



**NETMEDICA  
ITALIA**

***Guida ai Web Service***

## Sommario

<i>Introduzione</i> .....	1
<i>Informazioni accesso Web Service</i> .....	1
<i>Schema Web Service</i> .....	2
<i>Servizi di Accesso</i> .....	3
login .....	3
loginGuardiaMedica .....	4
logout .....	4
<i>Servizi di scrittura</i> .....	5
write .....	5
write_allergia .....	11
write_anagrafica .....	11
write_certificato .....	13
write_contatto .....	14
write_diario .....	14
write_esenzione .....	15
write_esito .....	16
write_farmaco .....	17
write_monitoraggio .....	18
write_patologia .....	19
write_periodoassistenza .....	19
write_prescrizione .....	20
write_ricovero .....	21
write_scadenziario .....	22
write_stiledivita .....	23
write_vaccinazione .....	23
expressfolder .....	24
<i>Servizi di consultazione</i> .....	26
search_pazienti .....	26
search_pazienti_small .....	28
search .....	30
search_allergie .....	35
search_certificati .....	36

# Manuale Web Service Netmedica Italia

---

search_contatti .....	37
search_diari.....	38
search_esenzioni .....	39
search_esiti.....	40
search_farmaci.....	41
search_gruppo .....	42
search_monitoraggi .....	42
search_num_paziente .....	43
search_patologie.....	44
search_periodiassistenza.....	45
search_prescrizioni .....	46
search_ricoveri.....	47
search_scadenziari .....	48
search_status_paziente .....	48
search_stilidivita .....	49
search_vaccinazioni .....	50
<i>Altri servizi</i> .....	52
elenco_Referti.....	52
invio_RefertoPaziente.....	52
lettura_RefertoPaziente.....	53
read_notifiche .....	54
delete_record .....	55
<i>Codice</i> .....	56

## *Introduzione*

Di seguito riportiamo una descrizione dei servizi offerti dal sistema Web Service sviluppati da NetmedicaItalia. Dopo una breve descrizione del sistema, verranno presentati i singoli servizi offerti, per una migliore consultazione abbiamo raggruppato i Web Service per funzionalità.

## *Informazioni accesso Web Service*

Riportiamo i parametri per accedere all'area di test prevista da Netmedica Italia per il testing delle applicazioni, la struttura della wsdl principale ovviamente è derivata da quella della palestra come descritto sotto:

Web Service: FIMMGwsdl

**Target Namespace:**urn:FIMMGwsdl

**Port** FIMMGwsdlPort

**Location:**<http://cloudtest.fimmg.org/wsdl.php>

**Protocol:**SOAP

**Default style:**rpc

**Transport protocol:**SOAP over HTTP

Per accedere ad un servizio ogni utente dovrà disporre di un nome utente e password (il rilascio avviene iscrivendosi al portale [www.netmedicaitali.it](http://www.netmedicaitali.it)).

Un secondo livello di accesso è regolato da una chiave rilasciata da Netmedica Italia, tale chiave **keycartella** viene rilasciata per avere un controllo più specifico sulle operazioni compiute dall'utente collegato.

Le **keycartella** vengono rilasciate in base alle esigenze dell'entità che ne richiede il rilascio, ad ogni **keycartella** è consentito utilizzare uno o più servizi offerti, questo ha un ovvio riflesso sul controllo della sicurezza di accesso da parte dei programmi alla consultazione/scrittura nel nostro Database(rimandiamo al documento specifico per la struttura dello stesso)

Utente prova :                   test1

Password prova:               pwdtest1

keycartella test:               07da8ae79afc3bfoe58b72bd165ofdc1

## Schema Web Service

FIMMGwsdlPortType	
▶	expressfolder
▶	login
▶	loginGuardiaMedica
▶	logout
▶	search_gruppo
▶	search_num_paziente
▶	search
▶	search_pazienti
▶	search_periodiassistenza
▶	search_contatti
▶	search_esenzioni
▶	search_stilidivita
▶	search_prescrizioni
▶	search_farmaci
▶	search_vaccinazioni
▶	search_certificati
▶	search_monitoraggi
▶	search_esiti
▶	search_patologie
▶	search_pazienti_small
▶	search_status_paziente
▶	search_diari
▶	search_allergie
▶	search_ricoveri
▶	search_scadenziari
▶	write
▶	write_periodoassistenza
▶	write_anagrafica
▶	write_contatto
▶	write_esenzione
▶	write_allergia
▶	write_stiledivita
▶	write_ricovero
▶	write_prescrizione
▶	write_farmaco
▶	write_vaccinazione
▶	write_certificato
▶	write_monitoraggio
▶	write_esito
▶	write_patologia
▶	write_diario
▶	write_scadenziario
▶	invio_RefertoPaziente
▶	elenco_RefertiPaziente
▶	lettura_RefertoPaziente
▶	read_notifiche
▶	delete_record

FIMMGwsdlBinding		soap	rpc
transport: <a href="http://schemas.xmlsoap.org/soap/http">http://schemas.xmlsoap.org/soap/http</a>			
▶	expressfolder	rpc	▼
▶	login	rpc	▼
▶	loginGuardiaMedica	rpc	▼
▶	logout	rpc	▼
▶	search_gruppo	rpc	▼
▶	search_num_paziente	rpc	▼
▶	search	rpc	▼
▶	search_pazienti	rpc	▼
▶	search_periodiassistenza	rpc	▼
▶	search_contatti	rpc	▼
▶	search_esenzioni	rpc	▼
▶	search_stilidivita	rpc	▼
▶	search_prescrizioni	rpc	▼
▶	search_farmaci	rpc	▼
▶	search_vaccinazioni	rpc	▼
▶	search_certificati	rpc	▼
▶	search_monitoraggi	rpc	▼
▶	search_esiti	rpc	▼
▶	search_patologie	rpc	▼
▶	search_pazienti_small	rpc	▼
▶	search_status_paziente	rpc	▼
▶	search_diari	rpc	▼
▶	search_allergie	rpc	▼
▶	search_ricoveri	rpc	▼
▶	search_scadenziari	rpc	▼
▶	write	rpc	▼
▶	write_periodoassistenza	rpc	▼
▶	write_anagrafica	rpc	▼
▶	write_contatto	rpc	▼
▶	write_esenzione	rpc	▼
▶	write_allergia	rpc	▼
▶	write_stiledivita	rpc	▼
▶	write_ricovero	rpc	▼
▶	write_prescrizione	rpc	▼
▶	write_farmaco	rpc	▼
▶	write_vaccinazione	rpc	▼
▶	write_certificato	rpc	▼
▶	write_monitoraggio	rpc	▼
▶	write_esito	rpc	▼
▶	write_patologia	rpc	▼
▶	write_diario	rpc	▼
▶	write_scadenziario	rpc	▼
▶	invio_RefertoPaziente	rpc	▼
▶	elenco_RefertiPaziente	rpc	▼
▶	lettura_RefertoPaziente	rpc	▼
▶	read_notifiche	rpc	▼
▶	delete_record	rpc	▼

FIMMGwsdl	
FIMMGwsdlPort	
location: <a href="http://cloud.fimmg.org/wsdl.php">http://cloud.fimmg.org/wsdl.php</a>	

## Servizi di Accesso

Sono previsti due servizi di accesso e rilascio del token, un login semplice ed un login per la continuità assistenziale, per la chiusura della sessione di consultazione abbiamo la funzione logout

### login

La funzione utilizza i tre parametri user, password e keycartella per l'accesso al Cloud. Dietro il rilascio del token l'utente potrà accedere agli altri servizi.

**Description:** FIMMG login al cloud fimmg

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl# login

**Input:**loginRequest (soap:body, use = encoded)

```
user type string
pwd type string
keycartella type string
```

**Output:**loginResponse (soap:body, use = encoded)

```
return type utente
type utente
  id_utente type positiveInteger
  user type string
  cognome type string
  nome type string
  cf type string
  professione type string
  postazioniGM type postazioniGM
    idpostazioniGM - optional;
    type ArrayOfpostazioneGM - array of type postazioneGM
  id_postazione type string
  nomepostazione type string
  utentisel type utentisel
    idutentisel - optional;
    type ArrayOfutentesel - array of type utentesel
    cognome type string
    nome type string
    cf type string
  token type string
  keycartella type string
```

## loginGuardiaMedica

La funzione viene utilizzata dal medico di continuità assistenziale per accedere al sistema e consultare i dati, oltre ai tre parametri user,password e keycartella vengono richiesti data inizio e data fine turno nonché la postazione di Guardia Medica da cui si collega

**Description:** FIMMG login gm al cloud fimmg

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl# loginGuardiaMedica

**Input:** loginGuardiaMedica Request (soap:body, use = encoded)

```
user type string
pwd type string
keycartella type string
dataInizio type string
dataFine type string
id_postazione type string
```

**Output:**

loginGuardiaMedicaResponse (soap:body, use = encoded)

```
return type string
```

## logout

Funzione di logout, accetta il token che viene disabilitato

**Description:** FIMMG logout cloud fimmg

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl# logout

**Input:** logout Request (soap:body, use = encoded)

```
token type string
```

**Output:**

logout Response (soap:body, use = encoded)

```
return type string
```

## Servizi di scrittura

Descriviamo i metodi di scrittura previsti dalla nostra interfaccia, sono previsti metodi di scrittura parziali o totale di una cartella clinica.

### write

La funzione write viene utilizzata per la scrittura di una intera cartella clinica in un'unica chiamata, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write ed il token risulta valido, vengono caricati i dati contenuti nella cartella, vengono valutate: anagrafica, periodo assistenza, esenzione etc..

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati cartella di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write

**Input:** writeRequest (soap:body, use = encoded)

---

token type string

keycartella type string

cartella type cartella

    paziente type paziente

        id\_paziente type string

        cfpaziente type string

        cfmedico type string

        cognome type string

        nome type string

        sesso type sesso - type string with restriction - enum { 'M', 'F' }

        datanascita type string

        istatcomune\_na type string

        descricomune\_na type string

        nazionalita\_na type string

        tipologia type string

        codregassistito type string

        indirizzo type string

        cap\_re type string

        comune\_re type string

        frazione\_re type string

        telefono type string

        ref\_cogn type string

        ref\_nom type string

---



```
ref_tel type string
sigla_res type string
distretto_cod type string
distretto_des type string
gruppo_san type string
rh type string
sottogruppo_san type string
email type string
profess_istat type string
professione type string
statocivile type string
titolo_studio type string
ssn type string
datadecesso type date
decessoICD9 type string
codicistologico_dec type string
desc_decess type string
note type string
privacy type integer
donatore type integer
accanimento type integer
codreg_ass type string
codasl_ass type string
codcartext type string
keycartella type string
versionecartella type string
datains type date
dataup type date
id_usr_agg type string
nome_medico type string
cognome_medico type string
professione_medico type string
periodiassistenza type periodiassistenza
idperiodiassistenza - optional; type ArrayOfperiodoassistenza - array of
type periodoassistenza
    id_periodoassistenza type string
    id_paziente type string
    id_utente type string
    datascelta type date
    datarevoca type date
    causarevoca type string
    datains type date
    dataup type date
    id_usr_agg type string
contatti type contatti
    idcontatti - optional; type ArrayOfcontatto - array of type contatto
    id_contatto type string
    id_paziente type string
    tipo type tipoContatto - type string with restriction - enum { 'A', 'B',
'C', 'D', 'E', 'F', 'G', 'Z' }
    cod_icd9 type string
    descr_prob type string
    data_contatto type date
    risoluzione type string
    note type string
    id_patologia type string
    datains type date
    dataup type date
esenzioni type esenzioni
    idesenzioni - optional; type ArrayOfesenzione - array of type esenzione
    id_esenzione type string
```

```
id_paziente type string
codese type string
desc_ese type string
cod_ese_reg type string
diagnosi type string
datastart type date
datastop type date
note type string
codICD9probl type string
id_patologia type string
datains type date
dataup type date
id_usr_agg type string
allergie type allergie
idallergie - optional; type ArrayOfallergia - array of type allergia
id_allergia type string
id_paziente type string
codatc type string
descr_molecola type string
farm_scat type string
descr_farm type string
princ_att type string
desc_princ_att type string
evento type string
data_ril type date
note type string
datains type date
dataup type date
id_usr_agg type string
stilidivita type stilidivita
idstilidivita - optional; type ArrayOfstiledivita - array of type stiledivita
id_stiledivita type string
id_paziente type positiveInteger
datavalutazione type date
attivafisica type string
statofumatore type string
quantitafumo type string
alcool type string
notealcool type string
datains type date
dataup type date
id_usr_agg type string
ricoveri type ricoveri
idricoveri - optional; type ArrayOfricovero - array of type ricovero
id_ricovero type string
id_problemaingresso type string
codiceICD9ingresso type string
diagnosingresso type string
nomestrutt type string
reparto type string
dataingresso type date
datadimissione type date
id_problemadimissione type string
codiceICD9dimiss type string
diagnosidimiss type string
note type string
id_paziente type string
datains type date
dataup type date
id_usr_agg type string
prescrizioni type prescrizioni
```

**idprescrizioni** - optional; type *ArrayOfprescrizione* - array of type *prescrizione*

- id\_paziente** type *string*
- id\_prescrizione** type *string*
- codplg** type *string*
- tipoplg** type *string*
- codnomcart** type *string*
- tipoprest** type *string*
- codnomnaz** type *string*
- codnomreg** type *string*
- tiponomen** type *string*
- descresame** type *string*
- dataprescr** type *date*
- dataesecuz** type *date*
- id\_patologia** type *string*
- codicd9prescr** type *string*
- diagnosi** type *string*
- tiporicetta** type *string*
- differibilita** type *string*
- quantitaesame** type *string*
- numesami** type *string*
- concedibilita** type *string*
- patologico** type *string*
- suggerimento** type *string*
- costoprest** type *string*
- codesenzione** type *string*
- followup** type *date*
- note** type *string*
- cup** type *string*
- data\_ins** type *date*
- data\_agg** type *date*
- id\_usr\_agg** type *string*

**farmaci** type *farmaci*

**idfarmaci** - optional; type *ArrayOffarmaco* - array of type *farmaco*

- id\_farmaco** type *string*
- id\_paziente** type *string*
- tipoplg** type *string*
- codplg** type *string*
- id\_patologia** type *string*
- codatc** type *string*
- descrizmolecola** type *string*
- cdoprinatt** type *string*
- descrprinatt** type *string*
- codaic** type *string*
- descrfarm** type *string*
- confezione** type *string*
- dataprescr** type *date*
- codicd9prescr** type *string*
- descrdiagnosi** type *string*
- tiporicetta** type *string*
- noteric** type *string*
- numconf** type *positiveInteger*
- concedibilita** type *string*
- continuativo** type *string*
- posologia** type *string*
- suggerimento** type *string*
- costo** type *string*
- codese** type *string*
- note** type *string*
- datains** type *date*
- dataup** type *date*

```
id_usr_agg type string
vaccinazioni type vaccinazioni
idvaccinazioni - optional; type ArrayOfvaccinazione - array of
type vaccinazione
    id_vaccinazione type string
    id_paziente type string
    datavacc type date
    codvaccino type string
    descvaccino type string
    minsanvaccino type string
    validita type string
    datarichiamo type date
    progressivo type string
    note type string
    data_ins type date
    data_agg type date
    id_usr_agg type string
certificati type certificati
idcertificati - optional; type ArrayOfcertificato - array of type certificato
    id_certificato type string
    id_paziente type string
    id_tipo type string
    datacert type date
    datainizio type date
    datafine type date
    tipocert type string
    id_patologia type string
    codicd9probl type string
    descrizione type string
    testo type string
    datains type date
    dataup type date
    id_usr_agg type string
monitoraggi type monitoraggi
idmonitoraggi - optional; type ArrayOfmonitoraggio - array of
type monitoraggio
    id_monitoraggio type string
    id_paziente type string
    datamon type date
    peso type positiveInteger
    altezza type positiveInteger
    circonferenza type positiveInteger
    sistolica type positiveInteger
    diastolica type positiveInteger
    misurazione type string
    frequenza type positiveInteger
    ritmo type string
    note type string
    datains type date
    dataup type date
    id_usr_agg type string
esiti type esiti
idesiti - optional; type ArrayOfesito - array of type esito
    id_esito type string
    id_paziente type string
    id_prescrizione type string
    codiceesito type string
    descrizioneesito type string
    dataesito type date
    valoreesito type string
    umesito type string
```

```
note type string
datains type date
dataup type date
id_usr_agg type string
patologie type patologie
  idpatologie - optional; type ArrayOfpatologia - array of type patologia
  id_paziente type string
  id_patologia type string
  icd9_problema type string
  descr_problema type string
  data_apertura type date
  data_chiusura type date
  risoluzione type string
  note type string
  datains type date
  dataup type date
  id_usr_agg type string
diari type diari
  iddiari - optional; type ArrayOfdiario - array of type diario
  id_diario type string
  id_paziente type string
  tipo type string
  descrizione type string
  data_rilev type date
  id_patologia type string
  codicd9probl type string
  datains type date
  dataup type date
  id_usr_agg type string
anamnesifamiliari type anamnesifamiliari
  idanamnesifamiliari - optional; type ArrayOfanamnesifamiliare - array of
  type anamnesifamiliare
  id_anamnesifamiliare type string
  id_paziente type string
  data_anamnesi type date
  grado_parentela type string
  id_patologia type string
  cod_icd9 type string
  descr_prob type string
  note type string
  datains type date
  dataup type date
  id_usr_agg type string
scadenziari type scadenziari
  idscadenziari - optional; type ArrayOfscadenziario - array of
  type scadenziario
  id_paziente type integer
  id_scadenziario type string
  data_inizio type date
  data_fine type date
  tipo_scadenziario type string
  codaic type string
  note type string
  datains type date
  dataup type date
  id_usr_agg type string
```

**Output:** writeResponse (soap:body, use = encoded)

---

return *type string*

---

## write\_allergia

La funzione write\_allergia viene utilizzata per la scrittura di una scheda di allergia, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_allergia ed il token risulta valido, vengono caricati i dati contenuti nella scheda di allergia

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi alle allergie di un paziente criptati, con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#write\_allergia

**Input:** write\_allergiaRequest (soap:body, use = encoded)

---

token *type string*

keycartella *type string*

allergia *type allergia*

id\_allergia *type string*

id\_paziente *type string*

codatc *type string*

descr\_molecola *type string*

farm\_scat *type string*

descr\_farm *type string*

princ\_att *type string*

desc\_princ\_att *type string*

evento *type string*

data\_ril *type date*

note *type string*

datains *type date*

dataup *type date*

id\_usr\_agg *type string*

---

**Output:** write\_allergiaResponse (soap:body, use = encoded)

---

return *type string*

---

## write\_anagrafica

La funzione write\_anagrafica viene utilizzata per la scrittura di una scheda anagrafica, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è

autorizzato ad utilizzare la funzione write\_anagrafica ed il token risulta valido, vengono caricati i dati contenuti nella scheda dei dati anagrafici

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo Scrivo nel db i dati relativi al paziente criptati, con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_anagrafica

**Input:** write\_anagraficaRequest (soap:body, use = encoded)

---

token type string  
keycartella type string  
paziente type paziente  
id\_paziente type string  
cfpaziente type string  
cfmedico type string  
cognome type string  
nome type string  
sesso type sesso - type string with restriction - enum { 'M', 'F' }  
datanascita type string  
istatcomune\_na type string  
descricomune\_na type string  
nazionalita\_na type string  
tipologia type string  
codregassistito type string  
indirizzo type string  
cap\_re type string  
comune\_re type string  
frazione\_re type string  
telefono type string  
ref\_cogn type string  
ref\_nom type string  
ref\_tel type string  
sigla\_res type string  
distretto\_cod type string  
distretto\_des type string  
gruppo\_san type string  
rh type string  
sottogruppo\_san type string  
email type string  
profess\_istat type string  
professione type string  
statocivile type string  
titolo\_studio type string  
ssn type string  
datadecesso type date  
decessoICD9 type string  
codicistologico\_dec type string  
desc\_decess type string  
note type string  
privacy type integer  
donatore type integer

---

```
accanimento type integer
codreg_ass type string
codasl_ass type string
codcartext type string
keycartella type string
versionecartella type string
datains type date
dataup type date
id_usr_agg type string
nome_medico type string
cognome_medico type string
professione_medico type string
```

**Output:** write\_anagraficaResponse (soap:body, use = encoded)

```
return type string
```

## write\_certificato

La funzione write\_certificato viene utilizzata per la scrittura di una scheda contenente un certificato, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_certificato ed il token risulta valido, vengono caricati i dati contenuti nella scheda del certificato

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo Scrivo nel db un certificato di un paziente, con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_certificato

**Input:** write\_certificatoRequest (soap:body, use = encoded)

```
token type string
keycartella type string
certificato type certificato
id_certificato type string
id_paziente type string
id_tipo type string
datacert type date
datainizio type date
datafine type date
tipocert type string
id_patologia type string
codicd9probl type string
descrizione type string
testo type string
datains type date
dataup type date
id_usr_agg type string
```

**Output:** write\_certificatoResponse (soap:body, use = encoded)



return type string

## write\_contatto

La funzione write\_contatto viene utilizzata per la scrittura di una scheda contenente un contatto, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_contatto ed il token risulta valido, vengono caricati i dati contenuti nella scheda del contatto

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo Scrivo nel db un contatto di un paziente, con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_contatto

**Input:** write\_contattoRequest (soap:body, use = encoded)

token type string

keycartella type string

contatto type contatto

id\_paziente type string

tipo type tipoContatto - type string with restriction - enum { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'Z' }

cod\_icd9 type string

descr\_prob type string

data\_contatto type date

risoluzione type string

note type string

id\_patologia type string

datains type date

dataup type date

**Output:** write\_contattoResponse (soap:body, use = encoded)

return type string

## write\_diario

La funzione write\_diario viene utilizzata per la scrittura di una scheda contenente un diario clinico, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_diario ed il token risulta valido, vengono caricati i dati contenuti nella scheda del diario

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la

scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo Scrivo nel db un diario di un paziente, con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_diario

**Input:** write\_diarioRequest (soap:body, use = encoded)

---

**token** type string  
**keycartella** type string  
**diario** type diario  
  **id\_paziente** type string  
  **tipo** type string  
  **descrizione** type string  
  **data\_rilev** type date  
  **id\_patologia** type string  
  **codicd9probl** type string  
  **datains** type date  
  **dataup** type date  
  **id\_usr\_agg** type string

---

**Output:** write\_diarioResponse (soap:body, use = encoded)

---

**return** type string

---

### write\_esenzione

La funzione write\_esenzione viene utilizzata per la scrittura di una scheda contenente una esenzione, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_esenzione ed il token risulta valido, vengono caricati i dati contenuti nella scheda dell'esenzione

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi alle esenzioni di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_esenzione

**Input:** write\_esenzioneRequest (soap:body, use = encoded)

---

**token** type string  
**keycartella** type string  
**esenzione** type esenzione  
  **id\_paziente** type string  
  **codese** type string

---

```
desc_ese type string
cod_ese_reg type string
diagnosi type string
datastart type date
datastop type date
note type string
codICD9probl type string
id_patologia type string
datains type date
dataup type date
id_usr_agg type string
```

**Output:** write\_esenzioneResponse (soap:body, use = encoded)

```
return type string
```

## write\_esito

La funzione write\_esito viene utilizzata per la scrittura di una scheda contenente un esito, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_esito ed il token risulta valido, vengono caricati i dati contenuti nella scheda di un esito di esame.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad un esito di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#write\_esito

**Input:** write\_esitoRequest (soap:body, use = encoded)

```
token type string
keycartella type string
esito type esito
id_paziente type string
id_prescrizione type string
codicesesito type string
descrizioneesito type string
dataesito type date
valoresesito type string
umesito type string
note type string
datains type date
dataup type date
id_usr_agg type string
```

**Output:** write\_esitoResponse (soap:body, use = encoded)

return *type string*

---

## write\_farmaco

La funzione write\_farmaco viene utilizzata per la scrittura di una scheda contenente dati di una ricetta medica, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_farmaco ed il token risulta valido, vengono caricati i dati contenuti nella scheda di una ricetta.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad una ricetta di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_farmaco

**Input:** write\_farmacoRequest (soap:body, use = encoded)

---

token *type string*

keycartella *type string*

farmaco *type farmaco*

id\_paziente *type string*

tipoplg *type string*

codplg *type string*

id\_patologia *type string*

codatc *type string*

descrizmolecola *type string*

cdoprinatt *type string*

descrprinatt *type string*

codaic *type string*

descrfarm *type string*

confezione *type string*

dataprescr *type date*

codicd9prescr *type string*

descrdiagnosi *type string*

tiporicetta *type string*

noteric *type string*

numconf *type positiveInteger*

concedibilita *type string*

continuativo *type string*

posologia *type string*

suggerimento *type string*

costo *type string*

codese *type string*

note *type string*

datains *type date*

dataup *type date*

id\_usr\_agg *type string*

---

**Output:** write\_farmacoResponse (soap:body, use = encoded)

return type string

## write\_monitoraggio

La funzione write\_monitoraggio viene utilizzata per la scrittura di una scheda di monitoraggio, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_monitoraggio ed il token risulta valido, vengono caricati i dati contenuti nella scheda di un monitoraggio.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad un monitoraggio di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#write\_monitoraggio

**Input:** write\_monitoraggioRequest (soap:body, use = encoded)

token type string

keycartella type string

monitoraggio type monitoraggio

id\_paziente type string

datamon type date

peso type positiveInteger

altezza type positiveInteger

circonferenza type positiveInteger

sistolica type positiveInteger

diastolica type positiveInteger

misurazione type string

frequenza type positiveInteger

ritmo type string

note type string

datains type date

dataup type date

id\_usr\_agg type string

**Output:** write\_monitoraggioResponse (soap:body, use = encoded)

return type string

## write\_patologia

La funzione write\_patologia viene utilizzata per la scrittura di una scheda di una patologia, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_patologia ed il token risulta valido, vengono caricati i dati contenuti nella scheda di una patologia.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad una patologia di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_patologia

**Input:** write\_patologiaRequest (soap:body, use = encoded)

---

```
token type string
keycartella type string
patologia type patologia
id_paziente type string
id_patologia type string
icd9_problema type string
descr_problema type string
data_apertura type date
data_chiusura type date
risoluzione type string
note type string
datains type date
dataup type date
id_usr_agg type string
```

---

**Output:** write\_patologiaResponse (soap:body, use = encoded)

---

```
return type string
```

---

## write\_periodoassistenza

La funzione write\_periodoassistenza viene utilizzata per la scrittura di una scheda di un periodo di assistenza, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_periodoassistenza ed il token risulta valido, vengono caricati i dati contenuti nella scheda di un periodo di assistenza, ovviamente un paziente avrà diversi periodi di assistenza anche associati a diversi medici.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la

scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad un periodo di assistenza di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_periodoassistenza

**Input:** write\_patologiaRequest (soap:body, use = encoded)

---

<b>token</b>	type string
<b>keycartella</b>	type string
<b>periodoassistenza</b>	type periodoassistenza
<b>id_paziente</b>	type string
<b>id_utente</b>	type string
<b>datascelta</b>	type date
<b>datarevoqa</b>	type date
<b>causarevoqa</b>	type string
<b>datains</b>	type date
<b>dataup</b>	type date
<b>id_usr_agg</b>	type string

---

**Output:** write\_periodoassistenzaResponse (soap:body, use = encoded)

---

<b>return</b>	type string
---------------	-------------

---

### write\_prescrizione

La funzione write\_patologia viene utilizzata per la scrittura di una scheda di una prescrizione, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_prescrizione ed il token risulta valido, vengono caricati i dati contenuti nella scheda di una prescrizione.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad una prescrizione di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_prescrizione

**Input:** write\_prescrizioneRequest (soap:body, use = encoded)

---

<b>token</b>	type string
<b>keycartella</b>	type string
<b>prescrizione</b>	type prescrizione
<b>id_paziente</b>	type string

---

```
id_prescrizione type string
codplg type string
tipoplg type string
codnomcart type string
tipoprest type string
codnomnaz type string
codnomreg type string
tiponomen type string
descresame type string
dataprescr type date
dataesecuz type date
id_patologia type string
codicd9prescr type string
diagnosi type string
tiporicetta type string
differibilita type string
quantitaesame type string
numesami type string
concedibilita type string
patologico type string
suggerimento type string
costoprest type string
codesenzione type string
followup type date
note type string
cup type string
data_ins type date
data_agg type date
id_usr_agg type string
```

**Output:** write\_prescrizioneResponse (soap:body, use = encoded)

```
return type string
```

## write\_ricovero

La funzione write\_ricovero viene utilizzata per la scrittura di una scheda di un ricovero, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_ricovero ed il token risulta valido, vengono caricati i dati contenuti nella scheda di ricovero

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad un ricovero di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_ricovero

**Input:** write\_ricoveroRequest (soap:body, use = encoded)

```
token type string
```



```
keycartella type string
rivocero type ricovero
id_problemaingresso type string
codicelCD9ingresso type string
diagnosingresso type string
nomestrutt type string
reparto type string
dataingresso type date
datadimissione type date
id_problemadimissione type string
codicelCD9dimiss type string
diagnosidimiss type string
note type string
id_paziente type string
datains type date
dataup type date
id_usr_agg type string
```

**Output:** write\_ricoveroResponse (soap:body, use = encoded)

```
return type string
```

## write\_scadenziario

La funzione write\_scadenziario viene utilizzata per la scrittura di una scheda di un piano terapeutico, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_scadenziario ed il token risulta valido, vengono caricati i dati contenuti nella scheda del piano terapeutico

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad un piano terapeutico di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_scadenziario

**Input:** write\_scadenziarioRequest (soap:body, use = encoded)

```
token type string
keycartella type string
scadenziari type scadenziari
  idscadenziari - optional; type ArrayOfscadenziario - array of type scadenziario
    id_paziente type integer
    id_scadenziario type string
    data_inizio type date
    data_fine type date
    tipo_scadenziario type string
    codaic type string
    note type string
    datains type date
```

```
dataup type date  
id_usr_agg type string
```

**Output:** write\_scadenziarioResponse (soap:body, use = encoded)

```
return type string
```

## write\_stiledivita

La funzione write\_stiledivita viene utilizzata per la scrittura di una scheda di uno stile di vita, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione write\_stiledivita ed il token risulta valido, vengono caricati i dati contenuti nella scheda di valutazione dello stile di vita

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad uno stile di vita di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#write\_stiledivita

**Input:** write\_stiledivitaRequest (soap:body, use = encoded)

```
token type string  
keycartella type string  
stiledivita type stiledivita  
id_paziente type positiveInteger  
datavalutazione type date  
attivafisica type string  
statofumatore type string  
quantitafumo type string  
alcohol type string  
notealcohol type string  
datains type date  
dataup type date  
id_usr_agg type string
```

**Output:** write\_stiledivitaResponse (soap:body, use = encoded)

```
return type string
```

## write\_vaccinazione

La funzione write\_vaccinazione viene utilizzata per la scrittura di una scheda di vaccinazione, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è

autorizzato ad utilizzare la funzione `write_vaccinazione` ed il token risulta valido, vengono caricati i dati contenuti nella scheda di una vaccinazione

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Scrivo nel db i dati relativi ad una vaccinazione di un paziente criptati con il keycartella

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#write\_vaccinazione

**Input:** write\_vaccinazioneRequest (soap:body, use = encoded)

---

```
token type string
keycartella type string
vaccinazione type vaccinazione
  id_paziente type string
  datavacc type date
  codvaccino type string
  descvaccino type string
  minsanvaccino type string
  validita type string
  datarichiamo type date
  progressivo type string
  note type string
  data_ins type date
  data_agg type date
  id_usr_agg type string
```

---

**Output:** write\_vaccinazioneResponse (soap:body, use = encoded)

---

```
return type string
```

---

### expressfolder

La funzione `expressfolder` viene utilizzata dal programma di estrazione scaricabile dal nostro sito, invia una scheda paziente completa al sistema in formato binario, analizzata e caricata nel sistema. Valutati token, keycartella si procede alla valutazione.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Funzione di invio dati da estrattore

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#expressfolder

**Input:** expressfolderRequest (soap:body, use = encoded)

**token** type *string*

**keycartella** type *string*

**data** type *comprexfolder*

**data** type *base64Binary*

**Output:** expressfolderResponse (soap:body, use = encoded)

**return** type *string*

## Servizi di consultazione

Descriviamo i metodi di consultazione previsti dalla nostra interfaccia, sono previsti metodi di richiesta parziali o totale di una cartella clinica.

### search\_pazienti

La funzione `search_pazienti` viene utilizzata per ricercare i pazienti che corrispondono ai criteri di ricerca. I criteri vengono valutati all'interno dell'insieme dei pazienti dell'utente connesso e degli utenti che condividono dati con lo stesso, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_paziente` ed il token risulta valido, vengono ricercati i dati che corrispondono ai parametri inseriti.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Ricerca un paziente per uno dei criteri (id paziente,cognome,nome,codice fiscale, codice tessera, data di nascita) e restituisce la cartella corrispondente se autorizzato

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_pazienti

**Input:** search\_pazientiRequest (soap:body, use = encoded)

---

```
token type string
keycartella type string
id_paziente type string
cognome type string
nome type string
cfpaziente type string
codicetessera type string
datadinascita type string
tiporicerca type string
utentisel type utentisel
  idutentisel - optional; type ArrayOfutentisel - array of type utentisel
    cognome type string
    nome type string
    cf type string
```

**Output:** search\_pazientiResponse (soap:body, use = encoded)

---

```
return type pazienti
```

---

**idpazienti** - optional; type *ArrayOfPaziente* - array of type *paziente*  
**paziente** type *paziente*  
**id\_paziente** type *string*  
**cfpaziente** type *string*  
**cfmedico** type *string*  
**cognome** type *string*  
**nome** type *string*  
 **Sesso** type  *Sesso* - type *string* with restriction - enum { 'M', 'F' }  
**datanascita** type *string*  
**istatcomune\_na** type *string*  
**descricomune\_na** type *string*  
**nazionalita\_na** type *string*  
**tipologia** type *string*  
**codregassistito** type *string*  
**indirizzo** type *string*  
**cap\_re** type *string*  
**comune\_re** type *string*  
**frazione\_re** type *string*  
**telefono** type *string*  
**ref\_cogn** type *string*  
**ref\_nom** type *string*  
**ref\_tel** type *string*  
**sigla\_res** type *string*  
**distretto\_cod** type *string*  
**distretto\_des** type *string*  
**gruppo\_san** type *string*  
**rh** type *string*  
**sottogruppo\_san** type *string*  
**email** type *string*  
**profess\_istat** type *string*  
**professione** type *string*  
**statocivile** type *string*  
**titolo\_studio** type *string*  
**ssn** type *string*  
**datadecesso** type *date*  
**decessoICD9** type *string*  
**codicistologico\_dec** type *string*  
**desc\_decess** type *string*  
**note** type *string*  
**privacy** type *integer*  
**donatore** type *integer*  
**accanimento** type *integer*  
**codreg\_ass** type *string*  
**codasl\_ass** type *string*  
**codcartext** type *string*  
**keycartella** type *string*  
**versionecartella** type *string*  
**datains** type *date*  
**dataup** type *date*  
**id\_usr\_agg** type *string*  
**nome\_medico** type *string*  
**cognome\_medico** type *string*  
**professione\_medico** type *string*

## search\_pazienti\_small

La funzione `search_pazienti_small` viene utilizzata per ricercare i pazienti che corrispondono ai criteri di ricerca. I criteri vengono valutati all'interno dell'insieme dei pazienti dell'utente connesso e degli utenti che condividono dati con lo stesso, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_paziente_small` ed il token risulta valido, vengono ricercati i dati che corrispondono ai parametri inseriti. La funzione riporta rispetto a `search_pazienti` un insieme di dati ridotto.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Ricerca un paziente per uno dei criteri (id paziente,cognome,nome,codice fiscale, codice tessera, data di nascita) e restituisce la cartella corrispondente se autorizzato

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#search\_pazienti\_small

**Input:** search\_pazienti\_smallRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
cognome type string
nome type string
cfpaziente type string
codicetessera type string
datadinascita type string
tiporicerca type string
utentisel type utentisel
  idutentisel - optional; type ArrayOfutentisel - array of type utentisel
    cognome type string
    nome type string
    cf type string
```

**Output:** search\_pazientiResponse (soap:body, use = encoded)

```
return type pazientiSmall
  idpazientiSmall - optional; type ArrayOfPazienteSmall - array of type pazienteSmall
  id_paziente type string
  cfpaziente type string
  cognome type string
  nome type string
  sesso type sesso - type string with restriction - enum { 'M', 'F' }
  datanascita type string
```





## search

La funzione search viene utilizzata per la richiesta di una intera cartella clinica, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituisce la cartella completa del pazienti desiderato

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#search

**Input:** searchRequest (soap:body, use = encoded)

token type string

keycartella type string

id\_paziente type string

**Output:** searchResponse (soap:body, use = encoded)

return type cartelle

idcartelle - optional; type *ArrayOfcartella* - array of type cartella

cartella type cartella

paziente type paziente

id\_paziente type string

cfpaziente type string

cfmedico type string

cognome type string

nome type string

sezzo type sesso - type string with restriction - enum { 'M', 'F' }

datanascita type string

istatcomune\_na type string

descricomune\_na type string

nazionalita\_na type string

tipologia type string

codregassistito type string

indirizzo type string

cap\_re type string

comune\_re type string

frazione\_re type string

telefono type string

ref\_cogn type string

ref\_nom type string

ref\_tel type string

sigla\_res type string

distretto\_cod type string

distretto\_des type string

gruppo\_san type string

rh type string

```

sottogruppo_san type string
email type string
profess_istat type string
professione type string
statocivile type string
titolo_studio type string
ssn type string
datadecesso type date
decessoICD9 type string
codicistologico_dec type string
desc_decess type string
note type string
privacy type integer
donatore type integer
accanimento type integer
codreg_ass type string
codasl_ass type string
codcartext type string
keycartella type string
versionecartella type string
datains type date
dataup type date
id_usr_agg type string
nome_medico type string
cognome_medico type string
professione_medico type string
periodiassistenza type periodiassistenza
idperiodiassistenza - optional; type ArrayOfperiodoassistenza -
array of type periodoassistenza
    id_periodoassistenza type string
    id_paziente type string
    id_utente type string
    datascelta type date
    datarevoca type date
    causarevoca type string
    datains type date
    dataup type date
    id_usr_agg type string
contatti type contatti
idcontatti - optional; type ArrayOfcontatto - array of type contatto
    id_contatto type string
    id_paziente type string
    tipo type tipoContatto - type string with restriction - enum {
'A', 'B', 'C', 'D', 'E', 'F', 'G', 'Z' }
    cod_icd9 type string
    descr_prob type string
    data_contatto type date
    risoluzione type string
    note type string
    id_patologia type string
    datains type date
    dataup type date
esenzioni type esenzioni
idesenzioni - optional; type ArrayOfesenzione - array of
type esenzione
    id_esenzione type string
    id_paziente type string
    codese type string
    desc_ese type string
    cod_ese_reg type string
    diagnosi type string

```

```
    datastart type date
    datastop type date
    note type string
    codICD9probl type string
    id_patologia type string
    datains type date
    dataup type date
    id_usr_agg type string
allergie type allergie
    idallergie - optional; type ArrayOfallergia - array of type allergia
    id_allergia type string
    id_paziente type string
    codatc type string
    descr_molecola type string
    farm_scat type string
    descr_farm type string
    princ_att type string
    desc_princ_att type string
    evento type string
    data_ril type date
    note type string
    datains type date
    dataup type date
    id_usr_agg type string
stilidivita type stilidivita
    idstilidivita - optional; type ArrayOfstiledivita - array of
type stiledivita
    id_stiledivita type string
    id_paziente type positiveInteger
    datavalutazione type date
    attivafisica type string
    statofumatore type string
    quantita fumo type string
    alcool type string
    notealcool type string
    datains type date
    dataup type date
    id_usr_agg type string
ricoveri type ricoveri
    idricoveri - optional; type ArrayOfricovero - array of type ricovero
    id_ricovero type string
    id_problemaingresso type string
    codicelCD9ingresso type string
    diagnosingresso type string
    nomestrutt type string
    reparto type string
    dataingresso type date
    datadimissione type date
    id_problemadimissione type string
    codicelCD9dimiss type string
    diagnosidimiss type string
    note type string
    id_paziente type string
    datains type date
    dataup type date
    id_usr_agg type string
prescrizioni type prescrizioni
    idprescrizioni - optional; type ArrayOfprescrizione - array of
type prescrizione
    id_paziente type string
    id_prescrizione type string
```

```
codplg type string
tipoplg type string
codnomcart type string
tipoprest type string
codnomnaz type string
codnomreg type string
tiponomen type string
descresame type string
dataprescr type date
dataesecuz type date
id_patologia type string
codicd9prescr type string
diagnosi type string
tiporicetta type string
differibilita type string
quantitaesame type string
numesami type string
concedibilita type string
patologico type string
suggerimento type string
costoprest type string
codesenzione type string
followup type date
note type string
cup type string
data_ins type date
data_agg type date
id_usr_agg type string
farmaci type farmaci
  idfarmaci - optional; type ArrayOffarmaco - array of type farmaco
  id_farmaco type string
  id_paziente type string
  tipoplg type string
  codplg type string
  id_patologia type string
  codatc type string
  descrizmolecola type string
  cdoprinatt type string
  descrprinatt type string
  codaic type string
  descrfarm type string
  confezione type string
  dataprescr type date
  codicd9prescr type string
  descrdiagnosi type string
  tiporicetta type string
  noteric type string
  numconf type positiveInteger
  concedibilita type string
  continuativo type string
  posologia type string
  suggerimento type string
  costo type string
  codese type string
  note type string
  datains type date
  dataup type date
  id_usr_agg type string
vaccinazioni type vaccinazioni
  idvaccinazioni - optional; type ArrayOfvaccinazione - array of
type vaccinazione
```

```
id_vaccinazione type string
id_paziente type string
datavacc type date
codvaccino type string
descvaccino type string
minsanvaccino type string
validita type string
datarichiamo type date
progressivo type string
note type string
data_ins type date
data_agg type date
id_usr_agg type string
certificati type certificati
idcertificati - optional; type ArrayOfcertificato - array of
type certificato
id_certificato type string
id_paziente type string
id_tipo type string
datacert type date
datainizio type date
datafine type date
tipocert type string
id_patologia type string
codicd9probl type string
descrizione type string
testo type string
datains type date
dataup type date
id_usr_agg type string
monitoraggi type monitoraggi
idmonitoraggi - optional; type ArrayOfmonitoraggio - array of
type monitoraggio
id_monitoraggio type string
id_paziente type string
datamon type date
peso type positiveInteger
altezza type positiveInteger
circonferenza type positiveInteger
sistolica type positiveInteger
diastolica type positiveInteger
misurazione type string
frequenza type positiveInteger
ritmo type string
note type string
datains type date
dataup type date
id_usr_agg type string
esiti type esiti
idesiti - optional; type ArrayOfesito - array of type esito
id_esito type string
id_paziente type string
id_prescrizione type string
codicesito type string
descrizioneesito type string
dataesito type date
valoreesito type string
umesito type string
note type string
datains type date
dataup type date
```

```
id_usr_agg type string
patologie type patologie
idpatologie - optional; type ArrayOfpatologia - array of
type patologia

id_paziente type string
id_patologia type string
icd9_problema type string
descr_problema type string
data_apertura type date
data_chiusura type date
risoluzione type string
note type string
datains type date
dataup type date
id_usr_agg type string
diari type diari
iddiari - optional; type ArrayOfdiario - array of type diario
id_diario type string
id_paziente type string
tipo type string
descrizione type string
data_rilev type date
id_patologia type string
codicd9probl type string
datains type date
dataup type date
id_usr_agg type string
anamnesifamiliari type anamnesifamiliari
idanamnesifamiliari - optional; type ArrayOfanamnesifamiliare -
array of type anamnesifamiliare
id_anamnesifamiliare type string
id_paziente type string
data_anamnesi type date
grado_parentela type string
id_patologia type string
cod_icd9 type string
descr_prob type string
note type string
datains type date
dataup type date
id_usr_agg type string
scadenziari type scadenziari
idscadenziari - optional; type ArrayOfscadenziario - array of
type scadenziario

id_paziente type integer
id_scadenziario type string
data_inizio type date
data_fine type date
tipo_scadenziario type string
codaic type string
note type string
datains type date
dataup type date
id_usr_agg type string
```

## search\_allergie

La funzione `search_allergie` viene utilizzata per la richiesta di tutte le allergie caricate e relative ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_allergie` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione delle allergie e intolleranze di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#search\_allergie

**Input:** search\_allergieRequest (soap:body, use = encoded)

---

**token** type string

**keycartella** type string

**id\_paziente** type string

**Output:** search\_allergieResponse (soap:body, use = encoded)

---

**return** type *allergia*

**idallergie** - optional; type *ArrayOfallergia* - array of type *allergia*

**id\_allergia** type string

**id\_paziente** type string

**codatc** type string

**descr\_molecola** type string

**farm\_scat** type string

**descr\_farm** type string

**princ\_att** type string

**desc\_princ\_att** type string

**evento** type string

**data\_ril** type date

**note** type string

**datains** type date

**dataup** type date

**id\_usr\_agg** type string

### search\_certificati

La funzione `search_certificati` viene utilizzata per la richiesta di tutti i certificati caricati e relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_certificati` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dei certificati di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_certificati

**Input:** search\_certificatiRequest (soap:body, use = encoded)

token type string

keycartella type string

id\_paziente type string

**Output:** search\_certificatiResponse (soap:body, use = encoded)

return type *certificati*

**idcertificati** - optional; type *ArrayOfcertificato* - array of type *certificato*

id\_certificato type string

id\_paziente type string

id\_tipo type string

datacert type date

datainizio type date

datafine type date

tipocert type string

id\_patologia type string

codicd9probl type string

descrizione type string

testo type string

datains type date

dataup type date

id\_usr\_agg type string

### search\_contatti

La funzione search\_contatti viene utilizzata per la richiesta di tutti i contatti caricati e relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_contatti ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dei contatti di un paziente



**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_contatti

**Input:** search\_contattiRequest (soap:body, use = encoded)

token type string

keycartella type string

id\_paziente type string

**Output:** search\_allergieResponse (soap:body, use = encoded)

return type contatti

idcontatti - optional; type ArrayOfcontatto - array of type contatto

id\_contatto type string

id\_paziente type string

tipo type tipoContatto - type string with restriction - enum { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'Z' }

cod\_icd9 type string

descr\_prob type string

data\_contatto type date

risoluzione type string

note type string

id\_patologia type string

datains type date

dataup type date

## search\_diari

La funzione search\_diari viene utilizzata per la richiesta di tutti i diari clinici caricati e relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_diari ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dei diari di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_diari

**Input:** search\_diariRequest (soap:body, use = encoded)

token type string

keycartella type string

id\_paziente type string

**Output:** search\_diariResponse (soap:body, use = encoded)

return type diari

```
iddiari - optional; type ArrayOfdiario - array of type diario
id_diario type string
id_paziente type string
tipo type string
descrizione type string
data_rilev type date
id_patologia type string
codicd9probl type string
datains type date
dataup type date
id_usr_agg type string
```

## search\_esenzioni

La funzione search\_esenzioni viene utilizzata per la richiesta di tutte le esenzioni caricati e relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_esenzioni ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione delle esenzioni di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_esenzioni

**Input:** search\_esenzioniRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
```

**Output:** search\_esenzioniiResponse (soap:body, use = encoded)

```
return type esenzione
id_esenzione type string
id_paziente type string
codese type string
desc_ese type string
cod_ese_reg type string
diagnosi type string
datastart type date
datastop type date
note type string
codICD9probl type string
id_patologia type string
datains type date
dataup type date
```

**id\_usr\_agg** type string

## search\_esiti

La funzione search\_esiti viene utilizzata per la richiesta di tutti gli esiti caricati e relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_esiti ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione degli esiti di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_esiti

**Input:** search\_esitiRequest (soap:body, use = encoded)

**token** type string

**keycartella** type string

**id\_paziente** type string

**Output:** search\_esitiResponse (soap:body, use = encoded)

**return** type *esiti*

**idesiti** - optional; type *ArrayOfesito* - array of type *esito*

**id\_esito** type string

**id\_paziente** type string

**id\_prescrizione** type string

**codiceesito** type string

**descrizioneesito** type string

**dataesito** type date

**valoreesito** type string

**umesito** type string

**note** type string

**datains** type date

**dataup** type date

**id\_usr\_agg** type string

## search\_farmaci

La funzione `search_farmaci` viene utilizzata per la richiesta di tutti i dati ricette caricate e relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_farmaci` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione delle ricette di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_farmaci

**Input:** search\_farmaciRequest (soap:body, use = encoded)

---

**token** type string

**keycartella** type string

**id\_paziente** type string

**Output:** search\_farmaciResponse (soap:body, use = encoded)

---

**return** type *farmaci*

**idfarmaci** - optional; type *ArrayOffarmaco* - array of type *farmaco*

**id\_farmaco** type string

**id\_paziente** type string

**tipoplg** type string

**codplg** type string

**id\_patologia** type string

**codatc** type string

**descrizmolecola** type string

**cdoprinatt** type string

**descrprinatt** type string

**codaic** type string

**descrfarm** type string

**confezione** type string

**dataprescr** type date

**codicd9prescr** type string

**descrdiagnosi** type string

**tiporicetta** type string

**noteric** type string

**numconf** type *positiveInteger*

**concedibilita** type string

**continuativo** type string

**posologia** type string

**suggerimento** type string

**costo** type string

**codese** type string

```
note type string
datains type date
dataup type date
id_usr_agg type string
```

## search\_gruppo

La funzione `search_gruppo` viene utilizzata per la richiesta di tutti i dati identificativi di base per determinare gli appartenenti ad un gruppo (i gruppi vengono creati sul portale [www.netmedicaitalia.it](http://www.netmedicaitalia.it)), vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_gruppo` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Ricerca i colleghi a cui sono collegato

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_gruppo

**Input:** search\_gruppoRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
```

**Output:** search\_gruppoResponse (soap:body, use = encoded)

```
return type utentisel
idutentisel - optional; type ArrayOfutentisel - array of type utentisel
cognome type string
nome type string
cf type string
```

## search\_monitoraggi

La funzione `search_monitoraggi` viene utilizzata per la richiesta di tutti i dati di monitoraggio relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_monitoraggi` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dei monitoraggi di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_monitoraggi

**Input:** search\_monitoraggiRequest (soap:body, use = encoded)

---

**token** type *string*

**keycartella** type *string*

**id\_paziente** type *string*

**Output:** search\_monitoraggiResponse (soap:body, use = encoded)

---

**return** type *monitoraggi*

**idmonitoraggi** - optional; type *ArrayOfmonitoraggio* - array of type *monitoraggio*

**id\_monitoraggio** type *string*

**id\_paziente** type *string*

**datamon** type *date*

**peso** type *positiveInteger*

**altezza** type *positiveInteger*

**circonferenza** type *positiveInteger*

**sistolica** type *positiveInteger*

**diastolica** type *positiveInteger*

**misurazione** type *string*

**frequenza** type *positiveInteger*

**ritmo** type *string*

**note** type *string*

**datains** type *date*

**dataup** type *date*

**id\_usr\_agg** type *string*

### search\_num\_paziente

La funzione search\_num\_paziente per un eventuale programma di visualizzazione per riportare le cardinalità delle schede e visualizzarle, evita al programma di caricare a priori l'intera cartella clinica del paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_num\_paziente ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituisce le cardinalità delle cartelle di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_num\_paziente

**Input:** search\_num\_pazienteRequest (soap:body, use = encoded)

---

token type string

keycartella type string

id\_paziente type string

data type string

**Output:** search\_num\_pazienteResponse (soap:body, use = encoded)

---

return type *numeroDatiPaziente*

id\_paziente type *positiveInteger*

num\_periodoassistenza type *positiveInteger*

num\_contatti type *positiveInteger*

num\_esenzioni type *positiveInteger*

num\_allergie type *positiveInteger*

num\_stilidivita type *positiveInteger*

num\_ricoveri type *positiveInteger*

num\_prescrizioni type *positiveInteger*

num\_farmaci type *positiveInteger*

num\_vaccinazioni type *positiveInteger*

num\_certificati type *positiveInteger*

num\_monitoraggi type *positiveInteger*

num\_esiti type *positiveInteger*

num\_patologie type *positiveInteger*

num\_diari type *positiveInteger*

num\_anamnesifamiliari type *positiveInteger*

---

## search\_patologie

La funzione search\_patologie viene utilizzata per la richiesta di tutti i dati delle patologie relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_patologie ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione delle patologie di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_patologie

**Input:** search\_patologieRequest (soap:body, use = encoded)

---

```
token type string
keycartella type string
id_paziente type string
```

**Output:** search\_patologieResponse (soap:body, use = encoded)

```
return type patologie
idpatologie - optional; typeArrayOfpatologia - array of type patologia
  id_paziente type string
  id_patologia type string
  icd9_problema type string
  descr_problema type string
  data_apertura type date
  data_chiusura type date
  risoluzione type string
  note type string
  datains type date
  dataup type date
  id_usr_agg type string
```

## search\_periodiassistenza

La funzione search\_periodiassistenza viene utilizzata per la richiesta di tutti i dati ai periodi di assistenza relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_periodiassistenza ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dei periodi di assistenza di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_periodiassistenza

**Input:** search\_periodiassistenzaRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
```

**Output:** search\_periodiassistenzaResponse (soap:body, use = encoded)

```
return type periodiassistenza
idperiodiassistenza - optional; typeArrayOfperiodoassistenza - array of
type periodoassistenza
  id_periodoassistenza type string
  id_paziente type string
```



```
id_utente type string
datascelta type date
datarevoca type date
causarevoca type string
datains type date
dataup type date
id_usr_agg type string
```

## search\_prescrizioni

La funzione search\_prescrizioni viene utilizzata per la richiesta di tutti i dati delle prescrizioni relative ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_prescrizioni ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione le prescrizioni di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_prescrizioni

**Input:** search\_prescrizioniRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
```

**Output:** search\_prescrizioniResponse (soap:body, use = encoded)

```
return type prescrizioni
idprescrizioni - optional; typeArrayOfprescrizione - array of type prescrizione
id_paziente type string
id_prescrizione type string
codplg type string
tipoplg type string
codnomcart type string
tipoprest type string
codnomnaz type string
codnomreg type string
tiponomen type string
descresame type string
dataprescr type date
dataesecuz type date
id_patologia type string
codicd9prescr type string
diagnosi type string
tiporicetta type string
differibilita type string
```

```
quantitaesame type string
numesami type string
concedibilita type string
patologico type string
suggerimento type string
costoprest type string
codesenzione type string
followup type date
note type string
cup type string
data_ins type date
data_agg type date
id_usr_agg type string
```

## search\_ricoveri

La funzione search\_ricoveri viene utilizzata per la richiesta di tutti i dati dei ricoveri relative ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_ricoveri ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione i ricoveri di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_ricoveri

**Input:** search\_ricoveriRequest (soap:body, use = encoded)

```
token type string
```

```
keycartella type string
```

```
id_paziente type string
```

**Output:** search\_ricoveriResponse (soap:body, use = encoded)

```
return type ricoveri
```

```
idricoveri - optional; type ArrayOfricovero - array of type ricovero
```

```
id_ricovero type string
```

```
id_problemaingresso type string
```

```
codiceICD9ingresso type string
```

```
diagnosingresso type string
```

```
nomestrutt type string
```

```
reparto type string
```

```
dataingresso type date
```

```
datadimissione type date
```

```
id_problemadimissione type string
```

```
codiceICD9dimiss type string
```

```
diagnosidimiss type string
```

```
note type string
id_paziente type string
datains type date
dataup type date
id_usr_agg type string
```

## search\_scadenziari

La funzione search\_scadenziari viene utilizzata per la richiesta di tutti i dati dei piani terapeutici relative ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_scadenziari ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dei piani terapeutici di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#search\_scadenziari

**Input:** search\_scadenziariRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
```

**Output:** search\_scadenziariResponse (soap:body, use = encoded)

```
return type scadenziari
idscadenziari - optional; type ArrayOfscadenziario - array of type scadenziario
id_paziente type integer
id_scadenziario type string
data_inizio type date
data_fine type date
tipo_scadenziario type string
codaic type string
note type string
datains type date
dataup type date
id_usr_agg type string
```

## search\_status\_paziente

La funzione search\_status\_paziente viene utilizzata per la richiesta dello stato di assistenza di un determinato paziente, vengono richiesti token e keycartella per

l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_status_paziente` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dello status di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_status\_paziente

**Input:** search\_status\_pazienteRequest (soap:body, use = encoded)

---

`token` type *string*

`keycartella` type *string*

`id_paziente` type *string*

**Output:** search\_status\_pazienteResponse (soap:body, use = encoded)

---

`return` type *statusPaziente*

`status` type *string*

`init_time` type *date*

`finish_time` type *date*

---

### search\_stilidivita

La funzione `search_stilidivita` viene utilizzata per la richiesta di tutti i dati di stile di vita relativi ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `search_stilidivita` ed il token risulta valido, viene ricercato l'`id_paziente` richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione degli stili di vita di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_stilidivita

**Input:** search\_stilidivitaRequest (soap:body, use = encoded)

---

`token` type *string*

**keycartella** type string

**id\_paziente** type string

**Output:** search\_stilidivitaResponse (soap:body, use = encoded)

**return** type stilidivita

**idstilidivita** - optional; type ArrayOfstiledivita - array of type stiledivita

**id\_stiledivita** type string

**id\_paziente** type positiveInteger

**datavalutazione** type date

**attivita fisica** type string

**statofumatore** type string

**quantitafumo** type string

**alcohol** type string

**notealcohol** type string

**datains** type date

**dataup** type date

**id\_usr\_agg** type string

## search\_vaccinazioni

La funzione search\_vaccinazioni viene utilizzata per la richiesta di tutti i dati delle vaccinazioni effettuate da un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione search\_vaccinazioni ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione delle vaccinazioni di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn: FIMMGwsdl#search\_vaccinazioni

**Input:** search\_vaccinazioniRequest (soap:body, use = encoded)

**token** type string

**keycartella** type string

**id\_paziente** type string

**Output:** search\_vaccinazioniResponse (soap:body, use = encoded)

**return** type vaccinazioni

**idvaccinazioni** - optional; type ArrayOfvaccinazione - array of type vaccinazione

**id\_vaccinazione** type string

**id\_paziente** type string

**datavacc** type date

**codvaccino** type string

**descvaccino** type string

**minsavaccino** type string

**validita** type string

**datarichiamo** *type date*  
**progressivo** *type string*  
**note** *type string*  
**data\_ins** *type date*  
**data\_agg** *type date*  
**id\_usr\_agg** *type string*

---

## Altri servizi

Abbiamo raggruppato qui altri servizi offerti dall'interfaccia, raggruppati qui per semplicità ma non meno importanti

### elenco\_Referti

La funzione elenco\_Referti viene utilizzata per la richiesta dell'elenco di tutti i referti di un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione elenco\_Referti ed il token risulta valido, viene ricercato l'id\_paziente richiesto (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'id\_paziente desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Restituzione dell'elenco dei referti di un paziente

**Style:** rpc

**Operation type:** Request-response. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#elenco\_RefertiPaziente

**Input:** elenco\_Referti\_PazienteRequest (soap:body, use = encoded)

token type string  
keycartella type string  
id\_paziente type string  
id\_tipodocumento type integer

**Output:** elenco\_Referti\_PazienteResponse (soap:body, use = encoded)

return type elenco\_referti  
idelencoReferto - optional; type ArrayOfdatoReferto - array of type datoReferto  
id\_referto type integer  
id\_paziente type string  
id\_utente type string  
tipodocumento type string  
identificativo type string  
data\_documento type string  
data type date  
id\_esito type string

### invio\_RefertoPaziente

La funzione `invio_RefertoPaziente` viene utilizzata per l'invio di un referto, il sistema si comporta da Repository per vari tipi di referto. Se il programma è autorizzato ad utilizzare la funzione `invio_RefertoPaziente` ed il token risulta valido, viene caricato il referto agganciandolo all'`id_paziente` evidenziato (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description** Invio di un referto di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#invio\_RefertoPaziente

**Input:** `invio_RefertoPazienteRequest` (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
id_tipodocumento type integer
id_tipodocumento type integer
identificativo type string
data_documento type date
referto type base64Binary
id_esito type integer
```

**Output:** `invio_RefertoPazienteResponse` (soap:body, use = encoded)

```
return type integer
```

### lettura\_RefertoPaziente

La funzione `lettura_RefertoPaziente` viene utilizzata per la richiesta del referto desiderato relativo ad un determinato paziente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `elenco_Referti` ed il token risulta valido, viene ricercato l'`id_paziente` ed `id_referto` richiesti (precedentemente il programma richiedente avrà consultato una delle funzioni di ricerca paziente per ottenere l'`id_paziente` desiderato ed `elenco_Referti` per l'`id_referto`)

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault



**Description** lettura di un referto di un paziente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#lettura\_RefertoPaziente

**Input:** lettura\_RefertoPazienteRequest (soap:body, use = encoded)

```
token type string
keycartella type string
id_paziente type string
id_referto type integer
```

**Output:** lettura\_RefertoPazienteResponse (soap:body, use = encoded)

```
return type base64Binary
```

## read\_notifiche

La funzione `read_notifiche`, viene utilizzata per leggere le varie notifiche memorizzate dal sistema (accesso alla consultazione di altri medici, schede incomplete inviate etc..) di un determinato utente, vengono richiesti token e keycartella per l'autorizzazione. Se il programma è autorizzato ad utilizzare la funzione `read_notifiche` ed il token risulta valido, vengono riportati i dati dell'utente desiderato.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

**Description:** Restituzione delle notifiche relative ad un utente utente

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#read\_notifiche

**Input:** read\_notificheRequest (soap:body, use = encoded) [Source code](#)

```
token type string
keycartella type string
id_utente type string
```

**Output:** read\_notificheResponse (soap:body, use = encoded) [Source code](#)

```
return type notifiche
  idnotifiche - optional; type ArrayOfnotifica - array of type notifica
    id_notifica type string
    id_utente_mitt type string
    cognome_mitt type string
    nome_mitt type string
    professione_mitt type string
    id_cartella_mitt type string
    cartella_mitt type string
    data_ins type string
    nuova type string
```

```
paziente type pazienteSmall
  id_paziente type string
  cfpaziente type string
  cognome type string
  nome type string
  sesso type sesso - type string with restriction - enum { 'M', 'F' }
  datanascita type string
id_tiponotifica type string
descrizione_tiponotifica type string
tipo_messaggio type string
messaggio_notifica type string
link_notifica type string
```

## delete\_record

La funzione delete\_record, viene utilizzata per cancellare un determinato dato nel sistema, questa operazione è consentita solo a determinate keycartella.

Lo schema sottostante descrive la chiamata della funzione, per la struttura rimandiamo al documento del DB, se l'operazione di parsing del documento XML e la scrittura dei dati nel DB vanno a buon fine viene dato un messaggio di avvenuta scrittura, in caso contrario un messaggio di fault

### Description:

Cancella un record inserito dal programma autorizzato

**Style:** rpc

**Operation type:** *Request-response*. The endpoint receives a message, and sends a correlated message.

**SOAP action:** urn:FIMMGwsdl#delete\_record

### Input:

delete\_recordRequest (soap:body, use = encoded)*Source code*

```
token type string
keycartella type string
id_paziente type string
nome_tabella type string
record type string
```

### Output:

delete\_recordResponse (soap:body, use = encoded)*Source code*

```
return type string
```

## Codice

```

<?xml version="1.0"?>
<definitions targetNamespace="urn:FIMMGwsdl"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:FIMMGwsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<types>
  <xsd:schema targetNamespace="urn:FIMMGwsdl">
    <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    <xsd:complexType name="utente">
      <xsd:all>
        <xsd:element name="id_utente" type="xsd:positiveInteger" />
        <xsd:element name="user" type="xsd:string" />
        <xsd:element name="cognome" type="xsd:string" />
        <xsd:element name="nome" type="xsd:string" />
        <xsd:element name="cf" type="xsd:string" />
        <xsd:element name="professione" type="xsd:string" />
        <xsd:element name="postazioniGM" type="tns:postazioniGM" />
        <xsd:element name="utentisel" type="tns:utentisel" />
        <xsd:element name="token" type="xsd:string" />
        <xsd:element name="keycartella" type="xsd:string" />
      </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="numeroDatiPaziente">
      <xsd:all>
        <xsd:element name="id_paziente" type="xsd:positiveInteger" />
        <xsd:element name="num_periodoassistenza" type="xsd:positiveInteger" />
        <xsd:element name="num_contatti" type="xsd:positiveInteger" />
        <xsd:element name="num_esenzioni" type="xsd:positiveInteger" />
        <xsd:element name="num_allergie" type="xsd:positiveInteger" />
        <xsd:element name="num_stilidivita" type="xsd:positiveInteger" />
        <xsd:element name="num_ricoveri" type="xsd:positiveInteger" />
        <xsd:element name="num_prescrizioni" type="xsd:positiveInteger" />
        <xsd:element name="num_farmaci" type="xsd:positiveInteger" />
        <xsd:element name="num_vaccinazioni" type="xsd:positiveInteger" />
        <xsd:element name="num_certificati" type="xsd:positiveInteger" />
        <xsd:element name="num_monitoraggi" type="xsd:positiveInteger" />
        <xsd:element name="num_esiti" type="xsd:positiveInteger" />
        <xsd:element name="num_patologie" type="xsd:positiveInteger" />
        <xsd:element name="num_diari" type="xsd:positiveInteger" />
        <xsd:element name="num_anamnesifamiliari" type="xsd:positiveInteger" />
      </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="postazioneGM">
      <xsd:all>
        <xsd:element name="id_postazione" type="xsd:string" />
        <xsd:element name="nomepostazione" type="xsd:string" />
      </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="postazioniGM">
      <xsd:sequence>
        <xsd:element name="idpostazioniGM" type="tns:ArrayOfpostazioneGM" minOccurs="0" maxOccurs="1" />
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ArrayOfpostazioneGM">
      <xsd:complexContent>
        <xsd:restriction base="SOAP-ENC:Array">
          <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:postazioneGM[]" minOccurs="0" maxOccurs="unbounded" />
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="scadenziario">
      <xsd:all>
        <xsd:element name="id_paziente" type="xsd:integer" />
        <xsd:element name="id_scadenziario" type="xsd:string" />
        <xsd:element name="data_inizio" type="xsd:date" />
        <xsd:element name="data_fine" type="xsd:date" />
        <xsd:element name="tipo_scadenziario" type="xsd:string" />
        <xsd:element name="codaic" type="xsd:string" />
        <xsd:element name="note" type="xsd:string" />
        <xsd:element name="datains" type="xsd:date" />
        <xsd:element name="dataup" type="xsd:date" />
      </xsd:all>
    </xsd:complexType>
  </xsd:schema>

```

```
<xsd:element name="id_usr_agg" type="xsd:string" />
</xsd:all>
</xsd:complexType>
<xsd:complexType name="scadenziari">
  <xsd:sequence>
    <xsd:element name="idscadenziari" type="tns:ArrayOfscadenziario" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfscadenziario">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:scadenziario[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="utentesel">
  <xsd:all>
    <xsd:element name="cognome" type="xsd:string" />
    <xsd:element name="nome" type="xsd:string" />
    <xsd:element name="cf" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="utentisel">
  <xsd:sequence>
    <xsd:element name="idutentisel" type="tns:ArrayOfutentesel" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfutentesel">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:utentesel[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="datoReferto">
  <xsd:all>
    <xsd:element name="id_referto" type="xsd:integer" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="id_utente" type="xsd:string" />
    <xsd:element name="tipodocumento" type="xsd:string" />
    <xsd:element name="identificativo" type="xsd:string" />
    <xsd:element name="data_documento" type="xsd:string" />
    <xsd:element name="data" type="xsd:date" />
    <xsd:element name="id_esito" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="elenco_referti">
  <xsd:sequence>
    <xsd:element name="idelencoReferto" type="tns:ArrayOfdatoReferto" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfdatoReferto">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:datoReferto[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="cartella">
  <xsd:all>
    <xsd:element name="paziente" type="tns:paziente" />
    <xsd:element name="periodiassistenza" type="tns:periodiassistenza" />
    <xsd:element name="contatti" type="tns:contatti" />
    <xsd:element name="esenzioni" type="tns:esenzioni" />
    <xsd:element name="allergie" type="tns:allergie" />
    <xsd:element name="stilidivita" type="tns:stilidivita" />
    <xsd:element name="ricoveri" type="tns:ricoveri" />
    <xsd:element name="prescrizioni" type="tns:prescrizioni" />
    <xsd:element name="farmaci" type="tns:farmaci" />
    <xsd:element name="vaccinazioni" type="tns:vaccinazioni" />
    <xsd:element name="certificati" type="tns:certificati" />
    <xsd:element name="monitoraggi" type="tns:monitoraggi" />
    <xsd:element name="esiti" type="tns:esiti" />
    <xsd:element name="patologie" type="tns:patologie" />
    <xsd:element name="diari" type="tns:diari" />
    <xsd:element name="anamnesifamiliari" type="tns:anamnesifamiliari" />
    <xsd:element name="scadenziari" type="tns:scadenziari" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="cartelle">
  <xsd:sequence>
    <xsd:element name="idcartelle" type="tns:ArrayOfcartella" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfcartella">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
```

```
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:cartella[]" minOccurs="0" maxOccurs="unbounded" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="paziente">
  <xsd:sequence>
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="cfpaziente" type="xsd:string" />
    <xsd:element name="cfmedico" type="xsd:string" />
    <xsd:element name="cognome" type="xsd:string" />
    <xsd:element name="nome" type="xsd:string" />
    <xsd:element name="sesso" type="tns:sesso" />
    <xsd:element name="datanascita" type="xsd:string" />
    <xsd:element name="istatcomune_na" type="xsd:string" />
    <xsd:element name="descricomune_na" type="xsd:string" />
    <xsd:element name="nazionalita_na" type="xsd:string" />
    <xsd:element name="tipologia" type="xsd:string" />
    <xsd:element name="codregassistito" type="xsd:string" />
    <xsd:element name="indirizzo" type="xsd:string" />
    <xsd:element name="cap_re" type="xsd:string" />
    <xsd:element name="comune_re" type="xsd:string" />
    <xsd:element name="frazione_re" type="xsd:string" />
    <xsd:element name="telefono" type="xsd:string" />
    <xsd:element name="ref_cogn" type="xsd:string" />
    <xsd:element name="ref_nom" type="xsd:string" />
    <xsd:element name="ref_tel" type="xsd:string" />
    <xsd:element name="sigla_res" type="xsd:string" />
    <xsd:element name="distretto_cod" type="xsd:string" />
    <xsd:element name="distretto_des" type="xsd:string" />
    <xsd:element name="gruppo_san" type="xsd:string" />
    <xsd:element name="rh" type="xsd:string" />
    <xsd:element name="sottogruppo_san" type="xsd:string" />
    <xsd:element name="email" type="xsd:string" />
    <xsd:element name="profess_istat" type="xsd:string" />
    <xsd:element name="professione" type="xsd:string" />
    <xsd:element name="statocivile" type="xsd:string" />
    <xsd:element name="titolo_studio" type="xsd:string" />
    <xsd:element name="ssn" type="xsd:string" />
    <xsd:element name="datadecesso" type="xsd:date" />
    <xsd:element name="decessoICD9" type="xsd:string" />
    <xsd:element name="codicistologico_dec" type="xsd:string" />
    <xsd:element name="desc_decess" type="xsd:string" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="privacy" type="xsd:integer" />
    <xsd:element name="donatore" type="xsd:integer" />
    <xsd:element name="accanimento" type="xsd:integer" />
    <xsd:element name="codreg_ass" type="xsd:string" />
    <xsd:element name="codasl_ass" type="xsd:string" />
    <xsd:element name="codcartext" type="xsd:string" />
    <xsd:element name="keycartella" type="xsd:string" />
    <xsd:element name="versionecartella" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
    <xsd:element name="nome_medico" type="xsd:string" />
    <xsd:element name="cognome_medico" type="xsd:string" />
    <xsd:element name="professione_medico" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="pazienti">
  <xsd:sequence>
    <xsd:element name="idpazienti" type="tns:ArrayOfPaziente" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfPaziente">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:paziente[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="pazienteSmall">
  <xsd:sequence>
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="cfpaziente" type="xsd:string" />
    <xsd:element name="cognome" type="xsd:string" />
    <xsd:element name="nome" type="xsd:string" />
    <xsd:element name="sesso" type="tns:sesso" />
    <xsd:element name="datanascita" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="pazientiSmall">
  <xsd:sequence>
    <xsd:element name="idpazientiSmall" type="tns:ArrayOfPazienteSmall" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfPazienteSmall">
```

```

<xsd:complexContent>
  <xsd:restriction base="SOAP-ENC:Array">
    <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:pazienteSmall[]" minOccurs="0" maxOccurs="unbounded" />
  </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="statusPaziente">
  <xsd:sequence>
    <xsd:element name="status" type="xsd:string" />
    <xsd:element name="init_time" type="xsd:date" />
    <xsd:element name="finish_time" type="xsd:date" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="periodoassistenza">
  <xsd:sequence>
    <xsd:element name="id_periodoassistenza" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="id_utente" type="xsd:string" />
    <xsd:element name="datascelta" type="xsd:date" />
    <xsd:element name="datarevoca" type="xsd:date" />
    <xsd:element name="causarevoca" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="periodiassistenza">
  <xsd:sequence>
    <xsd:element name="idperiodiassistenza" type="tns:ArrayOfperiodoassistenza" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfperiodoassistenza">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:periodoassistenza[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="contatto">
  <xsd:all>
    <xsd:element name="id_contatto" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="tipo" type="tns:tipoContatto" />
    <xsd:element name="cod_icd9" type="xsd:string" />
    <xsd:element name="descr_prob" type="xsd:string" />
    <xsd:element name="data_contatto" type="xsd:date" />
    <xsd:element name="risoluzione" type="xsd:string" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="id_patologia" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="contatti">
  <xsd:sequence>
    <xsd:element name="idcontatti" type="tns:ArrayOfcontatto" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfcontatto">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:contatto[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="esenzione">
  <xsd:sequence>
    <xsd:element name="id_esenzione" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="codese" type="xsd:string" />
    <xsd:element name="desc_ese" type="xsd:string" />
    <xsd:element name="cod_ese_reg" type="xsd:string" />
    <xsd:element name="diagnosi" type="xsd:string" />
    <xsd:element name="datastart" type="xsd:date" />
    <xsd:element name="datastop" type="xsd:date" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="codICD9probl" type="xsd:string" />
    <xsd:element name="id_patologia" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="esenzioni">
  <xsd:sequence>

```

```

<xsd:element name="idesenzioni" type="tns:ArrayOfesenzione" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfesenzione">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:esenzione[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="allergia">
  <xsd:sequence>
    <xsd:element name="id_allergia" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="codatc" type="xsd:string" />
    <xsd:element name="descr_molecola" type="xsd:string" />
    <xsd:element name="farm_scat" type="xsd:string" />
    <xsd:element name="descr_farm" type="xsd:string" />
    <xsd:element name="princ_att" type="xsd:string" />
    <xsd:element name="desc_princ_att" type="xsd:string" />
    <xsd:element name="evento" type="xsd:string" />
    <xsd:element name="data_ril" type="xsd:date" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="allergie">
  <xsd:sequence>
    <xsd:element name="idallergie" type="tns:ArrayOfallergia" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfallergia">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:allergia[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="stiledivita">
  <xsd:all>
    <xsd:element name="id_stiledivita" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:positiveInteger" />
    <xsd:element name="datavalutazione" type="xsd:date" />
    <xsd:element name="attivafisica" type="xsd:string" />
    <xsd:element name="statofumatore" type="xsd:string" />
    <xsd:element name="quantitafumo" type="xsd:string" />
    <xsd:element name="alcool" type="xsd:string" />
    <xsd:element name="notealcool" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="stilidivita">
  <xsd:sequence>
    <xsd:element name="idstilidivita" type="tns:ArrayOfstiledivita" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfstiledivita">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:stiledivita[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ricovero">
  <xsd:sequence>
    <xsd:element name="id_ricovero" type="xsd:string" />
    <xsd:element name="id_problemaingresso" type="xsd:string" />
    <xsd:element name="codiceICD9ingresso" type="xsd:string" />
    <xsd:element name="diagnosingresso" type="xsd:string" />
    <xsd:element name="nomestruitt" type="xsd:string" />
    <xsd:element name="reparto" type="xsd:string" />
    <xsd:element name="dataingresso" type="xsd:date" />
    <xsd:element name="datadimissione" type="xsd:date" />
    <xsd:element name="id_problemadimissione" type="xsd:string" />
    <xsd:element name="codiceICD9dimiss" type="xsd:string" />
    <xsd:element name="diagnosidimiss" type="xsd:string" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="ricoveri">
  <xsd:sequence>
    <xsd:element name="idricoveri" type="tns:ArrayOfricovero" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfricovero">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:ricovero[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="prescrizione">
  <xsd:sequence>
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="id_prescrizione" type="xsd:string" />
    <xsd:element name="codplg" type="xsd:string" />
    <xsd:element name="tipoplg" type="xsd:string" />
    <xsd:element name="codnomcart" type="xsd:string" />
    <xsd:element name="tipoprest" type="xsd:string" />
    <xsd:element name="codnomnaz" type="xsd:string" />
    <xsd:element name="codnomreg" type="xsd:string" />
    <xsd:element name="tiponomen" type="xsd:string" />
    <xsd:element name="descresame" type="xsd:string" />
    <xsd:element name="dataprescr" type="xsd:date" />
    <xsd:element name="dataesecuz" type="xsd:date" />
    <xsd:element name="id_patologia" type="xsd:string" />
    <xsd:element name="codicd9prescr" type="xsd:string" />
    <xsd:element name="diagnosi" type="xsd:string" />
    <xsd:element name="tiporicetta" type="xsd:string" />
    <xsd:element name="differibilita" type="xsd:string" />
    <xsd:element name="quantitaesame" type="xsd:string" />
    <xsd:element name="numesami" type="xsd:string" />
    <xsd:element name="concedibilita" type="xsd:string" />
    <xsd:element name="patologico" type="xsd:string" />
    <xsd:element name="suggerimento" type="xsd:string" />
    <xsd:element name="costoprest" type="xsd:string" />
    <xsd:element name="codesenzione" type="xsd:string" />
    <xsd:element name="followup" type="xsd:date" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="cup" type="xsd:string" />
    <xsd:element name="data_ins" type="xsd:date" />
    <xsd:element name="data_agg" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="prescrizioni">
  <xsd:sequence>
    <xsd:element name="idprescrizioni" type="tns:ArrayOfprescrizione" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfprescrizione">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:prescrizione[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="farmaco">
  <xsd:sequence>
    <xsd:element name="id_farmaco" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="tipoplg" type="xsd:string" />
    <xsd:element name="codplg" type="xsd:string" />
    <xsd:element name="id_patologia" type="xsd:string" />
    <xsd:element name="codatc" type="xsd:string" />
    <xsd:element name="descrizmolecola" type="xsd:string" />
    <xsd:element name="cdoprinatt" type="xsd:string" />
    <xsd:element name="descprinatt" type="xsd:string" />
    <xsd:element name="codaic" type="xsd:string" />
    <xsd:element name="descrfarm" type="xsd:string" />
    <xsd:element name="confezione" type="xsd:string" />
    <xsd:element name="dataprescr" type="xsd:date" />
    <xsd:element name="codicd9prescr" type="xsd:string" />
    <xsd:element name="descrdiagnosi" type="xsd:string" />
    <xsd:element name="tiporicetta" type="xsd:string" />
    <xsd:element name="noteric" type="xsd:string" />
    <xsd:element name="numconf" type="xsd:positiveInteger" />
    <xsd:element name="concedibilita" type="xsd:string" />
    <xsd:element name="continuativo" type="xsd:string" />
    <xsd:element name="posologia" type="xsd:string" />
    <xsd:element name="suggerimento" type="xsd:string" />
    <xsd:element name="costo" type="xsd:string" />
    <xsd:element name="codese" type="xsd:string" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
  </xsd:sequence>

```



```
<xsd:element name="id_usr_agg" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="farmaci">
  <xsd:sequence>
    <xsd:element name="idfarmaci" type="tns:ArrayOffarmaco" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOffarmaco">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:farmaco[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="vaccinazione">
  <xsd:sequence>
    <xsd:element name="id_vaccinazione" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="datavacc" type="xsd:date" />
    <xsd:element name="codvaccino" type="xsd:string" />
    <xsd:element name="descvaccino" type="xsd:string" />
    <xsd:element name="minsavaccino" type="xsd:string" />
    <xsd:element name="validita" type="xsd:string" />
    <xsd:element name="datarichiamo" type="xsd:date" />
    <xsd:element name="progressivo" type="xsd:string" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="data_ins" type="xsd:date" />
    <xsd:element name="data_agg" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="vaccinazioni">
  <xsd:sequence>
    <xsd:element name="idvaccinazioni" type="tns:ArrayOfvaccinazione" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfvaccinazione">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:vaccinazione[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="certificato">
  <xsd:sequence>
    <xsd:element name="id_certificato" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="id_tipo" type="xsd:string" />
    <xsd:element name="datacert" type="xsd:date" />
    <xsd:element name="datainizio" type="xsd:date" />
    <xsd:element name="datafine" type="xsd:date" />
    <xsd:element name="tipocert" type="xsd:string" />
    <xsd:element name="id_patologia" type="xsd:string" />
    <xsd:element name="codicd9probl" type="xsd:string" />
    <xsd:element name="descrizione" type="xsd:string" />
    <xsd:element name="testo" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="certificati">
  <xsd:sequence>
    <xsd:element name="idcertificati" type="tns:ArrayOfcertificato" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfcertificato">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:certificato[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="monitoraggio">
  <xsd:sequence>
    <xsd:element name="id_monitoraggio" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="datamon" type="xsd:date" />
    <xsd:element name="peso" type="xsd:positiveInteger" />
    <xsd:element name="altezza" type="xsd:positiveInteger" />
    <xsd:element name="circonferenza" type="xsd:positiveInteger" />
    <xsd:element name="sistolica" type="xsd:positiveInteger" />
    <xsd:element name="diastolica" type="xsd:positiveInteger" />
    <xsd:element name="misurazione" type="xsd:string" />
    <xsd:element name="frequenza" type="xsd:positiveInteger" />
    <xsd:element name="ritmo" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

```

<xsd:element name="note" type="xsd:string" />
<xsd:element name="datains" type="xsd:date" />
<xsd:element name="dataup" type="xsd:date" />
<xsd:element name="id_usr_agg" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="monitoraggi">
<xsd:sequence>
<xsd:element name="idmonitoraggi" type="tns:ArrayOfmonitoraggio" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfmonitoraggio">
<xsd:complexContent>
<xsd:restriction base="SOAP-ENC:Array">
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:monitoraggio[]" minOccurs="0" maxOccurs="unbounded" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="esito">
<xsd:sequence>
<xsd:element name="id_esito" type="xsd:string" />
<xsd:element name="id_paziente" type="xsd:string" />
<xsd:element name="id_prescrizione" type="xsd:string" />
<xsd:element name="codiceesito" type="xsd:string" />
<xsd:element name="descrizioneesito" type="xsd:string" />
<xsd:element name="dataesito" type="xsd:date" />
<xsd:element name="valoreesito" type="xsd:string" />
<xsd:element name="umesito" type="xsd:string" />
<xsd:element name="note" type="xsd:string" />
<xsd:element name="datains" type="xsd:date" />
<xsd:element name="dataup" type="xsd:date" />
<xsd:element name="id_usr_agg" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="esiti">
<xsd:sequence>
<xsd:element name="idesiti" type="tns:ArrayOfesito" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfesito">
<xsd:complexContent>
<xsd:restriction base="SOAP-ENC:Array">
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:esito[]" minOccurs="0" maxOccurs="unbounded" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="patologia">
<xsd:sequence>
<xsd:element name="id_paziente" type="xsd:string" />
<xsd:element name="id_patologia" type="xsd:string" />
<xsd:element name="icd9_problema" type="xsd:string" />
<xsd:element name="descr_problema" type="xsd:string" />
<xsd:element name="data_apertura" type="xsd:date" />
<xsd:element name="data_chiusura" type="xsd:date" />
<xsd:element name="risoluzione" type="xsd:string" />
<xsd:element name="note" type="xsd:string" />
<xsd:element name="datains" type="xsd:date" />
<xsd:element name="dataup" type="xsd:date" />
<xsd:element name="id_usr_agg" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="patologie">
<xsd:sequence>
<xsd:element name="idpatologie" type="tns:ArrayOfpatologia" minOccurs="0" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfpatologia">
<xsd:complexContent>
<xsd:restriction base="SOAP-ENC:Array">
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:patologia[]" minOccurs="0" maxOccurs="unbounded" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="diario">
<xsd:sequence>
<xsd:element name="id_diario" type="xsd:string" />
<xsd:element name="id_paziente" type="xsd:string" />
<xsd:element name="tipo" type="xsd:string" />
<xsd:element name="descrizione" type="xsd:string" />
<xsd:element name="data_rilev" type="xsd:date" />
<xsd:element name="id_patologia" type="xsd:string" />
<xsd:element name="codicd9probl" type="xsd:string" />
<xsd:element name="datains" type="xsd:date" />
<xsd:element name="dataup" type="xsd:date" />
<xsd:element name="id_usr_agg" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>

```

```
<xsd:complexType name="diari">
  <xsd:sequence>
    <xsd:element name="iddiari" type="tns:ArrayOfdiario" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfdiario">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:diario[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="anamnesifamiliare">
  <xsd:sequence>
    <xsd:element name="id_anamnesifamiliare" type="xsd:string" />
    <xsd:element name="id_paziente" type="xsd:string" />
    <xsd:element name="data_anamnesi" type="xsd:date" />
    <xsd:element name="grado_parentela" type="xsd:string" />
    <xsd:element name="id_patologia" type="xsd:string" />
    <xsd:element name="cod_icd9" type="xsd:string" />
    <xsd:element name="descr_prob" type="xsd:string" />
    <xsd:element name="note" type="xsd:string" />
    <xsd:element name="datains" type="xsd:date" />
    <xsd:element name="dataup" type="xsd:date" />
    <xsd:element name="id_usr_agg" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="anamnesifamiliari">
  <xsd:sequence>
    <xsd:element name="idanamnesifamiliari" type="tns:ArrayOfanamnesifamiliare" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfanamnesifamiliare">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:anamnesifamiliare[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="comprefolder">
  <xsd:sequence>
    <xsd:element name="data" type="xsd:base64Binary" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="notifica">
  <xsd:sequence>
    <xsd:element name="id_notifica" type="xsd:string" />
    <xsd:element name="id_utente_mitt" type="xsd:string" />
    <xsd:element name="cognome_mitt" type="xsd:string" />
    <xsd:element name="nome_mitt" type="xsd:string" />
    <xsd:element name="professione_mitt" type="xsd:string" />
    <xsd:element name="id_cartella_mitt" type="xsd:string" />
    <xsd:element name="cartella_mitt" type="xsd:string" />
    <xsd:element name="data_ins" type="xsd:string" />
    <xsd:element name="nuova" type="xsd:string" />
    <xsd:element name="paziente" type="tns:pazienteSmall" />
    <xsd:element name="id_tiponotifica" type="xsd:string" />
    <xsd:element name="descrizione_tiponotifica" type="xsd:string" />
    <xsd:element name="tipo_messaggio" type="xsd:string" />
    <xsd:element name="messaggio_notifica" type="xsd:string" />
    <xsd:element name="link_notifica" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="notifiche">
  <xsd:sequence>
    <xsd:element name="idnotifiche" type="tns:ArrayOfnotifica" minOccurs="0" maxOccurs="1" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfnotifica">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:notifica[]" minOccurs="0" maxOccurs="unbounded" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="tipoContatto">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="A" />
    <xsd:enumeration value="B" />
    <xsd:enumeration value="C" />
    <xsd:enumeration value="D" />
    <xsd:enumeration value="E" />
    <xsd:enumeration value="F" />
    <xsd:enumeration value="G" />
    <xsd:enumeration value="Z" />
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:simpleType>
<xsd:simpleType name="tipoDifferibilita">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="U" />
    <xsd:enumeration value="B" />
    <xsd:enumeration value="D" />
    <xsd:enumeration value="P" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="sesso">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="M" />
    <xsd:enumeration value="F" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="sino">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="S" />
    <xsd:enumeration value="N" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="tipologiaAssistito">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="UE" />
    <xsd:enumeration value="EE" />
    <xsd:enumeration value="NA" />
    <xsd:enumeration value="ND" />
    <xsd:enumeration value="NE" />
    <xsd:enumeration value="NX" />
    <xsd:enumeration value="AD" />
    <xsd:enumeration value="ST" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="tipoEventoAllergia">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="A" />
    <xsd:enumeration value="I" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="gruppoSanguigno">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="O" />
    <xsd:enumeration value="A" />
    <xsd:enumeration value="B" />
    <xsd:enumeration value="AB" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="rh">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="+" />
    <xsd:enumeration value="-" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="vf">
  <xsd:restriction base="xsd:integer">
    <xsd:enumeration value="1" />
    <xsd:enumeration value="0" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
</types>
<message name="expressfolderRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="data" type="tns:comprexfolder" />
</message>
<message name="expressfolderResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="loginRequest">
  <part name="user" type="xsd:string" />
  <part name="pwd" type="xsd:string" />
</message>
<message name="loginResponse">
  <part name="return" type="tns:utente" />
</message>
<message name="loginGuardiaMedicaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="dataInizio" type="xsd:string" />
  <part name="dataFine" type="xsd:string" />
  <part name="id_postazione" type="xsd:string" />
</message>
<message name="loginGuardiaMedicaResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="logoutRequest">
```

```
<part name="token" type="xsd:string" />
</message>
<message name="logoutResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="search_gruppoRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
</message>
<message name="search_gruppoResponse">
  <part name="return" type="tns:utentisel" />
</message>
<message name="search_num_pazienteRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
  <part name="data" type="xsd:string" />
</message>
<message name="search_num_pazienteResponse">
  <part name="return" type="tns:numeroDatiPaziente" />
</message>
<message name="searchRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="searchResponse">
  <part name="return" type="tns:cartelle" />
</message>
<message name="search_pazientiRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
  <part name="cognome" type="xsd:string" />
  <part name="nome" type="xsd:string" />
  <part name="cfpaziente" type="xsd:string" />
  <part name="codicetessera" type="xsd:string" />
  <part name="datadinascita" type="xsd:string" />
  <part name="tiporicerca" type="xsd:string" />
  <part name="utentisel" type="tns:utentisel" />
</message>
<message name="search_pazientiResponse">
  <part name="return" type="tns:pazienti" />
</message>
<message name="search_periodiassistenzaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_periodiassistenzaResponse">
  <part name="return" type="tns:periodiassistenza" />
</message>
<message name="search_contattiRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_contattiResponse">
  <part name="return" type="tns:contatti" />
</message>
<message name="search_esenzioniRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_esenzioniResponse">
  <part name="return" type="tns:esenzioni" />
</message>
<message name="search_stilidivitaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_stilidivitaResponse">
  <part name="return" type="tns:stilidivita" />
</message>
<message name="search_prescrizioniRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_prescrizioniResponse">
  <part name="return" type="tns:prescrizioni" />
</message>
<message name="search_farmaciRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
```

```
<part name="id_paziente" type="xsd:string" />
</message>
<message name="search_farmaciResponse">
  <part name="return" type="tns:farmaci" />
</message>
<message name="search_vaccinazioniRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_vaccinazioniResponse">
  <part name="return" type="tns:vaccinazioni" />
</message>
<message name="search_certificatiRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_certificatiResponse">
  <part name="return" type="tns:certificati" />
</message>
<message name="search_monitoraggiRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_monitoraggiResponse">
  <part name="return" type="tns:monitoraggi" />
</message>
<message name="search_esitiRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_esitiResponse">
  <part name="return" type="tns:esiti" />
</message>
<message name="search_patologieRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_patologieResponse">
  <part name="return" type="tns:patologie" />
</message>
<message name="search_pazienti_smallRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
  <part name="cognome" type="xsd:string" />
  <part name="nome" type="xsd:string" />
  <part name="cfpaziente" type="xsd:string" />
  <part name="datadinascita" type="xsd:string" />
  <part name="tiporicerca" type="xsd:string" />
  <part name="utentisel" type="tns:utentisel" />
</message>
<message name="search_pazienti_smallResponse">
  <part name="return" type="tns:pazientiSmall" />
</message>
<message name="search_status_pazienteRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_status_pazienteResponse">
  <part name="return" type="tns:statusPaziente" />
</message>
<message name="search_diariRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_diariResponse">
  <part name="return" type="tns:diari" />
</message>
<message name="search_allergieRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_allergieResponse">
  <part name="return" type="tns:allergie" />
</message>
<message name="search_ricoveriRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
```

```
</message>
<message name="search_ricoveriResponse">
  <part name="return" type="tns:ricoveri" />
</message>
<message name="search_scadenziariRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
</message>
<message name="search_scadenziariResponse">
  <part name="return" type="tns:scadenziari" />
</message>
<message name="writeRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="cartella" type="tns:cartella" />
</message>
<message name="writeResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_periodoassistenzaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="periodoassistenza" type="tns:periodoassistenza" />
</message>
<message name="write_periodoassistenzaResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_anagraficaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="paziente" type="tns:paziente" />
</message>
<message name="write_anagraficaResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_contattoRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="contatto" type="tns:contatto" />
</message>
<message name="write_contattoResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_esenzioneRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="esenzione" type="tns:esenzione" />
</message>
<message name="write_esenzioneResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_allergiaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="allergia" type="tns:allergia" />
</message>
<message name="write_allergiaResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_stiledivitaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="stiledivita" type="tns:stiledivita" />
</message>
<message name="write_stiledivitaResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_ricoveroRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="ricovero" type="tns:ricovero" />
</message>
<message name="write_ricoveroResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_prescrizioneRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="prescrizione" type="tns:prescrizione" />
</message>
<message name="write_prescrizioneResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_farmacoRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
```

```
<part name="farmaco" type="tns:farmaco" />
</message>
<message name="write_farmacoResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_vaccinazioneRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="vaccinazione" type="tns:vaccinazione" />
</message>
<message name="write_vaccinazioneResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_certificatoRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="certificato" type="tns:certificato" />
</message>
<message name="write_certificatoResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_monitoraggioRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="monitoraggio" type="tns:monitoraggio" />
</message>
<message name="write_monitoraggioResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_esitoRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="esito" type="tns:esito" />
</message>
<message name="write_esitoResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_patologiaRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="patologia" type="tns:patologia" />
</message>
<message name="write_patologiaResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_diarioRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="diario" type="tns:diario" />
</message>
<message name="write_diarioResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="write_scadenziarioRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="scadenziari" type="tns:scadenziari" />
</message>
<message name="write_scadenziarioResponse">
  <part name="return" type="xsd:string" />
</message>
<message name="invio_RefertoPazienteRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
  <part name="id_tipodocumento" type="xsd:integer" />
  <part name="identificativo" type="xsd:string" />
  <part name="data_documento" type="xsd:date" />
  <part name="referto" type="xsd:base64Binary" />
  <part name="id_esito" type="xsd:integer" />
</message>
<message name="invio_RefertoPazienteResponse">
  <part name="return" type="xsd:integer" />
</message>
<message name="elenco_RefertiPazienteRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
  <part name="id_tipodocumento" type="xsd:integer" />
</message>
<message name="elenco_RefertiPazienteResponse">
  <part name="return" type="tns:elenco_referti" />
</message>
<message name="lettura_RefertoPazienteRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
```



```
<part name="id_referto" type="xsd:integer" />
</message>
<message name="lettura_RefertoPazienteResponse">
  <part name="return" type="xsd:base64Binary" />
</message>
<message name="read_notificheRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_utente" type="xsd:string" />
</message>
<message name="read_notificheResponse">
  <part name="return" type="tns:notifiche" />
</message>
<message name="delete_recordRequest">
  <part name="token" type="xsd:string" />
  <part name="keycartella" type="xsd:string" />
  <part name="id_paziente" type="xsd:string" />
  <part name="nome_tabella" type="xsd:string" />
  <part name="record" type="xsd:string" />
</message>
<message name="delete_recordResponse">
  <part name="return" type="xsd:string" />
</message>
<portType name="FIMMGwsdlPortType">
  <operation name="expressfolder">
    <documentation>XXXX</documentation>
    <input message="tns:expressfolderRequest" />
    <output message="tns:expressfolderResponse" />
  </operation>
  <operation name="login">
    <documentation>FIMMG Test login a cloud fimmg</documentation>
    <input message="tns:loginRequest" />
    <output message="tns:loginResponse" />
  </operation>
  <operation name="loginGuardiaMedica">
    <documentation>FIMMG Test login gm a cloud fimmg</documentation>
    <input message="tns:loginGuardiaMedicaRequest" />
    <output message="tns:loginGuardiaMedicaResponse" />
  </operation>
  <operation name="logout">
    <documentation>FIMMG logout cloud fimmg</documentation>
    <input message="tns:logoutRequest" />
    <output message="tns:logoutResponse" />
  </operation>
  <operation name="search_gruppo">
    <documentation>FIMMG Ricerca i CF dei colleghi a cui sono collegato</documentation>
    <input message="tns:search_gruppoRequest" />
    <output message="tns:search_gruppoResponse" />
  </operation>
  <operation name="search_num_paziente">
    <documentation>Restituisce la cartella completa dei pazienti trovati.</documentation>
    <input message="tns:search_num_pazienteRequest" />
    <output message="tns:search_num_pazienteResponse" />
  </operation>
  <operation name="search">
    <documentation>Restituisce la cartella completa dei pazienti trovati.</documentation>
    <input message="tns:searchRequest" />
    <output message="tns:searchResponse" />
  </operation>
  <operation name="search_pazienti">
    <documentation>Ricerca un paziente per uno dei criteri (id paziente,cognome,nome,codice fiscale, codice tessera, data di nascita)
    e restituisce la cartella corrispondente se autorizzato.</documentation>
    <input message="tns:search_pazientiRequest" />
    <output message="tns:search_pazientiResponse" />
  </operation>
  <operation name="search_periodiassistenza">
    <documentation>Restituzione il periodo di assistenza di un paziente</documentation>
    <input message="tns:search_periodiassistenzaRequest" />
    <output message="tns:search_periodiassistenzaResponse" />
  </operation>
  <operation name="search_contatti">
    <documentation>Restituzione dei contatti di un paziente</documentation>
    <input message="tns:search_contattiRequest" />
    <output message="tns:search_contattiResponse" />
  </operation>
  <operation name="search_esenzioni">
    <documentation>Restituzione delle esenzioni di un paziente</documentation>
    <input message="tns:search_esenzioniRequest" />
    <output message="tns:search_esenzioniResponse" />
  </operation>
  <operation name="search_stilidivita">
    <documentation>Restituzione degli stili di vita di un paziente</documentation>
    <input message="tns:search_stilidivitaRequest" />
    <output message="tns:search_stilidivitaResponse" />
  </operation>
  <operation name="search_prescrizioni">
    <documentation>Restituzione delle prescrizioni di un paziente</documentation>
  </operation>
</portType>
```

```
<input message="tns:search_prescrizioniRequest" />
<output message="tns:search_prescrizioniResponse" />
</operation>
<operation name="search_farmaci">
  <documentation>Restituzione dei farmaci di un paziente</documentation>
  <input message="tns:search_farmaciRequest" />
  <output message="tns:search_farmaciResponse" />
</operation>
<operation name="search_vaccinazioni">
  <documentation>Restituzione delle vaccinazioni di un paziente</documentation>
  <input message="tns:search_vaccinazioniRequest" />
  <output message="tns:search_vaccinazioniResponse" />
</operation>
<operation name="search_certificati">
  <documentation>Restituzione dei certificati di un paziente</documentation>
  <input message="tns:search_certificatiRequest" />
  <output message="tns:search_certificatiResponse" />
</operation>
<operation name="search_monitoraggi">
  <documentation>Restituzione dei monitoraggi di un paziente</documentation>
  <input message="tns:search_monitoraggiRequest" />
  <output message="tns:search_monitoraggiResponse" />
</operation>
<operation name="search_esiti">
  <documentation>Restituzione degli esiti di un paziente</documentation>
  <input message="tns:search_esitiRequest" />
  <output message="tns:search_esitiResponse" />
</operation>
<operation name="search_patologie">
  <documentation>Restituzione delle patologie di un paziente</documentation>
  <input message="tns:search_patologieRequest" />
  <output message="tns:search_patologieResponse" />
</operation>
<operation name="search_pazienti_small">
  <documentation>Ricerca un paziente per uno dei criteri (id paziente,cognome,nome,codice fiscale, codice tessera, data di nascita)
  e restituisce la cartella corrispondente se autorizzato.</documentation>
  <input message="tns:search_pazienti_smallRequest" />
  <output message="tns:search_pazienti_smallResponse" />
</operation>
<operation name="search_status_paziente">
  <documentation>Ricerca lo status, data inizio e data fine rapporto con il medico.</documentation>
  <input message="tns:search_status_pazienteRequest" />
  <output message="tns:search_status_pazienteResponse" />
</operation>
<operation name="search_diari">
  <documentation>Restituzione del diario completo di un paziente</documentation>
  <input message="tns:search_diariRequest" />
  <output message="tns:search_diariResponse" />
</operation>
<operation name="search_allergie">
  <documentation>Restituzione delle allergie e intolleranze di un paziente</documentation>
  <input message="tns:search_allergieRequest" />
  <output message="tns:search_allergieResponse" />
</operation>
<operation name="search_ricoveri">
  <documentation>Restituzione dei ricoveri di un paziente</documentation>
  <input message="tns:search_ricoveriRequest" />
  <output message="tns:search_ricoveriResponse" />
</operation>
<operation name="search_scadenziari">
  <documentation>Restituzione dello scadenziario completo di un paziente</documentation>
  <input message="tns:search_scadenziariRequest" />
  <output message="tns:search_scadenziariResponse" />
</operation>
<operation name="write">
  <documentation>Scrivo nel db i dati cartella di un paziente criptati con il keycartella.</documentation>
  <input message="tns:writeRequest" />
  <output message="tns:writeResponse" />
</operation>
<operation name="write_periodoassistenza">
  <documentation>Scrivo nel db i dati relativi al periodo di assistenza di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_periodoassistenzaRequest" />
  <output message="tns:write_periodoassistenzaResponse" />
</operation>
<operation name="write_anagrafica">
  <documentation>Scrivo nel db i dati relativi al paziente criptati con il keycartella.</documentation>
  <input message="tns:write_anagraficaRequest" />
  <output message="tns:write_anagraficaResponse" />
</operation>
<operation name="write_contatto">
  <documentation>Scrivo nel db i dati relativi ai contatti di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_contattoRequest" />
  <output message="tns:write_contattoResponse" />
</operation>
<operation name="write_esenzione">
  <documentation>Scrivo nel db i dati relativi alle esenzioni di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_esenzioneRequest" />
  <output message="tns:write_esenzioneResponse" />
</operation>
```

```
</output message="tns:write_esenzioneResponse" />
</operation>
<operation name="write_allergia">
  <documentation>Scivo nel db i dati relativi alle allergie di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_allergiaRequest" />
  <output message="tns:write_allergiaResponse" />
</operation>
<operation name="write_stiledivita">
  <documentation>Scivo nel db i dati relativi agli stili di vita di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_stiledivitaRequest" />
  <output message="tns:write_stiledivitaResponse" />
</operation>
<operation name="write_ricovero">
  <documentation>Scivo nel db i dati relativi ai ricovero di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_ricoveroRequest" />
  <output message="tns:write_ricoveroResponse" />
</operation>
<operation name="write_prescrizione">
  <documentation>Scivo nel db i dati relativi alle prescrizioni di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_prescrizioneRequest" />
  <output message="tns:write_prescrizioneResponse" />
</operation>
<operation name="write_farmaco">
  <documentation>Scivo nel db i dati relativi a un farmaco di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_farmacoRequest" />
  <output message="tns:write_farmacoResponse" />
</operation>
<operation name="write_vaccinazione">
  <documentation>Scivo nel db i dati relativi ad una vaccinazione di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_vaccinazioneRequest" />
  <output message="tns:write_vaccinazioneResponse" />
</operation>
<operation name="write_certificato">
  <documentation>Scivo nel db i dati relativi a un certificato di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_certificatoRequest" />
  <output message="tns:write_certificatoResponse" />
</operation>
<operation name="write_monitoraggio">
  <documentation>Scivo nel db i dati relativi a un monitoraggio di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_monitoraggioRequest" />
  <output message="tns:write_monitoraggioResponse" />
</operation>
<operation name="write_esito">
  <documentation>Scivo nel db i dati relativi all esito di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_esitoRequest" />
  <output message="tns:write_esitoResponse" />
</operation>
<operation name="write_patologia">
  <documentation>Scivo nel db i dati relativi alla patologia di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_patologiaRequest" />
  <output message="tns:write_patologiaResponse" />
</operation>
<operation name="write_diario">
  <documentation>Scivo nel db i dati relativi al diario di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_diarioRequest" />
  <output message="tns:write_diarioResponse" />
</operation>
<operation name="write_scadenziario">
  <documentation>Scivo nel db i dati relativi al piano terapeutico di un paziente criptati con il keycartella.</documentation>
  <input message="tns:write_scadenziarioRequest" />
  <output message="tns:write_scadenziarioResponse" />
</operation>
<operation name="invio_RefertoPaziente">
  <documentation>Invia il referto di un paziente</documentation>
  <input message="tns:invio_RefertoPazienteRequest" />
  <output message="tns:invio_RefertoPazienteResponse" />
</operation>
<operation name="elenco_RefertiPaziente">
  <documentation>Restituzione delle notifiche relative ad un utente utente</documentation>
  <input message="tns:elenco_RefertiPazienteRequest" />
  <output message="tns:elenco_RefertiPazienteResponse" />
</operation>
<operation name="lettura_RefertoPaziente">
  <documentation> Lettura di un referto di un paziente</documentation>
  <input message="tns:lettura_RefertoPazienteRequest" />
  <output message="tns:lettura_RefertoPazienteResponse" />
</operation>
<operation name="read_notifiche">
  <documentation>Restituzione delle notifiche relative ad un utente utente</documentation>
  <input message="tns:read_notificheRequest" />
  <output message="tns:read_notificheResponse" />
</operation>
<operation name="delete_record">
  <documentation>Cancella un record inserito dal programma autorizzato</documentation>
  <input message="tns:delete_recordRequest" />
  <output message="tns:delete_recordResponse" />
</operation>
```

```
</portType>
<binding name="FIMMGwsdlBinding" type="tns:FIMMGwsdlPortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="expressfolder">
    <soap:operation soapAction="urn:FIMMGwsdl#expressfolder" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="login">
    <soap:operation soapAction="urn:FIMMGwsdl#login" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="loginGuardiaMedica">
    <soap:operation soapAction="urn:FIMMGwsdl#loginGuardiaMedica" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="logout">
    <soap:operation soapAction="urn:FIMMGwsdl#logout" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="search_gruppo">
    <soap:operation soapAction="urn:FIMMGwsdl#search_gruppo" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="search_num_paziente">
    <soap:operation soapAction="urn:FIMMGwsdl#search_num_paziente" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="search">
    <soap:operation soapAction="urn:FIMMGwsdl#search" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="search_pazienti">
    <soap:operation soapAction="urn:FIMMGwsdl#search_pazienti" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="search_periodiassistenza">
    <soap:operation soapAction="urn:FIMMGwsdl#search_periodiassistenza" style="rpc" />
    <input>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
  <operation name="search_contatti">
    <soap:operation soapAction="urn:FIMMGwsdl#search_contatti" style="rpc" />
    <input>
```

```
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
</output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_esenzioni">
<soap:operation soapAction="urn:FIMMGwsdl#search_esenzioni" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_stilidivita">
<soap:operation soapAction="urn:FIMMGwsdl#search_stilidivita" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_prescrizioni">
<soap:operation soapAction="urn:FIMMGwsdl#search_prescrizioni" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_farmaci">
<soap:operation soapAction="urn:FIMMGwsdl#search_farmaci" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_vaccinazioni">
<soap:operation soapAction="urn:FIMMGwsdl#search_vaccinazioni" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_certificati">
<soap:operation soapAction="urn:FIMMGwsdl#search_certificati" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_monitoraggi">
<soap:operation soapAction="urn:FIMMGwsdl#search_monitoraggi" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_esiti">
<soap:operation soapAction="urn:FIMMGwsdl#search_esiti" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="search_patologie">
<soap:operation soapAction="urn:FIMMGwsdl#search_patologie" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
```

```
<operation name="search_pazienti_small">
  <soap:operation soapAction="urn:FIMMGwsdl#search_pazienti_small" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="search_status_paziente">
  <soap:operation soapAction="urn:FIMMGwsdl#search_status_paziente" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="search_diari">
  <soap:operation soapAction="urn:FIMMGwsdl#search_diari" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="search_allergie">
  <soap:operation soapAction="urn:FIMMGwsdl#search_allergie" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="search_ricoveri">
  <soap:operation soapAction="urn:FIMMGwsdl#search_ricoveri" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="search_scadenziari">
  <soap:operation soapAction="urn:FIMMGwsdl#search_scadenziari" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="write">
  <soap:operation soapAction="urn:FIMMGwsdl#write" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="write_periodoassistenza">
  <soap:operation soapAction="urn:FIMMGwsdl#write_periodoassistenza" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="write_anagrafica">
  <soap:operation soapAction="urn:FIMMGwsdl#write_anagrafica" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </output>
</operation>
<operation name="write_contatto">
  <soap:operation soapAction="urn:FIMMGwsdl#write_contatto" style="rpc" />
  <input>
    <soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
  </input>
  <output>
```

```
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_esenzione">
<soap:operation soapAction="urn:FIMMGwsdl#write_esenzione" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_allergia">
<soap:operation soapAction="urn:FIMMGwsdl#write_allergia" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_stiledivita">
<soap:operation soapAction="urn:FIMMGwsdl#write_stiledivita" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_ricovero">
<soap:operation soapAction="urn:FIMMGwsdl#write_ricovero" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_prescrizione">
<soap:operation soapAction="urn:FIMMGwsdl#write_prescrizione" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_farmaco">
<soap:operation soapAction="urn:FIMMGwsdl#write_farmaco" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_vaccinazione">
<soap:operation soapAction="urn:FIMMGwsdl#write_vaccinazione" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_certificato">
<soap:operation soapAction="urn:FIMMGwsdl#write_certificato" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_monitoraggio">
<soap:operation soapAction="urn:FIMMGwsdl#write_monitoraggio" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_esito">
<soap:operation soapAction="urn:FIMMGwsdl#write_esito" style="rpc" />
<input>
```

```
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
</output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_patologia">
<soap:operation soapAction="urn:FIMMGwsdl#write_patologia" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_diario">
<soap:operation soapAction="urn:FIMMGwsdl#write_diario" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="write_scadenziario">
<soap:operation soapAction="urn:FIMMGwsdl#write_scadenziario" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="invio_RefertoPaziente">
<soap:operation soapAction="urn:FIMMGwsdl#invio_RefertoPaziente" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="elenco_RefertiPaziente">
<soap:operation soapAction="urn:FIMMGwsdl#elenco_RefertiPaziente" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="lettura_RefertoPaziente">
<soap:operation soapAction="urn:FIMMGwsdl#lettura_RefertoPaziente" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="read_notifiche">
<soap:operation soapAction="urn:FIMMGwsdl#read_notifiche" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="delete_record">
<soap:operation soapAction="urn:FIMMGwsdl#delete_record" style="rpc" />
<input>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="urn:FIMMGwsdl" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
<service name="FIMMGwsdl">
<port name="FIMMGwsdlPort" binding="tns:FIMMGwsdlBinding">
<soap:address location="http://cloudtest.fimmg.org/wsdl.php" />
</port>
</service>
</definitions>
```