²).

Inoltre è possibile utilizzare l'apposito comando per validare il messaggio.

1.1.2. Scenario acquisizione dati

In questo scenario è possibile simulare in maniera più realistica un reale caso d'uso. Infatti, la richiesta del fruitore sarà generata con dati prelevati dal db, o con dati da richiedere all'utente, o con campi costanti. [COMBINARE]

Prima di procedere è utile conoscere quali sono i dati che dovranno essere inseriti all'interno del messaggio di richiesta. Un modo possibile è quello di generare un messaggio di esempio con gli strumenti indicati in precedenza, e localizzare la giusta provenienza di ciascun dato.

Per prelevare i **dati dal db**, bisogna fornire la SELECT che il sistema utilizzerà per estrarre i dati. Nella maggior parte dei casi è conveniente utilizzare una query parametrica

```
SELECT * FROM tabella WHERE tabella.campo = ?
```

In fase di esecuzione dell'istanza, sarà richiesto all'utente di completare tutti i campi parametrizzati.

N.B: E' necessario specificare nella clausola FROM tutte le tabelle da cui prelevare i dati.

I **dati interattivi** sono dati che verranno richiesti all'utente ad ogni esecuzione dell'istanza.

E' inoltre possibile specificare dei **dati costanti**, che non variano ad ogni esecuzione dell'istanza.

¹ <http://www.soapui.org/>

² <http://www.altova.com/XMLSpy>

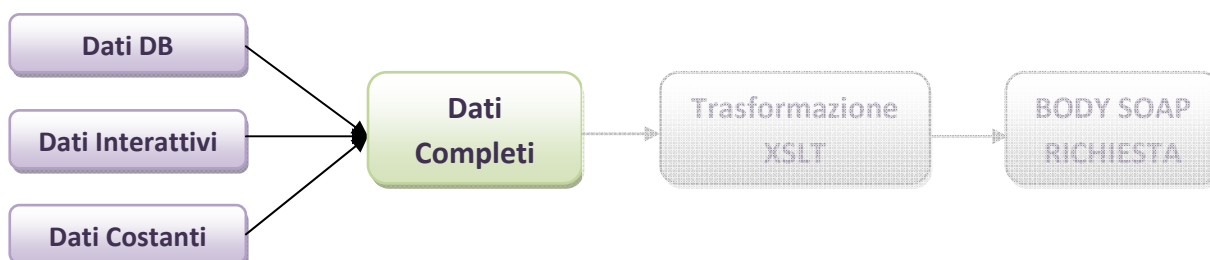
Esempio concreto di SegnalazioneFruizionePrestazioni del task AP1.

Campi della richiesta	Valori
IdentificazioneRegioneMittente	170
...	...
IdentificazioneRegioneDestinatario	100
...	...
CodiceFiscaleAssistito	AAAAAA00A00A000A
ImportoTicket	100,00

La regione mittente è un dato che non varia ad ogni esecuzione dell'istanza → **Dato Costante**

La regione del destinatario è un dato che varia ad ogni esecuzione dell'istanza → **Dato Interattivo**

I dati del ricovero sono prelevati dalla base di dati → **Dato da DB**

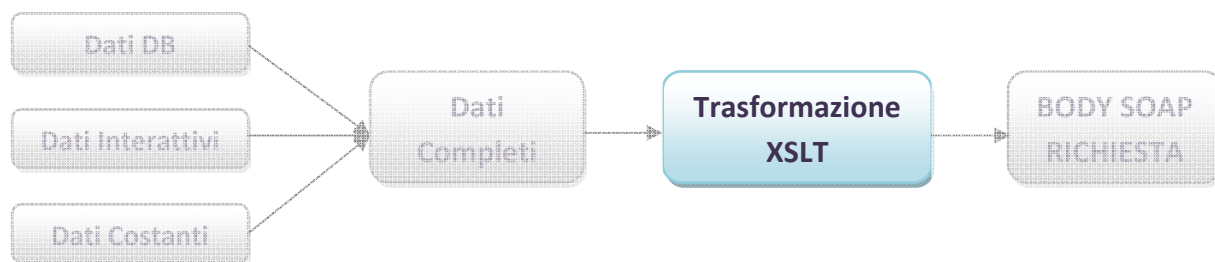


1.1.3. Definizione trasformazione richiesta

Una volta definiti tutti i dati che comporranno il messaggio di richiesta, è necessario definire come questi andranno a comporre il body del messaggio SOAP.

Per fare ciò bisogna definire un file XSLT che permette di trasformare il documento XML contenente i dati completi generato da SIL-VIO nella richiesta da inviare all'erogatore. Anche in questo caso il sistema offre supporto alla creazione del XSLT, fornendo gli schemi XSD sorgente e destinazione (dati completi → body richiesta). Per generare il file XSD della sorgente a partire dai dati da estrarre dal db, il sistema effettua la query fornita dall'utente escludendo la clausola WHERE. Il risultato della query verrà utilizzato per fornire la lista dei campi nel formato **tabella.campo**. Nel caso in cui la query non dovesse produrre alcun risultato, il sistema non sarà in grado di generare in maniera corretta il file XSD della sorgente. Ovviamente, l'utente può scrivere il file di trasformazione qualora conosca già i campi da inserire nella richiesta.

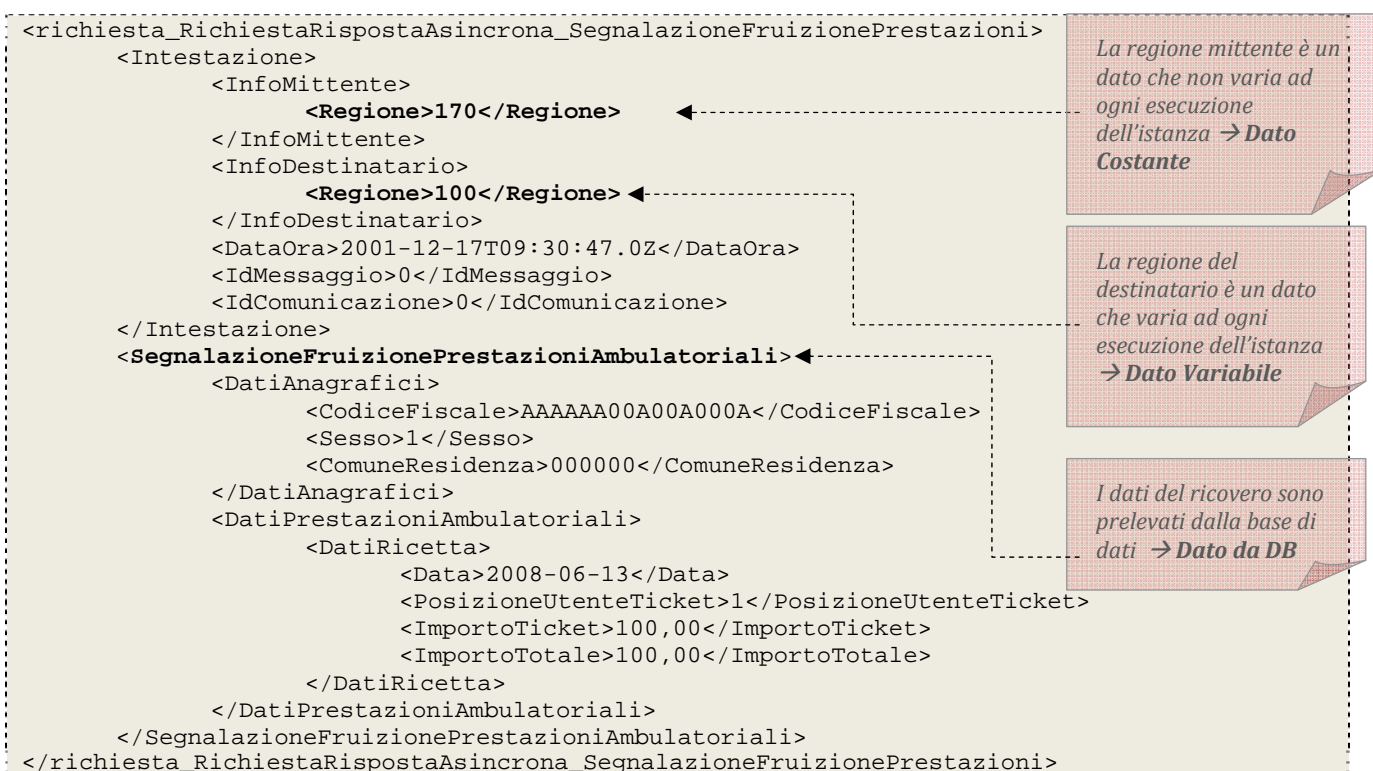
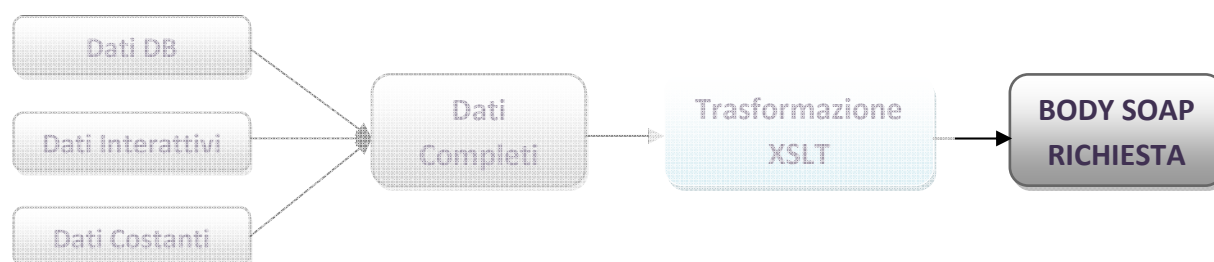
Le trasformazioni XSLT possono essere realizzate manualmente o con l'ausilio di strumenti automatici Open Source (Jamper³) o commerciali (Altova MapForce®⁴)



Nel caso di scenario con Payload Costante, questo passaggio non è previsto.

1.1.1. Creazione messaggio SOAP

A questo punto il sistema possiede tutti gli elementi per poter creare il messaggio body della richiesta. Di seguito ne è riportato un esempio.



1.1.2. Elaborazione della risposta

Una volta inviato il messaggio SOAP all'erogatore, il fruitore resta in attesa della risposta. Questa può contenere dati che il fruitore potrebbe decidere di salvare in una propria base di dati.

Per fare ciò, il sistema necessita di un file di trasformazione XSLT che permetta di recuperare i dati dalla risposta dell'erogatore e contemporaneamente creare la query per effettuarne l'inserimento nella base di dati. Anche in questo caso il sistema propone lo schema della risposta dell'erogatore.

Per produrre il file della trasformazione non è presente alcuno strumento sia proprietario che open source che possa offrire una qualsiasi forma di supporto. Questo comporta la scrittura manuale del file da parte dell'utente.



Esempio di risposta da parte dell'erogatore

Ecco un esempio di file XSLT che elabora la risposta dell'erogatore

```
<xsl:template match="/">
  INSERT INTO paziente (nome, cognome, codice_fiscale, data_nascita, sesso,
comune_residenza, codice_ssr) VALUES(
    <!-- elemento radice -->
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/DatiAnagraficiBase/Nome"/>',
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/DatiAnagraficiBase/Cognome"/>',
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/CodiceFiscale"/>',
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/DataDiNascita"/>',
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/Sesso"/>',
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/ComuneResidenza"/>',
    ' <xsl:apply-templates
select="/soapenv:Envelope/soapenv:Body/risposta_RichiestaRispostaSincrona_IdentificazioneAs
sistito/DatiAnagraficiAssistito/CodiceSSR"/>'
    );
</xsl:template>
```

L'utente dunque deve fornire la query per l'inserimento dei valori nella base di dati. Per poter estrarre i dati dalla richiesta viene utilizzata la sintassi XPATH attraverso la quale viene prelevato il valore di un elemento esplorando l'intero documento a partire dalla radice.

In alcuni casi, è possibile che dalla richiesta arrivino informazioni che verranno salvate su tabelle differenti dello stesso database e che sono correlate tra loro attraverso un identificatore (ad esempio, nel caso precedente la correlazione medico-paziente attraverso chiave esterna). Attraverso il file XSLT non è possibile specificare in alcun modo questo tipo di correlazione, né è possibile scrivere le query poiché non si conosce il valore delle chiavi, solitamente generate dal dbms.

ID	Cognome	Nome	Codice_Fiscale
01	Rossi	Mario	Rssmra56945poil
02	Verdi	Chiara	Vrdchr89034jilk
03	Bianchi	Enrico	Bncnrc78oagaet
...	...		

Tabella Medico

ID	Cognome	Nome	Medico_id
01	Celeste	Rossella	01
02	Colli	Andrea	02
03	Felice	Rocco	01
...	...		

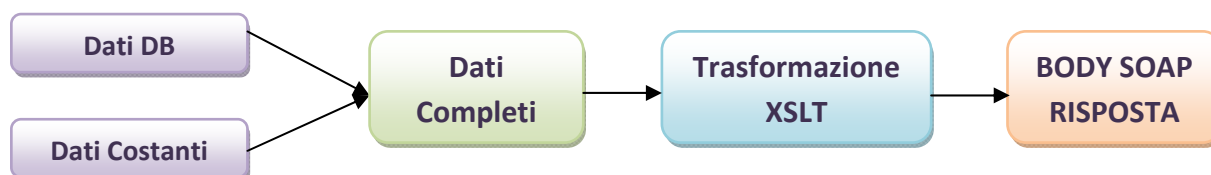
Tabella Paziente

Per superare il problema delle INSERT multiple il sistema offre la possibilità di utilizzare i *placeholder* attraverso una semplice sintassi. Qualora si voglia delegare la gestione dell'identificativo al sistema, basta utilizzare il placeholder **#{id.tabella}**; per correlare invece ennuple di tabelle differenti è necessario utilizzare il placeholder **#{refid.tabella}**.

1.1.3. Semantica di $\{id.tabella\}$ e $\{refId.tabella\}$

Per generare l'ID da attribuire alla query fornita dall'utente, il sistema effettua una SELECT sulla tabella in questione calcolando il MAX degli ID presenti; successivamente incrementa questo valore, lo attribuisce alla query fornita e la esegue. Se la query non va a buon fine, il sistema presuppone che siano stati violati alcuni vincoli di chiave; per ovviare a ciò, il sistema effettua una SELECT utilizzando tutti i campi a disposizione per verificare la reale presenza della ennupla nella tabella e ne estrae l'ID. Se la selezione restituisce una ennupla vuota o un risultato errato, il sistema solleva un'eccezione. In caso di esito positivo, l'ID estratto viene salvato ed eventualmente sostituito al posto del palceholder $\{refId.tabella\}$ qualora previsto.

1.2. Configurazione Dati Risposta Erogatore



1.2.1. Scenario Payload Costante

Per lo scenario in questione è necessario fornire a SIL-VIO il *body* del messaggio SOAP della risposta. Questo messaggio sarà utilizzato dall'istanza dell'erogatore generata automaticamente da SIL-VIO per inviare la risposta.

Per facilitare la generazione del messaggio, l'applicativo fornisce lo schema XSD a cui il messaggio dovrà essere conforme, elaborando i WSDL(e il file dei tipi ad esso associato) forniti in fase di configurazione dell'accordo di collaborazione. La generazione di un messaggio di esempio a partire da un XSD può essere realizzata automaticamente attraverso strumenti opensource (ex. soapUI⁵) o commerciali (ex. Altova XMLSpy^{® 6}).

Inoltre è possibile utilizzare l'apposito comando per validare il messaggio.

1.2.2. Elaborazione della risposta

L'erogatore che riceve una richiesta da parte del fruitore può decidere di elaborarla al fine di estrapolarne dei dati. Anche in questo caso l'erogatore necessita di un file XSLT nel quale viene fornita la INSERT per l'aggiornamento del proprio db. Per la scrittura di tale file non è presente alcuno strumento software in grado di generarlo automaticamente, ma è compito dell'utente scrivere a mano il file in questione. Il sistema fornisce soltanto il supporto per le INSERT multiple attraverso l'utilizzo dei *placeholder*.

⁵ <http://www.soapui.org/>

⁶ <http://www.altova.com/XMLSpy>

