

**RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE.
NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”**

DOCUMENTAZIONE TECNICA

Codice: A-PI3-PIS Versione: 01.0.19 Diciannovesima Revisione

Data di emissione:

	Firma	Data
Redatto:	L. Luciani, F. Veltri, A. Faillace, S. Frezza, C. Ferlito	09/04/2014
Rivisto:	L. Luciani, A. Lembo	30/04/2020
Approvato:		

Distribuzione elettronica a:

Sito aziendale

Distribuzione cartacea a:

Segreteria Direzione

**RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA
P.I.TRE.
NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”**

**P.I.TRE.
DOCUMENTAZIONE TECNICA**

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.
DOCUMENTAZIONE TECNICA

European Union Public Licence V. 1.2

EUPL © the European Union 2007, 2016

This European Union Public Licence (the ‘EUPL’) applies to the Work (as defined below) which is provided under the terms of this Licence. Any use of the Work, other than as authorised under this Licence is prohibited (to the extent such use is covered by a right of the copyright holder of the Work).

The Work is provided under the terms of this Licence when the Licensor (as defined below) has placed the following notice immediately following the copyright notice for the Work: “Licensed under the EUPL”, or has expressed by any other means his willingness to license under the EUPL.

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

PRINCIPALI MODIFICHE RISPETTO ALLA VERSIONE PRECEDENTE

Data	Versione	Modifiche apportate
02/04/2012	01.0.0 Prima Stesura	Prima stesura
06/09/2012	01.0.1 Seconda Stesura	Aggiunta la definizione dei servizi che restituiscono i tipi di filtro applicabili nelle ricerche; aggiornata la definizione dei servizi di ricerca che restituiscono elenchi con il numero totale di record trovati; aggiunta la definizione del nuovo servizio che crea un documento e lo aggiunge in un fascicolo (CreateDocumentAndAddInProject).
05/12/2012	01.0.2 Terza Stesura	Aggiunta la gestione delle eccezioni per tutti i servizi esposti; adeguato il valore di ritorno dei servizi di ricerca che non generano risultati; aggiunta la gestione della tipologia dei filtri per i servizi di ricerca; definizione del formato data per i servizi di ricerca.
16/01/2013	01.0.3 Quarta Stesura	Aggiunta delle sezioni riguardanti il servizio che gestisce i token di autenticazione. Inserimento dei nuovi tipi di eccezione riguardanti la visibilità dei template e dei loro campi nei metodi interessati. Modifica della classe Field e inserimento della proprietà Rights. Inserimento del campo AttachmentType nella Request di UploadFileToDocument.
28/01/2013	01.0.4 Quinta Stesura	Aggiunta dei nuovi metodi di trasmissione ExecuteTransmissionDocument e ExecuteTransmissionProject.
22/02/2013	01.0.5 Sesta Stesura	Aggiunto il campo GetFileWithSignature nella richiesta di GetDocument.
10/06/2013	01.0.6 Settima Stesura	Aggiunte descrizioni specifiche per la richiesta di UploadFileToDocument. Il campo Description è passato da opzionale a obbligatorio. Aggiornati i filtri nella SearchCorrespondent.
29/10/2013	01.0.10 revisione Decima	<p>Aggiunte descrizioni specifiche per la richiesta di FollowDocument (§ 3.5.4), FollowProject (§ 3.5.5) e SearchUsers (§ 3.5.6).</p> <p>Effettuate modifiche ai seguenti paragrafi:</p> <ul style="list-style-type: none"> • 2.1.1 - Modificata descrizione ArrivalDate. Tolto ArrivalTime. Tolto CreationTime. Aggiunto PersonalDocument • 2.1.3 - Spostato IsCommonAddress. Modificato NationalIdentificationNumber. Aggiunto VatNumber • 2.1.5 - Tolto IdRegistryOrRF • 2.1.6 - Modificata descrizione name. • 2.1.9 - Corretto IsRF • 2.1.12 - Aggiunto Register • 2.1.14 - Corretto DiagramId • 3.5 - Ordinamento alfabetico dei metodi
19/12/2013	01.0.11 Undicesima	Modificato campo HashFile per la richiesta di

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

	revisione	UploadFileToDocument. Il campo HashFile è passato da opzionale a obbligatorio in presenza di una specifica chiave di configurazione.
09/04/2014	01.0.12 Dodicesima revisione	Aggiunto campo AnnulmentDate per oggetto Document. Aggiunto campo OpeningDate per oggetto Project. Aggiunto oggetto Folder. Aggiunte e/o modificate relative immagini. Aggiunto sottocapitolo per Gestione Avanzata dei documenti Aggiunto metodo e sezione RemoveDocument Aggiunto metodo e sezione GiveUpRights Aggiunto metodo e sezione GetToken Aggiunto metodi e sezioni CreateFolder e GetProjectFolders Aggiunto metodo e sezione GetRolesForEnabledActions Aggiunta sezione Aggiunta di un documento in un sottofascicolo in AddDocInProject. Aggiunti nuovi filtri per ricerca documento. Aggiunti nuovi filtri per ricerca fascicoli.
10/06/2014	01.0.13 Tredicesima revisione	Aggiunto oggetto SendingResult Aggiunto metodo SendDocumentAdvanced Aggiunto filtro Template_Extraction per le ricerche di documenti e fascicoli, con relativa sezione che ne descrive il comportamento.
11/11/2014	01.0.14 Quattordicesima revisione	Aggiunto filtro EXACT_CODE per la ricerca di un corrispondente dato il suo codice esatto.
12/12/2017	01.0.15	Aggiunti medoti Libro Firma
05/10/2018	01.0.16	Modificate alcune sezioni riguardanti il Libro Firma.
06/03/2020	01.0.17	Aggiunte informazioni su Fattura Attiva. Modifiche sostanziali al paragrafo 3.4 Autenticazione Cambiato il logo
30/04/2020	01.0.18	Aggiungi i metodi GetLinkDocById e GetLinkPrjById
07/07/2020	01.0.19	Aggiunto il metodo ProtocolPredisposed

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

INDICE

1	INTRODUZIONE.....	1
1.1	PREMESSA	1
1.2	MODELLO DI COMUNICAZIONE REQUEST - RESPONSE	1
2	MODELLO DI DOMINIO	2
2.1	DIAGRAMMA DELLE CLASSI.....	2
2.1.1	CLASSE DOCUMENT	2
2.1.2	CLASSE CLASSIFICATIONSCHEME	5
2.1.3	CLASSE CORRESPONDENT	6
2.1.4	CLASSE TEMPLATE	8
2.1.5	CLASSE FIELD.....	8
2.1.6	CLASSE FILE.....	10
2.1.7	CLASSE FILTER	10
2.1.7.1	Enumerato FilterTypeEnum	11
2.1.8	CLASSE ROLE.....	11
2.1.9	CLASSE REGISTER.....	12
2.1.10	CLASSE USER	13
2.1.11	CLASSE NOTE	13
2.1.12	CLASSE PROJECT	14
2.1.13	CLASSE FOLDER.....	15
2.1.14	CLASSE STATEDIAGRAM	16
2.1.15	CLASSE STATEOFDIAGRAM.....	16
2.1.16	CLASSE TRANSMISSIONMODEL	17
2.1.17	CLASSE STAMP.....	18
2.1.18	CLASSE SENDINGRESULT.....	19
2.2	ENUMERAZIONI	20
2.2.1	ENUMERATO OPERATIONFOLLOW	20
3	DESCRIZIONE DEI SERVIZI	21
3.1	STRUTTURA DEI PACCHETTI DI REQUEST	21
3.2	STRUTTURA DEI PACCHETTI DI RESPONSE.....	21
3.3	GESTIONE DELLE ECCEZIONI.....	22
3.4	AUTENTICAZIONE	22
3.5	SERVIZI ESPOSTI	24
3.5.1	SERVIZI PER LA GESTIONE DEI RUOLI.....	24
3.5.2	SERVIZI PER LA GESTIONE DEI REGISTRI/RF.....	24
3.5.3	SERVIZI PER LA GESTIONE DEI TITOLARI	24
3.5.4	SERVIZI PER LA GESTIONE DEI DOCUMENTI	25
3.5.5	SERVIZI PER LA GESTIONE AVANZATA DEI DOCUMENTI	26
3.5.6	SERVIZI PER LA GESTIONE DEI FASCICOLI	26

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.5.7	SERVIZI PER LA GESTIONE DELLA RUBRICA.....	27
3.5.8	SERVIZI PER LA GESTIONE DEI TOKEN DI AUTENTICAZIONE.....	28
3.5.9	SERVIZI PER LA GESTIONE DELLE TRASMISSIONI	28
3.5.10	SERVIZI PER LA GESTIONE DEL LIBROFIRMA.....	28
3.6	SERVIZI PER LA GESTIONE DEI RUOLI	29
3.6.1	SERVIZIO “ GETROLE”	29
3.6.1.1	Descrizione oggetto “Request”	29
3.6.1.2	Descrizione oggetto “Response”	30
3.6.1.3	Esempio di utilizzo C#	30
3.6.1.4	Esempio di utilizzo Java.....	31
3.6.2	SERVIZIO “ GETROLES”	31
3.6.2.1	Descrizione oggetto “Request”	31
3.6.2.2	Descrizione oggetto “Response”	32
3.6.2.3	Esempio di utilizzo C#	32
3.6.2.4	Esempio di utilizzo Java.....	33
3.6.3	SERVIZIO “GETROLESFORENABLEDACTIONS”	33
3.6.3.1	Descrizione oggetto “Request”	33
3.6.3.2	Descrizione oggetto “Response”	34
3.6.3.3	Esempio di utilizzo C#	34
3.6.3.4	Esempio di utilizzo Java.....	35
3.6.4	SERVIZIO “ GETUSERSINROLE”	36
3.6.4.1	Descrizione oggetto “Request”	36
3.6.4.2	Descrizione oggetto “Response”	36
3.6.4.3	Esempio di utilizzo C#	37
3.6.4.4	Esempio di utilizzo Java.....	37
3.7	SERVIZI PER LA GESTIONE DEI REGISTRI	38
3.7.1	SERVIZIO “ GETREGISTERORRF”	38
3.7.1.1	Descrizione oggetto “Request”	38
3.7.1.2	Descrizione oggetto “Response”	38
3.7.1.3	Esempio di utilizzo C#	39
3.7.1.4	Esempio di utilizzo Java.....	39
3.7.2	SERVIZIO “ GETREGISTERSORRF”	40
3.7.2.1	Descrizione oggetto “Request”	40
3.7.2.2	Descrizione oggetto “Response”	40
3.7.2.3	Esempio di utilizzo C#	41
3.7.2.4	Esempio di utilizzo Java.....	41
3.8	SERVIZI PER LA GESTIONE DEI TITOLARI.....	42
3.8.1	SERVIZIO “GETALLCLASSIFICATIONSCHEMES”	42
3.8.1.1	Descrizione oggetto “Request”	42
3.8.1.2	Descrizione oggetto “Response”	43
3.8.1.3	Esempio di utilizzo C#	43
3.8.1.4	Esempio di utilizzo Java.....	43
3.8.2	SERVIZIO “GETACTIVECLASSIFICATIONSCHEME”	44
3.8.2.1	Descrizione oggetto “Request”	44

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.8.2.2	Descrizione oggetto “Response”	45
3.8.2.3	Esempio di utilizzo C#	45
3.8.2.4	Esempio di utilizzo Java.....	45
3.8.3	SERVIZIO “GETCLASSIFICATIONSCHEMEBYID”.....	46
3.8.3.1	Descrizione oggetto “Request”.....	46
3.8.3.2	Descrizione oggetto “Response”	47
3.8.3.3	Esempio di utilizzo C#	47
3.8.3.4	Esempio di utilizzo Java.....	48
3.9	SERVIZI PER LA GESTIONE DEI DOCUMENTI.....	48
3.9.1	SERVIZIO “CREATEDOCUMENT”.....	48
3.9.1.1	Descrizione oggetto “Request”.....	48
3.9.1.2	Creare un documento non protocollato	49
3.9.1.3	Creare un documento protocollato in entrata.....	52
3.9.1.4	Creare un documento protocollato in uscita.....	55
3.9.1.5	Creare un documento protocollato interno.....	59
3.9.1.6	Creare un documento predisposto	62
3.9.1.7	Descrizione oggetto “Response”	62
3.9.2	SERVIZIO “GETTEMPLATESDOCUMENTS”	64
3.9.2.1	Descrizione oggetto “Request”.....	64
3.9.2.2	Descrizione oggetto “Response”	64
3.9.2.3	Esempio di utilizzo C#	65
3.9.2.4	Esempio di utilizzo Java.....	65
3.9.3	SERVIZIO “GETTEMPLATEDOC”	66
3.9.3.1	Descrizione oggetto “Request”.....	66
3.9.3.2	Descrizione oggetto “Response”	66
3.9.3.3	Esempio di utilizzo C#	67
3.9.3.4	Esempio di utilizzo Java.....	67
3.9.4	SERVIZIO “GETDOCUMENT”	68
3.9.4.1	Descrizione oggetto “Request”.....	68
3.9.4.2	Descrizione oggetto “Response”	69
3.9.4.3	Esempio di utilizzo C#	69
3.9.4.4	Esempio di utilizzo Java.....	70
3.9.5	SERVIZIO “UPLOADFILETODOCUMENT”	70
3.9.5.1	Descrizione oggetto “Request”.....	70
3.9.5.2	Descrizione oggetto “Response”	71
3.9.5.3	Esempio di utilizzo C#	72
3.9.5.4	Esempio di utilizzo Java.....	72
3.9.6	SERVIZIO “GETFILEDOCUMENTBYID”	73
3.9.6.1	Descrizione oggetto “Request”.....	73
3.9.6.2	Descrizione oggetto “Response”	74
3.9.6.3	Esempio di utilizzo C#	75
3.9.6.4	Esempio di utilizzo Java.....	75
3.9.7	SERVIZIO “GETFILEWITHSIGNATUREORSTAMP”	76
3.9.7.1	Descrizione oggetto “Request”.....	76

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.9.7.2	Descrizione oggetto “Response”	76
3.9.7.3	Esempio di utilizzo C#	77
3.9.7.4	Esempio di utilizzo Java.....	77
3.9.8	SERVIZIO “ADDDOCINPROJECT”	78
3.9.8.1	Descrizione oggetto “Request”	78
3.9.8.2	Descrizione oggetto “Response”	79
3.9.8.3	Aggiunta del documento in un sottofascicolo.....	79
3.9.8.4	Esempio di utilizzo C#	80
3.9.8.5	Esempio di utilizzo Java.....	80
3.9.9	SERVIZIO “EDITDOCUMENT”	81
3.9.9.1	Descrizione oggetto “Request”	81
3.9.9.2	Descrizione oggetto “Response”	81
3.9.9.3	Esempio di utilizzo C#	82
3.9.9.4	Esempio di utilizzo Java.....	82
3.9.10	SERVIZIO “GETDOCUMENTSINPROJECT”	85
3.9.10.1	Descrizione oggetto “Request”	85
3.9.10.2	Descrizione oggetto “Response”	86
3.9.10.3	Esempio di utilizzo C#	87
3.9.10.4	Esempio di utilizzo Java	87
3.9.11	SERVIZIO “SEARCHDOCUMENTS”	88
3.9.11.1	Descrizione oggetto “Request”	88
3.9.11.2	Descrizione oggetto “Response”	89
3.9.11.3	Filtri disponibili.....	89
3.9.11.4	Estrazione dei campi profilati nella ricerca	91
3.9.11.5	Esempio di utilizzo C#	91
3.9.11.6	Esempio di utilizzo Java	92
3.9.12	SERVIZIO “EDITDOCSTATEDIAGRAM”	93
3.9.12.1	Descrizione oggetto “Request”	93
3.9.12.2	Descrizione oggetto “Response”	94
3.9.12.3	Esempio di utilizzo C#	94
3.9.12.4	Esempio di utilizzo Java	95
3.9.13	SERVIZIO “GETDOCUMENTSTATEDIAGRAM”	95
3.9.13.1	Descrizione oggetto “Request”	95
3.9.13.2	Descrizione oggetto “Response”	96
3.9.13.3	Esempio di utilizzo C#	96
3.9.13.4	Esempio di utilizzo Java	97
3.9.14	SERVIZIO “SENDDOCUMENT”	98
3.9.14.1	Descrizione oggetto “Request”	98
3.9.14.2	Descrizione oggetto “Response”	98
3.9.14.3	Esempio di utilizzo C#	99
3.9.14.4	Esempio di utilizzo Java	99
3.9.15	SERVIZIO “GETSTAMPANDSIGNATURE”	99
3.9.15.1	Descrizione oggetto “Request”	100
3.9.15.2	Descrizione oggetto “Response”	100

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.9.15.3	Esempio di utilizzo C#	101
3.9.15.4	Esempio di utilizzo Java.....	101
3.9.16	SERVIZIO “GETDOCUMENTFILTERS”.....	102
3.9.16.1	Descrizione oggetto “Request”	102
3.9.16.2	Descrizione oggetto “Response”	102
3.9.16.3	Esempio di utilizzo C#	102
3.9.16.4	Esempio di utilizzo Java.....	103
3.9.17	SERVIZIO “CREATEDOCUMENTANDADDINPROJECT”	103
3.9.17.1	Descrizione oggetto “Request”	104
3.9.17.2	Descrizione oggetto “Response”	104
3.9.17.3	Esempio di utilizzo C#	105
3.9.17.4	Esempio di utilizzo Java.....	105
3.9.18	SERVIZIO “FOLLOWDOCUMENT”	107
3.9.18.1	Descrizione oggetto “Request”	107
3.9.18.2	Descrizione oggetto “Response”	107
3.9.18.3	Esempio di utilizzo C#	108
3.9.18.4	Esempio di utilizzo Java.....	108
3.9.19	SERVIZIO “GETLINKDOCBYID”	109
3.9.19.1	Descrizione oggetto “Request”	109
3.9.19.2	Descrizione oggetto “Response”	109
3.9.19.3	Esempio di utilizzo C#	110
3.9.19.4	Esempio di utilizzo Java.....	110
3.10	SERVIZI PER LA GESTIONE AVANZATA DEI DOCUMENTI	111
3.10.1	SERVIZIO “REMOVEDOCUMENT”	111
3.10.1.1	Descrizione oggetto “Request”	111
3.10.1.2	Descrizione dell’oggetto “Response”.....	111
3.10.1.3	Esempio utilizzo C#	112
3.10.1.4	Esempio di utilizzo Java.....	112
3.10.2	SERVIZIO “SENDDOCUMENTADVANCED”	113
3.10.2.1	Descrizione oggetto “Request”	113
3.10.2.2	Descrizione dell’oggetto “Response”.....	114
3.10.2.3	Esempio utilizzo C#	115
3.10.2.4	Esempio di utilizzo Java.....	115
3.10.3	SERVIZIO “NUOVAFATTURAATTIVA”	116
3.10.3.1	Descrizione oggetto “Request”	116
3.10.3.2	Descrizione oggetto “Response”	117
3.10.3.3	Note riguardanti il servizio NuovaFatturaAttiva	118
3.10.4	SERVIZIO “NUOVOLOTTOATTIVO”	119
3.10.4.1	Descrizione oggetto “Request”	119
3.10.4.2	Descrizione oggetto “Response”	120
3.10.4.3	Note riguardanti il servizio NuovoLottoAttivo	121
3.11	SERVIZI PER LA GESTIONE DEI FASCICOLI	121
3.11.1	SERVIZIO “CREATEPROJECT”	121
3.11.1.1	Descrizione oggetto “Request”	121

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.11.1.2	Descrizione oggetto “Response”	122
3.11.1.3	Esempio di utilizzo C#.....	123
3.11.1.4	Esempio di utilizzo Java	124
3.11.2	SERVIZIO “GETTEMPLATESPROJECTS”	125
3.11.2.1	Descrizione oggetto “Request”.....	125
3.11.2.2	Descrizione oggetto “Response”	126
3.11.2.3	Esempio di utilizzo C#.....	126
3.11.2.4	Esempio di utilizzo Java	126
3.11.3	SERVIZIO “GETTEMPLATEPRJ”	127
3.11.3.1	Descrizione oggetto “Request”.....	127
3.11.3.2	Descrizione oggetto “Response”	128
3.11.3.3	Esempio di utilizzo C#.....	128
3.11.3.4	Esempio di utilizzo Java	129
3.11.4	SERVIZIO “GETPROJECT”	129
3.11.4.1	Descrizione oggetto “Request”.....	129
3.11.4.2	Descrizione oggetto “Response”	130
3.11.4.3	Esempio di utilizzo C#.....	130
3.11.4.4	Esempio di utilizzo Java	131
3.11.5	SERVIZIO “EDITPROJECT”	131
3.11.5.1	Descrizione oggetto “Request”.....	132
3.11.5.2	Descrizione oggetto “Response”	132
3.11.5.3	Esempio di utilizzo C#.....	133
3.11.5.4	Esempio di utilizzo Java	133
3.11.6	SERVIZIO “GETPROJECTSBYDOCUMENT”	134
3.11.6.1	Descrizione oggetto “Request”.....	134
3.11.6.2	Descrizione oggetto “Response”	134
3.11.6.3	Esempio di utilizzo C#.....	135
3.11.6.4	Esempio di utilizzo Java	135
3.11.7	SERVIZIO “EDITPRJSTATEDIAGRAM”	136
3.11.7.1	Descrizione oggetto “Request”.....	136
3.11.7.2	Descrizione oggetto “Response”	136
3.11.7.3	Esempio di utilizzo C#.....	137
3.11.7.4	Esempio di utilizzo Java	138
3.11.8	SERVIZIO “GETPROJECTSTATEDIAGRAM”	138
3.11.8.1	Descrizione oggetto “Request”.....	138
3.11.8.2	Descrizione oggetto “Response”	139
3.11.8.3	Esempio di utilizzo C#.....	140
3.11.8.4	Esempio di utilizzo Java	140
3.11.9	SERVIZIO “OPENCLOSEPROJECT”	141
3.11.9.1	Descrizione oggetto “Request”.....	141
3.11.9.2	Descrizione oggetto “Response”	141
3.11.9.3	Esempio di utilizzo C#.....	142
3.11.9.4	Esempio di utilizzo Java	143
3.11.10	SERVIZIO “SEARCHPROJECTS”	143

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.11.10.1	Descrizione oggetto “Request”	143
3.11.10.2	Descrizione oggetto “Response”	144
3.11.10.3	Filtri disponibili	144
3.11.10.4	Estrazione dei campi profilati nella ricerca	146
3.11.10.5	Esempio di utilizzo C#	146
3.11.10.6	Esempio di utilizzo Java	147
3.11.11	SERVIZIO “GETPROJECTFILTERS”	148
3.11.11.1	Descrizione oggetto “Request”	148
3.11.11.2	Descrizione oggetto “Response”	148
3.11.11.3	Esempio di utilizzo C#	149
3.11.11.4	Esempio di utilizzo Java	149
3.11.12	SERVIZIO “CREATEFOLDER”	150
3.11.12.1	Descrizione oggetto “Request”	150
3.11.12.2	Descrizione oggetto “Response”	151
3.11.12.3	Esempio di utilizzo C#	151
3.11.12.4	Esempio di utilizzo Java	152
3.11.13	SERVIZIO “GETPROJECTFOLDERS”	152
3.11.13.1	Descrizione oggetto “Request”	152
3.11.13.2	Descrizione oggetto “Response”	153
3.11.13.3	Esempio di utilizzo C#	154
3.11.13.4	Esempio di utilizzo Java	154
3.11.14	SERVIZIO “FOLLOWPROJECT”	155
3.11.14.1	Descrizione oggetto “Request”	155
3.11.14.2	Descrizione oggetto “Response”	156
3.11.14.3	Esempio di utilizzo C#	156
3.11.14.4	Esempio di utilizzo Java	157
3.11.15	SERVIZIO “GETLINKPRJBYID”	157
3.11.15.1	Descrizione oggetto “Request”	157
3.11.15.2	Descrizione oggetto “Response”	158
3.11.15.3	Esempio di utilizzo C#	158
3.11.15.4	Esempio di utilizzo Java	159
3.12	SERVIZI PER LA GESTIONE DELLA RUBRICA	159
3.12.1	SERVIZIO “ADDCORRESPONDENT”	159
3.12.1.1	Descrizione oggetto “Request”	159
3.12.1.2	Descrizione oggetto “Response”	161
3.12.1.3	Esempio di utilizzo C#	162
3.12.1.4	Esempio di utilizzo Java	162
3.12.2	SERVIZIO “DELETECORRESPONDENT”	163
3.12.2.1	Descrizione oggetto “Request”	163
3.12.2.2	Descrizione oggetto “Response”	164
3.12.2.3	Esempio di utilizzo C#	164
3.12.2.4	Esempio di utilizzo Java	165
3.12.3	SERVIZIO “EDITCORRESPONDENT”	165
3.12.3.1	Descrizione oggetto “Request”	165

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.12.3.2	Descrizione oggetto “Response”	167
3.12.3.3	Esempio di utilizzo C#.....	168
3.12.3.4	Esempio di utilizzo Java	168
3.12.4	SERVIZIO “GETCORRESPONDENT”	169
3.12.4.1	Descrizione oggetto “Request”	169
3.12.4.2	Descrizione oggetto “Response”	169
3.12.4.3	Esempio di utilizzo C#.....	170
3.12.4.4	Esempio di utilizzo Java	170
3.12.5	SERVIZIO “SEARCHUSERS”	171
3.12.5.1	Descrizione oggetto “Request”	171
3.12.5.2	Descrizione oggetto “Response”	171
3.12.5.3	Filtri disponibili.....	172
3.12.5.4	Esempio di utilizzo C#.....	172
3.12.5.5	Esempio di utilizzo Java	173
3.12.6	SERVIZIO “SEARCHCORRESPONDENTS”	173
3.12.6.1	Descrizione oggetto “Request”	173
3.12.6.2	Descrizione oggetto “Response”	174
3.12.6.3	Filtri disponibili.....	174
3.12.6.4	Esempio di utilizzo C#.....	175
3.12.6.5	Esempio di utilizzo Java	176
3.12.7	SERVIZIO “GETCORRESPONDENTFILTERS”	178
3.12.7.1	Descrizione oggetto “Request”	178
3.12.7.2	Descrizione oggetto “Response”	178
3.12.7.3	Esempio di utilizzo C#.....	178
3.12.7.4	Esempio di utilizzo Java	179
3.12.8	SERVIZIO “GETUSERFILTERS”	180
3.12.8.1	Descrizione oggetto “Request”	180
3.12.8.2	Descrizione oggetto “Response”	180
3.12.8.3	Esempio di utilizzo C#.....	180
3.12.8.4	Esempio di utilizzo Java	181
3.13	SERVIZI PER LA GESTIONE DEI TOKEN DI AUTENTICAZIONE.....	181
3.13.1	SERVIZIO “GETAUTHENTICATIONTOKEN”	182
3.13.1.1	Descrizione oggetto “Request”	182
3.13.1.2	Descrizione oggetto “Response”	182
3.13.1.3	Esempio di utilizzo C#.....	183
3.13.1.4	Esempio di utilizzo Java	183
3.13.2	SERVIZIO “GETTOKEN”	183
3.13.2.1	Descrizione oggetto “Request”	184
3.13.2.2	Descrizione oggetto “Response”	184
3.13.2.3	Esempio di utilizzo C#.....	185
3.13.2.4	Esempio di utilizzo Java	185
3.14	SERVIZI PER LA GESTIONE DEI MODELLI DI TRASMISSIONE	185
3.14.1	SERVIZIO “EXECUTETRANSMDOCMODEL”	185
3.14.1.1	Descrizione oggetto “Request”	186

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.14.1.2	Descrizione oggetto “Response”	186
3.14.1.3	Esempio di utilizzo C#	187
3.14.1.4	Esempio di utilizzo Java.....	187
3.14.2	SERVIZIO “EXECUTETRANSMPrjMODEL”	187
3.14.2.1	Descrizione oggetto “Request”	188
3.14.2.2	Descrizione oggetto “Response”	188
3.14.2.3	Esempio di utilizzo C#	189
3.14.2.4	Esempio di utilizzo Java.....	189
3.14.3	SERVIZIO “GETTRANSMISSIONMODEL”	190
3.14.3.1	Descrizione oggetto “Request”	190
3.14.3.2	Descrizione oggetto “Response”	190
3.14.3.3	Esempio di utilizzo C#	191
3.14.3.4	Esempio di utilizzo Java.....	191
3.14.4	SERVIZIO “GETTRANSMISSIONMODELS”	192
3.14.4.1	Descrizione oggetto “Request”	192
3.14.4.2	Descrizione oggetto “Response”	193
3.14.4.3	Esempio di utilizzo C#	193
3.14.4.4	Esempio di utilizzo Java.....	194
3.14.5	SERVIZIO “EXECUTETRANSMISSIONDOCUMENT”	194
3.14.5.1	Descrizione oggetto “Request”	194
3.14.5.2	Descrizione oggetto “Response”	195
3.14.5.3	Esempio di utilizzo C#	196
3.14.5.4	Esempio di utilizzo Java.....	197
3.14.6	SERVIZIO “EXECUTETRANSMISSIONPROJECT”	197
3.14.6.1	Descrizione oggetto “Request”	197
3.14.6.2	Descrizione oggetto “Response”	198
3.14.6.3	Esempio di utilizzo C#	199
3.14.6.4	Esempio di utilizzo Java.....	199
3.14.7	SERVIZIO “GIVEUPRIGHTS”	200
3.14.7.1	Descrizione oggetto “Request”	200
3.14.7.2	Descrizione oggetto “Response”	201
3.14.7.3	Esempio di utilizzo C#	201
3.14.7.4	Esempio di utilizzo Java.....	202
3.15	SERVIZI PER LA GESTIONE DEL LIBROFIRMA	202
3.15.1	SERVIZIO “GETSIGNATUREPROCESSES”	202
3.15.1.1	Descrizione oggetto “Request”	202
3.15.1.2	Descrizione oggetto “Response”	203
3.15.1.3	Caso d’uso : come ottenere la lista dei processi o modelli di processo visibili ad un ruolo.	204
3.15.2	SERVIZIO “STARTSIGNATUREPROCESS”	204
3.15.2.1	Descrizione oggetto “Request”	204
3.15.2.2	Descrizione oggetto “Response”	207
3.15.2.3	Caso d’uso avvio di un processo di firma inserendo un documento in esso.	208

RICHIESTE DI MANUTENZIONE EVOLUTIVA PER IL SISTEMA P.I.TRE. NUOVI SERVIZI WEB “PRODUCT INTEGRATION SERVICES”

P.I.TRE.

DOCUMENTAZIONE TECNICA

3.15.3	SERVIZIO “SEARCHSIGNPROCESSINSTANCES”	209
3.15.3.1	Descrizione oggetto “Request”	209
3.15.3.2	Descrizione oggetto “Response”	210
3.15.3.3	Caso d’uso del ws SearchSignProcessInstances	213
3.15.4	SERVIZIO “GETINSTANCESEARCHFILTERS”	213
3.15.4.1	Descrizione oggetto “Request”	213
3.15.4.2	Descrizione oggetto “Response”	214
3.15.5	SERVIZIO “GETSIGNATUREPROCESS”	216
3.15.5.1	Descrizione oggetto “Request”	216
3.15.5.2	Descrizione oggetto “Response”	217
3.15.6	SERVIZIO “GETSIGNPROCESSINSTANCE”	218
3.15.6.1	Descrizione oggetto “Request”	218
3.15.6.2	Descrizione oggetto “Response”	219
3.15.7	SERVIZIO “INTERRUPTSIGNATUREPROCESS”	222
3.15.7.1	Descrizione oggetto “Request”	222
3.15.7.2	Descrizione oggetto “Response”	222

1 INTRODUZIONE

1.1 PREMESSA

I servizi web “Product Integration Services” sono stati realizzati nell’ambito della manutenzione evolutiva del sistema P.I.Tre.

L’esigenza espressa dalla Committente è quella di offrire dei servizi ad applicazioni o prodotti di terze parti per integrarsi con il protocollo federato P.I.Tre..

Lo scopo del presente documento è quello di fornire una guida pratica di utilizzo per gli sviluppatori delle applicazioni verticali che necessitano di utilizzare i servizi web “Product Integration Services” (d’ora in avanti, “PiS”).

Per gli esempi Java sono state utilizzate le librerie di Axis2 versione 1.5.5.

1.2 MODELLO DI COMUNICAZIONE REQUEST - RESPONSE

I servizi PiS, coerentemente col paradigma SOA, sono:

- completamente indipendenti dai “tipi” esposti dagli altri servizi di P.I.Tre.;
- indipendenti dalla piattaforma sottostante e dalla tecnologia di sviluppo (es. Java, .NET);

Il modello di comunicazione utilizzato dai servizi è il Request – Response. Tutti i metodi esposti funzionano in maniera simile, coerente e semplice: per effettuare una richiesta è necessario che gli utilizzatori creino un “pacchetto” di invio contenente:

- i dati relativi all’autenticazione dell’utente (obbligatori)
- i parametri specifici richiesti dal metodo

Alla richiesta seguirà l’invio, da parte del server, di un “pacchetto” di risposta contenente i dati specifici restituiti dal servizio oppure, in caso di errore, un opportuno “pacchetto” contenente il codice identificativo dello stesso per consentire alle verticali di intercettare correttamente l’errore e applicare di conseguenza le opportune “logiche” condizionali.

2 MODELLO DI DOMINIO

2.1 DIAGRAMMA DELLE CLASSI

I seguenti diagrammi mostrano le classi introdotte per implementare il modello di dominio che gli sviluppatori delle applicazioni verticali utilizzeranno per interagire con il “mondo” del P.I.Tre.

2.1.1 CLASSE DOCUMENT

┌

Figura 1 – Classe Document

Nome campo	Tipologia di dato	Descrizione
Annulled	Boolean	Se il valore è “true” il documento è annullato.
AnnulmentDate	String	Data di annullamento di un protocollo.
ArrivalDate	String	Data di arrivo del protocollo in ingresso. Formato “dd/MM/yyyy HH:mm”.
Attachments	File[]	Allegati del documento
ConsolidationState	String	Se il documento è consolidato né indica lo stato. Può avere i seguenti valori: <ul style="list-style-type: none"> • CONS_CONT: il documento è consolidato nel contenuto. • CONS_CONT_META: il documento è consolidato nel contenuto e nei meta dati
CreationDate	String	Indica la data di creazione del documento.
DataProtocolSender	String	Indica la data del protocollo mittente.
DocNumber	String	Numero univoco del documento.
DocumentType	String	Tipologia del documento. Può assumere i seguenti valori: <ul style="list-style-type: none"> • “A”: protocollo o predisposto in arrivo. • “P”: protocollo o predisposto in partenza. • “T”: protocollo o predisposto interno. • “G”: documento non protocollato.
Id	String	System id univoco del documento.
IdParent	String	Se il documento è un allegato indica il docNumbef del documento principale
InBasket	Boolean	Se il valore è a “true” indica che il documento è nel cestino.
IsAttachments	Boolean	Se il valore è a “true” indica che il documento è un allegato.
MainDocument	File	Documento principale

MeansOfSending	String	Mezzo di spedizione per un protocollo/predisposto in arrivo. Può assumere i seguenti valori: <ul style="list-style-type: none"> • “Hand_delivery”: consegna a mano • “Fax”: fax • “Interoperability”: interoperabilità • “P_Interoperability”: interoperabilità Pitre • “Letter”: lettera • “Mail”: mail • “Certified_Mail”: raccomandata
MultipleSenders	Correspondent[]	Nel caso di protocollo in arrivo lista dei mittenti multipli
Note	Note[]	Note del documento
Object	String	Oggetto del documento.
PersonalDocument	Boolean	Se il valore è impostato a “true” il documento è visibile solo all’utente creatore.
Predisposed	Boolean	Se il valore è impostato a “true” il documento è un predisposto.
PrivateDocument	Boolean	Se il valore è impostato a “true” il documento è privato.
ProtocolDate	String	Data di protocollazione del documento.
ProtocolSender	String	Indica il protocollo mittente nel caso di un protocollo in arrivo.
Recipients	Correspondent[]	Destinatari del documento.
RecipientsCC	Correspondent[]	Destinatari per conoscenza.
Register	Register	Nel caso di protocollo il registro in cui è protocollato.
Sender	Correspondent	Mittente del documento.
Signature	String	Segnatura del protocollo.
Template	Template	Tipologia del documento.

2.1.2 CLASSE CLASSIFICATIONSCHEME



Figura 2 – Classe Classification Scheme

Nome campo	Tipologia di dato	Descrizione
Id	String	Id univoco del titolare.
Description	String	Descrizione del titolare.
Active	Boolean	Se il valore è impostato a “true” indica che il titolare è attivo.

2.1.3 CLASSE CORRESPONDENT

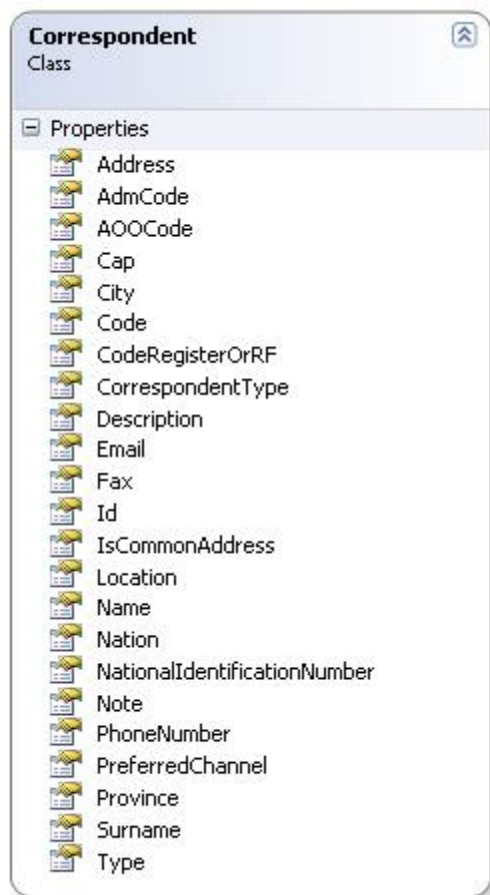


Figura 3 – Classe Correspondent

Nome campo	Tipologia di dato	Descrizione
Address	String	Indirizzo del corrispondente.
AdmCode	String	Codice dell'amministrazione.
AOOCode	String	Codice AOO.
Cap	String	Codice di avviamento postale.
City	String	Città del corrispondente.
Code	String	Codice rubrica all'interno del sistema.
CodeRegisterOrRf	String	Se il corrispondente è di registro o rf è valorizzato il codice.
CorrespondentType	String	Tipologia di corrispondente. Può assumere i seguenti valori: <ul style="list-style-type: none"> • "U": indica che il corrispondente

		<p>è una unità organizzativa.</p> <ul style="list-style-type: none"> • “R”: indica che il corrispondente è un ruolo. • “P”: indica che il corrispondente è una persona.
Description	String	Descrizione del corrispondente.
Email	String	Indirizzo mail del corrispondente.
Fax	String	Fax del corrispondente.
Id	String	System id univoco del corrispondente.
IsCommonAddress	Boolean	Indica se il corrispondente è memorizzato nella rubrica comune
Location	String	Località del corrispondente.
Name	String	Nome del corrispondente.
Nation	String	Nazione del corrispondente.
NationalIdentificationNumber	String	Codice fiscale del corrispondente.
Note	String	Note.
PhoneNumber	String	Numero di telefono del corrispondente.
PreferredChannel	String	Canale preferenziale.
Province	String	Provincia (sigla).
Surname	String	Cognome.
Type	String	<p>Tipo di corrispondente. Può assumere i seguenti valori:</p> <ul style="list-style-type: none"> • “I”: indica che il corrispondente è interno • “E”: indica che il corrispondente è esterno. • “O”: indica che il corrispondente è occasionale.
VatNumber	String	Partita Iva.

2.1.4 CLASSE TEMPLATE

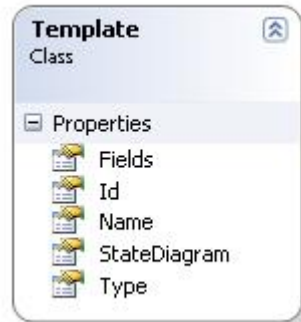


Figura 4 – Classe Template

Nome campo	Tipologia di dato	Descrizione
Fields	Field[]	Lista dei campi della tipologia.
Id	String	Id del documento.
Name	String	Descrizione della tipologia.
StateDiagram	StateDiagram	Diagramma di stato associato
Type	String	Indica la tipologia a cui è associato un Template, se a un documento oppure ad un fascicolo. D per Documento, F per Fascicolo.

2.1.5 CLASSE FIELD

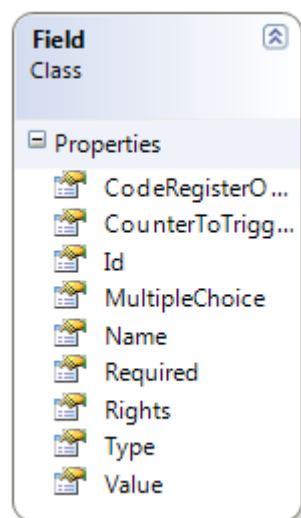


Figura 5 – Classe Field

Nome campo	Tipologia di dato	Descrizione
CodeRegisterOrRF	String	Codice del registro o rf nel caso di contatore che necessita di questi campi.
CounterToTrigger	Boolean	Se il campo è un contatore e il contatore deve scattare il valore è “true”.
Id	String	Id del campo.
MultipleChoice	String[]	Nel caso di campo di tipo “casella di selezione” con scelta multipla la lista sarà valorizzata con le selezioni.
Name	String	Descrizione del campo.
Required	Boolean	Se ha valore “true” indica che il campo è obbligatorio.
Rights	String	Diritti che il ruolo dell’utente ha sul campo del template. Può assumere i seguenti valori: <ul style="list-style-type: none"> • “NONE”: il ruolo non ha diritti sul campo • “SEARCH_AND_VIEW”: il ruolo ha solo diritti di visualizzazione • “INSERT_AND_MODIFY”: il ruolo ha diritti di modifica
Type	String	Tipo di oggetto. Può assumere i seguenti valori: <ul style="list-style-type: none"> • “Correspondent”: campo di tipo corrispondente • “TextField”: campo di testo • “MultipleChoice”: casella di selezione • “ExclusiveSelection”: selezione esclusiva • “DropDown”: menù a tendina • “Counter”: contatore
Value	String	Valore del campo. Nel caso di casella di selezione valorizzare MultipleChoice

2.1.6 CLASSE FILE

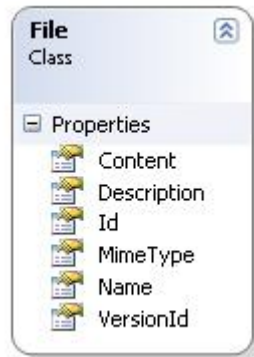


Figura 6 – Classe File

Nome campo	Tipologia di dato	Descrizione
Content	byte[]	Contenuto del file.
Description	String	Descrizione del file.
Id	String	Id del file.
MimeType	String	Mime del file.
Name	String	Nome del file e sua estensione.
VersionId	String	Numero di versione

2.1.7 CLASSE FILTER

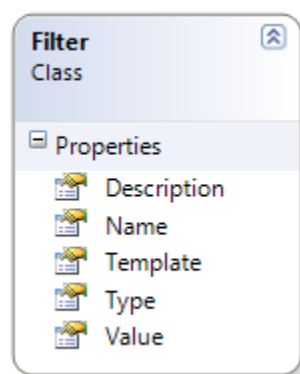


Figura 7 – Classe Filter

Nome campo	Tipologia di dato	Descrizione
Description	String	Descrizione del filtro

Name	String	Nome del filtro.
Template	String	Inserire i campi che si desidera ricercare di una tipologia di documento o fascicolo.
Type	FilterTypeEnum	Tipo del Filtro
Value	String	Valore della ricerca.

2.1.7.1 Enumerato FilterTypeEnum

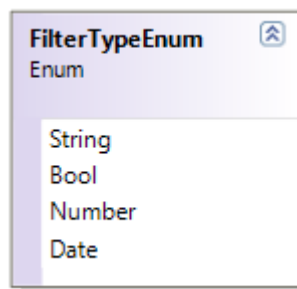


Figura 8.1 – Enum FilterTypeEnum

Nome campo	Tipologia di dato	Descrizione
String	EnumMember	Membro enumerato di tipo String.
Bool	EnumMember	Membro enumerato di tipo Bool.
Number	EnumMember	Membro enumerato di tipo Number.
Date	EnumMember	Membro enumerato di tipo Date nel formato dd/MM/yyyy.

2.1.8 CLASSE ROLE



Figura 9 – Classe Role

Nome campo	Tipologia di dato	Descrizione
Code	String	Codice del ruolo.
Description	String	Descrizione del ruolo.
Id	String	Id univo del ruolo.
Registers	Register[]	Registri/RF del ruolo.

2.1.9 CLASSE REGISTER



Figura 10 – Classe Register

Nome campo	Tipologia di dato	Descrizione
Code	String	Codice del registro.
Description	String	Descrizione del registro.
Id	String	Id univo del registro.
IsRF	Boolean	Indica se si tratta di un RF, in questo caso assume il valore “true”.
State	String	Stato del registro. Può assumere i seguenti valori: <ul style="list-style-type: none"> • “Open”: registro aperto • “Closed”: registro chiuso • “Intermediate”: registro in attesa.

2.1.10 CLASSE USER

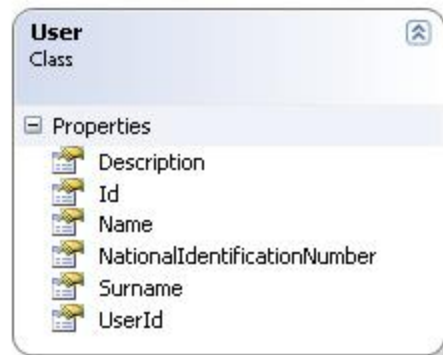


Figura 11 – Classe User

Nome campo	Tipologia di dato	Descrizione
Description	String	Descrizione dell'utente.
Id	String	Id univoco dell'utente.
Name	String	Nome dell'utente.
NationalIdentificationNumber	String	Codice fiscale/p.iva dell'utente.
Surname	String	Cognome dell'utente.
UserId	String	Codice dell'utente.

2.1.11 CLASSE NOTE



Figura 12 – Classe Note

Nome campo	Tipologia di dato	Descrizione
Description	String	Descrizione della nota.
Id	String	Id univoco della nota.

User	User	Creatore della nota.
------	------	----------------------

2.1.12 CLASSE PROJECT

E

Figura 13 – Classe Project

Nome campo	Tipologia di dato	Descrizione
ClassificationScheme	ClassificationScheme	Titolario del fascicolo.
ClosureDate	String	Data di chiusura del fascicolo.
Code	String	Codice del fascicolo.
CodeNodeClassification	String	Codice del nodo in cui è stato creato il fascicolo.
CollocationDate	String	Data di collocazione.
Controlled	Boolean	Indica se il fascicolo è controllato.
CreationDate	String	Data di creazione.

Description	String	Descrizione del fascicolo.
Id	String	Id univoco del fascicolo.
IdParent	String	Id del fascicolo padre.
Note	Note[]	Note del fascicolo.
Number	String	Numero del fascicolo.
Open	Boolean	Se il valore è “true” il fascicolo è aperto.
OpeningDate	String	Data di apertura di un fascicolo.
Paper	Boolean	Se il valore è “true” il fascicolo è cartaceo.
PhysicsCollocation	String	Collocazione fisica.
Private	Boolean	Se il valore è “true” il fascicolo è privato.
Register	Register	Registro di AOO a cui è collegato il fascicolo.
Template	Template	Tipologia del fascicolo.
Type	String	Indica se il fascicolo è G generale, P procedimentale.

2.1.13 CLASSE FOLDER

Figura 13.1 – Classe Folder

Nome Campo	Tipologia di Dato	Descrizione
Description	String	Nome del sottofascicolo
Id	String	Id del sottofascicolo
IdParent	String	Id del fascicolo/

		sottofascicolo padre
IdProject	String	Id del fascicolo di appartenenza

2.1.14 CLASSE STATEDIAGRAM



Figura 14 – Classe StateDiagram

Nome campo	Tipologia di dato	Descrizione
Description	String	Descrizione del diagramma.
Id	String	Id univoco del diagramma.
StateOfDiagram	StateOfDiagram[]	Stati del diagramma.

2.1.15 CLASSE STATEOFDIAGRAM

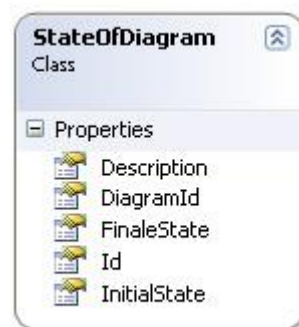


Figura 15 – Classe StateOfDiagram

Nome campo	Tipologia di dato	Descrizione
Description	String	Descrizione dello stato.
DiagramId	String	Id del diagramma al quale lo stato è

		associato.
FinaleState	Boolean	Se “true” indica lo stato finale.
Id	String	Id univoco dello stato.
InitialState	Boolean	Se “true” indica lo stato iniziale.

2.1.16 CLASSE TRANSMISSIONMODEL

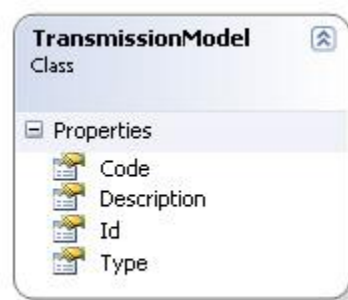


Figura 16 – Classe TransmissionModel

Nome campo	Tipologia di dato	Descrizione
Code	String	Codice del modello di trasmissione.
Description	String	Descrizione.
Id	String	Id univoco del modello.
Type	String	Tipo di modello, valore “D” per i modelli dei documenti e “F” per i modelli dei fascicoli.

2.1.17 CLASSE STAMP

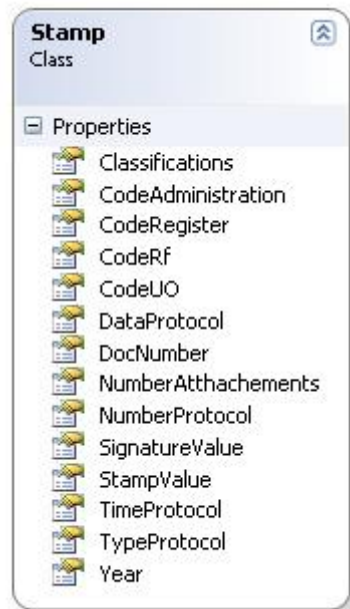


Figura 17 – Classe Stamp

Nome campo	Tipologia di dato	Descrizione
Classifications	String	Codici dei fascicoli in cui è classificato il documento.
CodeAdministration	String	Codice dell'amministrazione.
CodeRegister	String	Codice del registro.
CodeRf	String	Codice dell'RF.
CodeUO	String	Codice UO.
DataProtocol	String	Data protocollo.
DocNumber	String	DocNumber.
NumberAttachements	String	Numero di allegati.
NumberProtocol	String	Numero di protocollo.
SignatureValue	String	Valore della segnatura del protocollo.
StampValue	String	Valore del timbro del documento.
TimeProtocol	String	Ora della creazione del protocollo.
TypeProtocol	String	Tipo di documento (G per grigio, A per arrivo, P per partenza, I per interno).
Year	String	Anno di creazione.

2.1.18 CLASSE SENDINGRESULT

e

Nome campo	Tipologia di dato	Descrizione
CorrespondentDescription	String	Descrizione del corrispondente
CorrespondentId	String	Id del corrispondente
Mail	String	Mail destinataria della spedizione, se presente
PrefChannel	String	Canale preferenziale del corrispondente
SendingRes	String	Esito della spedizione

2.1.19

2.2 ENUMERAZIONI

Nel presente paragrafo vengono illustrate le enumerazioni introdotte per implementare il modello di dominio che gli sviluppatori delle applicazioni verticali utilizzeranno per interagire con il “mondo” del P.I.Tre.

2.2.1 ENUMERATO OPERATIONFOLLOW

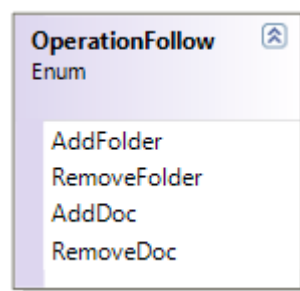


Figura 18 – Enum OperationFollow

Nome campo	Tipologia di dato	Descrizione
AddFolder	EnumMember	Membro enumerato di tipo int. Utilizzato per richiedere il monitoraggio di un determinato fascicolo.
RemoveFolder	EnumMember	Membro enumerato di tipo int. Utilizzato per richiedere al sistema che un determinato fascicolo non sia più monitorato.
AddDoc	EnumMember	Membro enumerato di tipo int. Utilizzato per richiedere il monitoraggio di un determinato documento.
RemoveDoc	EnumMember	Membro enumerato di tipo int. Utilizzato per richiedere al sistema che un determinato documento non sia più monitorato.

3 DESCRIZIONE DEI SERVIZI

3.1 STRUTTURA DEI PACCHETTI DI REQUEST

Coerentemente col modello di comunicazione introdotto nel paragrafo 3.1 ogni servizio richiederà un oggetto “Request” per il passaggio dei parametri di input.

Ogni metodo esposto dai servizi specializza un oggetto Request a seconda delle proprie esigenze. Tutti gli oggetti Request ereditano il proprio comportamento da una classe base astratta, come riportato nella figura seguente.

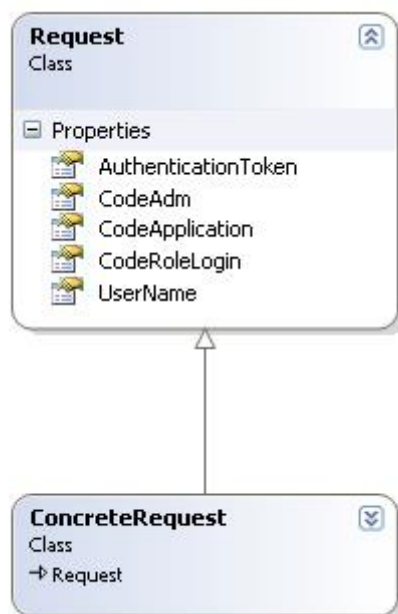


Figura 19 – Modello Request: classe base astratta

La classe base (e quindi tutte le request derivate) richiede (come dati obbligatori):

- il nome dell'utente che accede al servizio;
- il codice dell'amministrazione dell'utente che accede al servizio;

Le classi derivate estenderanno il comportamento della classe Request richiedendo le informazioni specifiche richieste dal servizio.

3.2 STRUTTURA DEI PACCHETTI DI RESPONSE

Ogni servizio restituirà alle applicazioni verticali un oggetto Response specializzato a seconda delle proprie esigenze. Tutti gli oggetti Response ereditano il proprio comportamento da una classe base astratta, come riportato nella figura seguente.

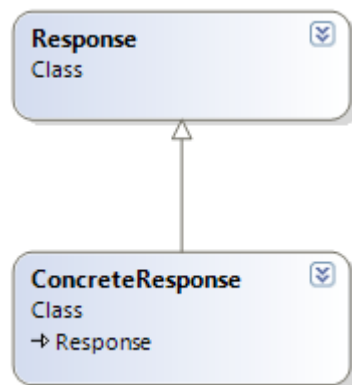


Figura 20 – Modello Response

Le classi derivate estenderanno il comportamento della classe Response restituendo le informazioni specifiche del servizio.

3.3 GESTIONE DELLE ECCEZIONI

Ogni oggetto Response è sottoposto ad un processo di verifica. In caso di errore viene lanciata un'eccezione di tipo PISFaultException in cui sono descritti il codice di errore e la sua descrizione.

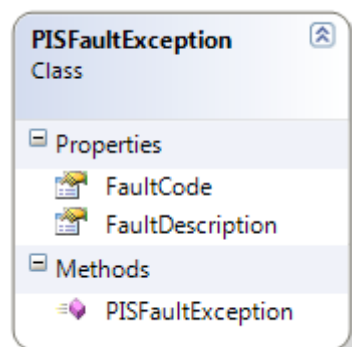


Figura 19 – PISFaultException

3.4 AUTENTICAZIONE

Al fine di poter dialogare con i servizi di P.I.Tre., verrà fornito un certificato per connettersi alla farm di test dei servizi, ed uno per connettersi alla farm di produzione.

Una volta installato il certificato sul client, sarà possibile caricare i wsdl dei servizi connettendosi agli URL:

Test
https://ws-t.pitre.tn.it/wcfrouting/wsdl/AddressBook.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/ClassificationSchemes.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/Documents.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/Projects.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/Token.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/Transmissions.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/Roles.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/Registers.wsdl https://ws-t.pitre.tn.it/wcfrouting/wsdl/LibroFirma.wsdl
Produzione
https://ws-p.pitre.tn.it/wcfrouting/wsdl/AddressBook.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/ClassificationSchemes.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/Documents.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/Projects.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/Token.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/Transmissions.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/Roles.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/Registers.wsdl https://ws-p.pitre.tn.it/wcfrouting/wsdl/LibroFirma.wsdl

In seguito, il certificato fornito dovrà essere censito per ogni amministrazione con la quale si intende interagire nella configurazione del routing dei servizi.

In caso di mancata o errata configurazione, il routing potrà restituire i seguenti messaggi di errore:

<i>Messaggio</i>	<i>Significato</i>
Amministrazione non configurata nel file locale	Il parametro CodeAdm non è stato configurato correttamente. Controllare la correttezza dello stesso.
Codice APPLICATION_NAME inserito non valido per il codice amministrazione inserito	Il certificato fornito non è stato configurato sull'amministrazione con la quale si sta cercando di interagire. Confrontarsi con un referente tecnico nel caso si ottenga questo errore.

Per il corretto utilizzo dei servizi, all'inizio di ogni sequenza di chiamate, sarà necessario chiamare il metodo `GetAuthenticationToken`, indicando il codice dell'amministrazione (`CodeAdm`), lo username fornito (`Username`), e il codice applicazione (`CodeApplication`).

Nel caso sia stato fornito, è necessario inserire anche il codice Ruolo (`CodeRoleLogin`).

Il metodo indicato fornirà un `AuthenticationToken` da inserire nelle seguenti chiamate.

Il Token di autenticazione ha una validità temporale di 20 minuti, quindi può essere utilizzato per più chiamate consecutive.

3.5 SERVIZI ESPOSTI

Le seguenti tabelle elencano tutti i metodi esposti dai PiS.

3.5.1 SERVIZI PER LA GESTIONE DEI RUOLI

<i>Servizio</i>	<i>Descrizione</i>
GetRole	Servizio per il reperimento dei dettagli di un ruolo dato il codice del ruolo o l'id.
GetRoles	Servizio per il reperimento di tutti i ruoli disponibili ad un utente.
GetRolesForEnabledActions	Servizio per il prelievo dei ruoli dato un utente ed una funzione richiesta.
GetUsersInRole	Servizio per il reperimento degli utenti in un ruolo.

3.5.2 SERVIZI PER LA GESTIONE DEI REGISTRI/RF

<i>Servizio</i>	<i>Descrizione</i>
GetRegisterOrRF	Servizio per il reperimento del dettaglio di un registro/RF dato il codice del registro/RF o l'id.
GetRegistersOrRF	Servizio per il reperimento di tutti gli RF o i registri disponibili per un ruolo.

3.5.3 SERVIZI PER LA GESTIONE DEI TITOLARI

<i>Servizio</i>	<i>Descrizione</i>
GetActiveClassificationScheme	Servizio per il reperimento del titolare attivo.
GetAllClassificationSchemes	Servizio per il reperimento di tutti i titolari.

GetClassificationSchemeById	Servizio per il reperimento dei dettagli di un titolare dato l'id.
------------------------------------	--

3.5.4 SERVIZI PER LA GESTIONE DEI DOCUMENTI

<i>Servizio</i>	<i>Descrizione</i>
AddDocInProject	Servizio per aggiungere un documento in un fascicolo.
CreateDocument	Servizio per la creazione di un documento non protocollato e/o protocollato o un predisposto.
CreateDocumentAndAddInProject	Servizio che crea un nuovo documento e lo inserisce nel relativo fascicolo indicato.
EditDocStateDiagram	Servizio per modificare lo stato del diagramma associato ad un documento.
EditDocument	Servizio per la modifica di un documento o protocollo.
ProtocolPredisposed	Servizio per la protocollazione di un predisposto senza effettuare modifiche.
FollowDocument	Servizio per la richiesta di “monitorare”/“non monitorare più” un documento presente nel Sistema.
GetDocument	Servizio per il reperimento del dettaglio di un documento data la segnatura o l'id.
GetDocumentFilters	Servizio che restituisce la lista dei filtri applicabili alla ricerca documenti.
GetDocumentsInProject	Servizio per il reperimento dei documenti contenuti in un fascicolo.
GetDocumentStateDiagram	Servizio per il reperimento dello stato di un documento.
GetFileDocumentById	Servizio per il reperimento di un file dato un id di un documento(versione)/allegato.
GetFileWithSignatureOrStamp	Servizio per il reperimento di un file con l'aggiunta della segnatura o del timbro. Etichetta PDF o timbro soprainpressi.
GetLinkDocById	Servizio per il recupero di un URL per l'apertura diretta del documento (previa

	autenticazione all'interfaccia PITRE)
GetStampAndSignature	Servizio per il reperimento dei dati del timbro e della segnatura.
GetTemplateDoc	Servizio per il reperimento del dettaglio di una tipologia di documento dato il nome o l'id.
GetTemplatesDocuments	Servizio per il reperimento di tutte le tipologie di documenti.
SearchDocuments	Servizio per la ricerca di documenti
SendDocument	Servizio per l'invio di un documento.
UploadFileToDocument	Servizio per aggiungere un file ad un documento, per creare un allegato o per aggiungere una nuova versione del file.

3.5.5 SERVIZI PER LA GESTIONE AVANZATA DEI DOCUMENTI

<i>Servizio</i>	<i>Descrizione</i>
RemoveDocument	Servizio per collocare un documento in cestino
SendDocumentAdvanced	Servizio per la spedizione di un protocollo in partenza, permettendo la scelta del registro, della mail di invio e dei destinatari. Restituisce l'esito delle singole spedizioni. Evoluzione del metodo SendDocument.
NuovaFatturaAttiva	Servizio per il caricamento di una fattura attiva.
NuovoLottoAttivo	Servizio per il caricamento di un lotto di fatture attive.

3.5.6 SERVIZI PER LA GESTIONE DEI FASCICOLI

<i>Servizio</i>	<i>Descrizione</i>
CreateFolder	Servizio per la creazione dei sottofascicoli.
CreateProject	Servizio per la creazione di un fascicolo.
EditPrjStateDiagram	Servizio per modificare lo stato del diagramma associato ad un fascicolo.
EditProject	Servizio per la modifica di un fascicolo.
FollowProject	Servizio per la richiesta di "monitorare"/"non monitorare più" un fascicolo presente nel sistema.
GetLinkPrjById	Servizio per il recupero di un URL per l'apertura diretta del fascicolo (previa autenticazione all'interfaccia PITRE)

GetProject	Servizio per il reperimento dei dati di un fascicolo dato un codice o l'id.
GetProjectFilters	Servizio che restituisce la lista dei filtri disponibili.
GetProjectFolders	Servizio che restituisce la lista dei sottofascicoli di un fascicolo
GetProjectsByDocument	Servizio per il reperimento dei fascicoli in cui è fascicolato un documento.
GetProjectStateDiagram	Servizio per il reperimento dello stato di un fascicolo.
GetTemplatePrj	Servizio per il reperimento del dettaglio di una tipologia di fascicolo dato il nome o l'id.
GetTemplatesProjects	Servizio per il reperimento di tutte le tipologie di fascicolo.
GetTemplatePrj	Servizio per il reperimento del dettaglio di una tipologia di fascicolo dato il nome o l'id.
OpenCloseProject	Servizio per chiudere/aprire un fascicolo
SearchProjects	Servizio per la ricerca di fascicoli.

3.5.7 SERVIZI PER LA GESTIONE DELLA RUBRICA

<i>Servizio</i>	<i>Descrizione</i>
AddCorrespondent	Servizio per la creazione di un nuovo corrispondente..
DeleteCorrespondent	Servizio per la cancellazione logica di un corrispondente esterno.
EditCorrespondent	Servizio per la modifica di un corrispondente.
GetCorrespondent	Servizio per il reperimento del dettaglio di un corrispondente dato il sistem_id
GetCorrespondentFilters	Metodo che restituisce la lista dei filtri applicabili nella ricerca dei corrispondenti.
GetUserFilters	Metodo che restituisce la lista dei filtri applicabili nella ricerca dei utenti.
SearchCorrespondents	Servizio per la ricerca di corrispondenti.
SearchUsers	Servizio per la ricerca degli utenti.

3.5.8 SERVIZI PER LA GESTIONE DEI TOKEN DI AUTENTICAZIONE

<i>Servizio</i>	<i>Descrizione</i>
GetAuthenticationToken	Servizio per il reperimento di un token di autenticazione per l'utente in una amministrazione.
GetToken	Servizio per il reperimento di un token di autenticazione e della durata dello stesso per l'utente di una amministrazione e per un utente di sistema esterno.

3.5.9 SERVIZI PER LA GESTIONE DELLE TRASMISSIONI

<i>Servizio</i>	<i>Descrizione</i>
ExecuteTransmDocModel	Servizio per la trasmissione di un documento tramite un modello di trasmissione.
ExecuteTransmissionDocument	Servizio per la trasmissione singola di un documento senza l'utilizzo di un modello di trasmissione.
ExecuteTransmissionProject	Servizio per la trasmissione singola di un fascicolo senza l'utilizzo di un modello di trasmissione.
ExecuteTransmPrjModel	Servizio per la trasmissione di un fascicolo. tramite un modello di trasmissione.
GetTransmissionModel	Servizio per il reperimento del dettaglio di un modello di trasmissione data la descrizione o il codice o l'id.
GetTransmissionModels	Servizio per il reperimento di tutti i modelli di trasmissione per documenti o fascicoli.
GiveUpRights	Servizio per la rinuncia del diritto di lettura e/o scrittura su un oggetto da parte di un utente di sistema esterno.

3.5.10 SERVIZI PER LA GESTIONE DEL LIBROFIRMA

<i>Servizio</i>	<i>Descrizione</i>
-----------------	--------------------

GetSignatureProcess	Servizio che restituisce la lista dei processi o dei modelli di processo visibili e quindi utilizzabili dal ruolo.
StartSignatureProcess	Servizio per inserire un documento in un processo di Libro Firma, avviando il processo stesso.
SearchSignProcessInstances	Servizio per ricercare e quindi monitorare lo stato delle istanze di processi di firma visibili ad un determinato ruolo.
GetInstanceSearchFilters	Servizio per ottenere la lista dei filtri utilizzabile nel metodo SearchSignProcessInstances .
GetSignatureProcess	Servizio per ottenere un processo a partire dal suo Id.
GetSignProcessInstance	Servizio per ottenere una istanza del processo a partire dal suo Id istanza.
InterruptSignatureProcess	Servizio per interrompere una istanza del processo a partire dal suo Id istanza.

3.6 SERVIZI PER LA GESTIONE DEI RUOLI

3.6.1 SERVIZIO “GETROLE”

Servizio per il reperimento dei dettagli di un ruolo dato il codice del ruolo o l'id. Inserire soltanto uno di questi due valori.

3.6.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “ GetRoleRequest”. Un parametro da CodeRole e IdRole è obbligatorio.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.

CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CodeRole	String	Opzionale. Codice del ruolo del quale si vuole vedere il dettaglio.
IdRole	String	Opzionale. Id del ruolo del quale si vuole vedere il dettaglio.

3.6.1.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "GetRoleResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Role	Role	Dettaglio del Ruolo del quale si desidera visualizzare il dettaglio.

In caso di errore, "GetRoleResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_ID_ROLE	Inserire il codice o l'id di un ruolo.
REQUIRED_ONLY_CODE_OR_ID_ROLE	Inserire soltanto il codice o il ruolo, ma non entrambi.
ROLE_NO_EXIST	Il ruolo non esiste.
APPLICATION_ERROR	Errore generico.

3.6.1.3 Esempio di utilizzo C#

```
Roles.RolesClient services = new Roles.RolesClient();
Roles.GetRoleRequest request = new Roles.GetRoleRequest();

request.IdRole = "Id del ruolo";
//Oppure inserire request.CodeRole = "Codice del ruolo";
```

```
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
Roles.GetRoleResponse response = services.GetRole(request);
Roles.Role result = response.Role;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.6.1.4 Esempio di utilizzo Java

```
RolesStub wws = new RolesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetRoleRequest getRoleRequest = GetRoleRequest.Factory.newInstance();
getRoleRequest.setCodeAdm("CodiceAmministrazione");
getRoleRequest.setUserName("UserName");
getRoleRequest.setCodeRole("CodiceRuolo");
//getRoleRequest.setIdRole("IdRuolo"); Opzionale al posto del Codice del ruolo
GetRoleDocument getRoleDocument = GetRoleDocument.Factory.newInstance();
GetRole getRole = GetRole.Factory.newInstance();
getRole.setRequest(getRoleRequest);
getRoleDocument.setGetRole(getRole);
GetRoleResponseDocument getRoleResponseDocument =
GetRoleResponseDocument.Factory.newInstance();
try
{
getRoleResponseDocument = wws.getRole(getRoleDocument);
Role role =
getRoleResponseDocument.getGetRoleResponse().getGetRoleResult().getRole();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.6.2 SERVIZIO “GETROLES”

Servizio per il reperimento di tutti i ruoli disponibili ad un utente.

3.6.2.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetRolesRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
-------------	------------------	--------------------

UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
UserID	String	Obbligatorio. User name dell'utente del quale si desiderano avere i ruoli.

3.6.2.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "GetRolesResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Roles	Role[]	Lista dei ruoli dell'utente.

In caso di errore, "GetRolesResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_USERID	UserID obbligatoria.
USER_NO_ROLES	L'utente non ha ruoli.
APPLICATION_ERROR	Errore generico.

3.6.2.3 Esempio di utilizzo C#

```
Roles.RolesClient services = new Roles.RolesClient();  
Roles.GetRolesRequest request = new Roles.GetRolesRequest();  
  
request.UserID = "UserId dell'utente";  
  
request.UserName = "UserName";
```



```
request.CodeAdm = "Codice amministrazione";
```

```
try
{
Roles.GetRolesResponse response = services.GetRoles(request);
Roles.Role[] result = response.Roles;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.6.2.4 Esempio di utilizzo Java

```
RolesStub wws = new RolesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetRolesRequest getRolesRequest = GetRolesRequest.Factory.newInstance();
getRolesRequest.setCodeAdm("CodiceAmministrazione");
getRolesRequest.setUserName("UserName");
getRolesRequest.setUserID("UserID");
GetRolesDocument getRolesDocument = GetRolesDocument.Factory.newInstance();
GetRoles getRoles = GetRoles.Factory.newInstance();
getRoles.setRequest(getRolesRequest);
getRolesDocument.setGetRoles(getRoles);
GetRolesResponseDocument getRolesResponseDocument =
GetRolesResponseDocument.Factory.newInstance();
try
{
getRolesResponseDocument = wws.getRoles(getRolesDocument);
Role[] roles= null;
roles =
getRolesResponseDocument.getGetRolesResponse().getGetRolesResult().getRoles().getRol
eArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.6.3 SERVIZIO “GETROLESFORENABLEDACTIONS”

Servizio per il reperimento dei ruoli dati un utente e una funzione.

3.6.3.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetRolesForEnabledActionsRequest”.

Struttura dell’oggetto:

Nome	Tipo Dato	Descrizione
------	-----------	-------------

UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
UserID	String	Obbligatorio. UserID dell'utente di cui si vogliono ottenere i ruoli.
CodiceTipoFunzione	String	Obbligatorio. Codice del tipo funzione con il quale filtrare i ruoli

3.6.3.2 Descrizione oggetto "Response"

L'oggetto restituito dal servizio è il "GetRolesForEnabledActionsResponse".

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Roles	Role[]	Lista degli ruoli dell'utente abilitati alla funzione inserita nella richiesta.

In caso di errore, "GetRolesForEnabledActionsResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
USER_NO_ROLES	L'utente non ha ruoli.
REQUIRED_USERID	UserID obbligatoria.
REQUIRED_CODTIPOFUNC	Codice tipo funzione obbligatorio.
APPLICATION_ERROR	Errore generico.

3.6.3.3 Esempio di utilizzo C#

```
Roles.RolesClient services = new Roles.RolesClient();
Roles.GetRolesForEnabledActionsRequest request = new
Roles.GetRolesForEnabledActionsRequest();
```

```
request.UserID = "UserId dell'utente";
request.CodiceTipoFunzione = "Codice della funzione da cercare";
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
```

```
try
{
    Roles.GetRolesForEnabledActionsResponse response =
    services.GetRolesForEnabledActions(request);
    Roles.Role[] result = response.Roles;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.6.3.4 Esempio di utilizzo Java

```
RolesStub wws = new RolesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetRolesForEnabledActionsRequest getRolesForEnabledActionsRequest =
GetRolesForEnabledActionsRequest.Factory.newInstance();
getRolesForEnabledActionsRequest.setCodeAdm("CodiceAmministrazione");
getRolesForEnabledActionsRequest.setUserName("UserName");
getRolesForEnabledActionsRequest.setUserID("UserID");
getRolesForEnabledActionsRequest.setCodiceTipoFunzione("CodiceTipoFunzione");

GetRolesForEnabledActionsDocument getRolesForEnabledActionsDocument =
GetRolesForEnabledActionsDocument.Factory.newInstance();
GetRolesForEnabledActions getRolesForEnabledActions =
GetRolesForEnabledActions.Factory.newInstance();
getRolesForEnabledActions.setRequest(getRolesForEnabledActionsRequest);
getRolesForEnabledActionsDocument.setGetRolesForEnabledActions(getRolesForEnabledAct
ions);
GetRolesResponseForEnabledActionsDocument getRolesResponseForEnabledActionsDocument
= GetRolesResponseForEnabledActionsDocument.Factory.newInstance();
try
{
    getRolesForEnabledActionsResponseDocument = wws.getRolesForEnabledActions
(getRolesForEnabledActionsDocument);
    Role[] roles= null;
    roles =
getRolesForEnabledActionsResponseDocument.getGetRolesResponse().getGetRolesResult().
getRoles().getRoleArray();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.6.4 SERVIZIO “GETUSERSINROLE”

Servizio per il reperimento degli utenti in un ruolo.

3.6.4.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetUsersInRoleRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CodeRole	String	Obbligatorio. Codice del ruolo del quale si desiderano gli utenti appartenenti.

3.6.4.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetUsersInRoleResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Users	User[]	Lista degli utenti del ruolo.

In caso di errore, “GetUsersInRoleResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_ROLE	Codice del ruolo obbligatorio.

ROLE_NO_EXIST	Il ruolo non esiste.
APPLICATION_ERROR	Errore generico.

3.6.4.3 Esempio di utilizzo C#

```
Roles.RolesClient services = new Roles.RolesClient();
Roles.GetUsersInRoleRequest request = new Roles.GetUsersInRoleRequest();

request.CodeRole = "Codice del ruolo";

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
    Roles.GetUsersInRoleResponse response = services.GetUsersInRole(request);
    Roles.User[] result = response.Users;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.6.4.4 Esempio di utilizzo Java

```
RolesStub wws = new RolesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetUsersInRoleRequest getUsersInRoleRequest =
GetUsersInRoleRequest.Factory.newInstance();
getUsersInRoleRequest.setCodeAdm("CodiceAmministrazione");
getUsersInRoleRequest.setUserName("UserName");
getUsersInRoleRequest.setCodeRole("CodiceDelRuolo");
GetUsersInRoleDocument getUsersInRoleDocument =
GetUsersInRoleDocument.Factory.newInstance();
GetUsersInRole getUsersInRole = GetUsersInRole.Factory.newInstance();
getUsersInRole.setRequest(getUsersInRoleRequest);
getUsersInRoleDocument.setGetUsersInRole(getUsersInRole);
GetUsersInRoleResponseDocument getUsersInRoleResponseDocument =
GetUsersInRoleResponseDocument.Factory.newInstance();
try
{
    getUsersInRoleResponseDocument = wws.getUsersInRole(getUsersInRoleDocument);
    User[] users = null;
    users =
getUsersInRoleResponseDocument.getGetUsersInRoleResponse().getGetUsersInRoleResult().
getUsers().getUserArray();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.7 SERVIZI PER LA GESTIONE DEI REGISTRI

3.7.1 SERVIZIO “GETREGISTERORRF”

Servizio per il reperimento del dettaglio di un registro/RF dato il codice del registro/RF o l'id. Obbligatorio a scelta l'id o il codice di un registro/RF.

3.7.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetRegisterOrRFRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CodeRegister	String	Opzionale. Codice del registro/RF.
IdRegister	String	Opzionale. Id del registro/RF.

3.7.1.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetRegisterOrRFResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Register	Register	Dettaglio del registro/RF.

In caso di errore, “GetRegisterOrRFResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
---------------	--------------------

MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_ID_REGISTER	Obbligatorio il codice del registro o l'id.
REQUIRED_ONLY_CODE_OR_ID_REGISTER	Inserire uno tra codice o id del registro.
APPLICATION_ERROR	Errore generico.

3.7.1.3 Esempio di utilizzo C#

```
Registers.RegistersClient services = new Registers.RegistersClient();
Registers.GetRegisterOrRFRequest request = new Registers.GetRegisterOrRFRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdRegister = "Id del registro";
//Oppure request.CodeRegister = "Codice del registro";

try
{
Registers.GetRegisterOrRFResponse response = services.GetRegisterOrRF(request);
Registers.Register result = response.Register;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.7.1.4 Esempio di utilizzo Java

```
RegistersStub wws = new RegistersStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetRegisterOrRFRequest getRegisterOrRFRequest =
GetRegisterOrRFRequest.Factory.newInstance();
getRegisterOrRFRequest.setCodeAdm("CodiceAmministrazione");
getRegisterOrRFRequest.setUserName("UserName");
getRegisterOrRFRequest.setCodeRegister("CodiceRegistroORF");
//getRegisterOrRFRequest.setIdRegister("IdDelRegistro"); Opzionale al posto del
//codice
GetRegisterOrRFDocument getRegisterOrRFDocument =
GetRegisterOrRFDocument.Factory.newInstance();
GetRegisterOrRF getRegisterOrRF = GetRegisterOrRF.Factory.newInstance();
getRegisterOrRF.setRequest(getRegisterOrRFRequest);
getRegisterOrRFDocument.setGetRegisterOrRF(getRegisterOrRF);
GetRegisterOrRFResponseDocument getRegisterOrRFResponseDocument =
GetRegisterOrRFResponseDocument.Factory.newInstance();
try
{
getRegisterOrRFResponseDocument = wws.getRegisterOrRF(getRegisterOrRFDocument);
Register register = null;
```

```
register =
getRegisterOrRFResponseDocument.getGetRegisterOrRFResponse().getGetRegisterOrRFResult().getRegister();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.7.2 SERVIZIO “GETREGISTERSORRF”

Servizio per il reperimento di tutti gli RF o i registri disponibili per un ruolo.

3.7.2.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetRegistersOrRFRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CodeRole	String	Opzionale. Codice del ruolo.
IdRole	String	Opzionale. Id del ruolo.
OnlyRegister	Boolean	Opzionale. Indica se reperire solo i registri.
OnlyRF	Boolean	Opzionale. Indica se reperire solo gli RF.

3.7.2.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetRegistersOrRFResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Registers	Register[]	Registri associati al ruolo.

In caso di errore, “GetRegistersOrRFResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_ID_ROLE	Richiesto il codice o l'id di un ruolo.
REQUIRED_ONLY_CODE_OR_ID_ROLE	Richiesto soltanto il codice o l'id di un ruolo.
APPLICATION_ERROR	Errore generico.

3.7.2.3 Esempio di utilizzo C#

```
Registers.RegistersClient services = new Registers.RegistersClient();
Registers.GetRegistersOrRFRequest request = new Registers.GetRegistersOrRFRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdRole = "Id del ruolo";
//Oppure request.CodeRole = "Codice del ruolo";

//Opzionale
request.OnlyRegisters = true; //Se vogliono anche i registri a true
request.OnlyRF = false; //Se si vogliono anche gli RF a true

try
{
Registers.GetRegistersOrRFResponse response = services.GetRegistersOrRF(request);
Registers.Register[] result = response.Registers;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.7.2.4 Esempio di utilizzo Java

```
RegistersStub wws = new RegistersStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
```

```
GetRegistersOrRFRequest getRegistersOrRFRequest =
GetRegistersOrRFRequest.Factory.newInstance();
getRegistersOrRFRequest.setCodeAdm("CodiceAmministrazione");
getRegistersOrRFRequest.setUserName("UserName");
getRegistersOrRFRequest.setCodeRole("CodiceDelRuolo");
//getRegistersOrRFRequest.setIdRole("IdDelRuolo"); Opzionale al posto del codice

GetRegistersOrRFDocument getRegistersOrRFDocument =
GetRegistersOrRFDocument.Factory.newInstance();
GetRegistersOrRF getRegistersOrRF = GetRegistersOrRF.Factory.newInstance();
getRegistersOrRF.setRequest(getRegistersOrRFRequest);
getRegistersOrRFDocument.setGetRegistersOrRF(getRegistersOrRF);
GetRegistersOrRFResponseDocument getRegistersOrRFResponseDocument =
GetRegistersOrRFResponseDocument.Factory.newInstance();
try
{
getRegistersOrRFResponseDocument = wws.getRegistersOrRF(getRegistersOrRFDocument);
Register[] register = null;
register =
getRegistersOrRFResponseDocument.getGetRegistersOrRFResponse().getGetRegistersOrRFRe
sult().getRegisters().getRegisterArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.8 SERVIZI PER LA GESTIONE DEI TITOLARI

3.8.1 SERVIZIO “GETALLCLASSIFICATIONSCHEMES”

Servizio per il reperimento di tutti i titolari.

3.8.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetAllClassificationSchemesRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.8.1.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetAllClassificationSchemesResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ClassificationSchemes	ClassificationScheme[]	Tutti i titolari disponibili.

In caso di errore, “GetAllClassificationSchemes” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
CLASSIFICATIONSCHEMES_NOT_FOUND	Titolari non presenti.
APPLICATION_ERROR	Errore generico.

3.8.1.3 Esempio di utilizzo C#

```
ClassificationSchemes.ClassificationSchemesClient services = new
ClassificationSchemes.ClassificationSchemesClient();
ClassificationSchemes.GetAllClassificationSchemesRequest request = new
ClassificationSchemes.GetAllClassificationSchemesRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
ClassificationSchemes.GetAllClassificationSchemesResponse response =
services.GetAllClassificationSchemes(request);
ClassificationSchemes.ClassificationScheme[] result = response.ClassificationSchemes;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.8.1.4 Esempio di utilizzo Java

```
ClassificationSchemesStub wws = new ClassificationSchemesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
```

```

GetAllClassificationSchemesRequest getAllClassificationSchemesRequest =
GetAllClassificationSchemesRequest.Factory.newInstance();
getAllClassificationSchemesRequest.setCodeAdm("CodiceAmministrazione");
getAllClassificationSchemesRequest.setUserName("UserName");
GetAllClassificationSchemesDocument getAllClassificationSchemesDocument =
GetAllClassificationSchemesDocument.Factory.newInstance();
GetAllClassificationSchemes getAllClassificationSchemes =
GetAllClassificationSchemes.Factory.newInstance();
getAllClassificationSchemes.setRequest(getAllClassificationSchemesRequest);
getAllClassificationSchemesDocument.setGetAllClassificationSchemes(getAllClassificationSchemes);
GetAllClassificationSchemesResponseDocument
getAllClassificationSchemesResponseDocument =
GetAllClassificationSchemesResponseDocument.Factory.newInstance();
try
{
getAllClassificationSchemesResponseDocument =
wvs.getAllClassificationSchemes(getAllClassificationSchemesDocument);
ClassificationScheme[] schemes = null;
schemes =
getAllClassificationSchemesResponseDocument.getGetAllClassificationSchemesResponse()
.getGetAllClassificationSchemesResult().getClassificationSchemes().getClassification
SchemeArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.8.2 SERVIZIO “GETACTIVECLASSIFICATIONSCHEME”

Servizio per il reperimento del titolare attivo.

3.8.2.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetActiveClassificationSchemeRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.8.2.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetActiveClassificationSchemeResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ClassificationScheme	ClassificationScheme	Titolario attivo

In caso di errore, “GetActiveClassificationSchemeResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
CLASSIFICATION_NOT_FOUND	Titolario non trovato.
APPLICATION_ERROR	Errore generico.

3.8.2.3 Esempio di utilizzo C#

```
ClassificationSchemes.ClassificationSchemesClient services = new
ClassificationSchemes.ClassificationSchemesClient();
ClassificationSchemes.GetActiveClassificationSchemeRequest request = new
ClassificationSchemes.GetActiveClassificationSchemeRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
    ClassificationSchemes.GetActiveClassificationSchemeResponse response =
    services.GetActiveClassificationScheme(request);
    ClassificationSchemes.ClassificationScheme result = response.ClassificationScheme;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.8.2.4 Esempio di utilizzo Java

```
ClassificationSchemesStub wws = new ClassificationSchemesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
```

```
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE_NAMESPACE_URI);
GetActiveClassificationSchemeRequest getActiveClassificationSchemeRequest =
GetActiveClassificationSchemeRequest.Factory.newInstance();
getActiveClassificationSchemeRequest.setCodeAdm("CodiceAmministrazione");
getActiveClassificationSchemeRequest.setUsername("UserName");
GetActiveClassificationSchemeDocument getActiveClassificationSchemeDocument =
GetActiveClassificationSchemeDocument.Factory.newInstance();
GetActiveClassificationScheme getActiveClassificationScheme =
GetActiveClassificationScheme.Factory.newInstance();
getActiveClassificationScheme.setRequest(getActiveClassificationSchemeRequest);
getActiveClassificationSchemeDocument.setGetActiveClassificationScheme(getActiveClassificationScheme);
GetActiveClassificationSchemeResponseDocument
getActiveClassificationSchemeResponseDocument =
GetActiveClassificationSchemeResponseDocument.Factory.newInstance();
try
{
getActiveClassificationSchemeResponseDocument =
wws.getActiveClassificationScheme(getActiveClassificationSchemeDocument);
ClassificationScheme schemes = null;
schemes =
getActiveClassificationSchemeResponseDocument.getGetActiveClassificationSchemeResponse().getGetActiveClassificationSchemeResult().getClassificationScheme();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.8.3 SERVIZIO “GETCLASSIFICATIONSCHEMEBYID”

Servizio per il reperimento dei dettagli di un titolare dato l'id.

3.8.3.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetClassificationSchemeByIdRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdClassificationScheme	String	Obbligatorio. Id del titolare di classificazione.

3.8.3.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetClassificationSchemeByIdResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ClassificationSchemes	ClassificationScheme	Dettaglio del titolare di interesse.

In caso di errore, “GetClassificationSchemeByIdResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
CLASSIFICATION_NOT_FOUND	Titolario non trovato.
APPLICATION_ERROR	Errore generico.

3.8.3.3 Esempio di utilizzo C#

```
ClassificationSchemes.ClassificationSchemesClient services = new
ClassificationSchemes.ClassificationSchemesClient();
ClassificationSchemes.GetClassificationSchemeByIdRequest request = new
ClassificationSchemes.GetClassificationSchemeByIdRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdClassificationScheme = "Id del titolare";

try
{
ClassificationSchemes.GetClassificationSchemeByIdResponse response =
services.GetClassificationSchemeById(request);
ClassificationSchemes.ClassificationScheme result = response.ClassificationScheme;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.8.3.4 Esempio di utilizzo Java

```
ClassificationSchemesStub wws = new ClassificationSchemesStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetClassificationSchemeByIdRequest getClassificationSchemeByIdRequest =
GetClassificationSchemeByIdRequest.Factory.newInstance();
getClassificationSchemeByIdRequest.setCodeAdm("CodiceAmministrazione");
getClassificationSchemeByIdRequest.setUserName("UserName");
getClassificationSchemeByIdRequest.setIdClassificationScheme("IdTitolario");
GetClassificationSchemeByIdDocument getClassificationSchemeByIdDocument =
GetClassificationSchemeByIdDocument.Factory.newInstance();
GetClassificationSchemeById getClassificationSchemeById =
GetClassificationSchemeById.Factory.newInstance();
getClassificationSchemeById.setRequest(getClassificationSchemeByIdRequest);
getClassificationSchemeByIdDocument.setGetClassificationSchemeById(getClassification
SchemeById);
GetClassificationSchemeByIdResponseDocument
getClassificationSchemeByIdResponseDocument =
GetClassificationSchemeByIdResponseDocument.Factory.newInstance();
try
{
getClassificationSchemeByIdResponseDocument =
wws.getClassificationSchemeById(getClassificationSchemeByIdDocument);
ClassificationScheme schemes = null;
schemes =
getClassificationSchemeByIdResponseDocument.getGetClassificationSchemeByIdResponse().
getGetClassificationSchemeByIdResult().getClassificationScheme();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9 SERVIZI PER LA GESTIONE DEI DOCUMENTI

3.9.1 SERVIZIO “CREATEDOCUMENT”

Il servizio permette la creazione di un documento non protocollato, protocollato in arrivo, in uscita, interno oppure un predisposto in arrivo, in uscita oppure interno.

3.9.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “CreateDocumentRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente

		censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Document	Document	Obbligatorio. Documento che deve essere creato. Vedere le tabelle successive per le varie tipologie di documento.
CodeRegister	String	Opzionale. Codice del registro, obbligatorio nel caso di documenti protocollati.
CodeRF	String	Opzionale. Codice dell'RF per la segnatura.

3.9.1.2 Creare un documento non protocollato

Struttura dell'oggetto Document nel caso di documento non protocollato.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
DocumentType	String	Obbligatorio. Indica la tipologia del documento. Inserire il valore "G".
Object	String	Obbligatorio. Oggetto del documento.
Note	Note[]	Opzionale. Possibilità di inserire una o più note nel documento.
Template	Template	Opzionale. Tipologia del documento che si vuole associare al documento con i relativi valori.
MainDocument	File	Opzionale. Documento principale.

3.9.1.2.1 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
```

```
Documents.CreateDocumentRequest request = new Documents.CreateDocumentRequest();
Documents.Document docRequest = new Documents.Document();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

docRequest.Object = "Oggetto";
docRequest.PrivateDocument = false; //Se privato a true
//Opzionale Nota
docRequest.Note = new Documents.Note[1];
docRequest.Note[0] = new Documents.Note();
docRequest.Note[0].Description = "Testo nota";
docRequest.DocumentType = "G";

request.Document = docRequest;

//Opzionale. Tipologia di documento
Documents.Template template = new Documents.Template();
template.Id = "Id del template";
template.Name = "Nome del template";

List<Documents.Field> campiProfilo = new List<Documents.Field>();

//Campo profilato della tipologia di document che si vuole valorizzare
Documents.Field campo = new Documents.Field();
campo.Name = "Nome del campo profilato";
campo.Value = "Valore del campo profilato";
campiProfilo.Add(campo);

request.Document.Template = template;
request.Document.Template.Fields = campiProfilo.ToArray();

//Allego il documento principale
byte[] content = new byte[this.uploadFile.PostedFile.InputStream.Length];
this.uploadFile.PostedFile.InputStream.Read(content, 0, content.Length);
request.Document.MainDocument = new Documents.File();
request.Document.MainDocument.Content = content;
request.Document.MainDocument.Name = this.uploadFile.PostedFile.FileName;
request.Document.MainDocument.MimeType = this.uploadFile.PostedFile.ContentType;
try
{
    Documents.CreateDocumentResponse response = services.CreateDocument(request);
    Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.1.2.2 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
```

```

wvs._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
CreateDocumentRequest createDocumentRequest =
CreateDocumentRequest.Factory.newInstance();
createDocumentRequest.setCodeAdm("CodiceAmministrazione");
createDocumentRequest.setUserName("UserName");
//Creazione documento
Document document = Document.Factory.newInstance();
document.setObject("Oggetto");
document.setDocumentType("G");
ArrayOfNote note = ArrayOfNote.Factory.newInstance();
Note nota = Note.Factory.newInstance();
nota.setDescription("Testo Nota");
Note[] arrayNota = new Note[1];
arrayNota[0] = nota;
note.setNoteArray(arrayNota);
document.setNote(note);
//Associo tipologia documento
Template template = Template.Factory.newInstance();
template.setName("Nome tipologia Template");
template.setId("IdTemplate");
ArrayOfField fields = ArrayOfField.Factory.newInstance();
Field[] fieldArray = new Field[2];
Field temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
temp.setValue("Valore");
fieldArray[0] = temp;
temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
temp.setValue("Valore");
fieldArray[1] = temp;
temp = Field.Factory.newInstance();
fields.setFieldArray(fieldArray);
template.setFields(fields);
document.setTemplate(template);
//Inserisco File
File file = new File("PercorsoFile");
byte fileContent[] = new byte[(int)file.length()];
FileInputStream fin = new FileInputStream(file);
int read = 0;
ByteArrayOutputStream ous = new ByteArrayOutputStream();
while((read = fin.read(fileContent)) != -1){
ous.write(fileContent, 0, read); }
File mainDocument = File.Factory.newInstance();
mainDocument.setContent(fileContent);
mainDocument.setName(file.getName());
mainDocument.setMimeType(new MimeTypeFileMap().getContentType(file));
document.setMainDocument(mainDocument);
//Fine creazione documento
createDocumentRequest.setDocument(document);
CreateDocumentDocument1 createDocumentDocument =
CreateDocumentDocument1.Factory.newInstance();
CreateDocument createDocument = CreateDocument.Factory.newInstance();
createDocument.setRequest(createDocumentRequest);
createDocumentDocument.setCreateDocument(createDocument);
CreateDocumentResponseDocument createDocumentResponseDocument =
CreateDocumentResponseDocument.Factory.newInstance();
try
{
createDocumentResponseDocument = wvs.createDocument(createDocumentDocument);
Document respdocument = null;
respdocument =
createDocumentResponseDocument.getCreateDocumentResponse().getCreateDocumentResult()
.getDocument();
}

```

```
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.9.1.3 Creare un documento protocollato in entrata

Struttura dell'oggetto Document nel caso di un protocollo in entrata.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
DocumentType	String	Obbligatorio. Indica la tipologia del documento. Inserire il valore "A".
Object	String	Obbligatorio. Oggetto del documento.
Note	Note[]	Opzionale. Possibilità di inserire una o più note nel documento.
Template	Template	Opzionale. Tipologia del documento che si vuole associare al documento con i relativi valori.
Sender	Correspondent	Obbligatorio. Indica il mittente del protocollo.
MultipleSenders	Correspondent[]	Opzionale. Lista di mittenti aggiuntivi.
MeansOfSending	String	Opzionale. Mezzo di spedizione.
ProtocolSender	String	Opzionale. Protocollo mittente.
ArrivalDate	String	Opzionale. Data di arrivo.
ArrivalTime	String	Opzionale. Ora di arrivo.
DataProtocolSender	String	Opzionale. Data protocollo mittente.
MainDocument	File	Opzionale. Documento principale.

3.9.1.3.1 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.CreateDocumentRequest request = new Documents.CreateDocumentRequest();
Documents.Document docRequest = new Documents.Document();
```

```
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

docRequest.Object = "Oggetto";
docRequest.PrivateDocument = false; //Se privato a true
//Opzionale Nota
docRequest.Note = new Documents.Note[1];
docRequest.Note[0] = new Documents.Note();
docRequest.Note[0].Description = "Testo nota";
docRequest.DocumentType = "A";

docRequest.Predisposed = false; //Nel caso di predisposto con valore true
//Creo un mittente occasionale
docRequest.Sender = new Documents.Correspondent();
docRequest.Sender.Description = "Nome del mittente";
docRequest.Sender.CorrespondentType = "0";
//Codice nel registro nel quale sarà protocollato il documento
request.CodeRegister = "Codice del registro";
//Opzionale. Inserisco un mittente multiplo occasionale
docRequest.MultipleSenders = new Documents.Correspondent[1];
docRequest.MultipleSenders[0] = new Documents.Correspondent();
docRequest.MultipleSenders[0].Description = "Descrizione mittente";
docRequest.MultipleSenders[0].CorrespondentType = "0";

request.Document = docRequest;

//Opzionale. Tipologia di documento
Documents.Template template = new Documents.Template();
template.Id = "Id del template";
template.Name = "Nome del template";

List<Documents.Field> campiProfilo = new List<Documents.Field>();

//Campo profilato della tipologia di document che si vuole valorizzare
Documents.Field campo = new Documents.Field();
campo.Name = "Nome del campo profilato";
campo.Value = "Valore del campo profilato";
campiProfilo.Add(campo);

request.Document.Template = template;
request.Document.Template.Fields = campiProfilo.ToArray();

//Allego il documento principale
byte[] content = new byte[this.uploadFile.PostedFile.InputStream.Length];
this.uploadFile.PostedFile.InputStream.Read(content, 0, content.Length);
request.Document.MainDocument = new Documents.File();
request.Document.MainDocument.Content = content;
request.Document.MainDocument.Name = this.uploadFile.PostedFile.FileName;
request.Document.MainDocument.MimeType = this.uploadFile.PostedFile.ContentType;

try
{
    Documents.CreateDocumentResponse response = services.CreateDocument(request);
    Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
}
```

```
//e.Code;  
//e.Message;  
}
```

3.9.1.3.2 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");  
//Timeout  
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);  
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V  
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);  
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE  
_NAMESPACE_URI);  
CreateDocumentRequest createDocumentRequest =  
CreateDocumentRequest.Factory.newInstance();  
createDocumentRequest.setCodeAdm("CodiceAmministrazione");  
createDocumentRequest.setUsername("UserName");  
//Codice nel registro nel quale sarà protocollato il documento  
createDocumentRequest.setCodeRegister("CodiceRegistro");  
//Creazione documento  
Document document = Document.Factory.newInstance();  
document.setObject("Oggetto");  
document.setDocumentType("A");  
ArrayOfNote note = ArrayOfNote.Factory.newInstance();  
Note nota = Note.Factory.newInstance();  
nota.setDescription("Testo Nota");  
Note[] arrayNota = new Note[1];  
arrayNota[0] = nota;  
note.setNoteArray(arrayNota);  
document.setNote(note);  
//Mittente occasionale  
Correspondent sender = Correspondent.Factory.newInstance();  
sender.setDescription("Nome del corrispondente");  
sender.setCorrespondentType("O");  
document.setSender(sender);  
//Aggiungo un mittente multiplo  
Correspondent multiple = Correspondent.Factory.newInstance();  
multiple.setId("93204"); //Id del mittente censito nel sistema  
ArrayOfCorrespondent multipleSenders = ArrayOfCorrespondent.Factory.newInstance();  
Correspondent[] correspondentArray = new Correspondent[1];  
correspondentArray[0] = multiple;  
multipleSenders.setCorrespondentArray(correspondentArray);  
document.setMultipleSenders(multipleSenders);  
//Associa tipologia documento  
Template template = Template.Factory.newInstance();  
template.setName("Nome tipologia Template");  
template.setId("IdTemplate");  
ArrayOfField fields = ArrayOfField.Factory.newInstance();  
Field[] fieldArray = new Field[2];  
Field temp = Field.Factory.newInstance();  
temp.setName("NomeCampoProfilato");  
temp.setValue("Valore");  
fieldArray[0] = temp;  
temp = Field.Factory.newInstance();  
temp.setName("NomeCampoProfilato");  
temp.setValue("Valore");  
fieldArray[1] = temp;  
temp = Field.Factory.newInstance();  
fields.setFieldArray(fieldArray);  
template.setFields(fields);  
document.setTemplate(template);  
//Inserisco File
```

```
File file = new File("PercorsoFile");
byte fileContent[] = new byte[(int)file.length()];
FileInputStream fin = new FileInputStream(file);
int read = 0;
ByteArrayOutputStream ous = new ByteArrayOutputStream();
while((read = fin.read(fileContent)) != -1){
    ous.write(fileContent, 0, read);
}
File mainDocument = File.Factory.newInstance();
mainDocument.setContent(fileContent);
mainDocument.setName(file.getName());
mainDocument.setMimeType(new MimeTypeMap().getContentType(file));
document.setMainDocument(mainDocument);
//Fine creazione documento
createDocumentRequest.setDocument(document);
CreateDocumentDocument1 createDocumentDocument =
CreateDocumentDocument1.Factory.newInstance();
CreateDocument createDocument = CreateDocument.Factory.newInstance();
createDocument.setRequest(createDocumentRequest);
createDocumentDocument.setCreateDocument(createDocument);
CreateDocumentResponseDocument createDocumentResponseDocument =
CreateDocumentResponseDocument.Factory.newInstance();
try
{
    createDocumentResponseDocument = wws.createDocument(createDocumentDocument);
    Document respdocument = null;
    respdocument =
createDocumentResponseDocument.getCreateDocumentResponse().getCreateDocumentResult()
.getDocument();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.9.1.4 Creare un documento protocollato in uscita

Struttura dell'oggetto Document nel caso di un protocollo in uscita.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
DocumentType	String	Obbligatorio. Indica la tipologia del documento. Inserire il valore "P".
Object	String	Obbligatorio. Oggetto del documento.
Note	Note[]	Opzionale. Possibilità di inserire una o più note nel documento.
Template	Template	Opzionale. Tipologia del documento che si vuole associare al documento con i relativi valori.
Sender	Correspondent	Opzionale. Indica il mittente del protocollo.

Recipients	Correspondent[]	Obbligatorio. Destinatari del protocollo, inserirne almeno uno di tipo interno o esterno.
RecipientsCC	Correspondent[]	Opzionale. Destinatari in conoscenza del protocollo di tipo interno o esterno.
MainDocument	File	Opzionale. Documento principale.

3.9.1.4.1 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.CreateDocumentRequest request = new Documents.CreateDocumentRequest();
Documents.Document docRequest = new Documents.Document();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

docRequest.Object = "Oggetto";
docRequest.PrivateDocument = false; //Se privato a true
//Opzionale Nota
docRequest.Note = new Documents.Note[1];
docRequest.Note[0] = new Documents.Note();
docRequest.Note[0].Description = "Testo nota";
docRequest.DocumentType = "P";

docRequest.Predisposed = false; //Nel caso di predisposto con valore true
//Creo un mittente occasionale
docRequest.Sender = new Documents.Correspondent();
docRequest.Sender.Description = "Nome del mittente";
docRequest.Sender.CorrespondentType = "0";
//Codice nel registro nel quale sarà protocollato il documento
request.CodeRegister = "Codice del registro";
//Inserisco un destinatario
docRequest.Recipients = new Documents.Correspondent[1];
docRequest.Recipients[0] = new Documents.Correspondent();
docRequest.Recipients[0].Id = "Id del corrispondente";
//Inserisco un destinatario in conoscenza
docRequest.RecipientsCC = new Documents.Correspondent[1];
docRequest.RecipientsCC[0] = new Documents.Correspondent();
docRequest.RecipientsCC[0].Id = "Id del corrispondente";

request.Document = docRequest;

//Opzionale. Tipologia di documento
Documents.Template template = new Documents.Template();
template.Id = "Id del template";
template.Name = "Nome del template";

List<Documents.Field> campiProfilo = new List<Documents.Field>();
```



```
//Campo profilato della tipologia di document che si vuole valorizzare
Documents.Field campo = new Documents.Field();
campo.Name = "Nome del campo profilato";
campo.Value = "Valore del campo profilato";
campiProfilo.Add(campo);

request.Document.Template = template;
request.Document.Template.Fields = campiProfilo.ToArray();

//Allego il documento principale
byte[] content = new byte[this.uploadFile.PostedFile.InputStream.Length];
this.uploadFile.PostedFile.InputStream.Read(content, 0, content.Length);
request.Document.MainDocument = new Documents.File();
request.Document.MainDocument.Content = content;
request.Document.MainDocument.Name = this.uploadFile.PostedFile.FileName;
request.Document.MainDocument.MimeType = this.uploadFile.PostedFile.ContentType;

try
{
Documents.CreateDocumentResponse response = services.CreateDocument(request);
Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.1.4.2 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
CreateDocumentRequest createDocumentRequest =
CreateDocumentRequest.Factory.newInstance();
createDocumentRequest.setCodeAdm("CodiceAmministrazione");
createDocumentRequest.setUsername("UserName");
//Codice nel registro nel quale sarà protocollato il documento
createDocumentRequest.setCodeRegister("CodiceRegistro");
//Creazione documento
Document document = Document.Factory.newInstance();
document.setObject("Oggetto");
document.setDocumentType("P");
ArrayOfNote note = ArrayOfNote.Factory.newInstance();
Note nota = Note.Factory.newInstance();
nota.setDescription("Testo Nota");
Note[] arrayNota = new Note[1];
arrayNota[0] = nota;
note.setNoteArray(arrayNota);
document.setNote(note);
//Mittente censito nel sistema
Correspondent sender = Correspondent.Factory.newInstance();
sender.setId("IdDelCorrispondente");
document.setSender(sender);
//Aggiungo un destinatario
Correspondent recipient = Correspondent.Factory.newInstance();
```

```
recipient.setId("IdDelCorrispondente");
ArrayOfCorrespondent recipients = ArrayOfCorrespondent.Factory.newInstance();
Correspondent[] correspondentArray = new Correspondent[1];
correspondentArray[0] = recipient;
recipients.setCorrespondentArray(correspondentArray);
document.setRecipients(recipients);
//Aggiungo un destinatario in conoscenza
Correspondent recipientCC = Correspondent.Factory.newInstance();
recipientCC.setId("IdDelCorrispondente");
ArrayOfCorrespondent recipientsCC = ArrayOfCorrespondent.Factory.newInstance();
Correspondent[] correspondentArrayCC = new Correspondent[1];
correspondentArrayCC[0] = recipientCC;
recipientsCC.setCorrespondentArray(correspondentArrayCC);
document.setRecipientsCC(recipientsCC);
//Associo tipologia documento
Template template = Template.Factory.newInstance();
template.setName("Nome tipologia Template");
template.setId("IdTemplate");
ArrayOfField fields = ArrayOfField.Factory.newInstance();
Field[] fieldArray = new Field[2];
Field temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
temp.setValue("Valore");
fieldArray[0] = temp;
temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
temp.setValue("Valore");
fieldArray[1] = temp;
temp = Field.Factory.newInstance();
fields.setFieldArray(fieldArray);
template.setFields(fields);
document.setTemplate(template);
//Inserisco File
File file = new File("PercorsoFile");
byte fileContent[] = new byte[(int)file.length()];
FileInputStream fin = new FileInputStream(file);
int read = 0;
ByteArrayOutputStream ous = new ByteArrayOutputStream();
while((read = fin.read(fileContent)) != -1){
    ous.write(fileContent, 0, read);
}
File mainDocument = File.Factory.newInstance();
mainDocument.setContent(fileContent);
mainDocument.setName(file.getName());
mainDocument.setMimeType(new MimeTypeFileMap().getContentType(file));
document.setMainDocument(mainDocument);
//Fine creazione documento
createDocumentRequest.setDocument(document);
CreateDocumentDocument1 createDocumentDocument =
CreateDocumentDocument1.Factory.newInstance();
CreateDocument createDocument = CreateDocument.Factory.newInstance();
createDocument.setRequest(createDocumentRequest);
createDocumentDocument.setCreateDocument(createDocument);
CreateDocumentResponseDocument createDocumentResponseDocument =
CreateDocumentResponseDocument.Factory.newInstance();

try
{
createDocumentResponseDocument = wws.createDocument(createDocumentDocument);
Document respdocument = null;
respdocument =
createDocumentResponseDocument.getCreateDocumentResponse().getCreateDocumentResult().
.getDocument();
}
catch (RemoteException e)
{
}
```

```
//e.getMessage();
//e.printStackTrace();
}
```

3.9.1.5 Creare un documento protocollato interno

Struttura dell'oggetto Document nel caso di un protocollo interno.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
DocumentType	String	Obbligatorio. Indica la tipologia del documento. Inserire il valore "T".
Object	String	Obbligatorio. Oggetto del documento.
Note	Note[]	Opzionale. Possibilità di inserire una o più note nel documento.
Template	Template	Opzionale. Tipologia del documento che si vuole associare al documento con i relativi valori.
Sender	Correspondent	Obbligatorio. Indica il mittente del protocollo.
Recipients	Correspondent[]	Obbligatorio. Destinatari del protocollo, inserirne almeno uno di tipo interno.
RecipientsCC	Correspondent[]	Opzionale. Destinatari in conoscenza del protocollo di tipo interno.
MainDocument	File	Opzionale. Documento principale.

3.9.1.5.1 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.CreateDocumentRequest request = new Documents.CreateDocumentRequest();
Documents.Document docRequest = new Documents.Document();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

docRequest.Object = "Oggetto";
docRequest.PrivateDocument = false; //Se privato a true
//Opzionale Nota
```

```
docRequest.Note = new Documents.Note[1];
docRequest.Note[0] = new Documents.Note();
docRequest.Note[0].Description = "Testo nota";
docRequest.DocumentType = "I";

docRequest.Predisposed = false; //Nel caso di predisposto con valore true
//Creo un mittente occasionale
docRequest.Sender = new Documents.Correspondent();
docRequest.Sender.Description = "Nome del mittente";
docRequest.Sender.CorrespondentType = "O";
//Codice nel registro nel quale sarà protocollato il documento
request.CodeRegister = "Codice del registro";
//Inserisco un destinatario
docRequest.Recipients = new Documents.Correspondent[1];
docRequest.Recipients[0] = new Documents.Correspondent();
docRequest.Recipients[0].Id = "Id del corrispondente";
//Inserisco un destinatario in conoscenza
docRequest.RecipientsCC = new Documents.Correspondent[1];
docRequest.RecipientsCC[0] = new Documents.Correspondent();
docRequest.RecipientsCC[0].Id = "Id del corrispondente";

request.Document = docRequest;

//Opzionale. Tipologia di documento
Documents.Template template = new Documents.Template();
template.Id = "Id del template";
template.Name = "Nome del template";

List<Documents.Field> campiProfilo = new List<Documents.Field>();

//Campo profilato della tipologia di document che si vuole valorizzare
Documents.Field campo = new Documents.Field();
campo.Name = "Nome del campo profilato";
campo.Value = "Valore del campo profilato";
campiProfilo.Add(campo);

request.Document.Template = template;
request.Document.Template.Fields = campiProfilo.ToArray();

//Allego il documento principale
byte[] content = new byte[this.uploadFile.PostedFile.InputStream.Length];
this.uploadFile.PostedFile.InputStream.Read(content, 0, content.Length);
request.Document.MainDocument = new Documents.File();
request.Document.MainDocument.Content = content;
request.Document.MainDocument.Name = this.uploadFile.PostedFile.FileName;
request.Document.MainDocument.MimeType = this.uploadFile.PostedFile.ContentType;

try
{
    Documents.CreateDocumentResponse response = services.CreateDocument(request);
    Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

}

3.9.1.5.2 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
CreateDocumentRequest createDocumentRequest =
CreateDocumentRequest.Factory.newInstance();
createDocumentRequest.setCodeAdm("CodiceAmministrazione");
createDocumentRequest.setUsername("UserName");
//Codice nel registro nel quale sarà protocollato il documento
createDocumentRequest.setCodeRegister("CodiceRegistro");
//Creazione documento
Document document = Document.Factory.newInstance();
document.setObject("Oggetto");
document.setDocumentType("I");
ArrayOfNote note = ArrayOfNote.Factory.newInstance();
Note nota = Note.Factory.newInstance();
nota.setDescription("Testo Nota");
Note[] arrayNota = new Note[1];
arrayNota[0] = nota;
note.setNoteArray(arrayNota);
document.setNote(note);
//Mittente censito nel sistema
Correspondent sender = Correspondent.Factory.newInstance();
sender.setId("IdDelCorrispondente");
document.setSender(sender);
//Aggiungo un destinatario
Correspondent recipient = Correspondent.Factory.newInstance();
recipient.setId("IdDelCorrispondente");
ArrayOfCorrespondent recipients = ArrayOfCorrespondent.Factory.newInstance();
Correspondent[] correspondentArray = new Correspondent[1];
correspondentArray[0] = recipient;
recipients.setCorrespondentArray(correspondentArray);
document.setRecipients(recipients);
//Aggiungo un destinatario in conoscenza
Correspondent recipientCC = Correspondent.Factory.newInstance();
recipientCC.setId("IdDelCorrispondente");
ArrayOfCorrespondent recipientsCC = ArrayOfCorrespondent.Factory.newInstance();
Correspondent[] correspondentArrayCC = new Correspondent[1];
correspondentArrayCC[0] = recipientCC;
recipientsCC.setCorrespondentArray(correspondentArrayCC);
document.setRecipientsCC(recipientsCC);
//Associo tipologia documento
Template template = Template.Factory.newInstance();
template.setName("Nome tipologia Template");
template.setId("IdTemplate");
ArrayOfField fields = ArrayOfField.Factory.newInstance();
Field[] fieldArray = new Field[2];
Field temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
temp.setValue("Valore");
fieldArray[0] = temp;
temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
temp.setValue("Valore");
fieldArray[1] = temp;
temp = Field.Factory.newInstance();
```

```
fields.setFieldArray(fieldArray);
template.setFields(fields);
document.setTemplate(template);
//Inserisco File
File file = new File("PercorsoFile");
byte fileContent[] = new byte[(int)file.length()];
FileInputStream fin = new FileInputStream(file);
int read = 0;
ByteArrayOutputStream ous = new ByteArrayOutputStream();
while((read = fin.read(fileContent))!= -1){
ous.write(fileContent, 0, read); }
File mainDocument = File.Factory.newInstance();
mainDocument.setContent(fileContent);
mainDocument.setName(file.getName());
mainDocument.setMimeType(new MimeTypeFileMap().getContentType(file));
document.setMainDocument(mainDocument);
//Fine creazione documento
createDocumentRequest.setDocument(document);
CreateDocumentDocument1 createDocumentDocument =
CreateDocumentDocument1.Factory.newInstance();
CreateDocument createDocument = CreateDocument.Factory.newInstance();
createDocument.setRequest(createDocumentRequest);
createDocumentDocument.setCreateDocument(createDocument);
CreateDocumentResponseDocument createDocumentResponseDocument =
CreateDocumentResponseDocument.Factory.newInstance();
try
{
createDocumentResponseDocument = wws.createDocument(createDocumentDocument);
Document respdocument = null;
respdocument =
createDocumentResponseDocument.getCreateDocumentResponse().getCreateDocumentResult()
.getDocument();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.1.6 Creare un documento predisposto

Per la tipologia arrivo, partenza, interno vedi i corrispondenti protocolli. Campi in più da valorizzare rispetto ai corrispondenti protocolli.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Predisposed	Bool	Obbligatorio. Con il valore “true” indica che il documento deve essere creato come predisposto.

3.9.1.7 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “CreateDocumentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Document	Il Documento creato.

In caso di errore, “CreateDocument” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
MISSING_TYPE_DOCUMENT	Tipo di documento mancante.
MISSING_OBJECT	Oggetto obbligatorio.
REQUIRED_REGISTER	Registro obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
TEMPLATE_NOT_FOUND	Template non trovato.
REQUIRED_SENDER	Mittente obbligatorio.
SENDER_NOT_FOUND	Mittente non trovato.
REQUIRED_RECIPIENT	Destinatario obbligatorio.
REGISTER_NOT_FOUND	Registro non trovato.
RF_NOT_FOUND	RF non trovato.
RECIPIENT_NOT_FOUND	Destinatario non trovato.
FILE_CREATION_ERROR	Errore nella creazione del file.
TEMPLATE_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del template non consentite al ruolo dell'utente
TEMPLATE_FIELD_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del campo del template non consentite al ruolo dell'utente
APPLICATION_ERROR	Errore generico.

3.9.2 SERVIZIO “GETTEMPLATESDOCUMENTS”

Servizio per il reperimento di tutte le tipologie di documenti.

3.9.2.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetTemplatesDocumentsRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.9.2.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetTemplatesDocumentsResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Templates	Template[]	Lista dei template disponibili.

In caso di errore, “GetTemplatesDocumentsResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
TEMPLATES_NOT_FOUND	Tipologie di documenti non presenti.
APPLICATION_ERROR	Errore generico.

3.9.2.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetTemplatesDocumentsRequest request = new
Documents.GetTemplatesDocumentsRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
Documents.GetTemplatesDocumentsResponse response =
services.GetTemplatesDocuments(request);
Documents.Template[] result = response.Templates;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.2.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetTemplatesDocumentsRequest getTemplatesDocumentsRequest =
GetTemplatesDocumentsRequest.Factory.newInstance();
getTemplatesDocumentsRequest.setCodeAdm("CodiceAmministrazione");
getTemplatesDocumentsRequest.setUsername("UserName");
GetTemplatesDocumentsDocument getTemplatesDocumentsDocument =
GetTemplatesDocumentsDocument.Factory.newInstance();
GetTemplatesDocuments getTemplatesDocuments =
GetTemplatesDocuments.Factory.newInstance();
getTemplatesDocuments.setRequest(getTemplatesDocumentsRequest);
getTemplatesDocumentsDocument.setGetTemplatesDocuments(getTemplatesDocuments);
GetTemplatesDocumentsResponseDocument getTemplatesDocumentsResponseDocument =
GetTemplatesDocumentsResponseDocument.Factory.newInstance();
try
{
getTemplatesDocumentsResponseDocument =
wws.getTemplatesDocuments(getTemplatesDocumentsDocument);
Template[] templates = null;
templates =
getTemplatesDocumentsResponseDocument.getGetTemplatesDocumentsResponse().getGetTempl
atesDocumentsResult().getTemplates().getTemplateArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.3 SERVIZIO “GETTEMPLATEDOC”

Servizio per il reperimento del dettaglio di una tipologia di documento dato il nome o l'id. Obbligatorio un valore tra descrizione o id della tipologia.

3.9.3.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetTemplateDocRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
DescriptionTemplate	String	Opzionale. Nome della tipologia di documento.
IdTemplate	String	Opzionale. Id della tipologia del documento.

3.9.3.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetTemplateDocResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Template	Template	Dettaglio della tipologia del documento.

In caso di errore, “GetTemplateDocResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.

USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_ID_TEMPLATE	Obbligatorio un valore tra nome o id della tipologia di documento.
REQUIRED_ONLY_CODE_OR_ID_TEMPLATE	Richiesto soltanto un valore tra il nome o l'id della tipologia.
TEMPLATE_NOT_FOUND	Template non trovato.
TEMPLATE_NOT_ROLE_VISIBLE	Visibilità del template non associata al ruolo dell'utente
APPLICATION_ERROR	Errore generico.

3.9.3.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetTemplateDocRequest request = new Documents.GetTemplateDocRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.DescriptionTemplate = "Descrizione del template";
//Oppure request.IdTemplate = "Id del template";

try
{
    Documents.GetTemplateDocResponse response = services.GetTemplateDoc(request);
    Documents.Template result = response.Template;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.3.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetTemplateDocRequest getTemplateDocRequest =
GetTemplateDocRequest.Factory.newInstance();
getTemplateDocRequest.setCodeAdm("CodiceAmministrazione");
getTemplateDocRequest.setUsername("UserName");
getTemplateDocRequest.setIdTemplate("IdTemplate");
//getTemplateDocRequest.setDescriptionTemplate("Descrizione template"); Opzionale al
//posto dell'id
GetTemplateDocDocument getTemplateDocDocument =
GetTemplateDocDocument.Factory.newInstance();
GetTemplateDoc getTemplateDoc = GetTemplateDoc.Factory.newInstance();
```

```
getTemplateDoc.setRequest(getTemplateDocRequest);
getTemplateDocDocument.setGetTemplateDoc(getTemplateDoc);
GetTemplateDocResponseDocument getTemplateDocResponseDocument =
GetTemplateDocResponseDocument.Factory.newInstance();
try
{
getTemplateDocResponseDocument = wws.getTemplateDoc(getTemplateDocDocument);
Template template = null;
template =
getTemplateDocResponseDocument.getGetTemplateDocResponse().getGetTemplateDocResult().
getTemplate();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.4 SERVIZIO “GETDOCUMENT”

Servizio per il reperimento del dettaglio di un documento data la segnatura o l'id. Obbligatorio un valore tra id del documento o segnatura.

3.9.4.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetDocumentRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale. Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. DocNumber del documento.
Signature	String	Opzionale. Segnatura del documento.
GetFile	Boolean	Opzionale. Se true indica che nel dettaglio sia presente anche il contenuto del documento principale e degli allegati.

GetFileWithSignature	String	Opzionale. Se il valore inserito è “1” consente di scaricare i file firmati (p7m, tsr, tsd, m7m) con il formato e l'estensione originale, e non il file contenuto al loro interno.
-----------------------------	--------	--

3.9.4.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetDocumentResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “GetDocumentResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Obbligatorio un valore tra id o segnatura del documento.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto soltanto un valore l'id o la segnatura.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.9.4.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetDocumentRequest request = new Documents.GetDocumentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
request.Signature = "Segnatura del documento";
request.GetFile = true; //Se true prende anche il contenuto dei file

try
{
    Documents.GetDocumentResponse response = services.GetDocument(request);
    Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
```

```
//e.Message;
}
```

3.9.4.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetDocumentRequest getDocumentRequest = GetDocumentRequest.Factory.newInstance();
getDocumentRequest.setCodeAdm("CodiceAmministrazione");
getDocumentRequest.setUsername("UserName");
getDocumentRequest.setIdDocument("IdDocumento");
//getDocumentRequest.setSignature("Segnatura"); opzionale al posto dell'id
GetDocumentDocument1 getDocumentDocument =
GetDocumentDocument1.Factory.newInstance();
GetDocument getDocument = GetDocument.Factory.newInstance();
getDocument.setRequest(getDocumentRequest);
getDocumentDocument.setGetDocument(getDocument);
GetDocumentResponseDocument getDocumentResponseDocument =
GetDocumentResponseDocument.Factory.newInstance();
try
{
getDocumentResponseDocument = wws.getDocument(getDocumentDocument);
Document document = null;
document =
getDocumentResponseDocument.getGetDocumentResponse().getGetDocumentResult().getDocum
ent();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.5 SERVIZIO “UPLOADFILETODOCUMENT”

Servizio per aggiungere un file ad un documento, per creare un allegato o per aggiungere una nuova versione del file.

3.9.5.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “UploadFileToDocumentRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice

		dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. DocNumber del documento.
File (*)	File	Obbligatorio. File da acquisire.
CreateAttachment	Boolean	Opzionale. Se a true, indica che deve essere creato il documento come nuovo allegato.
Description	String	Obbligatorio. Descrizione del file da acquisire.
CovertToPDFA	Boolean	Opzionale. Se a true converte il file in PDF/A.
AttachmentType	String	Opzionale. Indica il tipo dell'allegato creato quando la proprietà CreateAttachment è true. Se "E", l'allegato è di tipo esterno, se assente o "U", l'allegato è di tipo utente.
HashFile	String	Opzionale(**). Hash del file da acquisire. (**) Obbligatorio in presenza della chiave di configurazione attiva "HASH_OBBLIGATORIO". (chiave di web.config).

(*) La classe File (par 2.1.6) include il campo Name. La valorizzazione di tale field è obbligatoria. Il valore inserito nel campo Name può differire dal nome del file da acquisire (il cui contenuto, convertito in base64, è inserito nel campo Contents).

Il valore inserito nel campo Name è quello che gli utenti del sistema P.I.Tre. visualizzano come nome file.

3.9.5.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "UploadFileToDocumentResponse".

Estende la classe base astratta "Response".

In caso di errore, "UploadFileToDocumentResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID	Il valore dell'id del documento è obbligatorio.
REQUIRED_FILE	Il file è obbligatorio.
REQUIRED_DESCRIPTION	La descrizione del file allegato è obbligatoria.
FILE_CREATION_ERROR	Errore nella creazione del file.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.9.5.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.UploadFileToDocumentRequest request = new
Documents.UploadFileToDocumentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";

byte[] content = new byte[this.uploadFile.PostedFile.InputStream.Length];
this.uploadFile.PostedFile.InputStream.Read(content, 0, content.Length);
request.File = new Documents.File();
request.File.Content = content;
request.File.Name = this.uploadFile.PostedFile.FileName;
request.File.MimeType = this.uploadFile.PostedFile.ContentType;

request.Description = "Descrizione";//Campo obbligatorio
request.CreateAttachment = false; //True se si vuole creare un nuovo allegato

try
{
Documents.UploadFileToDocumentResponse response =
services.UploadFileToDocument(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.5.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
```



```
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
UploadFileToDocumentRequest uploadFileToDocumentRequest =
UploadFileToDocumentRequest.Factory.newInstance();
uploadFileToDocumentRequest.setCodeAdm("CodiceAmministrazione");
uploadFileToDocumentRequest.setUsername("UserName");
uploadFileToDocumentRequest.setIdDocument("IdDelDocumento");
//uploadFileToDocumentRequest.setCreateAttachment(false); Se true crea un nuovo
//allegato
uploadFileToDocumentRequest.setDescription("Descrizione");
//Inserisco File
File file = new File("PercorsoFisicoDelFile");
byte fileContent[] = new byte[(int)file.length()];
FileInputStream fin = new FileInputStream(file);
int read = 0;
ByteArrayOutputStream ous = new ByteArrayOutputStream();
while ( (read = fin.read(fileContent)) != -1 ) {
ous.write(fileContent, 0, read); }
File mainDocument = File.Factory.newInstance();
mainDocument.setContent(fileContent);
mainDocument.setName(file.getName());
mainDocument.setMimeType(new MimeTypeOfFileTypeMap().getContentType(file));
uploadFileToDocumentRequest.setFile(mainDocument);
UploadFileToDocumentDocument1 uploadFileToDocumentDocument =
UploadFileToDocumentDocument1.Factory.newInstance();
UploadFileToDocument uploadFileToDocument =
UploadFileToDocument.Factory.newInstance();
uploadFileToDocument.setRequest(uploadFileToDocumentRequest);
uploadFileToDocumentDocument.setUploadFileToDocument(uploadFileToDocument);
UploadFileToDocumentResponseDocument uploadFileToDocumentResponseDocument =
UploadFileToDocumentResponseDocument.Factory.newInstance();

try
{
uploadFileToDocumentResponseDocument =
wws.uploadFileToDocument(uploadFileToDocumentDocument);
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.6 SERVIZIO “GETFILEDOCUMENTBYID”

Servizio per il reperimento di un file dato un id di un documento/allegato e opzionalmente il numero di versione.

3.9.6.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetFileDocumentByIdRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. DocNumber del documento.
VersionId	String	Opzionale. Versione del documento da visualizzare.

3.9.6.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "GetFileDocumentByIdResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
File	File	Dettaglio del File richiesto.

In caso di errore, "GetFileDocumentByIdResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID	Il valore dell'id del documento è obbligatorio.
REQUIRED_INTEGER	Il numero di versione deve essere un intero.
FILE_VERSION_NOT_FOUND	La versione del file non è stata trovata.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION ERROR	Errore generico.

3.9.6.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetFileDocumentByIdRequest request = new
Documents.GetFileDocumentByIdRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
request.VersionId = "Numero di versione"; //Opzionale

try
{
Documents.GetFileDocumentByIdResponse response = services.GetFileDocumentById(request);
Documents.File result = response.File;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.6.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetFileDocumentByIdRequest getFileDocumentByIdRequest =
GetFileDocumentByIdRequest.Factory.newInstance();
getFileDocumentByIdRequest.setCodeAdm("CodiceAmministrazione");
getFileDocumentByIdRequest.setUserName("UserName");
getFileDocumentByIdRequest.setIdDocument("IdDocumento");
//getFileDocumentByIdRequest.setVersionId("NumeroVersione"); opzionale
GetFileDocumentByIdDocument getFileDocumentByIdDocument =
GetFileDocumentByIdDocument.Factory.newInstance();
GetFileDocumentById getFileDocumentById = GetFileDocumentById.Factory.newInstance();
getFileDocumentById.setRequest(getFileDocumentByIdRequest);
getFileDocumentByIdDocument.setGetFileDocumentById(getFileDocumentById);
GetFileDocumentByIdResponseDocument getFileDocumentByIdResponseDocument =
GetFileDocumentByIdResponseDocument.Factory.newInstance();
try
{
getFileDocumentByIdResponseDocument =
wws.getFileDocumentById(getFileDocumentByIdDocument);
File file = null;
file =
getFileDocumentByIdResponseDocument.getGetFileDocumentByIdResponse().getGetFileDocum
entByIdResult().getFile();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.9.7 SERVIZIO “GETFILEWITHSIGNATUREORSTAMP”

Servizio per il reperimento di un file con l’aggiunta della segnatura o del timbro. Etichetta PDF o timbro sopraimpressi. Obbligatorio un valore tra id del documento e segnatura.

3.9.7.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetFileWithSignatureOrStampRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. DocNumber del documento.
Signature	String	Opzionale. Segnatura del documento.
WithSignature	Boolean	Opzionale. Se true inserisce la segnatura sul documento.
WithStamp	Boolean	Opzionale. Se true inserisce il timbro sul documento

3.9.7.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetFileWithSignatureOrStampResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “GetFileWithSignatureOrStampResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Inserire un valore tra id o segnatura.

REQUIRED_ONLY_ID_OR_SIGNATURE	Inserire soltanto un valore tra id o segnatura.
REQUIRED_ONLY_SIGNATURE_OR_STAMP	Selezionare soltanto un valore tra segnatura e timbro.
REQUIRED_SIGNATURE_OR_STAMP	Un valore tra segnatuta e timbro deve essere true.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.9.7.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetFileWithSignatureOrStampRequest request = new
Documents.GetFileWithSignatureOrStampRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
//Oppure la segnatuta request.Signature = "Segnatuta";

request.WithSignature = true; //Se si vuole visualizzare la segnatuta
request.WithStamp = false; // true se si vuole visualizzare il timbro

try
{
Documents.GetFileWithSignatureOrStampResponse response =
services.GetFileWithSignatureOrStamp(request);
Documents.File result = response.File;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.7.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetFileWithSignatureOrStampRequest getFileWithSignatureOrStampRequest =
GetFileWithSignatureOrStampRequest.Factory.newInstance();
getFileWithSignatureOrStampRequest.setCodeAdm("CodiceAmministrazione");
getFileWithSignatureOrStampRequest.setUsername("UserName");
getFileWithSignatureOrStampRequest.setIdDocument("IdDocumento");
getFileWithSignatureOrStampRequest.setWithSignature(true);
//getFileWithSignatureOrStampRequest.setWithStamp(false); //Se il timbre a true
```

```
//getFileWithSignatureOrStampRequest.setSignature("segnatura"); //Opzionale al posto
//dell'id del documento
getFileWithSignatureOrStampDocument getFileWithSignatureOrStampDocument =
getFileWithSignatureOrStampDocument.Factory.newInstance();
getFileWithSignatureOrStamp getFileWithSignatureOrStamp =
getFileWithSignatureOrStamp.Factory.newInstance();
getFileWithSignatureOrStamp.setRequest(getFileWithSignatureOrStampRequest);
getFileWithSignatureOrStampDocument.setGetFileWithSignatureOrStamp(getFileWithSignatureOrStamp);
getFileWithSignatureOrStampResponseDocument
getFileWithSignatureOrStampResponseDocument =
getFileWithSignatureOrStampResponseDocument.Factory.newInstance();
try
{
getFileWithSignatureOrStampResponseDocument =
wvs.getFileWithSignatureOrStamp(getFileWithSignatureOrStampDocument);
File file = null;
file =
getFileWithSignatureOrStampResponseDocument.getGetFileWithSignatureOrStampResponse().
getGetFileWithSignatureOrStampResult().getFile();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.8 SERVIZIO “ADDDOCINPROJECT”

Servizio per aggiungere un documento in un fascicolo. Obbligatorio un valore tra id del fascicolo e codice del fascicolo. Il codice del fascicolo si riferisce soltanto a fascicoli presenti sul titolario attivo.

3.9.8.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “AddDocInProjectRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. Id del documento che si desidera inserire nel

CodeProject	String	fascicolo. Opzionale. Codice del fascicolo nel titolario attivo nel quale si desidera inserire il documento.
IdProject	String	Opzionale. Id del fascicolo nel quale si desidera inserire il documento.

3.9.8.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “AddDocInProjectResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “AddDocInProjectResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_AND_ID	Fascicolo e documento obbligatori.
REQUIRED_ONLY_CODE_OR_ID_PROJECT	Richiesto solo un valore tra codice fascicolo e id fascicolo.
DOCUMENT_NOT_FOUND	Documento non trovato.
MULTIPLE	Trovato più di un fascicolo con lo stesso codice.
PROJECT_NOT_FOUND	Fascicolo non trovato.
APPLICATION_ERROR	Errore generico.

3.9.8.3 Aggiunta del documento in un sottofascicolo.

Per effettuare l’inserimento del documento in un sottofascicolo, bisogna inserire nella richiesta nel parametro in ingresso CodeProject il codice del fascicolo, inserendo quindi i caratteri separatori “//”, e di seguito la descrizione del sottofascicolo.

Un esempio di stringa accettabile può essere “6.2-2014//SottoFasc1”.

Per inserire il documento in un sottofascicolo non di primo livello bisogna inserire nel parametro tutto il percorso per arrivare alla cartella desiderata. Esempio “6.2-2014//SottoFasc1//SottoFasc2”.

3.9.8.4 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.AddDocInProjectRequest request = new Documents.AddDocInProjectRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Oppure
//request.CodeProject = "Codice del fascicolo solo su titolatio attivo";
request.IdDocument = "Id del documento";

try
{
Documents.AddDocInProjectResponse response = services.AddDocInProject(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.8.5 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
AddDocInProjectRequest addDocInProjectRequest =
AddDocInProjectRequest.Factory.newInstance();
addDocInProjectRequest.setCodeAdm("PAT_PRE");
addDocInProjectRequest.setUserName("PR29058");
addDocInProjectRequest.setIdDocument("8572470");
addDocInProjectRequest.setIdProject("8568247");
//addDocInProjectRequest.setCodeProject("CodiceFascicolo"); Opzionale al posto
//dell'id fascicolo il codice consentito solo su titolare attivo
AddDocInProjectDocument addDocInProjectDocument =
AddDocInProjectDocument.Factory.newInstance();
AddDocInProject addDocInProject = AddDocInProject.Factory.newInstance();
addDocInProject.setRequest(addDocInProjectRequest);
addDocInProjectDocument.setAddDocInProject(addDocInProject);
AddDocInProjectResponseDocument addDocInProjectResponseDocument =
AddDocInProjectResponseDocument.Factory.newInstance();
try
{
addDocInProjectResponseDocument = wws.addDocInProject(addDocInProjectDocument);
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```


3.9.9 SERVIZIO “EDITDOCUMENT”

Servizio per la modifica di un documento o protocollo.

3.9.9.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “EditDocumentRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Document	Document	Obbligatorio. Documento da modificare.

3.9.9.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “EditDocumentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Documet	Documento modificato.

In caso di errore, “EditDocumentResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_DOCUMENT	Documento obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
TEMPLATE_NOT_FOUND	Template non trovato.

TEMPLATE_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del template non consentite al ruolo dell'utente
TEMPLATE_FIELD_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del campo del template non consentite al ruolo dell'utente
REQUIRED_SENDER	Mittente obbligatorio.
SENDER_NOT_FOUND	Mittente non trovato.
REQUIRED_RECIPIENT	Richiesto destinatario.
RECIPIENT_NOT_FOUND	Destinatario non trovato.
APPLICATION_ERROE	Errore generico.

3.9.9.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.EditDocumentRequest request = new Documents.EditDocumentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

Documents.Document docRequest = new Documents.Document();
docRequest.Id = "Id del documento";
docRequest.Object = "Oggetto";
request.Document = docRequest; //Documento da modificare
try
{
    Documents.EditDocumentResponse response = services.EditDocument(request);
    Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.9.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_VERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE_NAMESPACE_URI);
EditDocumentRequest editDocumentRequest = EditDocumentRequest.Factory.newInstance();
editDocumentRequest.setCodeAdm("CodiceAmministrazione");
editDocumentRequest.setUserName("UserName");
Document document = Document.Factory.newInstance();
document.setId("IdDelDocumento");
document.setObject("OggettoDaModificare");
```

```

editDocumentRequest.setDocument(document);
EditDocumentDocument1 editDocumentDocument =
EditDocumentDocument1.Factory.newInstance();
EditDocument editDocument = EditDocument.Factory.newInstance();
editDocument.setRequest(editDocumentRequest);
editDocumentDocument.setEditDocument(editDocument);
EditDocumentResponseDocument editDocumentResponseDocument =
EditDocumentResponseDocument.Factory.newInstance();
try
{
editDocumentResponseDocument = wws.editDocument(editDocumentDocument);
Document respDocument = null;
respDocument =
editDocumentResponseDocument.getEditDocumentResponse().getEditDocumentResult().getDo
cument();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.9.10 SERVIZIO “PROTOCOLPREDISPOSED”

Servizio per la protocollazione di un predisposto senza effettuare modifiche al documento.

3.9.10.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “ProtocolPredisposedRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. Id del documento da protocollare.
CodeRegister	String	Obbligatorio. Codice del registro di protocollazione
CodeRF	String	Opzionale. Codice del RF di protocollazione.

3.9.10.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “EditDocumentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Document	Documento modificato.

In caso di errore, il servizio potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_DOCUMENT	Documento obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
DOCUMENT_NOT_FOUND	Destinatario non trovato.
APPLICATION_ERROE	Errore generico.

3.9.10.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.ProtocolPredisposedRequest request = new Documents.
ProtocolPredisposedRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
try
{
    Documents.EditDocumentResponse response = services.ProtocolPredisposed(request);
    Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.10.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
ProtocolPreDisposedRequest protocolPreDisposedRequest =
ProtocolPreDisposedRequest.Factory.newInstance();
protocolPreDisposedRequest.setCodeAdm("CodiceAmministrazione");
protocolPreDisposedRequest.setUserName("UserName");
protocolPreDisposedRequest.setId("IdDelDocumento");
ProtocolPreDisposedDocument1 protocolPreDisposedDocument =
ProtocolPreDisposedDocument1.Factory.newInstance();
ProtocolPreDisposed protocolPreDisposed = ProtocolPreDisposed.Factory.newInstance();
protocolPreDisposed.setRequest(protocolPreDisposedRequest);
protocolPreDisposedDocument.setProtocolPreDisposed(protocolPreDisposed);
EditDocumentResponseDocument editDocumentResponseDocument =
EditDocumentResponseDocument.Factory.newInstance();
try
{
editDocumentResponseDocument = wws.
protocolPreDisposed(protocolPreDisposedDocument);
Document respDocument = null;
respDocument =
editDocumentResponseDocument.getEditDocumentResponse().getEditDocumentResult().getDo
cument();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.11 SERVIZIO “GETDOCUMENTSINPROJECT”

Servizio per il reperimento dei documenti contenuti in un fascicolo. Obbligatorio l'id del fascicolo oppure la coppia codice e id del titolare.

3.9.11.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetDocumentsInProjectRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.

CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CodeProject	String	Opzionale. Codice del fascicolo.
ClassificationSchemeId	String	Opzionale. Id del titolare.
PageNumber	String	Opzionale. Numero di pagina da visualizzare.
ElementsInPage	String	Opzionale. Numero di elementi per pagina.
IdProject	String	Opzionale. Id del fascicolo.
GetTotalDocumentsNumber	bool	Se impostato a true la relativa response conterrà il numero totale di documenti ottenuti nella ricerca.

3.9.11.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetDocumentsInProjectResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Documents	Document[]	Lista dei documenti nel fascicolo.
TotalDocumentsNumber	int	Numero totale dei documenti trovati.

In caso di errore, “GetDocumentsInProjectResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto codice o id del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto solo un valore tra id o codice del fascicolo.
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto il codice del fascicolo

INVALID_PAGINATION	e l'id del titolare. Paginazione non valida.
DOCUMENTS_NOT_FOUND	Documenti non trovati.
PROJECT_NOT_FOUND	Fascicolo non trovato.
APPLICATION_ERROR	Errore generico.

3.9.11.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetDocumentsInProjectRequest request = new
Documents.GetDocumentsInProjectRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Paginazione opzionale
//request.PageNumber = "Numero di pagina";
//request.ElementsInPage = "Elementi per pagina";

//Si possono inserire codice fascicolo e id titolare al posto dell'id del fascicolo
//request.ClassificationSchemeId = this.txtIdTitolario.Text;
//request.CodeProject = this.txtCodiceFascicolo.Text;

request.GetTotalDocumentsNumber = true;

try
{
Documents.GetDocumentsInProjectResponse response =
services.GetDocumentsInProject(request);
Documents.Document[] result = response.Documents;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.11.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetDocumentsInProjectRequest getDocumentsInProjectRequest =
GetDocumentsInProjectRequest.Factory.newInstance();
getDocumentsInProjectRequest.setCodeAdm("CodiceAmministrazione");
getDocumentsInProjectRequest.setUsername("UserName");
getDocumentsInProjectRequest.setIdProject("IdFascicolo");
getDocumentsInProjectRequest.setGetTotalDocumentsNumber(true);
```

```

GetDocumentsInProjectDocument getDocumentsInProjectDocument =
GetDocumentsInProjectDocument.Factory.newInstance();
GetDocumentsInProject getDocumentsInProject =
GetDocumentsInProject.Factory.newInstance();
getDocumentsInProject.setRequest(getDocumentsInProjectRequest);
getDocumentsInProjectDocument.setGetDocumentsInProject(getDocumentsInProject);
GetDocumentsInProjectResponseDocument getDocumentsInProjectResponseDocument =
GetDocumentsInProjectResponseDocument.Factory.newInstance();
try
{
getDocumentsInProjectResponseDocument =
wvs.getDocumentsInProject(getDocumentsInProjectDocument);
Document[] documents = null;
documents =
getDocumentsInProjectResponseDocument.getGetDocumentsInProjectResponse().getGetDocum
entsInProjectResult().getDocuments().getDocumentArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.9.12 SERVIZIO “SEARCHDOCUMENTS”

Servizio per la ricerca di documenti.

3.9.12.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “SearchDocumentsRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Filters	Filter[]	Obbligatoria. Filtri con cui effettuare la ricerca degli utenti.
PageNumber	String	Opzionale. Numero di pagina da visualizzare.

ElementsInPage	String	Opzionale. Numero di elementi per pagina.
GetTotalDocumentsNumber	bool	Se impostato a true la relativa response conterrà il numero totale di documenti ottenuti nella ricerca.

3.9.12.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “SearchDocumentsResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Documents[]	Documenti trovati con i filtri inseriti.
TotalDocumentsNumber	Int	Numero totale dei documenti trovati.

Nel caso in cui la ricerca non produca risultati, verrà restituita la lista dei documenti vuota.

In caso di errore, “SearchDocuments” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_FILTER	Richiesto almeno un filtro di ricerca.
INVALID_PAGINATION	Paginazione non valida.
APPLICATION_ERROR	Errore generico / Formato (Number/ Bool/ Date) non corretto.

3.9.12.3 Filtri disponibili

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
YEAR	Filter	Inserire il valore dell’anno dei documenti..
IN_PROTOCOL	Filter	Con valore true cerca i protocolli in entrata.
OUT_PROTOCOL	Filter	Con valore true cerca i protocolli in uscita.

INTERNAL_PROTOCOL	Filter	Con valore true cerca i protocolli interni.
NOT_PROTOCOL	Filter	Con valore true cerca i documenti non protocollati.
PREDISPOSED	Filter	Con valore true cerca i predisposti.
ATTACHMENTS	Filter	Con valore true cerca gli allegati.
PRINTS	Filter	Con valore true cerca le stampe.
NUM_PROTOCOL_FROM	Filter	Filtro utilizzato per la ricerca per intervallo su numero di protocollo, questo valore è il limite inferiore.
NUM_PROTOCOL_TO	Filter	Filtro utilizzato per la ricerca per intervallo su numero di protocollo, questo valore è il limite superiore.
CREATION_DATE_FROM	Filter	Filtro utilizzato per intervallo su data creazione, questo valore è il limite inferiore.
CREATION_DATE_TO	Filter	Filtro utilizzato per intervallo su data creazione, questo valore è il limite superiore.
PROTOCOL_DATE_FROM	Filter	Filtro utilizzato per intervallo su data protocollazione, questo valore è il limite inferiore.
PROTOCOL_DATE_TO	Filter	Filtro utilizzato per intervallo su data protocollazione, questo valore è il limite superiore.
SENDER_RECIPIENT	Filter	Filtro utilizzato sui mittenti/destinatari di un documento, inserire l'id del corrispondente.
TEMPLATE	Filter	Filtro utilizzato per la ricerca della tipologia dei documenti.
DOCNUMBER_FROM	Filter	Filtro utilizzato per intervallo sul docnumner questo valore è il limite inferiore.
DOCNUMBER_TO	Filter	Filtro utilizzato per intervallo sul docnumner questo valore è il limite superiore.
REGISTER	Filter	Filtro utilizzato con il codice del registro.

OBJECT	Filter	Filtro utilizzato per la ricerca per oggetto di un documento.
TEMPLATE_EXTRACTION	Filter	Filtro utilizzato per l'estrazione dei campi profilati nella ricerca.
FULL_TEXT_SEARCH	Filter	Filtro utilizzato per la ricerca full-text su documenti text-based.

3.9.12.4 Estrazione dei campi profilati nella ricerca

Se si presentasse la necessità di prelevare i valori dei campi di una tipologia da una ricerca, il filtro `TEMPLATE_EXTRACTION` permette l'estrazione dei campi profilati.

Per utilizzare questo filtro è necessario effettuare una ricerca inserendo un filtro `TEMPLATE`, e settando il filtro `TEMPLATE_EXTRACTION` a `TRUE`. Nel caso non siano verificate le condizioni, il filtro viene ignorato anche se presente.

3.9.12.5 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.SearchDocumentsRequest request = new Documents.SearchDocumentsRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.Filters = new Documents.Filter[5];

int indice = 0;
Documents.Filter filter = new Documents.Filter();
filter.Name = "YEAR";
filter.Value = this.txtAnno.Text;
filter.Type = Documents.FilterType.Number;
request.Filters.SetValue(filter, indice);
indice++;
filter = new Documents.Filter();
filter.Name = "IN_PROTOCOL";
filter.Value = "true";
filter.Type = Documents.FilterType.Bool;
request.Filters.SetValue(filter, indice);
indice++;
filter = new Documents.Filter();
filter.Name = "OUT_PROTOCOL";
filter.Value = this.chkPar.Checked.ToString();
filter.Type = Documents.FilterType.Bool;
request.Filters.SetValue(filter, indice);
indice++;
filter = new Documents.Filter();
filter.Name = "INTERNAL_PROTOCOL";
filter.Value = this.chkInt.Checked.ToString();
filter.Type = Documents.FilterType.Bool;
request.Filters.SetValue(filter, indice);
indice++;
```

```
filter = new Documents.Filter();
filter.Name = "NOT_PROTOCOL";
filter.Value = this.chkGrigio.Checked.ToString();
filter.Type = Documents.FilterType.Bool;
request.Filters.SetValue(filter, indice);

request.GetTotalDocumentsNumber = true;

try
{
Documents.SearchDocumentsResponse response = services.SearchDocuments(request);
Documents.Document[] result = response.Documents;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.12.6 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
SearchDocumentsRequest searchDocumentsRequest =
SearchDocumentsRequest.Factory.newInstance();
searchDocumentsRequest.setCodeAdm("PAT_PRE");
searchDocumentsRequest.setUserName("PR29058");
ArrayOfFilter filters = ArrayOfFilter.Factory.newInstance();
Filter[] filterArray = new Filter[5];
int indice = 0;
Filter filter = Filter.Factory.newInstance();
filter.setName("YEAR");
filter.setValue("2012");
filter.setType(FilterType.Number);
filterArray[indice] = filter;
indice++;
filter = Filter.Factory.newInstance();
filter.setName("IN_PROTOCOL");
filter.setValue("true");
filter.setType(FilterType.Bool);
filterArray[indice] = filter;
indice++;
filter = Filter.Factory.newInstance();
filter.setName("OUT_PROTOCOL");
filter.setValue("true");
filter.setType(FilterType.Bool);
filterArray[indice] = filter;
indice++;
filter = Filter.Factory.newInstance();
filter.setName("INTERNAL_PROTOCOL");
filter.setValue("true");
filter.setType(FilterType.Bool);
filterArray[indice] = filter;
indice++;
filter = Filter.Factory.newInstance();
filter.setName("NOT_PROTOCOL");
```

```

filter.setValue("true");
filter.setType(FilterType.Bool);
filterArray[indice] = filter;
filters.setFilterArray(filterArray);
searchDocumentsRequest.setFilters(filters);
searchDocumentsRequest.setGetTotalDocumentsNumber(true);
SearchDocumentsDocument searchDocumentsDocument =
SearchDocumentsDocument.Factory.newInstance();
SearchDocuments searchDocuments = SearchDocuments.Factory.newInstance();
searchDocuments.setRequest(searchDocumentsRequest);
searchDocumentsDocument.setSearchDocuments(searchDocuments);
SearchDocumentsResponseDocument searchDocumentsResponseDocument =
SearchDocumentsResponseDocument.Factory.newInstance();
try
{
searchDocumentsResponseDocument = wws.searchDocuments(searchDocumentsDocument);
Document[] documents = null;
documents =
searchDocumentsResponseDocument.getSearchDocumentsResponse().getSearchDocumentsResult().getDocuments().getDocumentArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.9.13 SERVIZIO “EDITDOCSTATEDIAGRAM”

Servizio per modificare lo stato del diagramma associato ad un documento. Obbligatorio un valore tra id del documento e segnatura.

3.9.13.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “EditDocStateDiagramRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. DocNumber del documento.
Signature	String	Opzionale. Segnatura del

StateOfDiagram	String	documento. Obbligatorio. Nome dello stato successivo del documento.
-----------------------	--------	--

3.9.13.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “EditDocStateDiagramResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “EditDocStateDiagramResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Richiesto id del documento o segnatura.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto solo un valore tra id o segnatura del documento.
DOCUMENT_NOT_FOUND	Documento non trovato.
DOC_STATEOFDIAGRAM_NOT_FOUND	Stato del diagramma non trovato.
DIAGRAM_NOT_FOUND	Diagramma non trovato.
TEMPLATE_NOT_FOUND	Template non trovato.
APPLICATION_ERROR	Errore generico.

3.9.13.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.EditDocStateDiagramRequest request = new
Documents.EditDocStateDiagramRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
//Oppure request.Signature = "Segnatura del documento";

request.StateOfDiagram = "Nome dello stato in cui verrà salvato il documento";

try
{
Documents.EditDocStateDiagramResponse response = services.EditDocStateDiagram(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
```

```
//e.Message;
}
```

3.9.13.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
EditDocStateDiagramRequest editDocStateDiagramRequest =
EditDocStateDiagramRequest.Factory.newInstance();
editDocStateDiagramRequest.setCodeAdm("CodiceAmministrazione");
editDocStateDiagramRequest.setUserName("UserName");
editDocStateDiagramRequest.setIdDocument("IdDocumento");
editDocStateDiagramRequest.setStateOfDiagram("NomeStato");
EditDocStateDiagramDocument editDocStateDiagramDocument =
EditDocStateDiagramDocument.Factory.newInstance();
EditDocStateDiagram editDocStateDiagram = EditDocStateDiagram.Factory.newInstance();
editDocStateDiagram.setRequest(editDocStateDiagramRequest);
editDocStateDiagramDocument.setEditDocStateDiagram(editDocStateDiagram);
EditDocStateDiagramResponseDocument editDocStateDiagramResponseDocument =
EditDocStateDiagramResponseDocument.Factory.newInstance();
try
{
editDocStateDiagramResponseDocument =
wws.editDocStateDiagram(editDocStateDiagramDocument);
//Modifica effettuata
}
catch(RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.14 SERVIZIO “GETDOCUMENTSTATEDIAGRAM”

Servizio per il reperimento dello stato di un documento. Obbligatorio un valore tra id del documento e segnatura.

3.9.14.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetDocumentStateDiagramRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.

CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. DocNumber del documento.
Signature	String	Opzionale. Segnatura del documento.

3.9.14.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "GetDocumentStateDiagramResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
StateOfDiagram	StateOfDiagram	Stato del diagramma del documento.

In caso di errore, "GetDocumentStateDiagramResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Richiesto id del documento o segnatura.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto solo un valore tra id o segnatura del documento.
DOCUMENT_NOT_FOUND	Documento non trovato.
DOC_STATEOFDIAGRAM_NOT_FOUND	Stato del diagramma non trovato.
DIAGRAM_NOT_FOUND	Diagramma non trovato.
TEMPLATE_NOT_FOUND	Template non trovato.
APPLICATION_ERROR	Errore generico.

3.9.14.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetDocumentStateDiagramRequest request = new
Documents.GetDocumentStateDiagramRequest();
```



```
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
//Oppure request.Signature = "Segnatura del documento";

try
{
Documents.GetDocumentStateDiagramResponse response =
services.GetDocumentStateDiagram(request);
Documents.StateOfDiagram result = response.StateOfDiagram;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.14.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetDocumentStateDiagramRequest getDocumentStateDiagramRequest =
GetDocumentStateDiagramRequest.Factory.newInstance();
getDocumentStateDiagramRequest.setCodeAdm("CodiceAmministrazione");
getDocumentStateDiagramRequest.setUserName("UserName");
getDocumentStateDiagramRequest.setIdDocument("IdDocumento");
GetDocumentStateDiagramDocument getDocumentStateDiagramDocument =
GetDocumentStateDiagramDocument.Factory.newInstance();
GetDocumentStateDiagram getDocumentStateDiagram =
GetDocumentStateDiagram.Factory.newInstance();
getDocumentStateDiagram.setRequest(getDocumentStateDiagramRequest);
getDocumentStateDiagramDocument.setGetDocumentStateDiagram(getDocumentStateDiagram);
GetDocumentStateDiagramResponseDocument getDocumentStateDiagramResponseDocument =
GetDocumentStateDiagramResponseDocument.Factory.newInstance();
try
{
getDocumentStateDiagramResponseDocument =
wws.getDocumentStateDiagram(getDocumentStateDiagramDocument);
StateOfDiagram stateOfDiagram = null;
stateOfDiagram =
getDocumentStateDiagramResponseDocument.getGetDocumentStateDiagramResponse().getGetD
ocumentStateDiagramResult().getStateOfDiagram();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.15 SERVIZIO “SENDDOCUMENT”

Servizio per l'invio di un documento. Obbligatorio un valore tra id del documento e segnatura.

3.9.15.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “SendDocumentRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. DocNumber del documento.
Signature	String	Opzionale. Segnatura del documento.

3.9.15.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “SendDocumentResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “SendDocumentResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Richiesto id del documento o segnatura.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto solo un valore tra id o segnatura del documento.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.9.15.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.SendDocumentRequest request = new Documents.SendDocumentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";
//Oppure request.Signature = "Segnatura del documento";

try
{
    Documents.SendDocumentResponse response = services.SendDocument(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.15.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
SendDocumentRequest sendDocumentRequest = SendDocumentRequest.Factory.newInstance();
sendDocumentRequest.setCodeAdm("CodiceAmministrazione");
sendDocumentRequest.setUserName("UserName");
sendDocumentRequest.setIdDocument("IdDocumento");
SendDocumentDocument1 sendDocumentDocument =
SendDocumentDocument1.Factory.newInstance();
SendDocument sendDocument = SendDocument.Factory.newInstance();
sendDocument.setRequest(sendDocumentRequest);
sendDocumentDocument.setSendDocument(sendDocument);
SendDocumentResponseDocument sendDocumentResponseDocument =
SendDocumentResponseDocument.Factory.newInstance();
try
{
    sendDocumentResponseDocument = wws.sendDocument(sendDocumentDocument);
    //Spedita
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.9.16 SERVIZIO “GETSTAMPANDSIGNATURE”

Servizio per il reperimento dei dati di segnatura e timbro del documento. Obbligatorio un valore tra id del documento o segnatura.

3.9.16.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetStampAndSignatureRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. DocNumber del documento.
Signature	String	Opzionale. Segnatura del documento.

3.9.16.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetStampAndSignatureResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “GetStampAndSignatureResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Richiesto id del documento o segnatura.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto solo un valore tra id o segnatura del documento.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.9.16.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetStampAndSignatureRequest request = new
Documents.GetStampAndSignatureRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdDocument = "Id del documento";
//Oppure request.Signature = "Segnatura del documento";

this.txtRequest.Text = this.SerializeAsXml(request);

try
{
Documents.GetStampAndSignatureResponse response =
services.GetStampAndSignature(request);
this.txtResponse.Text = this.SerializeAsXml(response);
Documents.Stamp result = response.Stamp;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.9.16.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetStampAndSignatureRequest getStampAndSignatureRequest=
GetStampAndSignatureRequest.Factory.newInstance();
getStampAndSignatureRequest.setCodeAdm("CodiceAmministrazione");
getStampAndSignatureRequest.setUserName("UserName");
getStampAndSignatureRequest.setIdDocument("IdDocumento");
GetStampAndSignatureDocument getStampAndSignatureDocument =
GetStampAndSignatureDocument.Factory.newInstance();
GetStampAndSignature getStampAndSignature =
GetStampAndSignature.Factory.newInstance();
getStampAndSignature.setRequest(getStampAndSignatureRequest);
getStampAndSignatureDocument.setGetStampAndSignature(getStampAndSignature);
GetStampAndSignatureResponseDocument getStampAndSignatureResponseDocument =
GetStampAndSignatureResponseDocument.Factory.newInstance();
try
{
getStampAndSignatureResponseDocument =
wws.getStampAndSignature(getStampAndSignatureDocument);
Stamp stamp = null;
stamp =
getStampAndSignatureResponseDocument.getGetStampAndSignatureResponse().getGetStampAn
dSignatureResult().getStamp();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.17 SERVIZIO “GETDOCUMENTFILTERS”

Servizio che restituisce la lista dei filtri applicabili alla ricerca documenti.

3.9.17.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetDocumentFiltersRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.9.17.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetDocumentFiltersResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Filters	Filter[]	Lista dei filtri applicabili.

In caso di errore, “GetDocumentFiltersResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.9.17.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();  
Documents.GetDocumentFiltersRequest request = new  
Documents.GetDocumentFiltersRequest();
```

```
request.UserName = this.txt_username.Text;
request.CodeAdm = this.txt_codAmm.Text;
request.CodeRoleLogin = this.txt_codice_ruolo.Text;

this.txtRequest.Text = this.SerializeAsXml(request);

try
{
Documents.GetDocumentFiltersResponse response = services.GetDocumentFilters(request);
this.txtResponse.Text = this.SerializeAsXml(response);
Documents.Filter[] result = response.Filters;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.17.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetDocumentFiltersRequest getDocumentFiltersRequest=
GetDocumentFiltersRequest.Factory.newInstance();
getDocumentFiltersRequest.setCodeAdm("CodiceAmministrazione");
getDocumentFiltersRequest.setUsername("UserName");
GetDocumentFiltersDocument getDocumentFiltersDocument =
GetDocumentFiltersDocument.Factory.newInstance();
GetDocumentFilters getDocumentFilters = GetDocumentFilters.Factory.newInstance();
getDocumentFilters.setRequest(getDocumentFiltersRequest);
getDocumentFiltersDocument.setDocumentFilters(getDocumentFilters);
GetDocumentFiltersResponseDocument getDocumentFiltersResponseDocument =
GetDocumentFiltersResponseDocument.Factory.newInstance();
try
{
getDocumentFiltersResponseDocument = wws.getDocumentFilters
(getDocumentFiltersDocument);
Filter[] filters = null;
filters =
getDocumentFiltersResponseDocument.getGetDocumentFiltersResponse().DocumentFiltersRe
sult().getFilters();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.9.18 SERVIZIO “CREATEDOCUMENTANDADDINPROJECT”

Servizio che crea un nuovo documento e lo inserisce nel relativo fascicolo indicato.

3.9.18.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “CreateDocumentAndAddInProjectRequest”.

Struttura dell'oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CodeRegister	String	Obbligatorio. Codice del registro di protocollo.
Document	Document	Obbligatorio. Documento da creare.
CodeRF	String	Opzionale. Codice dell'RF in cui si vuole protocollare.
CodeProject	String	Obbligatorio. Codice del fascicolo nel quale fascicolare il documento, il codice prende soltanto i fascicoli nei titolari attivi. Da utilizzare in coppia con ClassificationSchemaId.
IdProject	String	Obbligatorio. Id del fascicolo nel quale fascicolare il documento. Alternativo alla coppia CodeProject/ClassificationSchemaId
ClassificationSchemaId	String	Obbligatorio. Id del titolare.

3.9.18.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “CreateDocumentAndAddInProjectResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Document	Dettaglio del documento creato.

In caso di errore, “ CreateDocumentAndAddInProjectResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.9.18.3 Esempio di utilizzo C#

```
Documents.CreateDocumentAndAddInProjectRequest request_ = new
Documents.CreateDocumentAndAddInProjectRequest();
request_.Document = new Documents.Document();
request_.Document.Object = txtOggetto.Text;
request_.Document.DocumentType = rbl_radio.SelectedItem.Value;
request_.UserName = request.UserName;
request_.CodeAdm = request.CodeAdm;
request_.Document = request.Document;
request_.CodeRegister = request.CodeRegister;
request_.CodeRF = request.CodeRF;
request_.IdProject = txtIdFasc.Text;

// In alternativa a request_.IdProject:
// request_.CodeProject = txtCodiceFasc.Text;
// request_.ClassificationSchemeId = txtCodiceClassificazione.Text;

request_.CodeApplication = "TestPIS";

try
{
Documents.CreateDocumentAndAddInProjectResponse response =
services.CreateDocumentAndAddInProject(request_);
this.txtResponse.Text = this.SerializeAsXml(response);
Documents.Document result = response.Document;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.18.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
```

```
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE_NAMESPACE_URI);
CreateDocumentAndAddInProjectRequest createDocumentAndAddInProjectRequest=
CreateDocumentAndAddInProjectRequest.Factory.newInstance();
createDocumentAndAddInProjectRequest.setCodeAdm("CodiceAmministrazione");
createDocumentAndAddInProjectRequest.setUsername("UserName");
createDocumentAndAddInProjectRequest.setDocument(document);
createDocumentAndAddInProjectRequest.setCodeRegister("CodeRegister");
createDocumentAndAddInProjectRequest.setCodeRF("CodeRF");
createDocumentAndAddInProjectRequest.setIdProject("IdProject");

CreateDocumentAndAddInProjectDocument createDocumentAndAddInProjectDocument =
CreateDocumentAndAddInProjectDocument.Factory.newInstance();
CreateDocumentAndAddInProject createDocumentAndAddInProject =
CreateDocumentAndAddInProject.Factory.newInstance();
createDocumentAndAddInProject.setRequest(createDocumentAndAddInProjectRequest);
createDocumentAndAddInProjectDocument.setCreateDocumentAndAddInProject
(createDocumentAndAddInProject);
CreateDocumentAndAddInProjectResponseDocument
createDocumentAndAddInProjectResponseDocument =
CreateDocumentAndAddInProjectResponseDocument.Factory.newInstance();
try
{
createDocumentAndAddInProjectResponseDocument = wws.createDocumentAndAddInProject
(createDocumentAndAddInProjectDocument);
Document document = null;
document =
createDocumentAndAddInProjectResponseDocument.getCreateDocumentAndAddInProjectRespon
se().DocumentResult().getDocument();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.9.19 SERVIZIO “FOLLOWDOCUMENT”

Servizio per la richiesta di “monitorare”/“non monitorare più” un documento presente nel Sistema.

3.9.19.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “FollowRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdObject	String	Obbligatorio. Id dell’oggetto di dominio che si intende “monitorare”, “non monitorare più”. Nello specifico si tratta dell’id documento.
Operation	OperationFollow	Obbligatorio. Specifica il tipo di azione che si intende eseguire sul documento: AddDoc = “monitora il documento”, RemoveDoc= “non monitorare più il documento”.
CodeApplication	String	Obbligatorio. Codice dell'applicazione che ha richiesto il monitoraggio.

3.9.19.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “FollowResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “FollowResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
INVALID_TOKEN	Token non valido.
USER_NO_EXIST	Utente non trovato.
ROLE_NO_EXIST	Il ruolo non esiste
USER_NO_ROLE	L'utente non ha ruoli
APPLICATION_ERROR	Eccezione Generica: APPLICATION_ERROR I campi Operation, CodeApplication, IdObject sono obbligatori.
APPLICATION_ERROR	Eccezione Generica: APPLICATION_ERROR Il campo operation deve essere AddDocument o RemoveDocument.

3.9.19.3 Esempio di utilizzo C#

```
Documents.DocumentsClient service = new Documents.DocumentsClient();
Documents.FollowRequest request = new Documents.FollowRequest();
request.CodeApplication = "GDOC";
request.UserName = "SAM";
request.CodeRoleLogin = "S147SEGP";
request.CodeAdm = "PAT_COLL";
request.Operation = Documents.OperationFollowDomainObject.AddDoc;
//or
//followDocReq.Operation = Documents.OperationFollowDomainObject.RemoveDoc;
request.IdObject = "14086814";
try
{
    Documents.FollowResponse response = service.FollowDocument(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
catch (Exception exc)
{
    //exc.Message;
}
```

3.9.19.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
```

```

wvs._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wvs._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
FollowRequest followRequest = FollowRequest.Factory.newInstance();
followRequest.setCodeApplication("CodiceApplicazione");
followRequest.setUsername("UserName");
followRequest.setCodeRoleLogin("CodeRoleLogin");
followRequest.setCodeAdm("CodeAdministration");
followRequest.setIdObject("IdObject");
followRequest.setOperation(OperationFollowDomainObject.AddDoc);
FollowResponse followResponse = FollowResponse.Factory.newInstance();
try
{
followResponse = wvs.FollowDocument(followRequest);
}
catch(RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.9.20 SERVIZIO “GETLINKDOCBYID”

Servizio per il recupero di un URL per l’apertura diretta del documento (previa autenticazione all’interfaccia PITRE)

3.9.20.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetLinkDocByIdRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. DocNumber del documento.

3.9.20.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetLinkDocByIdResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
--------------------	-------------------------	---------------------------

DocumentLink	String	URL diretta per il documento.
---------------------	--------	-------------------------------

In caso di errore, “ GetLinkDocByIdResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID	Richiesto id del documento.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.9.20.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetLinkDocByIdRequest request = new Documents. GetLinkDocByIdRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdDocument = "Id del documento";

try
{
    Documents. GetLinkDocByIdResponse response = services. GetLinkDocById(request);
    string result = response.DocumentLink;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.9.20.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetLinkDocByIdRequest getLinkDocByIdRequest =
GetLinkDocByIdRequest.Factory.newInstance();
getLinkDocByIdRequest.setCodeAdm("CodiceAmministrazione");
getLinkDocByIdRequest.setUserName("UserName");
getLinkDocByIdRequest.setIdDocument("IdDocumento");
GetLinkDocById getLinkDocById= GetLinkDocById.Factory.newInstance();
getLinkDocById.setRequest(getLinkDocByIdRequest);
```

```

GetLinkDocByIdResponseDocument getLinkDocByIdResponseDocument =
GetLinkDocByIdResponseDocument.Factory.newInstance();
try
{
getLinkDocByIdResponseDocument = wws.getLinkDocById (getLinkDocByIdRequest);
string returnvalue = getLinkDocByIdResponseDocument.
getLinkDocByIdResponse().getDocumentLink();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.10 SERVIZI PER LA GESTIONE AVANZATA DEI DOCUMENTI

3.10.1 SERVIZIO “REMOVEDOCUMENT”

Il servizio permette la rimozione di un documento e la collocazione dello stesso nel cestino.

3.10.1.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “RemoveDocumentRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. Id del documento da cancellare.

3.10.1.2 Descrizione dell’oggetto “Response”.

L’oggetto response restituito dal servizio è il “RemoveDocumentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ResultMessage	String	Messaggio contenente l'esito dell'operazione

In caso di errore, “ RemoveDocumentResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
USER_NO_EXIST	Utente non trovato.
ROLE_NO_EXIST	Il ruolo non esiste.
REQUIRED_ID	Id del documento obbligatorio
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.10.1.3 Esempio utilizzo C#

```
DocumentsAdvanced.DocumentsAdvancedClient services = new
DocumentsAdvanced.DocumentsAdvancedClient();
DocumentsAdvanced.RemoveDocumentRequest request = new
DocumentsAdvanced.RemoveDocumentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdDocument = "Id del documento";

try
{
DocumentsAdvanced.RemoveDocumentResponse response = services.RemoveDocument(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.10.1.4 Esempio di utilizzo Java

```
DocumentsAdvancedStub wws = new DocumentsAdvancedStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
RemoveDocumentRequest removeDocumentRequest =
RemoveDocumentRequest.Factory.newInstance();
removeDocumentRequest.setCodeAdm("CodiceAmministrazione");
removeDocumentRequest.setUserName("UserName");
removeDocumentRequest.setIdDocument("IdDocumento");
```



```

RemoveDocumentDocument removeDocumentDocument =
RemoveDocumentDocument.Factory.newInstance();
RemoveDocument removeDocument = RemoveDocument.Factory.newInstance();
removeDocument.setRequest(removeDocumentRequest);
removeDocumentDocument.setRemoveDocument(removeDocument);
RemoveDocumentResponseDocument removeDocumentResponseDocument =
RemoveDocumentResponseDocument.Factory.newInstance();
try
{
removeDocumentResponseDocument = wws.removeDocument(removeDocumentDocument);
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.10.2 SERVIZIO “SENDDOCUMENTADVANCED”

Il servizio permette la spedizione di un documento permettendo la scelta del registro, della mail dalla quale inviare e dei destinatari.

3.10.2.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “SendDocumentAdvancedRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. Id del documento da cancellare. E’ obbligatorio inserire uno tra idDocument e Signature.
Signature	String	Opzionale. Segnatura del documento. E’ obbligatorio inserire uno tra idDocument e Signature.
CodeRegister	String	Opzionale. Codice del registro dal quale si spedisce. E’ obbligatorio inserire uno tra CodeRegister e IdRegister.

IdRegister	String	Opzionale. Id del registro dal quale si spedisce. E' obbligatorio inserire uno tra CodeRegister e IdRegister.
SenderMail	String	Opzionale. Mail dalla quale si spedisce: se omessa utilizza la mail default del registro.
Recipients	Correspondent[]	Opzionale. Array dei corrispondenti da inserire come destinatari della spedizione. Se omesso spedisce a tutti i destinatari.

3.10.2.2 Descrizione dell'oggetto "Response".

L'oggetto response restituito dal servizio è il " SendDocumentAdvancedResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
SendingResults	SendingResult[]	Esiti delle spedizioni per ogni destinatari.

In caso di errore, " SendDocumentAdvancedResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
USER_NO_EXIST	Utente non trovato.
ROLE_NO_EXIST	Il ruolo non esiste.
REQUIRED_ID_OR_SIGNATURE	Richiesto Id o Segnatura del documento.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto solo uno tra Id e Segnatura del documento.
REQUIRED_CODE_OR_ID_REGISTER	Richiesto Codice o Id del registro.
REQUIRED_ONLY_CODE_OR_ID_REGISTER	Richiesto solo uno tra Codice e Id del registro.
DOCUMENT_NOT_FOUND	Documento non trovato.

REGISTER_NOT_FOUND	Registro non trovato.
RECIPIENT_NOT_FOUND	Destinatario non trovato.
SEND_DOCUMENT_FAILED	Errore durante la spedizione.
APPLICATION_ERROR	Errore generico.

3.10.2.3 Esempio utilizzo C#

```
DocumentsAdvanced.DocumentsAdvancedClient services = new
DocumentsAdvanced.DocumentsAdvancedClient();
DocumentsAdvanced.SendDocumentAdvancedRequest request = new
DocumentsAdvanced.SendDocumentAdvancedRequest();
```

```
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdDocument = "Id del documento";
request.IdRegister = "Id del registro";
```

```
try
{
DocumentsAdvanced.SendDocumentAdvancedResponse response =
services.SendDocumentAdvanced.(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.10.2.4 Esempio di utilizzo Java

```
DocumentsAdvancedStub wws = new DocumentsAdvancedStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
SendDocumentAdvancedRequest sendDocumentAdvancedRequest =
SendDocumentAdvancedRequest.Factory.newInstance();
sendDocumentAdvancedRequest.setCodeAdm("CodiceAmministrazione");
sendDocumentAdvancedRequest.setUserName("UserName");
sendDocumentAdvancedRequest.setIdDocument("IdDocumento");
SendDocumentAdvancedDocument sendDocumentAdvancedDocument =
SendDocumentAdvancedDocument.Factory.newInstance();
SendDocumentAdvanced sendDocumentAdvanced =
SendDocumentAdvanced.Factory.newInstance();
sendDocumentAdvanced.setRequest(sendDocumentAdvancedRequest);
sendDocumentAdvancedDocument.set SendDocumentAdvanced(sendDocumentAdvanced);
SendDocumentAdvancedResponseDocument sendDocumentAdvancedResponseDocument =
SendDocumentAdvancedResponseDocument.Factory.newInstance();
try
{
sendDocumentAdvancedResponseDocument = wws.sendDocumentAdvanced
(sendDocumentAdvancedDocument);
}
catch (RemoteException e)
```

```
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.10.3 SERVIZIO “NUOVAFATTURAATTIVA”

Il servizio permette la creazione di una particolare tipologia di documento, la fattura attiva.

3.10.3.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “NuovaFatturaRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
AuthenticationToken	String	Token di autenticazione.
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Document	Document	Obbligatorio. Documento che deve essere creato. Vedere le tabelle successive per le varie tipologie di documento.
TrasmissionReceiver	Correspondent	Opzionale: corrispondente che deve ricevere la trasmissione della fattura attiva.

Il parametro Document deve essere popolato nei seguenti campi:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
DocumentType	String	Obbligatorio. Tipo del documento da valorizzare con “G”
Object	String	Obbligatorio. Descrizione della fattura.

Template	Template	Obbligatorio. Oggetto di tipo Template popolato con l'id della tipologia fattura attiva.
MainDocument	File	Obbligatorio. Oggetto di tipo File popolato nel campo Content di tipo binario, e nel campo Name di tipo stringa. Il file deve essere obbligatoriamente una fattura in formato XML o XML firmato.

L'oggetto TransmissionReceiver deve essere popolato nel suo parametro Code con il codice del corrispondente che deve ricevere la trasmissione.

3.10.3.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "NuovaFatturaResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Document	Il Documento creato.

In caso di errore, "NuovaFatturaAttiva" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
MISSING_TYPE_DOCUMENT	Tipo di documento mancante.
MISSING_OBJECT	Oggetto obbligatorio.
REQUIRED_REGISTER	Registro obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
TEMPLATE_NOT_FOUND	Template non trovato.
REQUIRED_SENDER	Mittente obbligatorio.
SENDER_NOT_FOUND	Mittente non trovato.

REQUIRED_RECIPIENT	Destinatario obbligatorio.
REGISTER_NOT_FOUND	Registro non trovato.
RF_NOT_FOUND	RF non trovato.
RECIPIENT_NOT_FOUND	Destinatario non trovato.
FILE_CREATION_ERROR	Errore nella creazione del file.
TEMPLATE_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del template non consentite al ruolo dell'utente
TEMPLATE_FIELD_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del campo del template non consentite al ruolo dell'utente
APPLICATION_ERROR	Errore generico.

3.10.3.3 Note riguardanti il servizio NuovaFatturaAttiva

Il servizio NuovaFatturaAttiva è stato sviluppato per venire incontro alla necessità di gestire una particolare tipologia documentale, Fattura Attiva, da parte delle integrazioni.

Il servizio comprende al suo interno le funzionalità di creazione di un documento, associazione della tipologia, compilazione dei campi, caricamento file principale, caricamento allegati, prelievo del titolare attivo, fascicolazione del documento e trasmissione del documento.

L'oggetto NuovaFatturaRequest è la fusione degli oggetti CreateDocumentRequest, UploadFileToDocument ed ExecuteTransmissionDocumentRequest, per come popolare i campi della richiesta, si rimanda ai paragrafi 3.9.1, 3.9.5 e 3.14.5 rispettivamente.

Il documento principale dovrà essere in formato XML, e dovrà essere una fattura elettronica nel formato SdI. Inoltre il fornitore, identificato dal suo codice fiscale e/o partita IVA, dovrà essere censito all'interno di PITRE in una tabella di configurazione nella quale sarà presente l'amministrazione sul quale il fornitore è abilitato a caricare le fatture attive, su quale registro stacca il numero di repertorio, e in quale fascicolo classifica le stesse. Nel caso il fornitore non sia correttamente censito, verrà restituito un errore APPLICATION ERROR.

Gli allegati, costituiti da oggetti di tipo File, saranno inseriti nell'array Request.Document.Attachments[].

Le informazioni di trasmissione si limitano all'indicazione del codice dell'ufficio e/o ruolo che riceverà la trasmissione, da inserire nel campo Request.TransmissionReceiver.Code. La ragione di trasmissione, il tipo di notifica vengono scegli automaticamente dal sistema.

3.10.4 SERVIZIO “NUOVOLOTTOATTIVO”

Il servizio permette la creazione di una particolare tipologia di documento, il lotto di fatture attive.

3.10.4.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “CaricaLottoInPi3Request”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
AuthenticationToken	String	Token di autenticazione.
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Document	Document	Obbligatorio. Documento che deve essere creato. Vedere le tabelle successive per le varie tipologie di documento.
TrasmissionReceiver	Correspondent	Opzionale: corrispondente che deve ricevere la trasmissione della fattura attiva.

Il parametro Document deve essere popolato nei seguenti campi:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
DocumentType	String	Obbligatorio. Tipo del documento da valorizzare con “G”
Object	String	Obbligatorio. Descrizione della fattura.
Template	Template	Obbligatorio. Oggetto di tipo Template popolato con l’id della tipologia fattura attiva.

MainDocument	File	Obbligatorio. Oggetto di tipo File popolato nel campo Content di tipo binario, e nel campo Name di tipo stringa. Il file deve essere obbligatoriamente una fattura in formato XML o XML firmato.
---------------------	------	--

L'oggetto TransmissionReceiver deve essere popolato nel suo parametro Code con il codice del corrispondente che deve ricevere la trasmissione.

3.10.4.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il " CaricaLottoInPi3Response".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Document	Document	Il Documento creato.

In caso di errore, "NuovoLottoAttivo" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
MISSING_TYPE_DOCUMENT	Tipo di documento mancante.
MISSING_OBJECT	Oggetto obbligatorio.
REQUIRED_REGISTER	Registro obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
TEMPLATE_NOT_FOUND	Template non trovato.
REQUIRED_SENDER	Mittente obbligatorio.
SENDER_NOT_FOUND	Mittente non trovato.
REQUIRED_RECIPIENT	Destinatario obbligatorio.
REGISTER_NOT_FOUND	Registro non trovato.
RF_NOT_FOUND	RF non trovato.

RECIPIENT_NOT_FOUND	Destinatario non trovato.
FILE_CREATION_ERROR	Errore nella creazione del file.
TEMPLATE_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del template non consentite al ruolo dell'utente
TEMPLATE_FIELD_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del campo del template non consentite al ruolo dell'utente
APPLICATION_ERROR	Errore generico.

3.10.4.3 Note riguardanti il servizio NuovoLottoAttivo

Si rimanda al paragrafo 3.10.3.3 per il funzionamento condiviso con il metodo NuovaFatturaAttiva, in questo paragrafo facciamo notare come il file XML caricato come documento principale debba essere un lotto attivo, quindi un file fattura elettronica con un unico tag FatturaElettronicaHeader, e più tag FatturaElettronicaBody.

3.11 SERVIZI PER LA GESTIONE DEI FASCICOLI

3.11.1 SERVIZIO “CREATEPROJECT”

Il servizio permette la creazione di un fascicolo.

3.11.1.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “CreateProject Request”.

Struttura dell’oggetto:

Nome	Tipo Dato	Descrizione
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Project	Project	Obbligatorio. Fascicolo che deve

		essere creato.
--	--	----------------

3.11.1.1.1 Campi dell'oggetto Project che possono essere valorizzati

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
CodeNodeClassification	String	Obbligatorio. Nodo del titolare nel quale creare il fascicolo.
ClassificationScheme	String	Obbligatorio. Id del titolare oin cui creare il fascicolo.
Description	String	Obbligatorio. Descrizione del fascicolo.
Template	Template	Opzionale. Tipologia del fascicolo.
Paper	Boolean	Opzionale. Indica se il fascicolo è cartaceo.
CollocationDate	String	Opzionale. Indica la data di collocazione del fascicolo.
PhysicsCollocation	String	Opzionale. Indica la collocazione fisica del fascicolo (inserire il system id).
Private	Boolean	Opzionale. Se true indica che il fascicolo è privato.
Note	Note	Opzionale. Nota del fascicolo.

3.11.1.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "CreateProjectResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Project	Project	Dettaglio del fascicolo creato.

In caso di errore, "CreateProjectResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_PROJECT	Inserire il fascicolo.
REQUIRED_GENERAL_NODE	Richiesto il nodo generale.
ID_CLASSIFICATIONSCHEME_REQUIRED	Richiesto l'id del titolare.
ROLE_NO_EXIST	Il ruolo non esiste.
CLASSIFICATION_NOT_FOUND	Titolario non trovato.
GENERAL_NODE_NOT_FOUND	Nodo generale non trovato.
TEMPLATE_NOT_FOUND	Template non trovato.
TEMPLATE_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del template non consentite al ruolo dell'utente
TEMPLATE_FIELD_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del campo del template non consentite al ruolo dell'utente
ERROR_PROJECT	Errore nella creazione del fascicolo.
APPLICATION_ERROR	Errore generico.

3.11.1.3 Esempio di utilizzo C#

```

Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.CreateProjectRequest request = new Projects.CreateProjectRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

Projects.Project fascRequest = new Projects.Project();

fascRequest.ClassificationScheme = new Projects.ClassificationScheme();
fascRequest.ClassificationScheme.Id = "Id del titolare";
fascRequest.Description = "Descrizione del fascicolo";
//Creazione di una nota
fascRequest.Note = new Projects.Note[1];
fascRequest.Note[0] = new Projects.Note();
fascRequest.Note[0].Description = this.txt_nota.Text;
fascRequest.CodeNodeClassification = "Nodo di classificazione";
request.Project = fascRequest;

Projects.Template template = new Projects.Template();
template.Id = "Id della tipologia del template";
template.Name = "descrizione della tipologia del template";

```

```
List<Projects.Field> campiProfilo = new List<Projects.Field>();
template.Id = "Id del template";
template.Name = "Nome del template";

List< Projects.Field> campiProfilo = new List< Projects.Field>();

//Campo profilato della tipologia di fascicolo che si vuole valorizzare
Projects.Field campo = new Projects.Field();
campo.Name = "Nome del campo profilato";
campo.Value = "Valore del campo profilato";
campiProfilo.Add(campo);
request.Project.Template = template;
request.Project.Template.Fields = campiProfilo.ToArray();

try
{
Projects.CreateProjectResponse response = services.CreateProject(request);
Projects.Project result = response.Project;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.11.1.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
CreateProjectRequest createProjectRequest =
CreateProjectRequest.Factory.newInstance();
createProjectRequest.setCodeAdm("CodiceAmministrazione");
createProjectRequest.setUserName("UserName");
Project project = Project.Factory.newInstance();
ClassificationScheme scheme = ClassificationScheme.Factory.newInstance();
scheme.setId("IdDelTitolario");
project.setClassificationScheme(scheme);
project.setDescription("Descrizione fascicolo");
project.setCodeNodeClassification("Nodo di classificazione");
ArrayOfNote note = ArrayOfNote.Factory.newInstance();
Note nota = Note.Factory.newInstance();
nota.setDescription("Testo Nota");
Note[] arrayNota = new Note[1];
arrayNota[0]= nota;
note.setNoteArray(arrayNota);
project.setNote(note);

//Associo tipologia fascicolo
Template template = Template.Factory.newInstance();
template.setName("NomeDelTemplate");
template.setId("IdDelTemplate");
ArrayOfField fields = ArrayOfField.Factory.newInstance();
Field[] fieldArray = new Field[1];
Field temp = Field.Factory.newInstance();
temp.setName("NomeCampoProfilato");
```

```
temp.setValue("valoreCampoProfilato");
fieldArray[0] = temp;
fields.setFieldArray(fieldArray);
template.setFields(fields);
project.setTemplate(template);
createProjectRequest.setProject(project);
CreateProjectDocument createProjectDocument =
CreateProjectDocument.Factory.newInstance();
CreateProject createProject = CreateProject.Factory.newInstance();
createProject.setRequest(createProjectRequest);
createProjectDocument.setCreateProject(createProject);
CreateProjectResponseDocument createProjectResponseDocument =
CreateProjectResponseDocument.Factory.newInstance();
try
{
createProjectResponseDocument = wws.createProject(createProjectDocument);
Project respProject = null;
respProject =
createProjectResponseDocument.getCreateProjectResponse().getCreateProjectResult().getProject();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.11.2 SERVIZIO “GETTEMPLATESPROJECTS”

Servizio per il reperimento di tutte le tipologie di fascicolo.

3.11.2.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetTemplatesProjectsRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.11.2.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetTemplatesProjectsResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Templates	Template[]	Tipologie fascicoli disponibili.

In caso di errore, “GetTemplatesProjectsResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
TEMPLATES_NOT_FOUND	Template non trovati.
APPLICATION_ERROR	Errore generico.

3.11.2.3 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.GetTemplatesProjectsRequest request = new
Projects.GetTemplatesProjectsRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
    Projects.GetTemplatesProjectsResponse response =
    services.GetTemplatesProjects(request);
    Projects.Template[] result = response.Templates;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.11.2.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
```

```

wvs._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetTemplatesProjectsRequest getTemplatesProjectsRequest =
GetTemplatesProjectsRequest.Factory.newInstance();
getTemplatesProjectsRequest.setCodeAdm("CodiceAmministrazione");
getTemplatesProjectsRequest.setUsername("UserName");
GetTemplatesProjectsDocument getTemplatesProjectsDocument =
GetTemplatesProjectsDocument.Factory.newInstance();
GetTemplatesProjects getTemplatesProjects =
GetTemplatesProjects.Factory.newInstance();
getTemplatesProjects.setRequest(getTemplatesProjectsRequest);
getTemplatesProjectsDocument.setGetTemplatesProjects(getTemplatesProjects);

GetTemplatesProjectsResponseDocument getTemplatesProjectsResponseDocument =
GetTemplatesProjectsResponseDocument.Factory.newInstance();

try
{
getTemplatesProjectsResponseDocument =
wvs.getTemplatesProjects(getTemplatesProjectsDocument);
Template[] templates = null;
templates =
getTemplatesProjectsResponseDocument.getGetTemplatesProjectsResponse().getGetTemplat
esProjectsResult().getTemplates().getTemplateArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.11.3 SERVIZIO “GETTEMPLATEPRJ”

Servizio per il reperimento del dettaglio di una tipologia di fascicolo dato il nome o l'id.

3.11.3.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetTemplatePrjRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
DescriptionTemplate	String	Opzionale. Nome della tipologia

IdTemplate	String	del fascicolo. Opzionale. Id della tipologia del fascicolo.
-------------------	--------	--

3.11.3.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetTemplatePrjResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “GetTemplatePrjResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_ID_TEMPLATE	Richiesto il nome o l'id del template.
REQUIRED_ONLY_CODE_OR_ID_TEMPLATE	Richiesto soltanto un valore tra nome e id del template.
TEMPLATE_NOT_FOUND	Template non trovato.
TEMPLATE_NOT_ROLE_VISIBLE	Visibilità del template non associata al ruolo dell'utente
APPLICATION_ERROR	Errore generico.

3.11.3.3 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.GetTemplatePrjRequest request = new Projects.GetTemplatePrjRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdTemplate = this.txtIdTemplate.Text;
//Invece dell'id request.DescriptionTemplate = this.txtDescTemplate.Text;

try
{
    Projects.GetTemplatePrjResponse response = services.GetTemplatePrj(request);
    Projects.Template result = response.Template;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```


3.11.3.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetTemplatePrjRequest getTemplatePrjRequest =
GetTemplatePrjRequest.Factory.newInstance();
getTemplatePrjRequest.setCodeAdm("CodiceAmministrazione");
getTemplatePrjRequest.setUsername("UserName");
getTemplatePrjRequest.setIdTemplate("IdTemplate");
GetTemplatePrjDocument getTemplatePrjDocument =
GetTemplatePrjDocument.Factory.newInstance();
GetTemplatePrj getTemplatePrj = GetTemplatePrj.Factory.newInstance();
getTemplatePrj.setRequest(getTemplatePrjRequest);
getTemplatePrjDocument.setGetTemplatePrj(getTemplatePrj);
GetTemplatePrjResponseDocument getTemplatePrjResponseDocument =
GetTemplatePrjResponseDocument.Factory.newInstance();
try
{
getTemplatePrjResponseDocument = wws.getTemplatePrj(getTemplatePrjDocument);
Template template = null;
template =
getTemplatePrjResponseDocument.getGetTemplatePrjResponse().getGetTemplatePrjResult()
.getTemplate();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.11.4 SERVIZIO “GETPROJECT”

Servizio per il reperimento dei dati di un fascicolo dato un codice o l'id. Obbligatorio l'id delo fascicolo oppure la coppia codice e id del titolare.

3.11.4.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetProjectRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con

IdProject	String	cui si vuole accedere al sistema.
ClassificationSchemeId	String	Opzionale. Id del fascicolo.
CodeProject	String	Opzionale. Id del titolare.
		Opzionale. Codice del fascicolo.

3.11.4.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetProjectResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Project	Project	Dettaglio del fascicolo.

In caso di errore, “GetProjectResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto il codice o l'id del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto soltanto un valore tra id e codice del fascicolo.
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto codice del fascicolo e id del titolare.
PROJECT_NOT_FOUND	Fascicolo non trovato.
APPLICATION_ERROR	Errore generico.

3.11.4.3 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.GetProjectRequest request = new Projects.GetProjectRequest();
```

```
request.UserName = “UserName”;
```

```
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Opzionalmente invece dell'id del fascicolo possibile inserire codice fascicolo e id
//del titolare
//request.CodeProject = "Codice del fascicolo";
//request.ClassificationSchemeId = "Id del titolare";

try
{
    Projects.GetProjectResponse response = services.GetProject(request);
    Projects.Project result = response.Project;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.11.4.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetProjectRequest getProjectRequest = GetProjectRequest.Factory.newInstance();
getProjectRequest.setCodeAdm("CodiceAmministrazione");
getProjectRequest.setUsername("UserName");
getProjectRequest.setIdProject("IdFascicolo");
GetProjectDocument getProjectDocument = GetProjectDocument.Factory.newInstance();
GetProject getProject = GetProject.Factory.newInstance();
getProject.setRequest(getProjectRequest);
getProjectDocument.setGetProject(getProject);
GetProjectResponseDocument getProjectResponseDocument =
GetProjectResponseDocument.Factory.newInstance();
try
{
    getProjectResponseDocument = wws.getProject(getProjectDocument);
    Project project = null;
    project =
getProjectResponseDocument.getGetProjectResponse().getGetProjectResult().getProject(
);
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.11.5 SERVIZIO “EDITPROJECT”

Servizio per la modifica di un fascicolo.

3.11.5.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “EditProjectRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Project	Project	Obbligatorio. Fascicolo da modificare.

3.11.5.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “EditProjectResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Project	Project	Dettaglio del fascicolo modificato.

In caso di errore, “EditProjectResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_PROJECT	Fascicolo obbligatorio.
TEMPLATE_NOT_FOUND	Template non trovato.
TEMPLATE_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del template non consentite al ruolo dell'utente

TEMPLATE_FIELD_NOT_ROLE_EDITABLE	Operazioni di Inserimento o modifica del campo del template non consentite al ruolo dell'utente
PROJECT_NOT_FOUND	Fascicolo non trovato.
APPLICATION_ERROR	Errore generico.

3.11.5.3 Esempio di utilizzo C#

```

Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.EditProjectRequest request = new Projects.EditProjectRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

Projects.Project fascRequest = new Projects.Project();
fascRequest.Description = this.txt_descrizione.Text;
fascRequest.Id = "Id del fascicolo";

request.Project = fascRequest;

try
{
    Projects.EditProjectResponse response = services.EditProject(request);
    Projects.Project result = response.Project;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.11.5.4 Esempio di utilizzo Java

```

ProjectsStub wws = new ProjectsStub("http://localhost/Vt-
Docs/DocsPaWS/VtDocsWS/WebServices/Projects.asmx");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
EditProjectRequest editProjectRequest = EditProjectRequest.Factory.newInstance();
editProjectRequest.setCodeAdm("PAT_PRE");
editProjectRequest.setUserName("PR29058");
Project project = Project.Factory.newInstance();
project.setId("8573066");
project.setDescription("Descrizione modificata");
editProjectRequest.setProject(project);
EditProjectDocument editProjectDocument = EditProjectDocument.Factory.newInstance();
EditProject editProject = EditProject.Factory.newInstance();
editProject.setRequest(editProjectRequest);
editProjectDocument.setEditProject(editProject);
EditProjectResponseDocument editProjectResponseDocument =
EditProjectResponseDocument.Factory.newInstance();
try
{
    editProjectResponseDocument = wws.editProject(editProjectDocument);
}

```

```
Project respProject = null;
respProject =
editProjectResponseDocument.getEditProjectResponse().getEditProjectResult().getProject();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.11.6 SERVIZIO “GETPROJECTSBYDOCUMENT”

Servizio per il reperimento dei fascicoli in cui è fascicolato un documento. Obbligatorio l'id o la segnatura del documento.

3.11.6.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetProjectsByDocumentRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Opzionale. Id del documento.
Signature	String	Opzionale. Segnatura del documento

3.11.6.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetProjectsByDocumentResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “GetProjectsByDocumentResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.

USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_OR_SIGNATURE	Richiesto id o segnatura del documento.
REQUIRED_ONLY_ID_OR_SIGNATURE	Richiesto solo un valore tra id o segnatura del documento.
DOCUMENT_NOT_FOUND	Documento non trovato.
PROJECTS_NOT_FOUND	Fascicoli non trovati.
APPLICATION_ERROR	Errore generico.

3.11.6.3 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();  
Projects.GetProjectsByDocumentRequest request = new  
Projects.GetProjectsByDocumentRequest();
```

```
request.UserName = "UserName";  
request.CodeAdm = "Codice amministrazione";
```

```
try  
{  
Projects.GetProjectsByDocumentResponse response =  
services.GetProjectsByDocument(request);  
Projects.Project[] result = response.Projects;  
}  
catch (System.ServiceModel.FaultException e)  
{  
    //e.Code;  
    //e.Message;  
}
```

3.11.6.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");  
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);  
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V  
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);  
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE  
_NAMESPACE_URI);  
GetProjectsByDocumentRequest getProjectsByDocumentRequest =  
GetProjectsByDocumentRequest.Factory.newInstance();  
getProjectsByDocumentRequest.setCodeAdm("CodiceAmministrazione");  
getProjectsByDocumentRequest.setUserName("UserName");  
getProjectsByDocumentRequest.setIdDocument("IdDocumento");  
GetProjectsByDocumentDocument1 getProjectsByDocumentDocument =  
GetProjectsByDocumentDocument1.Factory.newInstance();  
GetProjectsByDocument getProjectsByDocument =  
GetProjectsByDocument.Factory.newInstance();  
getProjectsByDocument.setRequest(getProjectsByDocumentRequest);  
getProjectsByDocumentDocument.setGetProjectsByDocument(getProjectsByDocument);  
GetProjectsByDocumentResponseDocument getProjectsByDocumentResponseDocument =  
GetProjectsByDocumentResponseDocument.Factory.newInstance();  
try  
{
```

```
getProjectsByDocumentResponseDocument =
wvs.getProjectsByDocument(getProjectsByDocumentDocument);
Project[] projects = null;
projects =
getProjectsByDocumentResponseDocument.getGetProjectsByDocumentResponse().getGetProjectsByDocumentResult().getProjects().getProjectArray();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.11.7 SERVIZIO “EDITPRJSTATEDIAGRAM”

Servizio per modificare lo stato del diagramma associato ad un fascicolo. Obbligatorio un valore tra id del fascicolo oppure codice e id del titolare.

3.11.7.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “EditPrjStateDiagramRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdProject	String	Opzionale. Id del fascicolo.
ClassificationSchemeId	String	Opzionale. Id del titolare.
CodeProject	String	Opzionale. Codice del fascicolo.
StateOfDiagram	String	Obbligatorio. Nome dello stato del fascicolo.

3.11.7.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “EditPrjStateDiagramResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “EditPrjStateDiagramResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto id o codice del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto solo un valore tra id o codice del fascicolo.
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto codice del fascicolo e id del titolare.
REQUIRED_STATE_OF_DIAGRAM	Richiesto lo stato del diagramma.
PROJECT_NOT_FOUND	Fascicolo non trovato.
STATEOFDIAGRAM_NOT_FOUND	Stato del diagramma non trovato.
DIAGRAM_NOT_FOUND	Diagramma non trovato.
TEMPLATE_NOT_FOUND	Template non trovato.
APPLICATION_ERROR	Errore generico.

3.11.7.3 Esempio di utilizzo C#

```

Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.EditPrjStateDiagramRequest request = new
Projects.EditPrjStateDiagramRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Opzionale invece dell'id del fascicolo il codice e l'id del titolare
//request.ClassificationSchemeId = "Id dell'id del titolare";
//request.CodeProject = "Codice del fascicolo";
request.StateOfDiagram = "Nome dello stato del fascicolo";

try
{
Projects.EditPrjStateDiagramResponse response = services.EditPrjStateDiagram(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.11.7.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebService");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
EditPrjStateDiagramRequest editPrjStateDiagramRequest =
EditPrjStateDiagramRequest.Factory.newInstance();
editPrjStateDiagramRequest.setCodeAdm("CodiceAmministrazione");
editPrjStateDiagramRequest.setUsername("UserName");
editPrjStateDiagramRequest.setIdProject("IdFascicolo");
editPrjStateDiagramRequest.setStateOfDiagram("StatoFascicolo");
EditPrjStateDiagramDocument editPrjStateDiagramDocument =
EditPrjStateDiagramDocument.Factory.newInstance();
EditPrjStateDiagram editPrjStateDiagram = EditPrjStateDiagram.Factory.newInstance();
editPrjStateDiagram.setRequest(editPrjStateDiagramRequest);
editPrjStateDiagramDocument.setEditPrjStateDiagram(editPrjStateDiagram);
EditPrjStateDiagramResponseDocument editPrjStateDiagramResponseDocument =
EditPrjStateDiagramResponseDocument.Factory.newInstance();
try
{
editPrjStateDiagramResponseDocument =
wws.editPrjStateDiagram(editPrjStateDiagramDocument);
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.11.8 SERVIZIO “GETPROJECTSTATEDIAGRAM”

Servizio per il reperimento dello stato di un fascicolo. Obbligatorio un valore tra id del fascicolo oppure codice e id del titolare.

3.11.8.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetProjectStateDiagramRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.

CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdProject	String	Opzionale. Id del fascicolo.
ClassificationSchemeId	String	Opzionale. Id del titolare.
CodeProject	String	Opzionale. Codice del fascicolo.

3.11.8.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetProjectStateDiagramResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
StateOfDiagram	StateOfDiagram	Stato del fascicolo.

In caso di errore, “GetProjectStateDiagramResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto id o codice del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto solo un valore tra id o codice del fascicolo.
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto codice del fascicolo e id del titolare.
REQUIRED_STATE_OF_DIAGRAM	Richiesto lo stato del diagramma.
PROJECT_NOT_FOUND	Fascicolo non trovato.
FASC_STATEOFDIAGRAM_NOT_FOUND	Stato del diagramma non trovato.
DIAGRAM_NOT_FOUND	Diagramma di stato non trovato.

TEMPLATE_NOT_FOUND	Template non trovato.
APPLICATION_ERROR	Errore generico.

3.11.8.3 Esempio di utilizzo C#

```

Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.GetProjectStateDiagramRequest request = new
Projects.GetProjectStateDiagramRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Opzionale invece dell'id del fascicolo il codice e l'id del titolare
//request.ClassificationSchemeId = "Id dell'id del titolare";
//request.CodeProject = "Codice del fascicolo";

try
{
    Projects.GetProjectStateDiagramResponse response =
    services.GetProjectStateDiagram(request);
    Projects.StateOfDiagram result = response.StateOfDiagram;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.11.8.4 Esempio di utilizzo Java

```

ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetProjectStateDiagramRequest getProjectStateDiagramRequest =
GetProjectStateDiagramRequest.Factory.newInstance();
getProjectStateDiagramRequest.setCodeAdm("CodiceAmm");
getProjectStateDiagramRequest.setUserName("UserName");
getProjectStateDiagramRequest.setIdProject("IdFascicolo");
GetProjectStateDiagramDocument getProjectStateDiagramDocument =
GetProjectStateDiagramDocument.Factory.newInstance();
GetProjectStateDiagram getProjectStateDiagram =
GetProjectStateDiagram.Factory.newInstance();
getProjectStateDiagram.setRequest(getProjectStateDiagramRequest);
getProjectStateDiagramDocument.setGetProjectStateDiagram(getProjectStateDiagram);
GetProjectStateDiagramResponseDocument getProjectStateDiagramResponseDocument =
GetProjectStateDiagramResponseDocument.Factory.newInstance();
try
{
    getProjectStateDiagramResponseDocument =
    wws.getProjectStateDiagram(getProjectStateDiagramDocument);
    StateOfDiagram state = null;
}

```

```
state =
getProjectStateDiagramResponseDocument.getGetProjectStateDiagramResponse().getGetPro
jectStateDiagramResult().getStateOfDiagram();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.11.9 SERVIZIO “OPENCLOSEPROJECT”

Servizio per chiudere/aprire un fascicolo. Obbligatorio un valore tra id del fascicolo oppure codice e id del titolare.

3.11.9.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “OpenCloseProjectRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdProject	String	Opzionale. Id del fascicolo.
ClassificationSchemeId	String	Opzionale. Id del titolare.
CodeProject	String	Opzionale. Codice del fascicolo.
Action	String	Obbligatorio. “O” per aprire oppure “C” per chiudere il fascicolo.

3.11.9.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “OpenCloseProjectResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “OpenCloseProjectResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto id o codice del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto solo un valore tra id o codice del fascicolo.
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto codice del fascicolo e id del titolare.
REQUIRED_OC_PROJECT	Richiesto O oppure C per aprire/chudere il fascicolo.
PROJECT_NOT_FOUND	Fascicolo non trovato.
PROJECT_ALREADY_CLOSE	Fascicolo già chiuso.
PROJECT_ALREADY_OPEN	Fascicolo già aperto.
APPLICATION_ERROR	Errore generico.

3.11.9.3 Esempio di utilizzo C#

```

Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.OpenCloseProjectRequest request = new Projects.OpenCloseProjectRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Opzionale invece dell'id del fascicolo il codice e l'id del titolare
//request.ClassificationSchemeId = "Id dell'id del titolare";
//request.CodeProject = "Codice del fascicolo";

request.Action = "C"; //C per chiudere e O per aprire il fascicolo

try
{
    Projects.OpenCloseProjectResponse response = services.OpenCloseProject(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.11.9.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebService");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
OpenCloseProjectRequest openCloseProjectRequest =
OpenCloseProjectRequest.Factory.newInstance();
openCloseProjectRequest.setCodeAdm("PAT_PRE");
openCloseProjectRequest.setUsername("PR29058");
openCloseProjectRequest.setAction("O"); //C per chiudere
openCloseProjectRequest.setIdProject("8573066");
OpenCloseProjectDocument openCloseProjectDocument =
OpenCloseProjectDocument.Factory.newInstance();
OpenCloseProject openCloseProject = OpenCloseProject.Factory.newInstance();
openCloseProject.setRequest(openCloseProjectRequest);
openCloseProjectDocument.setOpenCloseProject(openCloseProject);

OpenCloseProjectResponseDocument openCloseProjectResponseDocument =
OpenCloseProjectResponseDocument.Factory.newInstance();
try
{
openCloseProjectResponseDocument = wws.openCloseProject(openCloseProjectDocument);
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.11.10 SERVIZIO “SEARCHPROJECTS”

Servizio che permette la ricerca di fascicoli.

3.11.10.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “SearchProjectsRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con

Filters	Filter[]	cui si vuole accedere al sistema.
PageNumber	String	Oblbigatorio. Filtri di ricerca.
ElementsInPage	String	Opzionale. Numero di pagina.
GetTotalProjectsNumber	bool	Opzionale. Se true la relativa response conterrà il numero totale di fascicoli ottenuti nella ricerca.

3.11.10.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “SearchProjectsResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Projects	Project[]	Lista dei fascicoli trovati.
TotalProjectsNumber	int	Numero totale dei fascicoli trovati.

Nel caso in cui la ricerca non produca risultati, verrà restituita la lista dei fascicoli vuota.

In caso di errore, “SearchProjectsResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_FILTER	Richiesto almeno un filtro di ricerca.
INVALID_PAGINATION	Paginazione non valida.
APPLICATION_ERROR	Errore generico / Formato (Number/ Bool/ Date) non corretto.

3.11.10.3 Filtri disponibili

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
YEAR	Filter	Filtro per inserire l’anno.

CREATION_DATE_FROM	Filter	Filtro utilizzato per intervallo su data creazione, questo valore è il limite inferiore.
CREATION_DATE_TO	Filter	Filtro utilizzato per intervallo su data creazione, questo valore è il limite superiore.
CLOSING_DATE_FROM	Filter	Filtro utilizzato per intervallo su data chiusura, questo valore è il limite inferiore.
CLOSING_DATE_TO	Filter	Filtro utilizzato per intervallo su data chiusura, questo valore è il limite superiore.
OPENING_DATE_FROM	Filter	Filtro utilizzato per intervallo su data apertura, questo valore è il limite inferiore.
OPENING_DATE_TO	Filter	Filtro utilizzato per intervallo su data apertura, questo valore è il limite superiore.
STATE	Filter	Filtro utilizzato per cercare il fascicolo in base allo stato di chiusura/apertura. Il valore del filtro è "O" per aperto e "C" per chiuso.
TYPE_PROJECT	Filter	Filtro utilizzato per cercare il fascicolo in base al tipo procedimentale/generale. Inserire "P" per procedimentale, "G" per generale.
PROJECT_NUMBER	Filter	Filtro utilizzato per cercare il fascicolo per numero.
PROJECT_DESCRIPTION	Filter	Filtro utilizzato per cercare i fascicoli per descrizione.
TEMPLATE	Filter	Filtro che permette la ricerca per tipologia di fascicolo.
PROJECT_CODE	Filter	Filtro che permette la ricerca per codice di fascicolo.
CLASSIFICATION_SCHEME	Filter	Filtro che permette la ricerca per titolare.
REGISTER	Filter	Filtro per la ricerca dei fascicoli in registro.

CLASSIFICATION_CODE	Filter	Filtro per cercare per codice di classificazione.
SUBPROJECT	Filter	Filtro per cercare un fascicolo data la descrizione di un suo sottofascicolo.
TEMPLATE_EXTRACTION	Filter	Filtro utilizzato per l'estrazione dei campi profilati nella ricerca.

3.11.10.4 Estrazione dei campi profilati nella ricerca

Se si presentasse la necessità di prelevare i valori dei campi di una tipologia da una ricerca, il filtro **TEMPLATE_EXTRACTION** permette l'estrazione dei campi profilati.

Per utilizzare questo filtro è necessario effettuare una ricerca inserendo un filtro **TEMPLATE**, e settando il filtro **TEMPLATE_EXTRACTION** a **TRUE**. Nel caso non siano verificate le condizioni, il filtro viene ignorato anche se presente.

3.11.10.5 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.SearchProjectsRequest request = new Projects.SearchProjectsRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.Filters = new Projects.Filter[3];
int indice = 0;
Projects.Filter filter = new Projects.Filter();
filter.Name = "YEAR";
filter.Value = "Anno";
filter.Type = Projects.FilterType.Number;
request.Filters.SetValue(filter, indice);
indice++;
filter = new Projects.Filter();
filter.Name = "CREATION_DATE_FROM";
filter.Value = "Data di creazione limite inferiore";
filter.Type = Projects.FilterType.Date;
request.Filters.SetValue(filter, indice);
indice++;
filter = new Projects.Filter();
filter.Name = "CREATION_DATE_TO";
filter.Type = Projects.FilterType.Date;
filter.Value = "Data di creazione limite superiore";
request.Filters.SetValue(filter, indice);
indice++;

request.GetTotalProjectsNumber = true;

try
{
Projects.SearchProjectsResponse response = services.SearchProjects(request);
Projects.Project[] result = response.Projects;
```

```
}  
catch (System.ServiceModel.FaultException e)  
{  
    //e.Code;  
    //e.Message;  
}
```

3.11.10.6 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");  
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);  
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V  
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);  
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE  
_NAMESPACE_URI);  
SearchProjectsRequest searchProjectsRequest =  
SearchProjectsRequest.Factory.newInstance();  
searchProjectsRequest.setCodeAdm("CodiceAmministrazione");  
searchProjectsRequest.setUsername("UserName");  
searchProjectsRequest.setGetTotalProjectsNumber(true);  
ArrayOfFilter arrFilters = ArrayOfFilter.Factory.newInstance();  
Filter[] filters = new Filter[3];  
int indice = 0;  
Filter filter = Filter.Factory.newInstance();  
filter.setName("YEAR");  
filter.setValue("2012");  
filter.setType(FilterType.Number);  
filters[indice] = filter;  
indice++;  
filter = Filter.Factory.newInstance();  
filter.setName("CREATION_DATE_FROM");  
filter.setValue("01/01/2012");  
filter.setType(FilterType.Date);  
filters[indice] = filter;  
indice++;  
filter = Filter.Factory.newInstance();  
filter.setName("CREATION_DATE_TO");  
filter.setValue("01/06/2012");  
filter.setType(FilterType.Date);  
filters[indice] = filter;  
indice++;  
arrFilters.setFilterArray(filters);  
searchProjectsRequest.setFilters(arrFilters);  
SearchProjectsDocument searchProjectsDocument =  
SearchProjectsDocument.Factory.newInstance();  
SearchProjects searchProjects = SearchProjects.Factory.newInstance();  
searchProjects.setRequest(searchProjectsRequest);  
searchProjectsDocument.setSearchProjects(searchProjects);  
SearchProjectsResponseDocument searchProjectsResponseDocument =  
SearchProjectsResponseDocument.Factory.newInstance();  
try  
{  
    searchProjectsResponseDocument = wws.searchProjects(searchProjectsDocument);  
    Project[] projects = null;  
    projects =  
searchProjectsResponseDocument.getSearchProjectsResponse().getSearchProjectsResult()  
    .getProjects().getProjectArray();  
}  
catch (RemoteException e)  
{  
    //e.getMessage();  
    //e.printStackTrace();  
}
```

}

3.11.11 SERVIZIO “GETPROJECTFILTERS”

Servizio che restituisce la lista dei filtri applicabili alla ricerca dei fascicoli.

3.11.11.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetProjectFiltersRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.11.11.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetProjectFiltersResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Filters	Filter[]	Lista dei filtri applicabili.

In caso di errore, “GetProjectFiltersResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.11.11.3 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.GetProjectFiltersRequest request = new Projects.GetProjectFiltersRequest();

request.UserName = this.txt_username.Text;
request.CodeAdm = this.txt_codAmm.Text;
request.CodeRoleLogin = this.txt_codice_ruolo.Text;

this.txtRequest.Text = this.SerializeAsXml(request);

try
{
    Projects.GetProjectFiltersResponse response = services.GetProjectFilters(request);
    this.txtResponse.Text = this.SerializeAsXml(response);
    Projects.Filter[] result = response.Filters;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.11.11.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetProjectFiltersRequest getProjectFiltersRequest=
GetProjectFiltersRequest.Factory.newInstance();
getProjectFiltersRequest.setCodeAdm("CodiceAmministrazione");
getProjectFiltersRequest.setUserName("UserName");
GetProjectFiltersDocument getProjectFiltersDocument =
GetProjectFiltersDocument.Factory.newInstance();
GetProjectFilters getProjectFilters = GetProjectFilters.Factory.newInstance();
getProjectFilters.setRequest(getProjectFiltersRequest);
getProjectFiltersDocument.setProjectFilters(getProjectFilters);
GetProjectFiltersResponseDocument getProjectFiltersResponseDocument =
GetProjectFiltersResponseDocument.Factory.newInstance();
try
{
    getProjectFiltersResponseDocument = wws.getProjectFilters
(getProjectFiltersDocument);
    Filter[] filters = null;
    filters =
getProjectFiltersResponseDocument.getGetProjectFiltersResponse().ProjectFiltersResul
t().getFilters();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.11.12 SERVIZIO “CREATEFOLDER”

Servizio per la creazione di un sottofascicolo.

3.11.12.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “CreateFolderRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdProject	String	Opzionale. Id del fascicolo nel quale si vuole creare il sottofascicolo. Uno e solo uno tra IdProject e CodeProject è un parametro obbligatorio.
CodeProject	String	Opzionale. Codice del fascicolo nel quale si vuole creare il sottofascicolo. Uno e solo uno tra IdProject e CodeProject è un parametro obbligatorio.
ClassificationSchemeId	String	Opzionale. Id del titolare. Diviene obbligatorio in caso di immissione di CodeProject.
FolderDescription	String	Obbligatorio. Descrizione del sottofascicolo.
IdParentFolder	String	Opzionale. Id del fascicolo/sottofascicolo padre. Necessario per la creazione di un sottofascicolo non di primo livello.

3.11.12.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “CreateFolderResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Result	String	Stringa contenente l’esito della creazione.

In caso di errore, “CreateFolderResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
PROJECT_NOT_FOUND	Fascicolo non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto codice o id del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto solo uno tra il codice o l’id del fascicolo
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto l’id del titolare quando viene inserito il codice del fascicolo.
APPLICATION_ERROR	Errore generico.

3.11.12.3 Esempio di utilizzo C#

```
Projects.ProjectsClient services = new Projects.ProjectsClient();  
Projects.CreateFolderRequest request = new Projects.CreateFolderRequest();  
  
request.UserName = "UserName";  
request.CodeAdm = "Codice amministrazione";  
request.IdProject= "Id del fascicolo";  
request.FolderDescription = "Descrizione del sottofascicolo";  
try  
{  
Projects.CreateFolderResponse response = services.CreateFolder(request);
```

```

}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.11.12.4 Esempio di utilizzo Java

```

ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
CreateFolderRequest createFolderRequest = CreateFolderRequest.Factory.newInstance();
createFolderRequest.setCodeAdm("CodiceAmministrazione");
createFolderRequest.setUserName("UserName");
createFolderRequest.setIdProject("Id del fascicolo");
createFolderRequest.setFolderDescription("Descrizione del sottofascicolo");
CreateFolderDocument createFolderDocument =
CreateFolderDocument.Factory.newInstance();
CreateFolder createFolder = CreateFolder.Factory.newInstance();
createFolder.setRequest(createFolderRequest);
createFolderDocument.setCreateFolder(createFolder);
CreateFolderResponseDocument createFolderResponseDocument =
CreateFolderResponseDocument.Factory.newInstance();
try
{
    createFolderResponseDocument = wws.createFolder(createFolderDocument);
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}

```

3.11.13 SERVIZIO “GETPROJECTFOLDERS”

Servizio per il prelievo dei sottofascicoli all'interno di un fascicolo.

3.11.13.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetProjectFoldersRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.

CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdProject	String	Opzionale. Id del fascicolo nel quale si vuole creare il sottofascicolo. Uno e solo uno tra IdProject e CodeProject è un parametro obbligatorio.
CodeProject	String	Opzionale. Codice del fascicolo nel quale si vuole creare il sottofascicolo. Uno e solo uno tra IdProject e CodeProject è un parametro obbligatorio.
ClassificationSchemeId	String	Opzionale. Id del titolare. Diviene obbligatorio in caso di immissione di CodeProject.

3.11.13.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "GetProjectFoldersResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
IdProject	String	Id del fascicolo
ProjectDescription	String	Descrizione del fascicolo
Folders	Folder[]	Lista dei sottofascicoli.

In caso di errore, "GetProjectFoldersResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.

USER_NO_EXIST	Utente non trovato.
PROJECT_NOT_FOUND	Fascicolo non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto codice o id del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiesto solo uno tra il codice o l'id del fascicolo
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto l'id del titolare quando viene inserito il codice del fascicolo.
APPLICATION_ERROR	Errore generico.

3.11.13.3 Esempio di utilizzo C#

```

Projects.ProjectsClient services = new Projects.ProjectsClient();
Projects.GetProjectFoldersRequest request = new Projects.GetProjectFoldersRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
//Opzionalmente invece dell'id del fascicolo possibile inserire codice fascicolo e id
//del titolare
//request.CodeProject = "Codice del fascicolo";
//request.ClassificationSchemeId = "Id del titolare";

try
{
Projects.GetProjectFoldersResponse response = services.GetProjectFolders(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.11.13.4 Esempio di utilizzo Java

```

ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetProjectFoldersRequest getProjectFoldersRequest =
GetProjectFoldersRequest.Factory.newInstance();
getProjectFoldersRequest.setCodeAdm("CodiceAmministrazione");
getProjectFoldersRequest.setUserName("UserName");
getProjectFoldersRequest.setIdProject("IdFascicolo");

```

```

GetProjectFoldersDocument getProjectFoldersDocument =
GetProjectFoldersDocument.Factory.newInstance();
GetProjectFolders getProjectFolders = GetProjectFolders.Factory.newInstance();
getProjectFolders.setRequest(getProjectFoldersRequest);
getProjectFoldersDocument.setGetProjectFolders (getProjectFolders);
GetProjectFoldersResponseDocument getProjectFoldersResponseDocument =
GetProjectFoldersResponseDocument.Factory.newInstance();
try
{
getProjectFoldersResponseDocument =
wvs.getProjectFolders(getProjectFoldersDocument);
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.11.14 SERVIZIO “FOLLOWPROJECT”

Servizio per la richiesta di “monitorare”/“non monitorare più” un fascicolo presente nel Sistema.

3.11.14.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “FollowRequest”.

Struttura dell’oggetto in comune per tutti i tipi di fascicolo:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdObject	String	Obbligatorio. Id dell’oggetto di dominio che si intende “monitorare”, “non monitorare più”. Nello specifico si tratta dell’id fascicolo.
Operation	OperationFollow	Obbligatorio. Specifica il tipo di azione che si intende eseguire sul fascicolo: AddFolder = “monitora il fascicolo”, RemoveFolder= “non monitorare più il fascicolo”.
CodeApplication	String	Obbligatorio. Codice

		dell'applicazione che ha richiesto il monitoraggio.
--	--	---

3.11.14.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "FollowResponse".
Estende la classe base astratta "Response".

In caso di errore, "FollowResponse" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
INVALID_TOKEN	Token non valido.
USER_NO_EXIST	Utente non trovato.
ROLE_NO_EXIST	Il ruolo non esiste
USER_NO_ROLE	L'utente non ha ruoli
APPLICATION_ERROR	Eccezione Generica: APPLICATION_ERROR I campi Operation, CodeApplication, IdObject sono obbligatori.
APPLICATION_ERROR	Eccezione Generica: APPLICATION_ERROR Il campo operation deve essere AddDocument o RemoveDocument.

3.11.14.3 Esempio di utilizzo C#

```
Projects.ProjectsClient service = new Projects.ProjectsClient();
Documents.FollowRequest request = new Documents.FollowRequest();
request.CodeApplication = "GDOC";
request.UserName = "PR41175";
request.CodeRoleLogin = "S147SEGP";
request.CodeAdm = "PAT_COLL";
request.Operation = Documents.OperationFollowDomainObject.RemoveFolder;
//or
//request.Operation = Documents.OperationFollowDomainObject.AddFolder;
request.IdObject = " 14086810";
try
{
    Projects.FollowResponse response = service.FollowProject(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
```

```

        //e.Message;
    }
    catch (Exception exc)
    {
        //exc.Message;
    }

```

3.11.14.4 Esempio di utilizzo Java

```

DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
FollowRequest followRequest = FollowRequest.Factory.newInstance();
followRequest.setCodeApplication("CodiceApplicazione");
followRequest.setUsername("UserName");
followRequest.setCodeRoleLogin("CodeRoleLogin");
followRequest.setCodeAdm("CodeAdministration");
followRequest.setIdObject("IdObject");
followRequest.setOperation(OperationFollowDomainObject.DeleteFolder);
FollowResponse followResponse = FollowResponse.Factory.newInstance();
try
{
    followResponse = wws.FollowProject(followRequest);
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}

```

3.11.15 SERVIZIO “GETLINKPRJBYID”

Servizio per il recupero di un URL per l’apertura diretta del documento (previa autenticazione all’interfaccia PITRE)

3.11.15.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “GetLinkPrjByIdRequest”.

Struttura dell’oggetto in comune per tutti i tipi di documento:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell’amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con

IdProject	String	cui si vuole accedere al sistema.
ClassificationSchemeID	String	Opzionale. Id del fascicolo.
CodeProject	String	Opzionale. Id del Titolare.
		Opzionale. Codice del fascicolo.

3.11.15.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetLinkPrjByIdResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ProjectLink	String	URL diretta per il fascicolo.

In caso di errore, “GetLinkPrjResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CODE_OR_IDPROJECT	Richiesto id o codice del fascicolo.
REQUIRED_ONLY_IDPROJECT_OR_CODEPROJECT	Richiedo solo uno tra id e codice del fascicolo
REQUIRED_CODEPROJECT_AND_CLASSIFICATION	Richiesto id del titolare (obbligatorio quando si utilizza il codice del fascicolo)
PROJECT_NOT_FOUND	Fascicolo non trovato.
APPLICATION_ERROR	Errore generico.

3.11.15.3 Esempio di utilizzo C#

```
Documents.DocumentsClient services = new Documents.DocumentsClient();
Documents.GetLinkPrjByIdRequest request = new Documents.GetLinkPrjByIdRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
```

```
try
{
Documents. GetLinkPrjByIdResponse response = services. GetLinkPrjById(request);
string result = response.ProjectLink;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.11.15.4 Esempio di utilizzo Java

```
ProjectsStub wws = new ProjectsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetLinkPrjByIdRequest getLinkPrjByIdRequest =
GetLinkPrjByIdRequest.Factory.newInstance();
getLinkPrjByIdRequest.setCodeAdm("CodiceAmministrazione");
getLinkPrjByIdRequest.setUsername("UserName");
getLinkPrjByIdRequest.setIdProject("IdFascicolo");
GetLinkPrjById getLinkPrjById= GetLinkPrjById.Factory.newInstance();
getLinkPrjById.setRequest(getLinkPrjByIdRequest);
GetLinkPrjByIdResponseDocument getLinkPrjByIdResponseDocument =
GetLinkPrjByIdResponseDocument.Factory.newInstance();
try
{
getLinkPrjByIdResponseDocument = wws. getLinkPrjById (getLinkPrjByIdRequest);
string returnvalue = getLinkPrjByIdResponseDocument.
getLinkPrjByIdResponse().getProjectLink();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.12 SERVIZI PER LA GESTIONE DELLA RUBRICA

3.12.1 SERVIZIO “ADDCORRESPONDENT”

Servizio per la creazione di un nuovo corrispondente esterno.

3.12.1.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “AddCorrespondentRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Correspondent	Correspondent	Obbligatorio. Nuovo corrispondente. Vedere le tabelle successive per le varie tipologie di corrispondente

3.12.1.1.1 Creare un corrispondente esterno

Struttura dell'oggetto Correspondent.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Description	String	Obbligatorio. Descrizione del corrispondente.
Code	String	Obbligatorio. Codice del corrispondente ricercabile da rubrica.
CorrespondentType	String	Obbligatorio. Tipologia del corrispondente. Può assumere i valori "U" nel caso di Unità organizzativa e "P" nel caso di persona.
City	String	Opzionale. Città del corrispondente.
Province	String	Opzionale. Provincia del corrispondente.
Location	String	Opzionale. Località del corrispondente.
Nation	String	Opzionale. Nazione del corrispondente.
PhoneNumber	String	Opzionale. Numero di telefono del corrispondente.

Fax	String	Opzionale. Numero del fax del corrispondente.
NationalIdentificationNumber	String	Opzionale. Codice fiscale del corrispondente.
Email	String	Opzionale. Mail del corrispondente.
AOOCode	String	Opzionale. Codice AOO.
AdmCode	String	Opzionale. Codice amministrazione.
Note	String	Opzionale. Note
Address	String	Opzionale. Indirizzo del corrispondente.
Cap	String	Opzionale. Cap del corrispondente.
CodeRegisterOrRF	String	Opzionale. Se il corrispondente deve essere disponibile soltanto per un Registro/RF valorizzare con il codice del Registro/RF.
PreferredChannel	String	Opzionale. Canale preferenziale del corrispondente.

3.12.1.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “AddCorrespondentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

Nome	Tipo Dato	Descrizione
Correspondent	Correspondent	Dettaglio del corrispondente creato.

In caso di errore, “AddCorrespondentResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CORRESPONDENT	Richiesto corrispondente.
CODE_CORRESPONDENT_EXISTS	Il codice del corrispondente già esiste.
REQUIRED_DESCRIPTION_CORREPENDENT	Richiesta la descrizione del corrispondente.
REQUIRED_N_S_CORREPENDENT	Richiesto il nome e cognome del corrispondente.
APPLICATION_ERROR	Errore generico.

3.12.1.3 Esempio di utilizzo C#

```

AddressBook.AddressBook services = new AddressBook.AddressBook();
services.Timeout = System.Threading.Timeout.Infinite;
AddressBook.AddCorrespondentRequest request = new
AddressBook.AddCorrespondentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.Correspondent = new AddressBook.Correspondent();
request.Correspondent.Address = "Indirizzo";
request.Correspondent.Code = "Codice rubrica";
request.Correspondent.CodeRegisterOrRF = "PAT"; //Registro o RF
request.Correspondent.CorrespondentType = "P"; //tipo di corrispondente
request.Correspondent.Name = "Nome";
request.Correspondent.Surname = "Cognome";
request.Correspondent.Email = "Mail";
request.Correspondent.NationalIdentificationNumber = "Codice fiscale";

try
{
AddressBook.AddCorrespondentResponse response = services.AddCorrespondent(request);
AddressBook.Correspondent result = response.Correspondent;
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}

```

3.12.1.4 Esempio di utilizzo Java

```

AddressBookStub wws = new AddressBookStub("linkWebServices");

```

```

wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
AddCorrespondentRequest addCorrespondentRequest =
AddCorrespondentRequest.Factory.newInstance();
addCorrespondentRequest.setCodeAdm("CodiceAmministrazione");
addCorrespondentRequest.setUsername("UserName");
Correspondent correspondent = Correspondent.Factory.newInstance();
correspondent.setAddress("Indirizzo");
correspondent.setDescription("Descrizione");
correspondent.setEmail("Mail");
correspondent.setCode("CodiceRubrica");
correspondent.setNationalIdentificationNumber("CodiceFiscale");
correspondent.setCorrespondentType("U");//U per UO o P per persona
addCorrespondentRequest.setCorrespondent(correspondent);
AddCorrespondentDocument addCorrespondentDocument =
AddCorrespondentDocument.Factory.newInstance();
AddCorrespondent addCorrespondent = AddCorrespondent.Factory.newInstance();
addCorrespondent.setRequest(addCorrespondentRequest);
addCorrespondentDocument.setAddCorrespondent(addCorrespondent);
AddCorrespondentResponseDocument addCorrespondentResponseDocument =
AddCorrespondentResponseDocument.Factory.newInstance();

try
{
addCorrespondentResponseDocument = wws.addCorrespondent(addCorrespondentDocument);
Correspondent result = null;
result =
addCorrespondentResponseDocument.getAddCorrespondentResponse().getAddCorrespondentR
esult().getCorrespondent();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.12.2 SERVIZIO “DELETECORRESPONDENT”

Servizio che permette la cancellazione di un corrispondente esterno.

3.12.2.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “DeleteCorrespondentRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice

CodeRoleLogin	String	dell'amministrazione interessata. Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
CorrespondentId	String	Obbligatorio. Id del corrispondente che si vuole cancellare.

3.12.2.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "DeleteCorrespondentResponse".

Estende la classe base astratta "Response".

In caso di errore, "DeleteCorrespondent" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CORRESPONDENT	Richiesto corrispondente.
CORRESPONDENT_NOT_FOUND	Corrispondente non trovato.
APPLICATION_ERROR	Errore generico.

3.12.2.3 Esempio di utilizzo C#

```
AddressBook.AddressBook services = new AddressBook.AddressBook();
services.Timeout = System.Threading.Timeout.Infinite;
AddressBook.DeleteCorrespondentRequest request = new
AddressBook.DeleteCorrespondentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.CorrespondentId = "Id del corrispondente da cancellare";

try
{
AddressBook.DeleteCorrespondentResponse response =
services.DeleteCorrespondent(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
```

}

3.12.2.4 Esempio di utilizzo Java

```
AddressBookStub wws = new AddressBookStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
DeleteCorrespondentRequest deleteCorrespondentRequest =
DeleteCorrespondentRequest.Factory.newInstance();
deleteCorrespondentRequest.setCodeAdm("CodiceAmministrazione");
deleteCorrespondentRequest.setUsername("UserName");
deleteCorrespondentRequest.setCorrespondentId("IdCorrispondente");

DeleteCorrespondentDocument deleteCorrespondentDocument =
DeleteCorrespondentDocument.Factory.newInstance();
DeleteCorrespondent deleteCorrespondent = DeleteCorrespondent.Factory.newInstance();
deleteCorrespondent.setRequest(deleteCorrespondentRequest);
deleteCorrespondentDocument.setDeleteCorrespondent(deleteCorrespondent);
DeleteCorrespondentResponseDocument deleteCorrespondentResponseDocument =
DeleteCorrespondentResponseDocument.Factory.newInstance();

try
{
deleteCorrespondentResponseDocument =
wws.deleteCorrespondent(deleteCorrespondentDocument);
//Cancellato
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.12.3 SERVIZIO “EDITCORRESPONDENT”

Servizio che permette la modifica di un corrispondente esterno.

3.12.3.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “EditCorrespondentRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice

CodeRoleLogin	String	dell'amministrazione interessata. Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Correspondent	Correspondent	Obbligatorio. Corrispondente esistente. Vedere le tabelle successive per le varie tipologie di corrispondente

3.12.3.1.1 Modificare un corrispondente esterno

Struttura dell'oggetto Correspondent nel caso di corrispondente esterno.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Description	String	Obbligatorio. Descrizione del corrispondente.
Code	String	Obbligatorio. Codice del corrispondente ricercabile da rubrica.
CorrespondentType	String	Obbligatorio. Tipologia del corrispondente. Può assumere i valori "U" nel caso di Unità organizzativa e "P" nel caso di persona.
City	String	Opzionale. Città del corrispondente.
Province	String	Opzionale. Provincia del corrispondente.
Location	String	Opzionale. Località del corrispondente.
Nation	String	Opzionale. Nazione del corrispondente.
PhoneNumber	String	Opzionale. Numero di telefono del corrispondente.
Fax	String	Opzionale. Numero del fax del corrispondente.
NationalIdentificationNumber	String	Opzionale. Codice fiscale del corrispondente.

Email	String	Opzionale. Mail del corrispondente.
AOOCode	String	Opzionale. Codice AOO.
AdmCode	String	Opzionale. Codice amministrazione.
Note	String	Opzionale. Note
Address	String	Opzionale. Indirizzo del corrispondente.
Cap	String	Opzionale. Cap del corrispondente.
CodeRegisterOrRF	String	Opzionale. Se il corrispondente deve essere disponibile soltanto per un Registro/RF valorizzare con il codice del Registro/RF.
PreferredChannel	String	Opzionale. Canale preferenziale del corrispondente.

3.12.3.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “EditCorrespondentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

Nome	Tipo Dato	Descrizione
Correspondent	Correspondent	Dettaglio del corrispondente creato.

In caso di errore, “EditCorrespondentResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

Codice	Descrizione
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CORRESPONDENT	Richiesto corrispondente.

CORRESPONDENT_NOT_FOUND	Corrispondente non trovato.
APPLICATION_ERROR	Errore generico.

3.12.3.3 Esempio di utilizzo C#

```
AddressBook.AddressBookClient services = new AddressBook.AddressBookClient();
AddressBook.EditCorrespondentRequest request = new
AddressBook.EditCorrespondentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

AddressBook.Correspondent corrRequest = new AddressBook.Correspondent();

corrRequest.Id = "Id del corrispondente";
corrRequest.Email = "Email da modificare";

request.Correspondent = corrRequest;

try
{
AddressBook.EditCorrespondentResponse response = services.EditCorrespondent(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.12.3.4 Esempio di utilizzo Java

```
AddressBookStub wws = new AddressBookStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
EditCorrespondentRequest editCorrespondentRequest =
EditCorrespondentRequest.Factory.newInstance();
editCorrespondentRequest.setCodeAdm("CodiceAmministrazione");
editCorrespondentRequest.setUserName("UserName");
Correspondent correspondent = Correspondent.Factory.newInstance();
correspondent.setId("IdDelcorrispondenteDaModificare");
//Campi che si vuole modificare
correspondent.setAddress("Indirizzo modificato");
editCorrespondentRequest.setCorrespondent(correspondent);
EditCorrespondentDocument editCorrespondentDocument =
EditCorrespondentDocument.Factory.newInstance();
EditCorrespondent editCorrespondent = EditCorrespondent.Factory.newInstance();
editCorrespondent.setRequest(editCorrespondentRequest);
editCorrespondentDocument.setEditCorrespondent(editCorrespondent);
EditCorrespondentResponseDocument editCorrespondentResponseDocument =
EditCorrespondentResponseDocument.Factory.newInstance();
try
{
}
```



```
editCorrespondentResponseDocument =
wvs.editCorrespondent(editCorrespondentDocument);
Correspondent result = null;
result =
editCorrespondentResponseDocument.getEditCorrespondentResponse().getEditCorresponde
ntResult().getCorrespondent();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.12.4 SERVIZIO “GETCORRESPONDENT”

Servizio che permette il reperimento del dettaglio di un corrispondente dato l'id.

3.12.4.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetCorrespondentRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdCorrespondent	String	Obbligatorio. Id del corrispondente.

3.12.4.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetCorrespondentResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Correspondent	Correspondent[]	Dettaglio del corrispondente cercato.

In caso di errore, “GetCorrespondentResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_CORRESPONDENT	Richiesto corrispondente.
CORRESPONDENT_NOT_FOUND	Corrispondente non trovato.
APPLICATION_ERROR	Errore generico.

3.12.4.3 Esempio di utilizzo C#

```
AddressBook.AddressBookClient services = new AddressBook.AddressBookClient();
AddressBook.GetCorrespondentRequest request = new
AddressBook.GetCorrespondentRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdCorrespondent = "Id del corrispondente";

try
{
    AddressBook.GetCorrespondentResponse response = services.GetCorrespondent(request);
    AddressBook.Correspondent result = response.Correspondent;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.12.4.4 Esempio di utilizzo Java

```
AddressBookStub wws = new AddressBookStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetCorrespondentRequest getCorrespondentRequest =
GetCorrespondentRequest.Factory.newInstance();
getCorrespondentRequest.setCodeAdm("CodiceAmministrazione");
getCorrespondentRequest.setUserName("UserName");
getCorrespondentRequest.setIdsCorrespondent("IdDelCorrispondente");
GetCorrespondentDocument getCorrespondentDocument =
GetCorrespondentDocument.Factory.newInstance();
GetCorrespondent getCorrespondent = GetCorrespondent.Factory.newInstance();
getCorrespondent.setRequest(getCorrespondentRequest);
getCorrespondentDocument.setGetCorrespondent(getCorrespondent);
```

```

GetCorrespondentResponseDocument getCorrespondentResponseDocument =
GetCorrespondentResponseDocument.Factory.newInstance();
try
{
getCorrespondentResponseDocument = wws.getCorrespondent(getCorrespondentDocument);
Correspondent result = null;
result =
getCorrespondentResponseDocument.getGetCorrespondentResponse().getGetCorrespondentRe
sult().getCorrespondent();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.12.5 SERVIZIO “SEARCHUSERS”

Servizio per il reperimento di utenti interni all'applicazione.

3.12.5.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “SearchUsersRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Filters	Filter	Obbligatorio. Filtri di ricerca.

3.12.5.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “SearchUsersResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Correspondents	Correspondent[]	Corrispondenti interni cercati.

Nel caso in cui la ricerca non produca risultati, verrà restituita la lista degli utenti vuota.

In caso di errore, “SearchUsersResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_FILTER	Richiesto almeno un filtro di ricerca.
APPLICATION_ERROR	Errore generico. / Formato (Number/ Bool/ Date) non corretto.

3.12.5.3 Filtri disponibili

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
NATIONAL_IDENTIFICATION_NUMBER	Filter	Filtro per codice fiscale.
USER_MAIL	Filter	Filtro per mail.
USER_NAME	Filter	Filtro per nome.
USER_SURNAME	Filter	Filtro per cognome.

3.12.5.4 Esempio di utilizzo C#

```
AddressBook.AddressBookClient services = new AddressBook.AddressBookClient();
AddressBook.SearchUsersRequest request = new AddressBook.SearchUsersRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.Filters = new AddressBook.Filter[1];
int indice = 0;
AddressBook.Filter filter = new AddressBook.Filter();
filter.Name = "NATIONAL_IDENTIFICATION_NUMBER";
filter.Value = "Codice fiscale";
filter.Type = AddressBook.FilterType.String;
request.Filters.SetValue(filter, indice);
indice++;

try
{
    AddressBook.SearchUsersResponse response = services.SearchUsers(request);
    AddressBook.User[] result = response.Users;
}
catch (System.ServiceModel.FaultException e)
{
}
```

```
//e.Code;
//e.Message;
}
```

3.12.5.5 Esempio di utilizzo Java

```
AddressBookStub wws = new AddressBookStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
SearchUsersRequest searchUsersRequest = SearchUsersRequest.Factory.newInstance();
searchUsersRequest.setCodeAdm("CodiceAmministrazione");
searchUsersRequest.setUsername("UserName");
Filter[] filterArray = new Filter[1];
ArrayOfFilter filters = ArrayOfFilter.Factory.newInstance();
Filter filter = Filter.Factory.newInstance();
filter.setName("NATIONAL_IDENTIFICATION_NUMBER");
filter.setValue("ValoreDelCodiceFiscale");
filter.setType(FilterType.String);
filterArray[0] = filter;
filters.setFilterArray(filterArray);
searchUsersRequest.setFilters(filters);
SearchUsersDocument searchUsersDocument = SearchUsersDocument.Factory.newInstance();
SearchUsers searchUsers = SearchUsers.Factory.newInstance();
searchUsers.setRequest(searchUsersRequest);
searchUsersDocument.setSearchUsers(searchUsers);
SearchUsersResponseDocument searchUsersResponseDocument =
SearchUsersResponseDocument.Factory.newInstance();
try
{
searchUsersResponseDocument = wws.searchUsers(searchUsersDocument);
User[] users = null;
users =
searchUsersResponseDocument.getSearchUsersResponse().getSearchUsersResult().getUsers
().getUserArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.12.6 SERVIZIO “SEARCHCORRESPONDENTS”

Servizio per la ricerca di corrispondenti.

3.12.6.1 Descrizione oggetto “Request”

L’oggetto Request necessario per effettuare la chiamata al servizio è il “SearchCorrespondentsRequest”.

Struttura dell’oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
--------------------	-------------------------	---------------------------

UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Filter	Filter[]	Obbligatorio. Lista di filtri per la ricerca.

3.12.6.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "SearchCorrespondentsResponse".

Estende la classe base astratta "Response" e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Correspondents	Correspondent[]	Corrispondenti cercati.

Nel caso in cui la ricerca non produca risultati, verrà restituita la lista dei corrispondenti vuota.

In caso di errore, "SearchCorrespondents" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_FILTER	Richiesto almeno un filtro di ricerca.
APPLICATION_ERROR	Errore generico. / Formato (Number/ Bool/ Date) non corretto.

3.12.6.3 Filtri disponibili

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
TYPE	Filter	Filtro utilizzato per reperire il tipo di corrispondente. Inserire il valore "INTERNAL" per gli interni, "EXTERNAL" per gli esterni, "GLOBAL"

		per tutti i corrispondenti.
OFFICES	Filter	Filtro utilizzato per reperire i corrispondenti di tipo UO con valore "true".
USERS	Filter	Filtro utilizzato per reperire i corrispondenti di tipo Persona con valore "true".
ROLES	Filter	Filtro utilizzato per reperire i corrispondenti di tipo Ruoli con valore "true".
COMMON_ADDRESSBOOK	Filter	Filtro utilizzato per reperire i corrispondenti anche in rubrica comune con valore "true".
RF	Filter	Filtro utilizzato per reperire i corrispondenti anche di RF con valore "true".
CODE	Filter	Filtro utilizzato per la ricerca di corrispondenti dato il codice rubrica.
DESCRIPTION	Filter	Filtro utilizzato per la ricerca di corrispondenti data la descrizione.
CITY	Filter	Filtro utilizzato per la ricerca di corrispondenti data la città.
LOCALITY	Filter	Filtro utilizzato per la ricerca di corrispondenti indicata al località.
REGISTRY_OR_RF	Filter	Filtro utilizzato per la ricerca di corrispondenti soltanto in un determinato Registro/RF dato il codice del Registro/RF.
MAIL	Filter	Filtro utilizzato per la ricerca di corrispondenti indicando l'indirizzo mail.
NATIONAL_IDENTIFICATION_NUMBER	Filter	Filtro utilizzato per la ricerca di corrispondenti indicando il codice fiscale.
VAT_NUMBER	Filter	Filtro utilizzato per la ricerca di corrispondenti indicando la partita IVA.
EXACT_CODE	Filter	Filtro utilizzato per la ricerca di un corrispondente dato il codice rubrica esatto.

3.12.6.4 Esempio di utilizzo C#

```
AddressBook.AddressBookClient services = new AddressBook.AddressBookClient();
```

```
AddressBook.SearchCorrespondentsRequest request = new
AddressBook.SearchCorrespondentsRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
int indice = 0;

request.Filters = new AddressBook.Filter[5];
AddressBook.Filter filter = new AddressBook.Filter();
filter = new AddressBook.Filter();
filter.Name = "OFFICES";
filter.Value = "True";
filter.Type = AddressBook.FilterType.Bool;
request.Filters.SetValue(filter, indice);
indice++;
filter = new AddressBook.Filter();
filter.Name = "USERS";
filter.Value = "True";
filter.Type = AddressBook.FilterType.Bool;
request.Filters.SetValue(filter, indice);
indice++;
filter = new AddressBook.Filter();
filter.Name = "ROLES";
filter.Value = "True";
filter.Type = AddressBook.FilterType.Bool;
request.Filters.SetValue(filter, indice);
indice++;
filter = new AddressBook.Filter();
filter.Name = "TYPE";
filter.Value = "EXTERNAL";
filter.Type = AddressBook.FilterType.String;
request.Filters.SetValue(filter, indice);
indice++;
filter = new AddressBook.Filter();
filter.Name = "DESCRIPTION";
filter.Value = "Descrizione";
filter.Type = AddressBook.FilterType.String;
request.Filters.SetValue(filter, indice);

try
{
    AddressBook.SearchCorrespondentsResponse response =
    services.SearchCorrespondents(request);
    AddressBook.Correspondent[] result = response.Correspondents;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.12.6.5 Esempio di utilizzo Java

```
AddressBookStub wws = new AddressBookStub("linkWebServices");
```



```
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
SearchCorrespondentsRequest searchCorrespondentsRequest =
SearchCorrespondentsRequest.Factory.newInstance();
searchCorrespondentsRequest.setCodeAdm("CodiceAmministrazione");
searchCorrespondentsRequest.setUsername("UserName");
Filter[] filterArray = new Filter[5];
ArrayOfFilter filters = ArrayOfFilter.Factory.newInstance();
Filter filter = Filter.Factory.newInstance();
filter.setName("OFFICES");
filter.setValue("True");
filter.setType(FilterType.Bool);
filterArray[0] = filter;
filter = Filter.Factory.newInstance();
filter.setName("USERS");
filter.setValue("True");
filter.setType(FilterType.Bool);
filterArray[1] = filter;
filter = Filter.Factory.newInstance();
filter.setName("ROLES");
filter.setValue("True");
filter.setType(FilterType.Bool);
filterArray[2] = filter;
filter = Filter.Factory.newInstance();
filter.setName("TYPE");
filter.setValue("EXTERNAL");
filter.setType(FilterType.String);
filterArray[3] = filter;
filter = Filter.Factory.newInstance();
filter.setName("DESCRIPTION");
filter.setValue("Descrizione");
filter.setType(FilterType.String);
filterArray[4] = filter;
filters.setFilterArray(filterArray);
searchCorrespondentsRequest.setFilters(filters);
SearchCorrespondentsDocument searchCorrespondentsDocument =
SearchCorrespondentsDocument.Factory.newInstance();
SearchCorrespondents searchCorrespondents =
SearchCorrespondents.Factory.newInstance();
searchCorrespondents.setRequest(searchCorrespondentsRequest);
searchCorrespondentsDocument.setSearchCorrespondents(searchCorrespondents);
SearchCorrespondentsResponseDocument searchCorrespondentsResponseDocument =
SearchCorrespondentsResponseDocument.Factory.newInstance();
try
{
searchCorrespondentsResponseDocument =
wws.searchCorrespondents(searchCorrespondentsDocument);
Correspondent[] result = null;
result =
searchCorrespondentsResponseDocument.getSearchCorrespondentsResponse().getSearchCorr
espondentsResult().getCorrespondents().getCorrespondentArray();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.12.7 SERVIZIO “GETCORRESPONDENTFILTERS”

Servizio che restituisce la lista dei filtri applicabili alla ricerca corrispondenti.

3.12.7.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetCorrespondentFiltersRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.12.7.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetCorrespondentFiltersResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Filters	Filter[]	Lista dei filtri applicabili.

In caso di errore, “GetCorrespondentFiltersResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.12.7.3 Esempio di utilizzo C#

```
AddressBook.AddressBookClient services = new AddressBook.AddressBookClient();
AddressBook.GetCorrespondentFiltersRequest request = new
AddressBook.GetCorrespondentFiltersRequest();
```

```
request.UserName = this.txt_username.Text;
request.CodeAdm = this.txt_codAmm.Text;
request.CodeRoleLogin = this.txt_codice_ruolo.Text;

this.txtRequest.Text = this.SerializeAsXml(request);

try
{
    AddressBook.GetCorrespondentFiltersResponse response =
    services.GetCorrespondentFilters(request);

    this.txtResponse.Text = this.SerializeAsXml(response);

    AddressBook.Filter[] result = response.Filters;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.12.7.4 Esempio di utilizzo Java

```
AddressBookStub wws = new AddressBookStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetCorrespondentFiltersRequest getCorrespondentFiltersRequest=
GetCorrespondentFiltersRequest.Factory.newInstance();
getCorrespondentFiltersRequest.setCodeAdm("CodiceAmministrazione");
getCorrespondentFiltersRequest.setUserName("UserName");
GetCorrespondentFiltersDocument getCorrespondentFiltersDocument =
GetCorrespondentFiltersDocument.Factory.newInstance();
GetCorrespondentFilters getCorrespondentFilters =
GetCorrespondentFilters.Factory.newInstance();
getCorrespondentFilters.setRequest(getCorrespondentFiltersRequest);
getCorrespondentFiltersDocument.setGetCorrespondentFilters(getCorrespondentFilters);
GetCorrespondentFiltersResponseDocument getCorrespondentFiltersResponseDocument =
GetCorrespondentFiltersResponseDocument.Factory.newInstance();
try
{
    getCorrespondentFiltersResponseDocument = wws.getCorrespondentFilters
(getCorrespondentFiltersDocument);
    Filter[] filters = null;
    filters =
getCorrespondentFiltersResponseDocument.getGetCorrespondentFiltersResponse().getGetC
orrespondentFiltersResult().getFilters().getFilterArray();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.12.8 SERVIZIO “GETUSERFILTERS”

Servizio che restituisce la lista dei filtri applicabili alla ricerca utenti.

3.12.8.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetUserFiltersRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.

3.12.8.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetUserFiltersResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Filters	Filter[]	Lista dei filtri applicabili.

In caso di errore, “GetUserFiltersResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.12.8.3 Esempio di utilizzo C#

```
AddressBook.AddressBookClient services = new AddressBook.AddressBookClient();  
AddressBook.GetUserFiltersRequest request = new AddressBook.GetUserFiltersRequest();  
  
request.UserName = this.txt_username.Text;
```

```
request.CodeAdm = this.txt_codAmm.Text;
request.CodeRoleLogin = this.txt_codice_ruolo.Text;

this.txtRequest.Text = this.SerializeAsXml(request);

try
{
    AddressBook.GetUserFiltersResponse response = services.GetUserFilters(request);

    this.txtResponse.Text = this.SerializeAsXml(response);

    AddressBook.Filter[] result = response.Filters;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.12.8.4 Esempio di utilizzo Java

```
DocumentsStub wws = new DocumentsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetUserFiltersRequest getUserFiltersRequest=
    GetUserFiltersRequest.Factory.newInstance();
getUserFiltersRequest.setCodeAdm("CodiceAmministrazione");
getUserFiltersRequest.setUserName("UserName");
GetUserFiltersDocument getUserFiltersDocument =
    GetUserFiltersDocument.Factory.newInstance();
GetUserFilters getUserFilters = GetUserFilters.Factory.newInstance();
getUserFilters.setRequest(getUserFiltersRequest);
getUserFiltersDocument.setUserFilters(getUserFilters);
GetUserFiltersResponseDocument getUserFiltersResponseDocument =
    GetUserFiltersResponseDocument.Factory.newInstance();
try
{
    getUserFiltersResponseDocument = wws.getUserFilters (getUserFiltersDocument);
    Filter[] filters = null;
    filters =
        getUserFiltersResponseDocument.getGetUserFiltersResponse().UserFiltersResult().getFi
        lters();
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.13 SERVIZI PER LA GESTIONE DEI TOKEN DI AUTENTICAZIONE

3.13.1 SERVIZIO “GETAUTHENTICATIONTOKEN”

Servizio per il reperimento di un token di autenticazione per un utente all'interno di un'amministrazione.

3.13.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetAuthenticationTokenRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.

3.13.1.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetAuthenticationTokenResponse”.

Estende la classe base astratta “Response”.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
AuthenticationToken	String	Stringa contenente il token di autenticazione.

In caso di errore, “GetAuthenticationTokenModel” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.13.1.3 Esempio di utilizzo C#

```
Token.TokenClient services = new Token.TokenClient ();
Token.GetAuthenticationTokenRequest request = new Token.GetAuthenticationTokenRequest
();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
    Token.GetAuthenticationTokenResponse response =
    services.GetAuthenticationToken(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.13.1.4 Esempio di utilizzo Java

```
TokenStub wws = new TokenStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetAuthenticationTokenRequest getAuthenticationTokenRequest =
GetAuthenticationTokenRequest.Factory.newInstance();
getAuthenticationTokenRequest.setCodeAdm("CodiceAmministrazione");
getAuthenticationTokenRequest.setUserName("UserName");
GetAuthenticationTokenDocument getAuthenticationTokenDocument =
GetAuthenticationTokenDocument.Factory.newInstance();
GetAuthenticationToken getAuthenticationToken=
GetAuthenticationToken.Factory.newInstance();
getAuthenticationToken.setRequest(getAuthenticationTokenRequest);
getAuthenticationTokenDocument.setGetAuthenticationToken(getAuthenticationToken);
GetAuthenticationTokenResponseDocument getAuthenticationTokenResponseDocument =
GetAuthenticationTokenResponseDocument.Factory.newInstance();
try
{
    getAuthenticationTokenResponseDocument =
    wws.executeGetAuthenticationToken(executeGetAuthenticationTokenDocument);
    //Token prelevato
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.13.2 SERVIZIO “GETTOKEN”

Servizio per il reperimento di un token di autenticazione e della durata dello stesso per un utente all'interno di un'amministrazione e per un utente di sistema esterno.

3.13.2.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetTokenRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio per utente di sistema. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeApplication	String	Obbligatorio per sistema esterno. Codice applicazione esterna censito nel sistema di gestione documentale.

3.13.2.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetTokenResponse”.

Estende la classe base astratta “Response”.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
AuthenticationToken	String	Stringa contenente il token di autenticazione.
TokenDuration	String	Durata in minuti del token di autenticazione.
Username	String	Username che un utente di sistema esterno deve utilizzare nelle chiamate al sistema.

In caso di errore, “GetTokenResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.

USER_NO_EXIST	Utente non trovato.
APPLICATION_ERROR	Errore generico.

3.13.2.3 Esempio di utilizzo C#

```
Token.TokenClient services = new Token.TokenClient ();
Token.GetTokenRequest request = new Token.GetTokenRequest ();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

try
{
    Token.GetTokenResponse response = services.GetToken(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.13.2.4 Esempio di utilizzo Java

```
TokenStub wws = new TokenStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetTokenRequest getTokenRequest = GetTokenRequest.Factory.newInstance();
getTokenRequest.setCodeAdm("CodiceAmministrazione");
getTokenRequest.setUserName("UserName");
GetTokenDocument getTokenDocument = GetTokenDocument.Factory.newInstance();
GetToken getToken= GetToken.Factory.newInstance();
getToken.setRequest(getTokenRequest);
getTokenDocument.setGetToken(getToken);
GetTokenResponseDocument getTokenResponseDocument =
GetTokenResponseDocument.Factory.newInstance();
try
{
    getTokenResponseDocument = wws.executeGetToken(executeGetTokenDocument);
    //Token prelevato
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.14 SERVIZI PER LA GESTIONE DEI MODELLI DI TRASMISSIONE

3.14.1 SERVIZIO “EXECUTE TRANSM DOC MODEL”

Servizio per la trasmissione di un documento tramite un modello di trasmissione.

3.14.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “ExecuteTransmDocModelRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
DocumentId	String	Obbligatorio. Id del documento che si desidera trasmettere.
CodeModel	String	Obbligatorio. Codice del modello di trasmissione.

3.14.1.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “ExecuteTransmDocModelResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “ExecuteTransmDocModel” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID	Id del documento obbligatorio.
REQUIRED_ID_MODEL	Richiesto l'id del modello di trasmissione.
DOCUMENT_NOT_FOUND	Documento non trovato.
TRANSMISSION_MODEL_NOT_FOUND	Modello non trovato.
APPLICATION_ERROR	Errore generico.

3.14.1.3 Esempio di utilizzo C#

```
Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.ExecuteTransmDocModelRequest request = new
Transmissions.ExecuteTransmDocModelRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.DocumentId = "Id del documento";
request.IdModel = "Id del modello";

try
{
    Transmissions.ExecuteTransmDocModelResponse response =
    services.ExecuteTransmDocModel(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.14.1.4 Esempio di utilizzo Java

```
TransmissionsStub wws = new TransmissionsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
ExecuteTransmDocModelRequest executeTransmDocModelRequest =
ExecuteTransmDocModelRequest.Factory.newInstance();
executeTransmDocModelRequest.setCodeAdm("CodiceAmministrazione");
executeTransmDocModelRequest.setUserName("UserName");
executeTransmDocModelRequest.setDocumentId("IdDocumento");
executeTransmDocModelRequest.setIdModel("IdModelloTrasmissione");
ExecuteTransmDocModelDocument executeTransmDocModelDocument =
ExecuteTransmDocModelDocument.Factory.newInstance();
ExecuteTransmDocModel executeTransmDocModel =
ExecuteTransmDocModel.Factory.newInstance();
executeTransmDocModel.setRequest(executeTransmDocModelRequest);
executeTransmDocModelDocument.setExecuteTransmDocModel(executeTransmDocModel);
ExecuteTransmDocModelResponseDocument executeTransmDocModelResponseDocument =
ExecuteTransmDocModelResponseDocument.Factory.newInstance();
try
{
    executeTransmDocModelResponseDocument =
    wws.executeTransmDocModel(executeTransmDocModelDocument);
    //Trasmesso
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.14.2 SERVIZIO “EXECUTETRANSPRJMODEL”

Servizio per la trasmissione di un fascicolo tramite un modello di trasmissione.

3.14.2.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “ExecuteTransmPrjModelRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
ProjectId	String	Obbligatorio. Id del fascicolo che si desidera trasmettere.
ModelCode	String	Obbligatorio. Codice del modello di trasmissione.

3.14.2.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “ExecuteTransmPrjModelResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “ExecuteTransmPrjModel” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_PROJECT	Id del fascicolo obbligatorio.
REQUIRED_ID_MODEL	Richiesto l'id del modello di trasmissione.
PROJECT_NOT_FOUND	Fascicolo non trovato.
TRANSMISSION_MODEL_NOT_FOUND	Modello non trovato.
APPLICATION_ERROR	Errore generico.

3.14.2.3 Esempio di utilizzo C#

```
Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.ExecuteTransmPrjModelRequest request = new
Transmissions.ExecuteTransmPrjModelRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdProject = "Id del fascicolo";
request.IdModel = "Id del modello";

try
{
    Transmissions.ExecuteTransmPrjModelResponse response =
    services.ExecuteTransmPrjModel(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.14.2.4 Esempio di utilizzo Java

```
TransmissionsStub wws = new TransmissionsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
ExecuteTransmPrjModelRequest executeTransmPrjModelRequest =
ExecuteTransmPrjModelRequest.Factory.newInstance();
executeTransmPrjModelRequest.setCodeAdm("CodiceAmministrazione");
executeTransmPrjModelRequest.setUsername("UserName");
executeTransmPrjModelRequest.setIdProject("IdProject");
executeTransmPrjModelRequest.setIdModel("IdModello");
ExecuteTransmPrjModelDocument executeTransmPrjModelDocument =
ExecuteTransmPrjModelDocument.Factory.newInstance();
ExecuteTransmPrjModel executeTransmPrjModel =
ExecuteTransmPrjModel.Factory.newInstance();
executeTransmPrjModel.setRequest(executeTransmPrjModelRequest);
executeTransmPrjModelDocument.setExecuteTransmPrjModel(executeTransmPrjModel);
ExecuteTransmPrjModelResponseDocument executeTransmPrjModelResponseDocument =
ExecuteTransmPrjModelResponseDocument.Factory.newInstance();
try
{
    executeTransmPrjModelResponseDocument =
    wws.executeTransmPrjModel(executeTransmPrjModelDocument);
    //Trasmesso
}
catch (RemoteException e)
{
    //e.getMessage();
    //e.printStackTrace();
}
```

3.14.3 SERVIZIO “GETTRANSMISSIONMODEL”

Servizio per il reperimento del dettaglio di un modello di trasmissione dato il codice o l'id del modello. Obbligatorio almeno un valore tra codice e id.

3.14.3.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetTransmissionModelRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdModel	String	Opzionale. Id del modello di trasmissione.
CodeModel	String	Opzionale. Codice del modello di trasmissione.

3.14.3.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “GetTransmissionModelResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall'oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
TransmissionModel	TransmissionModel	Dettaglio del modello di trasmissione.

In caso di errore, “GetTransmissionModelResponse” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ONLY_TM_ID_OR_CODE	Richiesto un valore tra id o codice del modello di trasmissione.
REQUIRED_TM_ID_OR_CODE	Richiesto un valore tra il codice o il modello di trasmissione.
TRANSMISSION_MODEL_NOT_FOUND	Modello di trasmissione non trovato.
APPLICATION_ERROR	Errore generico.

3.14.3.3 Esempio di utilizzo C#

```

Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.GetTransmissionModelRequest request = new
Transmissions.GetTransmissionModelRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.IdModel = "Id del modello";
//Opzionale al posto dell'id
// request.CodeModel = "Codice del modello";

try
{
    Transmissions.GetTransmissionModelResponse response =
    services.GetTransmissionModel(request);
    Transmissions.TransmissionModel result = response.TransmissionModel;
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}

```

3.14.3.4 Esempio di utilizzo Java

```

TransmissionsStub wws = new TransmissionsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GetTransmissionModelRequest getTransmissionModelRequest =
GetTransmissionModelRequest.Factory.newInstance();
getTransmissionModelRequest.setCodeAdm("CodiceAmministrazione");
getTransmissionModelRequest.setUsername("UserName");
getTransmissionModelRequest.setIdModel("IdDelModello");

```

```

GetTransmissionModelDocument getTransmissionModelDocument =
GetTransmissionModelDocument.Factory.newInstance();
GetTransmissionModel getTransmissionModel =
GetTransmissionModel.Factory.newInstance();
getTransmissionModel.setRequest(getTransmissionModelRequest);
getTransmissionModelDocument.setGetTransmissionModel(getTransmissionModel);
GetTransmissionModelResponseDocument getTransmissionModelResponseDocument =
GetTransmissionModelResponseDocument.Factory.newInstance();
try
{
getTransmissionModelResponseDocument =
wvs.getTransmissionModel(getTransmissionModelDocument);
TransmissionModel result = null;
result =
getTransmissionModelResponseDocument.getGetTransmissionModelResponse().getGetTransmi
ssionModelResult().getTransmissionModel();
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.14.4 SERVIZIO “GETTRANSMISSIONMODELS”

Servizio per il reperimento di tutti i modelli di trasmissione per documenti o fascicoli.

3.14.4.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetTransmissionModelsRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
Type	String	Obbligatorio. Inserire “D” per i modelli dei documenti, “F” per i modelli dei fascicoli.
Registers	Register[]	Obbligatorio. Obbligatorio almeno un codice di un registro.

3.14.4.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “GetTransmissionModelsResponse”.

Estende la classe base astratta “Response” e restituisce i parametri aggiuntivi.

La tabella seguente riporta i parametri restituiti dall’oggetto Response.

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
TransmissionModels	TransmissionModel[]	Modelli di trasmissione cercati.

In caso di errore, “GetTransmissionModelsResponse” potrà restituire un’eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REGISTER_NOT_FOUND	Registro non trovato.
REQUIRED_REGISTER	Richiesto registro.
TRANSMISSION_MODEL_TYPE	Inserire il tipo di modello D o F.
APPLICATION_ERROR	Errore generico.

3.14.4.3 Esempio di utilizzo C#

```
Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.GetTransmissionModelsRequest request = new
Transmissions.GetTransmissionModelsRequest();

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";

request.Registers = new Transmissions.Register[1];
Transmissions.Register register = new Transmissions.Register();
register.Code = "Codice del registro";
request.Registers[0] = register;
request.Type = "D"; //D per documenti F per fascicoli

try
{
    Transmissions.GetTransmissionModelsResponse response =
    services.GetTransmissionModels(request);
    Transmissions.TransmissionModel[] result = response.TransmissionModels;
}
catch (System.ServiceModel.FaultException e)
{
}
```

```
//e.Code;  
//e.Message;  
}
```

3.14.4.4 Esempio di utilizzo Java

```
TransmissionsStub wws = new TransmissionsStub("linkWebServices");  
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);  
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V  
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);  
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE  
_NAMESPACE_URI);  
GetTransmissionModelsRequest getTransmissionModelsRequest =  
GetTransmissionModelsRequest.Factory.newInstance();  
getTransmissionModelsRequest.setCodeAdm("CodiceAmministrazione");  
getTransmissionModelsRequest.setUsername("UserName");  
ArrayOfRegister registers = ArrayOfRegister.Factory.newInstance();  
Register[] registerArray = new Register[1];  
Register reg = Register.Factory.newInstance();  
reg.setCode("CodiceRegistro");  
registerArray[0] = reg;  
registers.setRegisterArray(registerArray);  
getTransmissionModelsRequest.setRegisters(registers);  
getTransmissionModelsRequest.setType("D"); //Tipo di modello D Documento  
GetTransmissionModelsDocument getTransmissionModelsDocument =  
GetTransmissionModelsDocument.Factory.newInstance();  
GetTransmissionModels getTransmissionModels =  
GetTransmissionModels.Factory.newInstance();  
getTransmissionModels.setRequest(getTransmissionModelsRequest);  
getTransmissionModelsDocument.setGetTransmissionModels(getTransmissionModels);  
GetTransmissionModelsResponseDocument getTransmissionModelsResponseDocument =  
GetTransmissionModelsResponseDocument.Factory.newInstance();  
try  
{  
    getTransmissionModelsResponseDocument =  
    wws.getTransmissionModels(getTransmissionModelsDocument);  
    TransmissionModel[] result = null;  
    result =  
    getTransmissionModelsResponseDocument.getGetTransmissionModelsResponse().getGetTrans  
missionModelsResult().getTransmissionModels().getTransmissionModelArray();  
}  
catch (RemoteException e)  
{  
    //e.getMessage();  
    //e.printStackTrace();  
}
```

3.14.5 SERVIZIO “EXECUTE TRANSMISSION DOCUMENT”

Servizio per la trasmissione singola di un documento senza l'utilizzo di un modello di trasmissione.

3.14.5.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “ExecuteTransmissionDocumentRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdDocument	String	Obbligatorio. Id del documento che si desidera trasmettere.
Receiver	Correspondent	Obbligatorio. Destinatario della trasmissione. E' obbligatorio l'inserimento della sua proprietà Id o Code.
TransmissionReason	String	Obbligatorio. Codice della ragione di trasmissione con la quale si invia il documento.
CodeReg	String	Opzionale. Codice del registro.
Notify	Boolean	Opzionale. <ul style="list-style-type: none"> • Se true, notifica gli utenti della trasmissione. • Se false o non inserito, non invia le notifiche.
TransmissionType	String	Opzionale. Definisce il tipo di trasmissione: <ul style="list-style-type: none"> • Se "T", la trasmissione è di tipo "Tutti". • Qualsiasi altro valore (compreso nullo) la trasmissione è di tipo "Uno"

3.14.5.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "ExecuteTransmissionDocumentResponse".

Estende la classe base astratta “Response”.

In caso di errore, “ExecuteTransmissionDocument” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID	Id del documento obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
REQUIRED_CORRESPONDENT	Corrispondente obbligatorio.
REQUIRED_CODE_OR_ID_RECEIVER	Codice o id destinatario richiesto.
CORRESPONDENT_NOT_FOUND	Corrispondente non trovato.
REQUIRED_TRANSMISSION_REASON	Ragione di trasmissione obbligatoria.
TRANSMISSION_REASON_NOT_FOUND	Ragione di trasmissione non trovata.
DOCUMENT_NOT_FOUND	Documento non trovato.
APPLICATION_ERROR	Errore generico.

3.14.5.3 Esempio di utilizzo C#

```
Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.ExecuteTransmissionDocumentRequest request = new
Transmissions.ExecuteTransmissionDocumentRequest();
```

```
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdDocument = "Id del documento";
request.TransmissionType= "T"; //oppure vuoto se deve essere di tipo "Uno"
request.Notify= "true";
request.TransmissionReason= "Codice ragione di trasmissione";
Transmission.Correspondent receiver= new Transmissions.Correspondent();
receiver.Code= "Codice destinatario";
request.Receiver= receiver;
```

```
try
{
Transmissions.ExecuteTransmissionDocumentResponse response =
services.ExecuteTransmissionDocumentModel(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}
```

3.14.5.4 Esempio di utilizzo Java

```

TransmissionsStub wws = new TransmissionsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeoutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
ExecuteTransmissionDocumentRequest executeTransmissionDocumentRequest =
ExecuteTransmissionDocumentRequest.Factory.newInstance();
executeTransmissionDocumentRequest.setCodeAdm("CodiceAmministrazione");
executeTransmissionDocumentRequest.setUsername("UserName");
executeTransmissionDocumentRequest.setIdDocument("IdDocumento");
executeTransmissionDocumentRequest.setTransmissionType("T");
executeTransmissionDocumentRequest.setNotify("true");
executeTransmissionDocumentRequest.setTransmissionReason("codice della ragione di
trasmissione.");
Correspondent receiver= new Correspondent.Factory.newInstance();
receiver.setCode("Codice del destinatario");
executeTransmissionDocumentRequest.setReceiver(receiver);
ExecuteTransmissionDocumentDocument executeTransmissionDocumentDocument =
ExecuteTransmissionDocumentDocument.Factory.newInstance();
ExecuteTransmissionDocument executeTransmissionDocument =
ExecuteTransmissionDocument.Factory.newInstance();
executeTransmissionDocument.setRequest(executeTransmissionDocumentRequest);
executeTransmissionDocumentDocument.setExecuteTransmissionDocument
(executeTransmissionDocument);
ExecuteTransmissionDocumentResponseDocument
executeTransmissionDocumentResponseDocument =
ExecuteTransmissionDocumentResponseDocument.Factory.newInstance();
try
{
executeTransmissionDocumentResponseDocument = wws.executeTransmissionDocument
(executeTransmissionDocumentDocument);
//Trasmesso
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.14.6 SERVIZIO “EXECUTE TRANSMISSION PROJECT”

Servizio per la trasmissione singola di un fascicolo senza l'utilizzo di un modello di trasmissione.

3.14.6.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “ExecuteTransmissionProjectRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione

CodeAdm	String	documentale. Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
IdProject	String	Obbligatorio. Id del progetto che si desidera trasmettere.
Receiver	Correspondent	Obbligatorio. Destinatario della trasmissione. E' obbligatorio l'inserimento della sua proprietà Id o Code.
TransmissionReason	String	Obbligatorio. Codice della ragione di trasmissione con la quale si invia il documento.
CodeReg	String	Opzionale. Codice del registro.
Notify	Boolean	Opzionale. <ul style="list-style-type: none"> • Se true, notifica gli utenti della trasmissione. • Se false o non inserito, non invia le notifiche.
TransmissionType	String	Opzionale. Definisce il tipo di trasmissione: <ul style="list-style-type: none"> • Se "T", la trasmissione è di tipo "Tutti". • Qualsiasi altro valore (compreso nullo) la trasmissione è di tipo "Uno"

3.14.6.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "ExecuteTransmissionProjectResponse".

Estende la classe base astratta "Response".

In caso di errore, "ExecuteTransmissionProject" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
REQUIRED_ID_PROJECT	Id del fascicolo obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
REQUIRED_CORRESPONDENT	Corrispondente obbligatorio.
REQUIRED_CODE_OR_ID_RECEIVER	Codice o id destinatario richiesto.
CORRESPONDENT_NOT_FOUND	Corrispondente non trovato.
REQUIRED_TRANSMISSION_REASON	Ragione di trasmissione obbligatoria.
TRANSMISSION_REASON_NOT_FOUND	Ragione di trasmissione non trovata.
PROJECT_NOT_FOUND	Fascicolo non trovato.
APPLICATION_ERROR	Errore generico.

3.14.6.3 Esempio di utilizzo C#

```

Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.ExecuteTransmissionProjectRequest request = new
Transmissions.ExecuteTransmissionProjectRequest();

```

```

request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdProject = "Id del fascicolo";
request.TransmissionType= "T"; //oppure vuoto se deve essere di tipo "Uno"
request.Notify= "true";
request.TransmissionReason= "Codice ragione di trasmissione";
Transmission.Correspondent receiver= new Transmissions.Correspondent();
receiver.Code= "Codice destinatario";
request.Receiver= receiver;

```

```

try
{
Transmissions.ExecuteTransmissionProjectResponse response =
services.ExecuteTransmissionProject(request);
}
catch (System.ServiceModel.FaultException e)
{
//e.Code;
//e.Message;
}

```

3.14.6.4 Esempio di utilizzo Java

```

TransmissionsStub wws = new TransmissionsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);

```

```
ExecuteTransmissionProjectRequest executeTransmissionProjectRequest =
ExecuteTransmissionProjectRequest.Factory.newInstance();
executeTransmissionProjectRequest.setCodeAdm("CodiceAmministrazione");
executeTransmissionProjectRequest.setUserName("UserName");
executeTransmissionProjectRequest.setIdProject ("Id del fascicolo");
executeTransmissionProjectRequest.setRequest.setTransmissionType("T");
executeTransmissionProjectRequest.setNotify("true");
executeTransmissionProjectRequest.setTransmissionReason("codice della ragione di
trasmissione.");
Correspondent receiver= new Correspondent.Factory.newInstance();
receiver.setCode("Codice del destinatario");
executeTransmissionProjectRequest.setReceiver(receiver);
ExecuteTransmissionProjectDocument executeTransmissionProjectDocument =
ExecuteTransmissionProjectDocument.Factory.newInstance();
ExecuteTransmissionProject executeTransmissionProject =
ExecuteTransmissionProject.Factory.newInstance();
executeTransmissionProject.setRequest(executeTransmissionProjectRequest);
executeTransmissionProjectDocument.setExecuteTransmissionProject
(executeTransmissionProject);
ExecuteTransmissionProjectResponseDocument
executeTransmissionProjectResponseDocument =
ExecuteTransmissionProjectResponseDocument.Factory.newInstance();
try
{
executeTransmissionProjectResponseDocument = wws.executeTransmissionProject
(executeTransmissionProjectDocument);
//Trasmesso
}
catch (RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}
```

3.14.7 SERVIZIO “GIVEUPRIGHTS”

Servizio per la cessione del diritto di scrittura/lettura su un oggetto da parte di un utente di sistema esterno dopo averne ceduto la proprietà.

3.14.7.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GiveUpRightsRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con

RightToKeep	String	cui si vuole accedere al sistema. Obbligatorio. Diritto da mantenere. Può avere 3 valori: WRITE, READ e NONE.
IdObject	String	Obbligatorio. Id dell'oggetto sul quale si vogliono perdere i diritti.

3.14.7.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il " GiveUpRightsResponse".

Estende la classe base astratta "Response".

In caso di errore, "GiveUpRights" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
ROLE_NO_EXIST	Il ruolo non esiste.
APPLICATION_ERROR	Errore generico. Alcuni errori interni particolari vengono gestiti tramite eccezione generica ma con messaggi espliciti.

3.14.7.3 Esempio di utilizzo C#

```
Transmissions.TransmissionsClient services = new Transmissions.TransmissionsClient();
Transmissions.GiveUpRightsRequest request = new Transmissions. GiveUpRightsRequest();
```

```
request.UserName = "UserName";
request.CodeAdm = "Codice amministrazione";
request.IdObject = "Id dell'oggetto";
request.RightToKeep= "READ";
```

```
try
{
    Transmissions.GiveUpRightsResponse response = services.GiveUpRights(request);
}
catch (System.ServiceModel.FaultException e)
{
    //e.Code;
    //e.Message;
}
```

3.14.7.4 Esempio di utilizzo Java

```

TransmissionsStub wws = new TransmissionsStub("linkWebServices");
wws._getServiceClient().getOptions().setTimeOutInMilliseconds(6 * 60 * 1000);
wws._getServiceClient().getOptions().setProperty(AddressingConstants.WS_ADDRESSING_V
ERSION, AddressingConstants.Submission.WSA_NAMESPACE);
wws._getServiceClient().getOptions().setSoapVersionURI(SOAP11Constants.SOAP_ENVELOPE
_NAMESPACE_URI);
GiveUpRightsRequest giveUpRightsRequest = GiveUpRightsRequest.Factory.newInstance();
giveUpRightsRequest.setCodeAdm("CodiceAmministrazione");
giveUpRightsRequest.setUserName("UserName");
giveUpRightsRequest.setIdObject("Id dell'oggetto");
giveUpRightsRequest.setRightToKeep("READ");
GiveUpRightsDocument giveUpRightsDocument =
GiveUpRightsDocument.Factory.newInstance();
GiveUpRights giveUpRights= GiveUpRights.Factory.newInstance();
giveUpRights.setRequest(giveUpRightsRequest);
giveUpRightsDocument.setGiveUpRights(giveUpRights);
GiveUpRightsResponseDocument giveUpRightsResponseDocument =
GiveUpRightsResponseDocument.Factory.newInstance();
try
{
giveUpRightsResponseDocument = wws.giveUpRights(giveUpRightsDocument);
//Trasmesso
}
catch(RemoteException e)
{
//e.getMessage();
//e.printStackTrace();
}

```

3.15 SERVIZI PER LA GESTIONE DEL LIBROFIRMA

3.15.1 SERVIZIO “GETSIGNATUREPROCESSES”

Servizio che restituisce la lista dei processi o dei modelli di processo visibili e quindi utilizzabili dal ruolo.

3.15.1.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetSignatureProcessesRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
AuthenticationToken	String	Obbligatorio. Ottenibile

CodeApplication	String	<p>richiamato il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.</p> <p>Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.</p>
------------------------	--------	--

3.15.1.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il " GetSignatureProcessesResponse".

Estende la classe base astratta "Response".

In caso di errore, " GetSignatureProcesses" potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
APPLICATION_ERROR	Errore generico.

reponse

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Processes	SignatureProcess[]	Array dei processi o modelli di processo di firma visibili al Ruolo passato in input nel paramentro CodeRoleLogin .
TotalProcessesNumber	Int	Numero totale dei processi trovati

--	--	--

3.15.1.3 Caso d'uso : come ottenere la lista dei processi o modelli di processo visibili ad un ruolo.

Dopo avere ottenuto un token di autenticazione richiamando il servizio GEtAuthenticationToken (3.13.1) è possibile richiamare il WS GetSignatureProcesses fornendo come input il codRoleLogin ovvero il codice ruolo associato al ruolo d'interesse.

Se non si possiede il codice ruolo può essere ricercato utilizzando il WS SearchCorrespondent (3.12.6) oppure può essere richiesto a Informatica Trentina.

Il GetSignatureProcesses risponde riportando un array contenente tutti i processi o i modelli di processo visibili al ruolo.

Scorrendo l'array è possibile trovare il processo d'interesse conoscendo il NOME dello stesso.

3.15.2 SERVIZIO “STARTSIGNATUREPROCESS”

Servizio per inserire un documento in un processo di Libro Firma, avviando il processo stesso.

3.15.2.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “StartSignatureProcessRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al

		sistema.
AuthenticationToken	String	Obbligatorio. Ottenibile richiamato il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.
CodeApplication	String	Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.
IdDocument	String	Obbligatorio. Id del documento che si desidera utilizzare per inserirlo nel processo del Libro firma. (id documento è ottenibile con i servizi SearchDocument o in risposta ai i servizi come CreateDocument)
EndGeneratesNote	Boolean	Opzionale. Se True, Abilita la ricezione delle notifiche al ruolo passato in input con CodeRoleLogin di conclusione processo.
InterruptionGeneratesNote	Boolean	Opzionale. Se True, abilita la ricezione delle notifiche al ruolo passato in input come CodeRoleLogin di interruzione del processo.
Note	String(2000 char max)	Opzionale. Permette di aggiungere delle note al processo che saranno poi visibili nel libro firma.
SignatureProcess	SignatureProcess	Processo di Libro firma in cui inserire il documento
EndGeneratesNote	Boolean	Opzionale. Se True, Abilita la ricezione delle notifiche al ruolo passato in input con CodeRoleLogin di conclusione processo.

InterruptionGeneratesNote	Boolean	Opzionale. Se True, abilita la ricezione delle notifiche al ruolo passato in input come CodeRoleLogin di interruzione del processo.
Note	String(2000 char max)	Opzionale. Permette di aggiungere delle note al processo che saranno poi visibili nel libro firma.
SignatureProcess	SignatureProcess	Processo di Libro firma in cui inserire il documento

SignatureProcess

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
AuthorRoleId	String	Id ruolo del ruolo creatore del Libro Firma.
AuthorRoleId	String	Id people del utente creatore del Libro Firma.
IdProcess	String	Id univoco del processo di firma
IsProcessModel	Boolean	Se true allora si tratta di un modello di processo
Name	String	Nome del processo
SignatureStep	SignatureStep[]	Passi del processo

SignatureStep

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
IdProcess	String	Id univoco del processo di firma
IdStep	String	Id univoco del passo del processo di firma
InvolvedRole	Role	Ruolo coinvolto nel passo, Ruolo

		che deve effettuare il passo
InvolvedUser	User	Utente coinvolto nel passo, Utente che deve effettuare il passo
IsModel	Boolean	Indica se è un passo di modello
SequenceNumber	Int	Numero di sequenza del passo
Note	String	Note del passo

3.15.2.2 Descrizione oggetto “Response”

L’oggetto response restituito dal servizio è il “ StartSignatureProcessResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “ StartSignatureProcess” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

Codice	Descrizione
MISSING_PARAMETER	UserName o CodeAdm mancanti.
USER_NO_EXIST	Utente non trovato.
DOCUMENT_NOT_FOUND	Id del documento obbligatorio.
ROLE_NO_EXIST	Il ruolo non esiste.
DOCUMENTO BLOCCATO	Id documento corrisponde ad un documento bloccato perché già presente in altro processo di firma o perché già in corso di firma in altro processo di firma.
DOCUMENTO CONSOLIDATO	Id documento corrisponde ad un documento consolidato e non più firmabile.
DOCUMENTO GIA IN LIBRO FIRMA	Id documento corrisponde ad un documento bloccato perché già presente in altro processo di firma o perché già in corso di firma in altro processo di firma.
FILE NON AMMESSO ALLA FIRMA	File di formato non ammesso alla firma.
ERRORE GENERICO	Errore generico.

3.15.2.3 Caso d'uso avvio di un processo di firma inserendo un documento in esso.

Per inserire in documento in un Libro firma è necessario utilizzare il metodo `StartSignatureProcess`.

Per poter utilizzare il metodo è necessario prima avere ottenuto in token di autenticazione richiamando il metodo `GetAuthenticationToken` che richiede come parametri `Userid` dell'utente utilizzato per richiamare i WS PIS e il `codAmm` codice amministrazione forniti entrambi da Informatica Trentina. Il codice amministrazione è il codice dell'ente con il quale si desidera interagire attraverso i WS PIS.

Il WS `StartSignatureProcess` utilizza il ruolo di default associato alla `userid` passato come input. Se l'utenza fosse associata a più ruoli allora è necessario passare in input il parametro `CodeRoleLogin` utilizzando il Codice Ruolo che se non noto sarà fornito dai referenti informatici dell'ente o da Informatica Trentina.

L'ultimo parametro necessario è `id document`. `Id document` è l'identificativo numerico univoco del documento che si desidera inserire nel processo di Libro firma.

Per ottenerlo:

aver utilizzato precedentemente uno dei WS per creare documenti:

- `CreateDocumentAndAddInProject`
- `CreateDocument`

Che restituiscono `id document` del documento creato, oppure aver ricercato il documento utilizzando:

- `SearchDocuments`

In caso , fosse un allegato a dover essere inserito LF, una volta ottenuto `id document` del documento padre dell'allegato è possibile ottenere la lista di tutti gli allegati (prelevando poi quello ricercato) di un documento utilizzando il ws `getdocument` (3.9.4) passando in input `id document`.

Ora è necessario identificare il processo di Firma cui inserire il documento. È necessario conoscere il nome del processo di firma, se non lo si conosce può essere richiesto ai referenti informatici dell'ente o a Informatica Trentina.

Conoscendo il nome del processo è possibile ricercarlo utilizzando il metodo `GetsignatureProcess` (3.14.8).

Il processo può essere di due tipologie: processo o modello

Tale distinzione è riconoscibile dal fatto che nell'oggetto `SignatureProcess` è presente la variabile **`IsProcessModel`**.

Nel caso di processo, basta selezionare l'oggetto SignatureProcess prelevandolo dall'array dei risultati del GetSignatureProcess ed utilizzarlo come input nel StartSignatureProcess.

Se non si hanno errori, allora il documento sarà stato inserito con successo nel Libro Firma avviando il processo stesso.

Nel caso di Modello di processo è necessario identificare i passi del modello di processo andando a riempire i parametri relativi al ruolo (e utente opzionale, solo se richiesto dall'ente) che dovrà eseguire il passo.

Per fare ciò è necessario selezionare i passi del SignatureProcess identificati dalla array **SignatureStep**, prelevando quelli che riportano la variabile boolean IsModel a true.

Una volta identificati i passi di modello sarà necessario inserire un ruolo (e utente opzionale, solo se richiesto dall'ente) nei parametri

InvolvedRole

InvolvedUser.

Per poter inserire il ruolo e l'utente è necessario avere a disposizione gli oggetti Role e User che possono essere ricercati attraverso l'utilizzo del WS SearchCorrespondent (3.12.6) o conoscendo id correspondent o codice è possibile utilizzare direttamente il metodo GetCorrespondent (3.12.4).

Una volta inseriti i parametri è possibile invocare il metodo StartSignatureProcess.

Se non si hanno errori, allora il documento sarà stato inserito con successo nel Libro Firma avviando il processo stesso.

3.15.3 SERVIZIO “SEARCHSIGNPROCESSINSTANCES”

Servizio per ricercare e quindi monitorare lo stato delle istanze di processi di firma visibili ad un determinato ruolo.

3.15.3.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “SearchSignProcessInstancesRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.

CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
AuthenticationToken	String	Obbligatorio. Ottenibile richiamato il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.
CodeApplication	String	Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.
ElementsinPage	String	Numero di elementi per pagina di risultati ricerca, la ricerca è paginata. Di default il valore è 20.
PageNumber	String	Numero della pagina desiderata dei risultati della ricerca. La prima pagina è la 1.
Filters	Filter[]	Filtri per la ricerca

Filter

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Name	String	Nome filtro
Value	String	Valore da impostare per la ricerca
IdProcess	String	Id univoco del processo di firma
Descrizione	String	Descrizione filtro

Per conoscere i filtri può essere utilizzato il metodo GetInstanceSearchFilters (3.14.11).

3.15.3.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “ SearchSignProcessInstancesResponse”.

Estende la classe base astratta "Response".

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
SignatureProcessInstance	SignatureProcessInstance[]	Lista delle istanze di processi di firma in avviate visibili al ruolo passato in input con CodRoleLogin.
TotalNumber	Int	Numero totale elementi trovati.

SignatureProcessInstance

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ActionDate	String	Data attivazione
AttachmentDoc	String	
AttachmentNunumber	String	Id allegato
ClosureDate	String	Data di chiusura
CreationDate	String	Data di creazione
Description	String	Descrizione processo
DocNumber	String	Id documento
EndingNotification	Boolean	Se true è attiva la notifica di conclusione.
IdProcess	String	idProcesso
IdProcessInstance	String	Id dell'istanza di processo
InterruptedBy	String	Ruolo / utente che ha interrotto il processo. Se vuoto non ci sono state

		interruzioni.
InterruptedNotification	Boolean	Se true è attiva la notifica di Interruzione.
Object	String	Oggetto del documento o descrizione dell'allegato
ProcessState	String	Stato del processo. Valori (CLOSED, CLOSED_WITH_CUT(troncato), IN_EXEC(in esecuzione), STOPPED (interrotto).)
ProposerRole	Role	Ruolo proponente
ProposerUser	User	Utente proponente
ProtocolDate	String	Data protocollo se il documento ha subito un evento di protocollazione
ProtocolNumber	String	Numero di protocollo. Se il documento ha subito un evento di protocollazione
RefusalReason	String	Note di respingimento.
RepertorySignature	String	Segnatura di repertorio. Se il documento ha subito in evento di repertorizzazione
SignStepInstances	SignStepInstances[]	Array contenente tutti i passi del processo.
StartingNote	String	Note di avvio.
Truncated	boolean	Se true indica che il processo è stato troncato.
VersionId	String	Id versione del documento o dell'allegato.

VersionNumber	String	Numero di versione del documento o dell'allegato.
----------------------	--------	---

3.15.3.3 Caso d'uso del ws SearchSignProcessInstances

Per monitorare i processi di firma in corso e necessario trovare tutte le istanze di processi avviate visibili al ruolo.

Per fare questo si deve utilizzare il metodo SearchSignProcessInstances.

Questo metodo esegue una ricerca sulle istanze di processi visibili al ruolo e restituisce i risultati in modo paginato.

Per poter utilizzare il SearchSignProcessInstances è necessario utilizzare i filtri che sono a disposizione per poter inserire le chiavi di ricerca. Per poter utilizzare i filtri è necessario utilizzare il WS GetInstanceSearchFilters che permette di conoscere la definizione di tutti i filtri possibili.

Una volta impostate le chiavi di ricerca desiderate (ad es, vogliamo ricercare tutti le istanze associate al processo che ha NOME="processo di firma".., oppure vogliamo ricercare in quale stato è il processo avviato sul documento con ID documento =123345 utilizzando il filtro DOC_NUMBER..) è possibile avviare la ricerca.

Il risultato della ricerca è un array di SignProcessInstances. Dagli attributi dell'oggetto SignProcessInstances è possibile conoscere tutti i dettagli che permettono di monitorare il processi e il loro stato.

3.15.4 SERVIZIO "GETINSTANCESEARCHFILTERS"

Servizio per ottenere la lista dei filtri utilizzabile nel metodo SearchSignProcessInstances .

3.15.4.1 Descrizione oggetto "Request"

L'oggetto Request necessario per effettuare la chiamata al servizio è il "GetInstanceSearchFiltersRequest".

Struttura dell'oggetto:

Nome	Tipo Dato	Descrizione
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice

CodeRoleLogin	String	dell'amministrazione interessata. Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
AuthenticationToken	String	Obbligatorio. Ottenibile richiamato il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.
CodeApplication	String	Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.

Filter

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
Name	String	Nome filtro
Value	String	Valore da impostare per la ricerca
IdProcess	String	Id univoco del processo di firma
Descrizione	String	Descrizione filtro

3.15.4.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il "GetInstanceSearchFiltersResponse".

Estende la classe base astratta "Response". E contiene un array di Filters[] i cui valori ammessi sono i seguenti:

```
{ Name = "PROCESS_ID", Description = "ID del processo associato all'istanza di firma", Type = Domain.FilterTypeEnum.Number };
```

{ Name = "DOC_NUMBER", Description = "ID del documento associato all'istanza di firma", Type = Domain.FilterTypeEnum.Number };

{ Name = "NOTE", Description = "Stringa contenuta nelle note dell'istanza", Type = Domain.FilterTypeEnum.String };

{ Name = "START_DATE", Description = "Data di avvio dell'istanza di firma", Type = Domain.FilterTypeEnum.Date };

{ Name = "START_DATE_FROM", Description = "Data di avvio dell'istanza di firma successiva a", Type = Domain.FilterTypeEnum.Date };

{ Name = "START_DATE_TO", Description = "Data di avvio dell'istanza di firma precedente il", Type = Domain.FilterTypeEnum.Date };

{ Name = "START_NOTES", Description = "Stringa contenuta nelle note di avvio dell'istanza", Type = Domain.FilterTypeEnum.String };

{ Name = "END_DATE", Description = "Data di conclusione dell'istanza di firma", Type = Domain.FilterTypeEnum.Date };

{ Name = "END_DATE_FROM", Description = "Data di conclusione dell'istanza di firma successiva a", Type = Domain.FilterTypeEnum.Date };

{ Name = "END_DATE_TO", Description = "Data di conclusione dell'istanza di firma precedente il", Type = Domain.FilterTypeEnum.Date };

{ Name = "INTERRUPTION_DATE", Description = "Data di interruzione dell'istanza di firma", Type = Domain.FilterTypeEnum.Date };

{ Name = "INTERRUPTION_DATE_FROM", Description = "Data di interruzione dell'istanza di firma successiva a", Type = Domain.FilterTypeEnum.Date };

{ Name = "INTERRUPTION_DATE_TO", Description = "Data di interruzione dell'istanza di firma precedente il", Type = Domain.FilterTypeEnum.Date };

{ Name = "REFUSAL_NOTE", Description = "Stringa contenuta nelle note di respingimento dell'istanza", Type = Domain.FilterTypeEnum.String };

```
{ Name = "IN_EXECUTION", Description = "Processo di firma in esecuzione", Type = Domain.FilterTypeEnum.Bool };
```

```
{ Name = "INTERRUPTED", Description = "Processo di firma interrotto", Type = Domain.FilterTypeEnum.Bool };
```

```
{ Name = "ENDED", Description = "Processo di firma concluso", Type = Domain.FilterTypeEnum.Bool };
```

```
{ Name = "TRUNCATED", Description = "Processo di firma troncato", Type = Domain.FilterTypeEnum.Bool };
```

In caso di errore, “GetInstanceSearchFilters” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
USER_NO_EXIST	Utente non trovato.
ERRORE GENERICO	Errore generico.

3.15.5 SERVIZIO “GETSIGNATUREPROCESS”

Servizio per ottenere un processo a partire dal suo Id.

3.15.5.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “GetSignatureProcessRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al

AuthenticationToken	String	sistema. Obbligatorio. Ottenibile richiamato il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.
CodeApplication	String	Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.
IdProcess	String	Id del processo.

3.15.5.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il " GetSignatureProcessResponse".

Estende la classe base astratta "Response". L'oggetto response contiene un oggetto del tipo **SignatureProcess**

SignatureProcess

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
AuthorRoleId	String	Id ruolo del ruolo creatore del Libro Firma.
AuthorRoleId	String	Id people del utente creatore del Libro Firma.
IdProcess	String	Id univoco del processo di firma
IsProcessModel	Boolean	Se true allora si tratta di un modello di processo
Name	String	Nome del processo
SignatureStep	SignatureStep[]	Passi del processo

SignatureStep

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
IdProcess	String	Id univoco del processo di firma
IdStep	String	Id univoco del passo del processo di firma
InvolvedRole	Role	Ruolo coinvolto nel passo, Ruolo che deve effettuare il passo
InvolvedUser	User	Utente coinvolto nel passo, Utente che deve effettuare il passo
IsModel	Boolean	Indica se è un passo di modello
SequenceNumber	Int	Numero di sequenza del passo
Note	String	Note del passo

In caso di errore, “ GetSignatureProcess” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
USER_NO_EXIST	Utente non trovato.
ERRORE GENERICO	Errore generico.

3.15.6 SERVIZIO “GETSIGNPROCESSINSTANCE”

Servizio per ottenere una istanza del processo a partire dal suo Id istanza.

.

3.15.6.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “ GetSignProcessInstanceRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
AuthenticationToken	String	Obbligatorio. Ottenibile richiamato il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.
CodeApplication	String	Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.
IdProcessInstance	String	Id istanza di processo.

3.15.6.2 Descrizione oggetto "Response"

L'oggetto response restituito dal servizio è il " GetSignatureProcessResponse".

Estende la classe base astratta "Response". L'oggetto response contiene un oggetto del tipo **SignatureProcessInstance**

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
ActionDate	String	Data attivazione
AttachmentDoc	String	

AttachmentNumber	String	Id allegato
ClosureDate	String	Data di chiusura
CreationDate	String	Data di creazione
Description	String	Descrizione processo
DocNumber	String	Id documento
EndingNotification	Boolean	Se true è attiva la notifica di conclusione.
IdProcess	String	idProcesso
IdProcessInstance	String	Id dell'istanza di processo
InterruptedBy	String	Ruolo / utente che ha interrotto il processo. Se vuoto non ci sono state interruzioni.
InterruptedNotification	Boolean	Se true è attiva la notifica di Interruzione.
Object	String	Oggetto del documento o descrizione dell'allegato
ProcessState	String	Stato del processo. Valori (CLOSED, CLOSED_WITH_CUT(troncato), IN_EXEC(in esecuzione), STOPPED (interrotto).)
ProposerRole	Role	Ruolo proponente

ProposerUser	User	Utente proponente
ProtocolDate	String	Data protocollo se il documento ha subito un evento di protocollazione
ProtocolNumber	String	Numero di protocollo. Se il documento ha subito un evento di protocollazione
RefusalReason	String	Note di respingimento.
RepertorySignature	String	Segnatura di repertorio. Se il documento ha subito in evento di repertorizzazione
SignStepInstances	SignStepInstances[]	Array contenente tutti i passi del processo.
StartingNote	String	Note di avvio.
Truncated	boolean	Se true indica che il processo è stato troncato.
VersionId	String	Id versione del documento o dell'allegato.
VersionNumber	String	Numero di versione del documento o dell'allegato.

In caso di errore, “ GetSignProcessInstance” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
USER_NO_EXIST	Utente non trovato.
ERRORE GENERICO	Errore generico.

3.15.7 SERVIZIO “INTERRUPTSIGNATUREPROCESS”

Servizio per interrompere una istanza del processo a partire dal suo Id istanza.

3.15.7.1 Descrizione oggetto “Request”

L'oggetto Request necessario per effettuare la chiamata al servizio è il “InterruptSignatureProcessRequest”.

Struttura dell'oggetto:

<i>Nome</i>	<i>Tipo Dato</i>	<i>Descrizione</i>
UserName	String	Obbligatorio. Nome utente censito nel sistema di gestione documentale.
CodeAdm	String	Obbligatorio. Codice dell'amministrazione interessata.
CodeRoleLogin	String	Opzionale: Codice del ruolo con cui si vuole accedere al sistema.
AuthenticationToken	String	Obbligatorio. Ottenibile richiamando il servizio GetAuthenticationToken descritto nel paragrafo 3.13.1.
CodeApplication	String	Opzionale. Dare un nome qualsiasi all'applicazione chiamante per poi utilizzarlo in tutte le chiamate passandolo con questa stringa.
IdProcessInstance	String	Id istanza di processo.

3.15.7.2 Descrizione oggetto “Response”

L'oggetto response restituito dal servizio è il “InterruptSignatureProcessResponse”.

Estende la classe base astratta “Response”.

In caso di errore, “InterruptSignatureProcess” potrà restituire un'eccezione di tipo PISFaultException con i codici di errore riepilogati nella seguente tabella:

<i>Codice</i>	<i>Descrizione</i>
----------------------	---------------------------

USER_NO_EXIST	Utente non trovato.
ERRORE GENERICO	Errore generico.