



Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



in collaborazione tecnico-scientifica con
l'Università degli Studi di Salerno – Dipartimento di Informatica

Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013

PATIENT DATA WEB SERVICE
PER LA TRASMISSIONE DELLE INFORMAZIONI DI BASE
DEGLI ASSISTITI ARRUOLATI DAL MMG
NELLA GESTIONE INTEGRATA DEL PAZIENTE DIABETICO

VERSIONE 3.0





Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



SOMMARIO

Revisioni del documento	3
Notazioni adottate nel documento	3
Contatti	4
1 INTRODUZIONE	4
2 SERVIZIO PER LA TRASMISSIONE DEI DATI	5
2.1 Endpoint	5
2.2 Operation del servizio	6
2.2.1 Operation getAuthorizationToken	6
2.2.2 Operation doWriteAnagraficData	7
2.2.3 Operation doWritePatientEnrollmentDate.....	8
2.2.4 Operation doWritePatientDisenrollmentDate.....	8
2.2.5 Operation doWriteAnamnesticData	9
2.2.6 Operation doWriteExemptionCode.....	10
2.2.7 Operation doWriteComplications	10
2.2.8 Operation getVersion	11
2.3 Tipi di dati complessi	12
2.3.1 Tipo authenticationToken.....	12
2.3.2 Tipo authorizationToken.....	12
2.3.3 Tipo patientAnagraficData.....	12
2.3.4 Tipo patientAnagraficDataResult.....	14
2.3.5 Tipo patientAnamnesticData	14
2.3.6 Tipo patientExemption.....	17
2.3.7 Tipo patientComplications.....	17
2.4 Tipi di eccezioni	20
2.4.1 Illegal argument exception	20
2.4.2 AIR application exception	20
2.4.3 Data validation exception.....	20
3 WSDL	21

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

Revisioni del documento



In base a ciò che viene modificato nel documento viene inserita la motivazione dell'aggiornamento, in modo che il lettore possa immediatamente sapere:

- se sono state variate le specifiche tecniche (AGGIORNAMENTO TECNICO) e, di conseguenza, deve variare il software affinché sia funzionante (ad esempio cambiamenti nei tracciati record, nuovi valori di campi flag, etc.),
- se sono stati pubblicati nuovi servizi (AGGIORNAMENTO PER NUOVO SERVIZIO) o aggiunte nuove funzionalità di un servizio già sviluppato, non presenti nelle versioni precedenti: tutto ciò esposto in precedenza rimane invariato,
- se sono stati solamente meglio specificati alcuni argomenti già trattati nelle versioni precedenti (AGGIORNAMENTO CONCETTUALE), che non hanno però riflesso nella produzione del software (ad es. nuovo flusso del processo).

VERSIONE	DATA MODIFICA	DESCRIZIONE
1.0	24 giugno 2015	Prima pubblicazione del documento
2.0	4 gennaio 2016	AGGIORNAMENTO PER NUOVO SERVIZIO (<i>getAuthorizationToken</i>) AGGIORNAMENTO TECNICO (su tutte le operation)
3.0	31 marzo 2016	AGGIORNAMENTO TECNICO (aggiunta di campi nel tipo <i>patientComplications</i>)
4.0	29 agosto 2016	AGGIORNAMENTO DGR 163/2016

Notazioni adottate nel documento

Leggenda obbligatorietà dei campi	
OBB	Sempre obbligatorio
FAC	Sempre facoltativo
NBB	Obbligatorio se si verificano le condizioni indicate

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

Contatti

Per ogni eventualità si prega di scrivere a: report@soresa.it

1 INTRODUZIONE

Con il Decreto commissariale n. 87 del 24 luglio 2013, pubblicato sul BURC n. 46 del 22 agosto 2013, è stato approvato, ai sensi dell'art. 1 del decreto legge 13 settembre 2012, n. 158, convertito con la legge 8 novembre 2012, n. 189 l'Accordo Integrativo Regionale per la Medicina Generale (AIR).

Tale Accordo, definisce all'art. 9 sul Disease Management del Diabete Mellito uno specifico percorso diagnostico-terapeutico-assistenziale per il follow-up del paziente diabetico di tipo 2 che include, tra l'altro, le modalità di effettuazione della Gestione Integrata del Diabete a livello regionale per i pazienti privi di complicanze d'organo o con complicanze stabilizzate.



L'Agenzia Regionale Sanitaria (ARSan) era stata individuata per l'accoglienza dei dati inviati dai Medici di Medicina Generale ed ha realizzato, in collaborazione tecnico-scientifica con il Dipartimento di Informatica dell'Università degli Studi di Salerno, un sistema informatizzato per il monitoraggio della Gestione Integrata del Diabete, secondo quanto previsto dall'art. 9 dell'Accordo Integrativo Regionale per la Medicina Generale.

La legge regionale 23 dicembre 2015, n. 20, pubblicata sul BURC n. 78 del 23.12.2015, ha previsto la soppressione dell'ARSan, assegnando le relative funzioni alle competenti strutture amministrative della Regione Campania.

Il team di progetto ha garantito le attività di implementazione e messa in esercizio ha proseguito nella gestione del sistema informatizzato installato presso il Datacenter della ex ARSan sotto la direzione della competente UOD Assistenza Territoriale della Direzione Generale Tutela della Salute ed il Coordinamento del Sistema Sanitario Regionale, giusta nota prot. 898992/2015 della Direzione Generale Tutela della Salute.

La Regione Campania con la deliberazione n. 163 del 19 aprile 2016 ha disposto che la Società So.Re.Sa. S.p.a. svolga funzioni di supporto all'attività istruttoria per conto della Direzione Generale Tutela della Salute e Coordinamento del SSR, nel funzionamento del Sistema Informativo Sanitario in ordine anche alla gestione dei dati di cui alla Gestione integrata del diabete.

Il seguente documento illustra la modalità con cui tramite appositi web service i report individuali dei pazienti diabetici arruolati attestanti la partecipazione del Medico di Medicina Generale alla Gestione Integrata del paziente diabetico possono essere inviati al sistema di accoglienza della Regione Campania.

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

2 SERVIZIO PER LA TRASMISSIONE DEI DATI

Il servizio consente ai Medici di Medicina Generale di inviare e di mantenere aggiornati sulla piattaforma regionale i dati anagrafici ed anamnestici dei pazienti reclutati nella Gestione Integrata del Diabete.

Il caricamento dei dati sulla piattaforma regionale avviene in modo semi-automatizzato, attraverso componenti di integrazione appositamente predisposti per il proprio sistema gestionale ambulatoriale.

Le comunicazioni con i componenti di integrazione avvengono su un canale di comunicazione criptato https al fine di garantire la riservatezza delle informazioni scambiate, nel rispetto della normativa sulla privacy.



2.1 Endpoint

Endpoint di test

<https://air.di.unisa.it:8443/AIR-EJBWS/PatientDataService/PatientDataWebService?wsdl>

Endpoint di esercizio

<https://airdiabete.soresa.campania.it:8443/AIR-EJBWS/PatientDataService/PatientDataWebService?wsdl>

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

2.2 Operation del servizio

Le operation del servizio, dettagliate nei paragrafi che seguono, sono descritte sinteticamente nella Tabella che segue.

Si precisa che nel codice WSDL i campi dei messaggi e dei tipi di dati seguono sempre l'ordine alfabetico e tale ordine deve essere rispettato nella realizzazione dei componenti di integrazione; nelle descrizioni che seguono tuttavia talvolta tale ordine potrebbe non essere stato rispettato al fine di favorire la leggibilità della struttura del messaggio o del tipo di dati nella sua interezza.



OPERATION	DESCRIZIONE
getAuthorizationToken	Consente di inviare le credenziali di autenticazione (authentication token) al fine di poter acquisire il token di autorizzazione (authorization token) necessario per poter eseguire tutte le altre operation
doWriteAnagraficData	Consente di registrare per la prima volta e di aggiornare successivamente i dati anagrafici del singolo assistito arruolato nel programma di Gestione Integrata del diabete
doWritePatientEnrollmentDate	Consente di indicare la data di ingresso del singolo assistito nel programma di Gestione Integrata del diabete
doWritePatientDisenrollmentDate	Consente di indicare la data di uscita del singolo assistito dal programma di Gestione Integrata del diabete
doWriteAnamnesticData	Consente di inviare/aggiornare i dati anamnestic del singolo assistito arruolato nel programma di Gestione Integrata del diabete
doWriteExemptionCode	Consente di inviare/aggiornare le esenzioni del singolo assistito arruolato nel programma di Gestione Integrata del diabete.
doWriteComplications	Consente di inviare/aggiornare le informazioni sulle complicazioni del singolo assistito arruolato nel programma di Gestione Integrata del diabete
getVersion	Consente di verificare la versione del web service in esercizio

2.2.1 Operation getAuthorizationToken

Consente di ottenere l'authentication token necessario di inviare/aggiornare i dati anagrafici del singolo paziente arruolato nel programma di Gestione Integrata del diabete

2.2.1.1 Input: message getAuthorizationToken

Descrizione degli elementi costitutivi del messaggio di richiesta

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

CAMPO	DESCRIZIONE	VINCOLI
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
AuthenticationToken :authenticationToken	Contiene i campi necessari all'autenticazione del MMG e del componente di integrazione utilizzato	OBB

2.2.1.2 Output: message getAuthorizationTokenResponse

Descrizione degli elementi costitutivi del messaggio di risposta

CAMPO	DESCRIZIONE	VINCOLI
return :authorizationToken	Contiene l'authorization token necessario per poter eseguire tutte le altre operation	NBB (Obbligatorio solo in caso di esito positivo dell'operation)

2.2.2 Operation doWriteAnagraficData

Consente di registrare per la prima volta e di aggiornare successivamente i dati anagrafici del singolo paziente arruolato nel programma di Gestione Integrata del diabete



2.2.2.1 Input: message doWriteAnagraficData

Descrizione degli elementi costitutivi del messaggio di richiesta

CAMPO	DESCRIZIONE	VINCOLI
securityToken :authorizationToken	Contiene il token di autorizzazione per l'invio dei dati	OBB
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
patientAnagraphicData :patientAnagraphicData	Contiene i dati anagrafici del singolo paziente	OBB

2.2.2.2 Output: message doWriteAnagraficDataResponse

Descrizione degli elementi costitutivi del messaggio di risposta

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

CAMPO	DESCRIZIONE	VINCOLI
return :patientAnagraficDataResult	Contiene l'esito della registrazione per la prima volta o dell'aggiornamento dei dati anagrafici del singolo paziente	OBB

2.2.3 Operation doWritePatientEnrollmentDate

Consente di indicare la data di ingresso del singolo assistito nel programma di Gestione Integrata del diabete

2.2.3.1 Input: message doWritePatientEnrollmentDate

Descrizione degli elementi costitutivi del messaggio di richiesta

CAMPO	DESCRIZIONE	VINCOLI
securityToken :authorizationToken	Contiene il token di autorizzazione per l'invio dei dati	OBB
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
citizenID :string	Contiene il codice di pseudonimizzazione dell'assistito	OBB
enrollDate :dateTime	Contiene la data di inclusione dell'assistito nel programma di Gestione Integrata del diabete	OBB

2.2.3.2 Output: message doWritePatientEnrollmentDateResponse

Descrizione degli elementi costitutivi del messaggio di risposta



CAMPO	DESCRIZIONE	VINCOLI
return :boolean	Contiene true se l'invio dei dati è andato a buon fine, false altrimenti	OBB

2.2.4 Operation doWritePatientDisenrollmentDate

Consente di indicare la data di uscita del singolo assistito dal programma di Gestione Integrata del diabete

2.2.4.1 Input: message doWritePatientDisenrollmentDate

Descrizione degli elementi costitutivi del messaggio di richiesta

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

CAMPO	DESCRIZIONE	VINCOLI
securityToken :authorizationToken	Contiene il token di autorizzazione per l'invio dei dati	OBB
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
citizenID :string	Contiene il codice di pseudonimizzazione dell'assistito	OBB
disenrollDate :dateTime	Contiene la data di uscita dell'assistito nel programma di Gestione Integrata del diabete	OBB

2.2.4.2 Output: message doWritePatientDisenrollmentDateResponse

Descrizione degli elementi costitutivi del messaggio di risposta

CAMPO	DESCRIZIONE	VINCOLI
return :boolean	Contiene true se l'invio dei dati è andato a buon fine, false altrimenti	OBB



2.2.5 Operation doWriteAnamnesicData

Consente ad un MMG di inviare i dati del report individuale di startup del singolo paziente arruolato nel programma di Gestione Integrata del diabete.

2.2.5.1 Input: message doWriteAnamnesicData

Descrizione degli elementi costitutivi del messaggio di richiesta

CAMPO	DESCRIZIONE	VINCOLI
securityToken :authorizationToken	Contiene il token di autorizzazione per l'invio dei dati	OBB
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
patientAnamnesicData :patientAnamnesicData	Contiene i dati anamnestici del singolo paziente	OBB

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

2.2.5.2 doWriteAnamnesticDataResponse

Descrizione degli elementi costitutivi del messaggio di risposta

CAMPO	DESCRIZIONE	VINCOLI
return :boolean	Contiene true se l'invio dei dati è andato a buon fine, false altrimenti	OBB

2.2.6 Operation doWriteExemptionCode

Consente ad un MMG di inviare/aggiornare un'esenzione del singolo paziente arruolato nel programma di Gestione Integrata del diabete

2.2.6.1 Input: message doWriteExemptionCode

Descrizione degli elementi costitutivi del messaggio di richiesta

CAMPO	DESCRIZIONE	VINCOLI
securityToken :authorizationToken	Contiene il token di autorizzazione per l'invio dei dati	OBB
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
exemption :patientExemption	Contiene l'esenzione del singolo paziente	OBB

2.2.6.2 doWriteExemptionCodeResponse

Descrizione degli elementi costitutivi del messaggio di risposta



CAMPO	DESCRIZIONE	VINCOLI
return :boolean	Contiene true se il caricamento dati è andato a buon fine, false altrimenti	OBB

2.2.7 Operation doWriteComplications

Consente ad un MMG di inviare/aggiornare i dati sulle complicazioni del singolo paziente arruolato nel programma di Gestione Integrata del diabete

2.2.7.1 Input: message doWriteComplications

Descrizione degli elementi costitutivi del messaggio di richiesta

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

CAMPO	DESCRIZIONE	VINCOLI
securityToken :authorizationToken	Contiene il token di autorizzazione per l'invio dei dati	OBB
MMGCodFisc :string	Contiene il codice fiscale del MMG	OBB
patientComplications :patientComplications	Contiene le informazioni sulle complicazioni del singolo paziente	OBB

2.2.7.2 doWriteComplicationsResponse

Descrizione degli elementi costitutivi del messaggio di risposta

CAMPO	DESCRIZIONE	VINCOLI
return :boolean	Contiene true se il caricamento dati è andato a buon fine, false altrimenti	OBB

2.2.8 Operation getVersion

Consente di verificare la versione del web service.in esercizio



2.2.8.1 Input: message getVersion

Il messaggio di richiesta è vuoto

2.2.8.2 Output: message getVersionResponse

Descrizione degli elementi costitutivi del messaggio di risposta

CAMPO	DESCRIZIONE	VINCOLI
return :string	Contiene la versione del webservice in esercizio	OBB

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

2.3 Tipi di dati complessi

2.3.1 Tipo authenticationToken

Descrive l'insieme di informazioni da fornire nell'authentication token

CAMPO	DESCRIZIONE	VINCOLI
clientSecret :string	Contiene il codice unico, identificativo della software house che ha sviluppato il componente di integrazione*	OBB
username :string	Contiene la username dell'MMG Il valore è al momento uguale al Codice Fiscale del MMG	OBB
password :string	Contiene la password dell'MMG trasformata come segue: lowercase (hex (md5(<password MMG>)))	OBB
role :string	Contiene il ruolo dell'utente, nel caso di MMG la codifica è "M"	OBB

* Per ottenere il proprio codice unico identificativo, le software house devono inviare un email di richiesta a report@soresa.it

2.3.2 Tipo authorizationToken

Descrive l'insieme di informazioni presenti nell'authorization token.

Da un punto di vista operativo, l'authorization token ha validità per singola sessione di lavoro. Più precisamente scade dopo una latenza di 15 secondi successivi all'ultima operazione invocata.

CAMPO	DESCRIZIONE	VINCOLI
token :stringa	Contiene un codice unico per sessione di lavoro, generato dalla piattaforma regionale in caso di esito positivo dell'operazione getAuthorizationToken e necessario per poter eseguire tutte le altre operazioni	OBB

2.3.3 Tipo patientAnagraficData

Descrive i dati anagrafici del singolo paziente arruolato nella Gestione Integrata del diabete



CAMPO	DESCRIZIONE	VINCOLI
patientCodFisc :string	Contiene il Codice Fiscale del paziente da valorizzare al primo invio	OBB



Accordo Integrativo Regionale per la Medicina Generale
 Decreto Commissariale n. 87 del 24 Luglio 2013
 BURC n. 46 del 22 Agosto 2013



	Da valorizzare con il Codice Fiscale dell'assistito nel caso di prima registrazione; da impostare a null in caso di aggiornamento.	Nillable
citizenID :string	Contiene il codice unico di pseudonimizzazione dell'assistito Da impostare a null nel caso di prima registrazione dell'assistito, da valorizzare con il citizenID; in caso di aggiornamento.	OBB Nillable
name :string	Contiene il nome del paziente Da valorizzare con il Nome dell'assistito nel caso di prima registrazione; da impostare a null in caso di aggiornamento.	OBB Nillable
surname :string	Contiene il cognome del paziente Da valorizzare con il Cognome dell'assistito nel caso di prima registrazione; da impostare a null in caso di aggiornamento.	OBB Nillable
asl :int	Indica l'ASL di residenza del paziente secondo la seguente codifica 201=ASL Avellino 202=ASL Benevento 203=ASL Caserta 204=ASL Napoli 1 Centro 205=ASL Napoli 2 Nord 206=ASL Napoli 3 Sud 207=ASL Salerno	OBB
birthdate :dateTime	Contiene la data di nascita del paziente	OBB
nationality :string	Contiene la cittadinanza del paziente	OBB
city :string	Contiene il comune di residenza del paziente	FAC

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

road :string	Contiene la via di residenza del paziente	FAC
houseNumber :string	Contiene il numero civico della via di residenza del paziente	FAC

2.3.4 Tipo patientAnagraficDataResult

Descrive il risultato dell'operazione del singolo paziente arruolato nella Gestione Integrata del diabete

CAMPO	DESCRIZIONE	VINCOLI
citizenID :string	Nel caso di registrazione per la prima volta contiene il codice unico di pseudonimizzazione del paziente	NBB (obbligatorio solo in caso di prima registrazione con esito positivo della operation)
result: boolean	Contiene true se l'invio dei dati è andato a buon fine, false altrimenti	OBB

2.3.5 Tipo patientAnamnesiticData

Descrive i dati anamnestici del singolo paziente arruolato nella Gestione Integrata del diabete

CAMPO	DESCRIZIONE	VINCOLI
citizenId :string	Contiene il codice unico di pseudonimizzazione dell'assistito	OBB
firstDiabetesDiagnosis :int	Contiene l'anno di prima diagnosi del diabete per il paziente	FAC
currentTherapy :(string)	Indica una o più terapie in corso secondo la codifica che segue 0=non registrato 1=solo dieta 2=terapia insulinica 3=incretine 4=metf.	OBB



	<p>5=sulf. 6=glindi 7=acarbose 8=glitazioni 9=ass.precost.</p> <p>Nel caso di più valori essi andranno inviati in una unica stringa separati dal carattere ‘;’</p>	
<p>maritalStatus :int</p>	<p>Indica lo stato civile secondo la seguente codifica:</p> <p>0=non registrato 1=coniugato o convivente 2=nubile/celibe 3=divorziato o separato legalmente 4=vedovo 5=altra condizione</p>	FAC
<p>educationLevel :int</p>	<p>Indica il titolo di studio del paziente secondo la seguente codifica:</p> <p>0=non registrato 1=nessuno 2=elementare 3=media inferiore 4=licenza professionale o diploma superiore 5=laurea</p>	FAC
<p>workingPosition :int</p>	<p>Indica la condizione professionale secondo la seguente codifica:</p> <p>0=non registrato 1=occupato 2=disoccupato 3=ritirato dal lavoro 4=casalinga 5=altra condizione</p>	FAC
<p>workingCategory :int</p>	<p>Indica la posizione professionale secondo la seguente codifica:</p> <p>0=non registrato 1=imprenditore 2=libero professionista 3=dirigente 4=impiegato 5=lavoratore in proprio, commerciante 6=operaio, lavoratore manuale, apprendista</p>	FAC
<p>alcoholUser</p>	<p>Indica se il paziente fa uso di alcolici (true) o meno (false)</p>	FAC



Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



:boolean		
alcoholCheckDate :dateTime	Indica la data di rilevazione delle abitudini sul consumo di alcolici	FAC
alcoholKind :int	Indica il tipo di alcolici consumati abitualmente secondo la seguente codifica: 0=non registrato 1=vino 2=birra 3=super alcolici	FAC
alcoholQuantity :int	Indica il consumo giornaliero di alcolici, espresso in ml	FAC
physicalActivity :boolean	Indica se il paziente pratica abitualmente attività sportive (true) o meno (false)	FAC
physicalActivityCheckDate :dateTime	Indica la data di rilevazione delle informazioni sulla attività fisica praticata	FAC
physicalActivityLevel :int	Indica il livello di intensità dell'attività fisica praticata secondo la seguente codifica: 0=non registrato 1=lieve 2=moderata 3=intensa	FAC
smokingCurrentAttitude :boolean	Indica se il paziente è un fumatore (true) o meno (false)	FAC
smokingPreviousAttitude :boolean	Indica se il paziente è stato in passato un fumatore (true) o meno (false)	FAC
smokingCheckDate :dateTime	Indica la data di rilevazione delle informazioni sulle abitudini al fumo	FAC
smokingLastYear :int	Indica l'anno in cui il paziente ex fumatore ha smesso di fumare	FAC
smokingType	Indica il tipo di tabacco fumato secondo la seguente	FAC

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

:int	codifica: 0=non registrato 1=sigarette 2=sigari 3=altro	
smokingDailyQuantity :int	Indica la quantità di tabacco giornaliero consumato del tipo indicato nel campo smokingType	FAC

2.3.6 Tipo patientExemption

Descrive i dati di interesse per una singola esenzione del Ticket riconosciuto al singolo paziente arruolato nella Gestione Integrata del diabete

CAMPO	DESCRIZIONE	VINCOLI
citizenID	Contiene il codice unico di pseudonimizzazione dell'assistito	OBB
exemptionCode	Contiene il codice dell'esenzione del paziente	OBB
exemptionDate	Contiene la data di rilascio del codice di esenzione del paziente	FAC

2.3.7 Tipo patientComplications

Descrive i dati di interesse per le complicazioni diagnosticate al singolo paziente arruolato nella Gestione Integrata del diabete

CAMPO	DESCRIZIONE	VINCOLI
citizenId :string	Contiene il codice unico di pseudonimizzazione dell'assistito	OBB
amputation_value :boolean	Indica se l'assistito soffre della complicazione Amputazione (true) o meno (false)	OBB
amputation :dateTime	Contiene la data di diagnosi della complicazione Amputazione	FAC
angina_value :boolean	Indica se l'assistito soffre della complicazione Angina (true) o meno (false)	OBB
angina	Contiene la data di diagnosi della complicazione Angina	FAC



Accordo Integrativo Regionale per la Medicina Generale
 Decreto Commissariale n. 87 del 24 Luglio 2013
 BURC n. 46 del 22 Agosto 2013





:dateTime		
blindness_value :dateTime	Indica se l'assistito soffre della complicazione Cecità (true) o meno (false)	OBB
blindness :dateTime	Contiene la data di diagnosi della complicazione Cecità	FAC
claudicatio_value :boolean	Indica se l'assistito soffre della complicazione Claudicatio (true) o meno (false)	OBB
claudicatio :dateTime	Contiene la data di diagnosi della complicazione Claudicatio	FAC
dialysis_value :boolean	Indica se l'assistito soffre della complicazione Dialisi (true) o meno (false)	OBB
dialysis :dateTime	Contiene la data di diagnosi della complicazione Dialisi	FAC
fundus_value :boolean	Indica se l'assistito soffre della complicazione Fondo Oculare (true) o meno (false)	OBB
fundus :dateTime	Contiene la data di diagnosi della complicazione Fondo Oculare	FAC
ictus_value :boolean	Indica se l'assistito soffre della complicazione Ictus (true) o meno (false)	OBB
ictus :dateTime	Contiene la data di diagnosi della complicazione Ictus	FAC
ima_value :boolean	Indica se l'assistito soffre della complicazione IMA (true) o meno (false)	OBB
ima :dateTime	Contiene la data di diagnosi della complicazione IMA	FAC
ischemicHeartCondition_value	Indica se l'assistito soffre della complicazione Cardiopatia Ischemica (true) o meno (false)	OBB



Accordo Integrativo Regionale per la Medicina Generale
 Decreto Commissariale n. 87 del 24 Luglio 2013
 BURC n. 46 del 22 Agosto 2013



:boolean		
ischemicHeartCondition :dateTime	Contiene la data di diagnosi della complicazione Cardiopatia Ischemica	FAC
nephropathy_value :boolean	Indica se l'assistito soffre della complicazione Nefropatia (true) o meno (false)	OBB
nephropathy :dateTime	Contiene la data di diagnosi della complicazione Nefropatia	FAC
retinopathy_value :boolean	Indica se l'assistito soffre della complicazione Retinopatia (true) o meno (false)	OBB
rethinopathy :dateTime	Contiene la data di diagnosi della complicazione Retinopatia	FAC
revascularization_value :boolean	Indica se l'assistito soffre della complicazione Rivascolarizzazione (true) o meno (false)	OBB
revascularization :dateTime	Contiene la data di diagnosi della complicazione Rivascolarizzazione	FAC
tia_value :boolean	Indica se l'assistito soffre della complicazione TIA (true) o meno (false)	OBB
tia :dateTime	Contiene la data di diagnosi della complicazione TIA	FAC
ulcerations_value :boolean	Indica se l'assistito soffre della complicazione Ulcerazioni (true) o meno (false)	OBB
ulcerations :dateTime	Contiene la data di diagnosi della complicazione Ulcerazioni	FAC
ipertension_value :boolean	Indica se l'assistito soffre della complicazione Ipertensione (true) o meno (false)	OBB
ipertension :dateTime	Contiene la data di diagnosi della complicazione Ipertensione	FAC

	<p>Accordo Integrativo Regionale per la Medicina Generale Decreto Commissariale n. 87 del 24 Luglio 2013 BURC n. 46 del 22 Agosto 2013</p>	
---	--	---

2.4 Tipi di eccezioni

2.4.1 Illegal argument exception

CAMPO	DESCRIZIONE	VINCOLI
message :string	Contiene un messaggio di errore correlato ad incoerenze tra i dati inviati e la segnatura dell'operazione	FAC

2.4.2 AIR application exception

CAMPO	DESCRIZIONE	VINCOLI
errorCode :string	Contiene un codice correlato ad errori manifestatisi lato server, ed è necessario contattare l'amministratore del sistema	FAC
message :string	Contiene un messaggio correlato ad errori manifestatisi lato server, ed è necessario contattare l'amministratore del sistema	FAC

2.4.3 Data validation exception

CAMPO	DESCRIZIONE	VINCOLI
errorCode :string	Contiene un codice correlato ad errori rispetto ai vincoli di integrità dei dati inviati	FAC
message :string	Contiene un messaggio correlato ad errori rispetto ai vincoli di integrità dei dati inviati	FAC

3 WSDL

```
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://air.di.unisa.it/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/http" name="PatientDataService"
targetNamespace="http://air.di.unisa.it/">
<wsdl:types>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://air.di.unisa.it/" attributeFormDefault="unqualified"
elementFormDefault="unqualified" targetNamespace="http://air.di.unisa.it/">
<xs:element name="doWriteAnagraficData" type="tns:doWriteAnagraficData" />
<xs:element name="doWriteAnagraficDataResponse"
type="tns:doWriteAnagraficDataResponse" />
<xs:element name="doWriteAnamnesticData" type="tns:doWriteAnamnesticData" />
<xs:element name="doWriteAnamnesticDataResponse"
type="tns:doWriteAnamnesticDataResponse" />
<xs:element name="doWriteComplications" type="tns:doWriteComplications" />
<xs:element name="doWriteComplicationsResponse"
type="tns:doWriteComplicationsResponse" />
<xs:element name="doWriteExemptionCode" type="tns:doWriteExemptionCode" />
<xs:element name="doWriteExemptionCodeResponse"
type="tns:doWriteExemptionCodeResponse" />
<xs:element name="doWritePatientDisenrollmentDate"
type="tns:doWritePatientDisenrollmentDate" />
<xs:element name="doWritePatientDisenrollmentDateResponse"
type="tns:doWritePatientDisenrollmentDateResponse" />
<xs:element name="doWritePatientEnrollmentDate"
type="tns:doWritePatientEnrollmentDate" />
<xs:element name="doWritePatientEnrollmentDateResponse"
type="tns:doWritePatientEnrollmentDateResponse" />
<xs:element name="getAuthorizationToken" type="tns:getAuthorizationToken" />
<xs:element name="getAuthorizationTokenResponse"
type="tns:getAuthorizationTokenResponse" />
<xs:element name="getVersion" type="tns:getVersion" />
<xs:element name="getVersionResponse" type="tns:getVersionResponse" />
<xs:complexType name="doWriteExemptionCode">
<xs:sequence>
<xs:element name="securityToken" type="tns:authorizationToken" />
<xs:element name="MMGCodFisc" type="xs:string" />
<xs:element name="patientExemption" type="tns:patientExemption" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="authorizationToken">
<xs:sequence>
<xs:element name="token" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="patientExemption">
<xs:sequence>
<xs:element name="citizenId" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:schema>
</wsdl:types>
</wsdl:definitions>
```

```
<xs:element name="exemptionCode" type="xs:string"/>
<xs:element minOccurs="0" name="exemptionDate" nillable="true"
type="xs:dateTime"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteExemptionCodeResponse">
<xs:sequence>
<xs:element name="return" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteAnagraficData">
<xs:sequence>
<xs:element name="securityToken" type="tns:authorizationToken"/>
<xs:element name="MMGCodFisc" type="xs:string"/>
<xs:element name="patientAnagraficData" type="tns:patientAnagraficData"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="patientAnagraficData">
<xs:sequence>
<xs:element name="asl" type="xs:int"/>
<xs:element name="birthdate" type="xs:dateTime"/>
<xs:element minOccurs="0" name="city" type="xs:string"/>
<xs:element name="citizenId" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="houseNumber" type="xs:string"/>
<xs:element name="name" nillable="true" type="xs:string"/>
<xs:element name="nationality" type="xs:string"/>
<xs:element minOccurs="0" name="road" type="xs:string"/>
<xs:element name="patientCodFisc" nillable="true" type="xs:string"/>
<xs:element name="surname" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteAnagraficDataResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="tns:patientAnagraficDataResult"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="patientAnagraficDataResult">
<xs:sequence>
<xs:element minOccurs="0" name="citizenId" type="xs:string"/>
<xs:element name="result" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWritePatientEnrollmentDate">
<xs:sequence>
<xs:element name="securityToken" type="tns:authorizationToken"/>
<xs:element name="MMGCodFisc" type="xs:string"/>
<xs:element name="citizenId" type="xs:string"/>
<xs:element name="enrollDate" type="xs:dateTime"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWritePatientEnrollmentDateResponse">
<xs:sequence>
```

```
<xs:element name="return" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteComplications">
<xs:sequence>
<xs:element name="securityToken" type="tns:authorizationToken"/>
<xs:element name="MMGCodFisc" type="xs:string"/>
<xs:element name="patientComplications" type="tns:patientComplications"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="patientComplications">
<xs:sequence>
<xs:element minOccurs="0" name="amputation" type="xs:dateTime"/>
<xs:element name="amputation_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="angina" type="xs:dateTime"/>
<xs:element name="angina_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="blindness" type="xs:dateTime"/>
<xs:element name="blindness_value" type="xs:boolean"/>
<xs:element name="citizenId" type="xs:string"/>
<xs:element minOccurs="0" name="claudicatio" type="xs:dateTime"/>
<xs:element name="claudicatio_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="dialysis" type="xs:dateTime"/>
<xs:element name="dialysis_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="fundus" type="xs:dateTime"/>
<xs:element name="fundus_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="ictus" type="xs:dateTime"/>
<xs:element name="ictus_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="ima" type="xs:dateTime"/>
<xs:element name="ima_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="ipertension" type="xs:dateTime"/>
<xs:element name="ipertension_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="ischemicHeartCondition" type="xs:dateTime"/>
<xs:element name="ischemicHeartCondition_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="nephropathy" type="xs:dateTime"/>
<xs:element name="nephropathy_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="retinopathy" type="xs:dateTime"/>
<xs:element name="retinopathy_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="revascularization" type="xs:dateTime"/>
<xs:element name="revascularization_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="tia" type="xs:dateTime"/>
<xs:element name="tia_value" type="xs:boolean"/>
<xs:element minOccurs="0" name="ulcerations" type="xs:dateTime"/>
<xs:element name="ulcerations_value" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteComplicationsResponse">
<xs:sequence>
<xs:element name="return" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="getVersion">
<xs:sequence/>
</xs:complexType>
```




Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



```
</xs:complexType>
<xs:complexType name="getVersionResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteAnamnesticData">
<xs:sequence>
<xs:element name="securityToken" type="tns:authorizationToken"/>
<xs:element name="MMGCodFisc" type="xs:string"/>
<xs:element name="patientAnamnesticData" type="tns:patientAnamnesticData"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="patientAnamnesticData">
<xs:sequence>
<xs:element minOccurs="0" name="alcoholCheckDate" type="xs:dateTime"/>
<xs:element name="alcoholKind" type="xs:int"/>
<xs:element name="alcoholQuantity" type="xs:int"/>
<xs:element name="citizenId" type="xs:string"/>
<xs:element name="currentTherapy" type="xs:string"/>
<xs:element name="educationLevel" type="xs:int"/>
<xs:element minOccurs="0" name="enrollDate" type="xs:dateTime"/>
<xs:element name="firstDiabetesDiagnosis" type="xs:int"/>
<xs:element name="marital_status" type="xs:int"/>
<xs:element name="physicalActivity" type="xs:boolean"/>
<xs:element minOccurs="0" name="physicalActivityCheckDate" type="xs:dateTime"/>
<xs:element name="physicalActivityLevel" type="xs:int"/>
<xs:element minOccurs="0" name="smokingCheckDate" type="xs:dateTime"/>
<xs:element name="smokingCurrentAttitude" type="xs:boolean"/>
<xs:element name="smokingDailyQuantity" type="xs:int"/>
<xs:element name="smokingLastYear" type="xs:int"/>
<xs:element name="smokingPreviousAttitude" type="xs:boolean"/>
<xs:element name="smokingType" type="xs:int"/>
<xs:element name="alcoholUser" type="xs:boolean"/>
<xs:element name="workingCategory" type="xs:int"/>
<xs:element name="workingPosition" type="xs:int"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWriteAnamnesticDataResponse">
<xs:sequence>
<xs:element name="return" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWritePatientDisenrollmentDate">
<xs:sequence>
<xs:element name="securityToken" type="tns:authorizationToken"/>
<xs:element name="MMGCodFisc" type="xs:string"/>
<xs:element name="citizenId" type="xs:string"/>
<xs:element name="disenrollmentDate" type="xs:dateTime"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="doWritePatientDisenrollmentDateResponse">
```




Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



```
<xs:sequence>
<xs:element name="return" type="xs:boolean"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="getAuthorizationToken">
<xs:sequence>
<xs:element name="MMGCodFisc" type="xs:string"/>
<xs:element name="AuthenticationToken" type="tns:authenticationToken"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="authenticationToken">
<xs:sequence>
<xs:element name="clientSecret" type="xs:string"/>
<xs:element name="password" type="xs:string"/>
<xs:element name="role" type="xs:string"/>
<xs:element name="username" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="getAuthorizationTokenResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="tns:authorizationToken"/>
</xs:sequence>
</xs:complexType>
<xs:element name="IllegalArgumentExeption"
type="tns:IllegalArgumentExeption"/>
<xs:complexType name="IllegalArgumentExeption">
<xs:sequence>
<xs:element minOccurs="0" name="message" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="AIRApplicationException" type="tns:AIRApplicationException"/>
<xs:complexType name="AIRApplicationException">
<xs:sequence>
<xs:element minOccurs="0" name="errorCode" type="xs:string"/>
<xs:element minOccurs="0" name="message" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="DataValidationException" type="tns:DataValidationException"/>
<xs:complexType name="DataValidationException">
<xs:sequence>
<xs:element minOccurs="0" name="errorCode" type="xs:string"/>
<xs:element minOccurs="0" name="message" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="getAuthorizationTokenResponse">
<wsdl:part element="tns:getAuthorizationTokenResponse"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteExemptionCode">
<wsdl:part element="tns:doWriteExemptionCode" name="parameters"></wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="doWritePatientEnrollmentDateResponse">
<wsdl:part element="tns:doWritePatientEnrollmentDateResponse"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="DataValidationException">
<wsdl:part element="tns:DataValidationException"
name="DataValidationException"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteComplicationsResponse">
<wsdl:part element="tns:doWriteComplicationsResponse"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteComplications">
<wsdl:part element="tns:doWriteComplications" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="getVersion">
<wsdl:part element="tns:getVersion" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="AIRApplicationException">
<wsdl:part element="tns:AIRApplicationException"
name="AIRApplicationException"></wsdl:part>
</wsdl:message>
<wsdl:message name="getVersionResponse">
<wsdl:part element="tns:getVersionResponse" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWritePatientDisenrollmentDateResponse">
<wsdl:part element="tns:doWritePatientDisenrollmentDateResponse"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteAnagraficDataResponse">
<wsdl:part element="tns:doWriteAnagraficDataResponse"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteAnagraficData">
<wsdl:part element="tns:doWriteAnagraficData" name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWritePatientEnrollmentDate">
<wsdl:part element="tns:doWritePatientEnrollmentDate"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="IllegalArgumentException">
<wsdl:part element="tns:IllegalArgumentException"
name="IllegalArgumentException"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteAnamnesticDataResponse">
<wsdl:part element="tns:doWriteAnamnesticDataResponse"
name="parameters"></wsdl:part>
</wsdl:message>
<wsdl:message name="doWriteAnamnesticData">
<wsdl:part element="tns:doWriteAnamnesticData" name="parameters"></wsdl:part>
</wsdl:message>
```



Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



```
<wsdl:message name="doWriteExemptionCodeResponse" >
<wsdl:part element="tns:doWriteExemptionCodeResponse"
name="parameters" ></wsdl:part>
</wsdl:message>
<wsdl:message name="doWritePatientDisenrollmentDate" >
<wsdl:part element="tns:doWritePatientDisenrollmentDate"
name="parameters" ></wsdl:part>
</wsdl:message>
<wsdl:message name="getAuthorizationToken" >
<wsdl:part element="tns:getAuthorizationToken" name="parameters" ></wsdl:part>
</wsdl:message>
<wsdl:portType name="PatientDataWebService" >
<wsdl:operation name="doWriteExemptionCode" >
<wsdl:input message="tns:doWriteExemptionCode"
name="doWriteExemptionCode" ></wsdl:input>
<wsdl:output message="tns:doWriteExemptionCodeResponse"
name="doWriteExemptionCodeResponse" ></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException" ></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException" ></wsdl:fault>
<wsdl:fault message="tns:IllegalArgumentException"
name="IllegalArgumentException" ></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWriteAnagraficData" >
<wsdl:input message="tns:doWriteAnagraficData"
name="doWriteAnagraficData" ></wsdl:input>
<wsdl:output message="tns:doWriteAnagraficDataResponse"
name="doWriteAnagraficDataResponse" ></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException" ></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException" ></wsdl:fault>
<wsdl:fault message="tns:IllegalArgumentException"
name="IllegalArgumentException" ></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWritePatientEnrollmentDate" >
<wsdl:input message="tns:doWritePatientEnrollmentDate"
name="doWritePatientEnrollmentDate" ></wsdl:input>
<wsdl:output message="tns:doWritePatientEnrollmentDateResponse"
name="doWritePatientEnrollmentDateResponse" ></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException" ></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException" ></wsdl:fault>
<wsdl:fault message="tns:IllegalArgumentException"
name="IllegalArgumentException" ></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWriteComplications" >
<wsdl:input message="tns:doWriteComplications"
name="doWriteComplications" ></wsdl:input>
<wsdl:output message="tns:doWriteComplicationsResponse"
```

```
name="doWriteComplicationsResponse"></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException"></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException"></wsdl:fault>
<wsdl:fault message="tns:IllegalArgumentException"
name="IllegalArgumentException"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getVersion">
<wsdl:input message="tns:getVersion" name="getVersion"></wsdl:input>
<wsdl:output message="tns:getVersionResponse"
name="getVersionResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="doWriteAnamnesticData">
<wsdl:input message="tns:doWriteAnamnesticData"
name="doWriteAnamnesticData"></wsdl:input>
<wsdl:output message="tns:doWriteAnamnesticDataResponse"
name="doWriteAnamnesticDataResponse"></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException"></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException"></wsdl:fault>
<wsdl:fault message="tns:IllegalArgumentException"
name="IllegalArgumentException"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWritePatientDisenrollmentDate">
<wsdl:input message="tns:doWritePatientDisenrollmentDate"
name="doWritePatientDisenrollmentDate"></wsdl:input>
<wsdl:output message="tns:doWritePatientDisenrollmentDateResponse"
name="doWritePatientDisenrollmentDateResponse"></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException"></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException"></wsdl:fault>
<wsdl:fault message="tns:IllegalArgumentException"
name="IllegalArgumentException"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getAuthorizationToken">
<wsdl:input message="tns:getAuthorizationToken"
name="getAuthorizationToken"></wsdl:input>
<wsdl:output message="tns:getAuthorizationTokenResponse"
name="getAuthorizationTokenResponse"></wsdl:output>
<wsdl:fault message="tns:DataValidationException"
name="DataValidationException"></wsdl:fault>
<wsdl:fault message="tns:AIRApplicationException"
name="AIRApplicationException"></wsdl:fault>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PatientDataServiceSoapBinding"
type="tns:PatientDataWebService">
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
```

```
<wsdl:operation name="doWriteExemptionCode">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="doWriteExemptionCode">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="doWriteExemptionCodeResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException">
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException">
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap:fault name="IllegalArgumentException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWriteAnagraficData">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="doWriteAnagraficData">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="doWriteAnagraficDataResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException">
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException">
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap:fault name="IllegalArgumentException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWritePatientEnrollmentDate">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="doWritePatientEnrollmentDate">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="doWritePatientEnrollmentDateResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException">
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException">
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap:fault name="IllegalArgumentException" use="literal"/>
</wsdl:fault>
```

```
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWriteComplications">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="doWriteComplications">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="doWriteComplicationsResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException">
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException">
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap:fault name="IllegalArgumentException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getVersion">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="getVersion">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="getVersionResponse">
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="doWriteAnamnesticData">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="doWriteAnamnesticData">
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="doWriteAnamnesticDataResponse">
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException">
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException">
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException">
<soap:fault name="IllegalArgumentException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="doWritePatientDisenrollmentDate">
<soap:operation soapAction="" style="document"/>
<wsdl:input name="doWritePatientDisenrollmentDate">
<soap:body use="literal"/>
</wsdl:input>
```




Accordo Integrativo Regionale per la Medicina Generale
Decreto Commissariale n. 87 del 24 Luglio 2013
BURC n. 46 del 22 Agosto 2013



```
<wsdl:output name="doWritePatientDisenrollmentDateResponse" >
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException" >
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException" >
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="IllegalArgumentException" >
<soap:fault name="IllegalArgumentException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getAuthorizationToken" >
<soap:operation soapAction="" style="document"/>
<wsdl:input name="getAuthorizationToken" >
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output name="getAuthorizationTokenResponse" >
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="DataValidationException" >
<soap:fault name="DataValidationException" use="literal"/>
</wsdl:fault>
<wsdl:fault name="AIRApplicationException" >
<soap:fault name="AIRApplicationException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="PatientDataService" >
<wsdl:port binding="tns:PatientDataServiceSoapBinding"
name="PatientDataServicePort" >
<soap:address location="https://air.di.unisa.it:8443/AIR-
EJBWS/PatientDataService/PatientDataWebService" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```