LegalDoc – Servizio di Conservazione a norma

SPT/NDOC – Specifiche tecniche per l'integrazione di LegalDoc

1.Introduzione al documento....5
1.1Novità introdotte rispetto la precedente emissione....5
1.2Scopo e campo di applicazione del documento....5
1.3Riferimenti....5
1.4Termini e definizioni....5
2.Introduzione....9
2.1Descrizione del servizio....9
2.2Il documento in LegalDoc....9
3.Architettura generale....10
3.1Flusso di conservazione....10
3.2Dettaglio dei passi di conservazione....11
3.2.1L'indice di conservazione....12
4.Architettura tecnica....13
4.1Prerequisiti tecnici....13
4.2Esposizione dei servizi....13
4.3Formato dei dati di scambio....13
4.4Risposta del sistema....14
5.Operazioni preliminari per l'utilizzo di LegalDoc....15
5.1Dati di configurazione....15
5.1.1Informazioni di configurazione definite in fase contrattuale....15
5.2Dimensionamento dei dati inviati dal Cliente alla conservazione....15
5.3Altre specifiche....16
5.3.1Lunghezze massime valori dei parametri di conservazione....16
5.3.2Caratteri ammessi nei nomi dei file....16
5.3.3Numero massimo di documenti estratti dalla funzione di ricerca....16
5.4Per ottenere assistenza....16
6.Policy LegalDoc....18
7.Funzioni di autenticazione....19
7.1Prerequisiti, url e invocazione del servizio....19
7.1.1Input – Output di autenticazione....19
7.2Login....19
7.2.1Input....19
7.2.2Output....20
7.3Logout....23
7.3.1Input....23
7.3.2Output....24
7.4Verifica dello stato della sessione....25
7.4.1Input....26
7.4.2Output....26

8.Funzione di conservazione....29
8.1Prerequisiti, url e invocazione del servizio....29
8.2Input – output del processo di conservazione....29
8.2.1Input....29
8.2.2Output....30
9.Il file dei parametri di conservazione....35
10.Il file di indice....38
10.1Conservazione del file di indici per la ricerca....38
10.2Regole per la conservazione del file di indici per la ricerca....39
10.3Controlli sul file di indice....42
11.Richiesta di esibizione....44
11.1Prerequisiti, url e invocazione del servizio....44
11.2Input – output del processo di esibizione....44
11.2.1Input....44
11.2.2Output....45
12.Funzione di Rettifica....47
12.1Che cosa si intende per Rettifica di un documento....47
12.2Esibizione di un documento rettificato e di un documento rettificante....47
12.3Input – output del processo di rettifica....48
13.Funzione di Cancellazione....49
13.1Che cosa si intende per cancellazione di un documento....49
13.2Esibizione di un documento cancellato e di un documento cancellante....49
13.3Input – output del processo di cancellazione....49
14.Funzioni di ricerca....51
14.1Prerequisiti, url e invocazione del servizio....51
14.1.1Input....51
14.1.2Output....53
15.Gestione dei visualizzatori....55
15.1Regole relative al caricamento dei visualizzatori per i formati predefiniti....55
15.2Regole relative al caricamenti dei visualizzatori per formati aggiuntivi....55
15.3Aggiornamento dei visualizzatori a nuove versioni....56
15.4Caricamento di visualizzatori mediante servizio REST....56
15.4.1Prerequisiti, url e invocazione del servizio....56
15.4.2Input....56
15.4.3Output....57
15.4.4Il file dei parametri di upload....61
15.4.5Controlli sul file dei parametri di upload....62
15.4.6Download dei visualizzatori....63

15.4.7Prerequisiti, url e invocazione del servizio....63
15.4.8Input....63
15.4.9Output....64

16.Invocazione Web Services LegalDoc....65

16.1Servizi esposti....65
16.2Composizione dell'URL da invocare per accedere a LegalDoc....65
16.3Esempi di invocazione dei servizi in Java....67
16.3.1Librerie....67
16.3.2Esempio metodo chiamata ws generico....67
16.3.3Esempio metodo chiamata ws login....68
16.3.4 Esempio di logout....68
16.3.5Esempio di conservazione....69
16.3.6Esempio di esibizione zip....69
16.3.7Esempio di esibizione multipart....70
16.3.8Esempio di ricerca....70

17.Appendice A: XSD file utilizzati in LegalDoc....71

17.1XSD del file dei parametri di conservazione....71
17.2XSD del file di indice....73
17.3XSD del file IdC....74

1. Introduzione al documento

Il presente documento di specifiche del servizio di conservazione è parte integrante del Manuale della conservazione, di InfoCert in quanto soggetto accreditato da Agid.

1.1 Novità introdotte rispetto la precedente emissione

Aggiunta descrizione del tag moreInfo nell'XSD del file IdC nel paragrafo 17.3

1.2 Scopo e campo di applicazione del documento

Il documento ha l'obiettivo di:

- Fornire la descrizione generale del prodotto LegalDoc e del flusso di conservazione
- Fornire i dettagli tecnici dei passi da eseguire per integrare LegalDoc in un generico Gestore Documentale

Il documento si applica a LegalDoc, servizio di Conservazione dei documenti digitali

1.3 Riferimenti

[1] Scheda dati tecnici per l'attivazione di LegalDoc
[2] Condizioni generali di contratto LegalDoc
[3] Allegato Tecnico al contratto LegalDoc
[4] Descrizione dei codici di errore di LegalDoc
[5] Manuale dei processi di conservazione LegalDoc
[6] Home Page dell'Agenzia delle Entrate: http://www.agenziaentrate.gov.it

1.4 Termini e definizioni

I termini e le definizioni, presenti in questo documento di specifiche tecniche, sono solo una parte delle definizioni presenti nel glossario del Manuale della conservazione.

AgID = Agenzia per l'Italia Digitale[1]

ESIBIZIONE (*esibizione a norma*) = estrazione di un documento conservato: il documento richiesto viene ottenuto direttamente dal servizio di conservazione ed esibito assieme alle informazioni che qualificano il processo di conservazione stessa, dando così garanzia della sua opponibilità a terzi

[1] Già DigitPA e precedentemente CNIPA.

EVIDENZA INFORMATICA = una sequenza di simboli binari (*bit*) che può essere elaborata da una procedura informatica (Definizione presente nell'Allegato 1 del DPCM DPCM del 3 dicembre 2013)

FILE DEI PARAMETRI DI CONSERVAZIONE = file in formato XML creato dal GD del Cliente al momento dell'invio alla conservazione di un documento, contenente fra l'altro informazioni relative alla composizione e alla classificazione del documento ai fini della conservazione (tipologia, file di dati, file di indici, percorso di conservazione, impronte di tutti i file che lo compongono)

FILE DEI PARAMETRI DI RETTIFICA = file in formato XML creato dal GD del Cliente al momento dell'invio alla conservazione di un documento, contenente fra l'altro informazioni relative alla composizione e alla classificazione del documento ai fini della conservazione (tipologia, file di dati, file di indici, percorso di conservazione, impronte di tutti i file che lo compongono)

FILE IdC = file creato da LegalDoc a seguito di una richiesta di conservazione e inviato in risposta a una richiesta di conservazione; il file attesta la corretta conservazione da parte di LegalDoc. E' in formato P7M e contiene la firma elettronica con certificato software apposta da LegalDoc. Rispetta lo standard UniSincro. La norma fa riferimento a tale file con il termine IPdA, ovvero Indice del Pacchetto di Archiviazione

FIRMA DIGITALE = così come definita all'art. 1 comma 1 lettera n DPR 445 del 28 dicembre 2000

FIRMA ELETTRONICA = così come definita all'art. 1 comma 1 lettera cc DPR 445 del 28 dicembre 2000

FUNZIONE DI HASH = una funzione matematica che genera, a partire da una evidenza informatica, una impronta in modo tale che risulti di fatto impossibile, a partire da questa, ricostruire l'evidenza informatica originaria e generare impronte uguali a partire da evidenze informatiche differenti (Definizione presente in Allegato 1 del DPCM DPCM del 3 dicembre 2013)

ID DI SESSIONE = valore ottenibile mediante la login al sistema LegalDoc, necessaria per invocare le funzionalità del sistema LegalDoc.

IdC = *Indice di conservazione*. Detto anche File IdC

IETF = *Internet Engineering Task Force* Ente preposto alla definizione di protocolli standard su Internet

IMPRONTA (o HASH) = la sequenza dei simboli binari (*bit*) di lunghezza predefinita generata mediante l'applicazione alla prima di una opportuna funzione di *hash* (Definizione presente in Allegato 1 del DPCM del 3 dicembre 2013)

MARCA TEMPORALE = così come definita all'art.1 comma 1 lettera i DPCM del 13 gennaio 2004

PACCHETTO DI ARCHIVIAZIONE = pacchetto informativo composto dalla trasformazione di uno o più pacchetti di versamento secondo le specifiche contenute nell'allegato 4 del decreto DPCM del 3 dicembre 2013 e secondo le modalità riportate nel manuale di conservazione (Definizione presente in Allegato 1 del DPCM del 3 dicembre 2013)

PACCHETTO DI DISTRIBUZIONE = pacchetto informativo inviato dal sistema di conservazione all'utente in risposta ad una sua richiesta (Definizione presente in Allegato 1 del DPCM del 3 dicembre 2013)

PACCHETTO DI VERSAMENTO = pacchetto informativo inviato dal produttore al sistema di conservazione secondo un formato predefinito e concordato descritto nel manuale di conservazione (Definizione presente in Allegato 1 del DPCM del 3 dicembre 2013)

PACCHETTO INFORMATIVO = contenitore che racchiude uno o più oggetti da conservare (documenti informatici, fascicoli informatici, aggregazioni documentali informatiche), oppure anche i soli metadati riferiti agli oggetti da conservare (Definizione presente in Allegato 1 del DPCM del 3 dicembre 2013)

PDF = *Portable Document Format*. E' un formato di file che cattura, in formato elettronico, tutti gli elementi di un documento stampato in modo che sia possibile visualizzarli, stamparli e scambiarli fra più utenti. E' un formato proprietario di Adobe e per creare file PDF sono necessari Adobe Acrobat, Adobe Capture o prodotti simili. Viceversa per visualizzare un file PDF è necessario Adobe Reader disponibile gratuitamente

PDV = *Pacchetto di versamento*. Pacchetto informativo inviato dal produttore al sistema di conservazione secondo un formato predefinito e concordato descritto nel manuale di conservazione

PRODUTTORE = persona fisica o giuridica, di norma diversa dal soggetto che ha formato il documento, che produce il pacchetto di versamento ed è responsabile del trasferimento del suo contenuto nel sistema di conservazione. Nelle pubbliche amministrazioni, tale figura si identifica con responsabile della gestione documentale (Definizione presente in Allegato 1 del DPCM del 3 dicembre 2013)

RESPONSABILE DELLA CONSERVAZIONE: Il soggetto cui sono attribuite funzioni, adempimenti, attività e responsabilità relative al servizio di conservazione, previsto all'art. 7 del DPCM 03/12/2013.

RESPONSABILE DEL SERVIZIO DI CONSERVAZIONE: Il soggetto che su delega del Responsabile della Conservazione gestisce le politiche generali del sistema di conservazione, nel rispetto del modello organizzativo esplicitato nel presente Manuale e di quanto previsto nelle Condizioni Generali del servizio di conservazione LegalDoc , di seguito per brevità chiamato anche “Contratto”, perfezionato tra InfoCert ed il Soggetto Produttore medesimo.

REST = *Representational State Transfer.* REST si riferisce ad un insieme di principi di architetture di rete, i quali delineano come le risorse sono definite e indirizzate. Un concetto importante in REST è l'esistenza di **risorse** (fonti di informazioni), a cui si può accedere tramite un identificatore globale (un URI). Per utilizzare le risorse, le *componenti* di una rete (componenti client e server) comunicano attraverso una interfaccia standard (ad es. HTTP) e si scambiano *rappresentazioni* di queste risorse.

TOKEN LEGALDOC = identificativo univoco di un documento generato da LegalDoc all'atto della richiesta di conservazione, rettifica e cancellazione. Tale identificativo viene utilizzato dal GD del Cliente per ogni successiva richiesta a LegalDoc.

VIEWER = un programma software che permette di leggere, e usualmente anche stampare (ma non modificare), un file nel proprio formato nativo (ovvero il formato usato internamente per rappresentare l'informazione)

XML = *Extensible Markup Language*. Linguaggio derivato dall’SGML (Standard Generalized Markup Language) il metalinguaggio, che permette di creare altri linguaggi. Mentre l’HTML è un’istanza specifica dell’SGML, XML costituisce a sua volta un metalinguaggio, più semplice dell’SGML, largamente utilizzato per la descrizione di documenti sul Web. L’XML viene utilizzato per definire le strutture dei dati invece

che per descrivere come questi ultimi devono essere presentati. Tali strutture vengono definite utilizzando dei marcatori (markup tags). Diversamente dall'HTML, l'XML consente all'utente di definire marcatori personalizzati, dandogli il controllo completo sulla struttura di un documento. Si possono definire liberamente anche gli attributi dei singoli marcatori.

2. Introduzione

2.1 Descrizione del servizio

Il servizio **LegalDoc** offre a pubbliche amministrazioni, imprese e professionisti un sistema di conservazione per mantenere e garantire nel tempo l'integrità e la disponibilità di un documento informatico, nel rispetto della normativa vigente.

All'interno del ciclo di vita di un documento, LegalDoc gestisce la sua fase finale relativa alla conservazione digitale ai fini del mantenimento della validità nel tempo.

In questo documento si vuole:

- fornire una presentazione funzionale del sistema e del flusso di conservazione
- affrontare tematiche più tecniche, spiegando in dettaglio i passi e gli skill necessari per scrivere un modulo di integrazione al servizio.

2.2 Il documento in LegalDoc

In LegalDoc un documento è un insieme di più file digitali.

Ogni documento, infatti, è composto da:

- un file XML descrittivo (***file dei parametri di conservazione***, descritto in seguito)
- un file XML di indici di ricerca (***file di indici del documento***, descritto in seguito)
- un file di dati (***file di dati del documento***, descritto in seguito)

Ad ogni documento, inoltre, è associato un identificativo univoco generato da LegalDoc ("**Token LegalDoc**"). In LegalDoc un documento rappresenta l'unità minima di elaborazione nel senso che viene memorizzato ed esibito come un tutt'uno. Non è possibile estrarre da LegalDoc parti singole di un documento.

3. Architettura generale

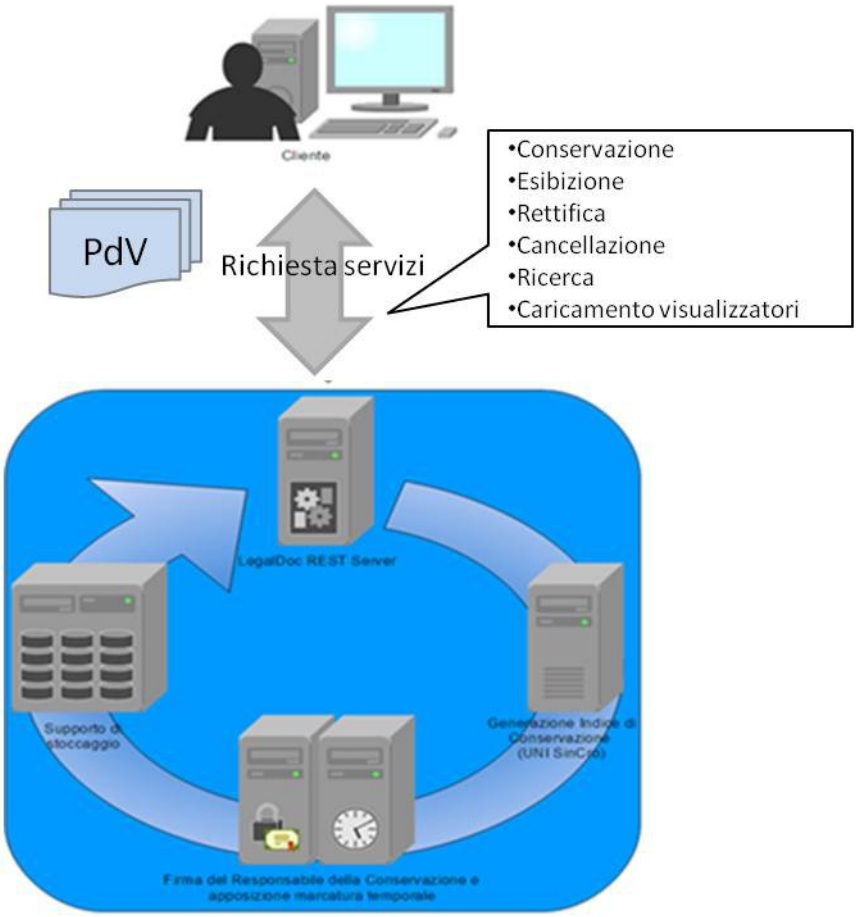

Figura 1Architettura generale LegalDoc

La Figura 1 mostra l'architettura generale del sistema LegalDoc, evidenziando gli attori che interagiscono con il sistema.

3.1 Flusso di conservazione

In Figura [2]è disegnato il flusso applicativo di conservazione.

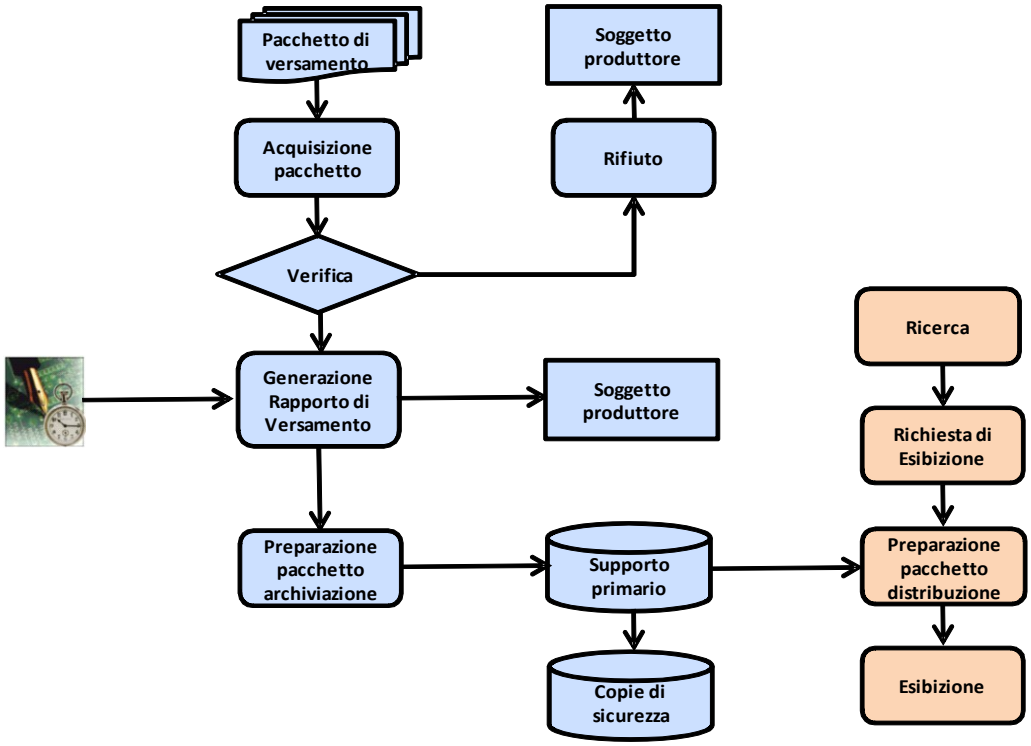

Figura 2 Flusso di conservazione

3.2 Dettaglio dei passi di conservazione

Vediamo in dettaglio i passi del flusso di conservazione (per la corretta lettura è consigliabile avere come riferimento le figure [1] e [2]):

Il cliente, lavorando sulla propria workstation, utilizza un Gestore Documentale (sistema di archiviazione generico, di seguito GD)per operare con il proprio documento. Per esempio potrebbe trattare documenti cartacei. Utilizzando uno scanner procede all'acquisizione di un documento e al suo salvataggio su file in formato elettronico (per esempio in formato PDF)

1. Il GD fornirà con opportuna interfaccia una modalità di fruizione dei servizi LegalDoc. Per esempio un GD su Web potrebbe avere un link del tipo: ***"Conserva in LegalDoc…"***
2. L'utente interagisce con l'interfaccia del sistema per richiedere i servizi LegalDoc. Riprendendo l'esempio del punto precedente fa click con il mouse sul link citato
3. Il GD crea una sessione di lavoro fornendo le credenziali di autenticazione al sistema LegalDoc e avendo cura di recuperare l'identificativo della sessione di lavoro stessa che viene restituito da LegalDoc
4. Il GD genera il documento formato, come detto in precedenza, dal file XML dei parametri di conservazione, il file XML di indici di ricerca e il file di dati (nell'esempio il file PDF

del punto 1) e lo spedisce a LegalDoc mediante una chiamata REST (nella chiamata verrà anche incluso l'identificativo di sessione ottenuto al punto 4)

5. LegalDoc acquisisce la richiesta di conservazione
6. LegalDoc preleva i file costituenti il documento ed esegue una serie di verifiche sulla completezza e la correttezza delle informazioni presenti nei file XML dei parametri di conservazione e indici di ricerca. In particolare, tra i diversi controlli eseguiti da LegalDoc, vi è quello sull'integrità dei dati inviati eseguita generando l'impronta di ogni file del documento e confrontata con la corrispondente impronta inviata all'interno del file XML dei parametri di conservazione
7. Se le verifiche falliscono il documento viene respinto e viene spedito un messaggio HTTP al GD descrivente l'errore intercorso. Il flusso in questo caso termina
8. LegalDoc memorizza il documento sul supporto primario previsto dal Manuale della Conservazione, lo trasferisce automaticamente presso il sito di disaster recovery e restituisce al GD il file xml indice di conservazione (denominato nel seguito 'IdC') firmato dal Responsabile della Conservazione e marcato temporalmente, contenente al suo interno le informazioni sul processo di conservazione (fra cui anche il Token LegalDoc ovvero l'identificativo univoco del documento appena conservato all'interno del sistema LegalDoc)
9. Il GD salva il file indice di conservazione ricevuto da LegalDoc, e, qualora avesse esaurito le operazioni di conservazione da eseguire verso LegalDoc effettua il logout dal sistema LegalDoc per chiudere correttamente la sessione di lavoro aperta al punto 4

3.2.1 L'indice di conservazione

Il file indice di conservazione è un file xml che rispetta lo standard SInCRO - Supporto all'Interoperabilità nella Conservazione e nel Recupero degli Oggetti digitali (UNI 11386:2010), che è lo standard nazionale riguardante la struttura dell'insieme dei dati a supporto del processo di conservazione. In analogia allo standard SInCRO, tale file prevede una specifica articolazione per mezzo del linguaggio formale XML, e individua gli elementi informativi necessari alla creazione dell'indice del pacchetto di archiviazione.

4. Architettura tecnica

4.1 Prerequisiti tecnici

Per poter integrare LegalDoc in un'applicazione GD sono necessari:

- XML
- Web Service (protocollo HTTPS con paradigma REST)

4.2 Esposizione dei servizi

LegalDoc offre ad un generico cliente i seguenti servizi:

- conservazione di un documento
- esibizione di un documento
- cancellazione di un documento
- rettifica di un documento
- ricerca di documenti indicizzati
- caricamento dei visualizzatori

utilizzando la tecnologia di interoperabilità più promettente attualmente disponibile sul mercato, la tecnologia dei Web Services.

I Web Services di LegalDoc sono implementati secondo REST su protocollo HTTPS.

Nel capitolo 16.1 vengono riassunti i metodi da invocare per ognuno dei servizi appena citati.

Nei successivi capitoli si descriveranno in dettaglio le strutture dei file scambiati e le modalità di fruizione dei servizi LegalDoc.

4.3 Formato dei dati di scambio

Lo scambio di dati e di informazioni tra il GD e LegalDoc avviene prevalentemente in due modalità che si completano vicendevolmente:

- file in formato XML
- informazioni nelle strutture previste dal protocollo HTTP

In Appendice A vengono pubblicati gli XSD dei "messaggi" XML utilizzati (vedi capitolo 17 Appendice A: XSD file utilizzati in LegalDoc).

4.4 Risposta del sistema

Il cliente che invoca i servizi offerti da LegalDoc ottiene sempre un codice HTTP (HTTP status code) indicante l'esito della chiamata. In aggiunta, a seconda del servizio invocato, può ricevere in risposta o un messaggio XML o un file.

E' a cura del GD la memorizzazione delle informazioni contenute nelle risposte di LegalDoc, nel caso possano essere utilizzate in successive richieste al sistema.

5. Operazioni preliminari per l'utilizzo di LegalDoc

5.1 Dati di configurazione

L'utilizzo di LegalDoc da parte di un GD richiede la definizione, in via preliminare, di informazioni specifiche fornite dal Cliente e concordate con InfoCert in fase contrattuale. A tale scopo è predisposto un opportuno modulo di richiesta attivazione (vedi rif. [1]). In tale modulo sono definiti e valorizzati i parametri necessari ad attivare il servizio. Nel seguente paragrafo vengono riportati, per comodità del lettore e per completezza di trattazione, alcuni dei parametri di configurazione succitati, ma l'elenco non è esaustivo. Il riferimento ufficiale è il documento [1] ed eventuali discrepanze fra questo elenco e il documento [1] si risolvono a favore del documento [1].

5.1.1 Informazioni di configurazione definite in fase contrattuale

- Identificativo GD: username identificativa del gestore documentale
- Password GD: password di autenticazione della username del GD
- Spazio di conservazione: bucket associato al GD in cui verranno ospitati i suoi documenti
- Regole di trattamento dei dati: saranno definite uno o più policy di trattamento dei dati (vedi capitolo 6.)
- Insieme dei tipi di file (MIME/TYPE) accettabili
- Classi documentali: saranno configurate le classi documentali attribuibili ai documenti inviati in conservazione

5.2 Dimensionamento dei dati inviati dal Cliente alla conservazione

Di seguito sono indicate le limitazioni massime relative al numero e al dimensionamento dei file costituenti i documenti inviati dal Cliente al servizio Legaldoc. Si specifica che il “file dei parametri di conservazione”, che deve essere allegato a cura del Cliente per ogni documento inviato alla conservazione, non entra a far parte del conteggio oggetto delle limitazioni:

Dimensione massima file di dati (in MB)	*Dimensione massima file XML di indici (in MB)*
1024	1

Tabella 1Dimensionamento dei documenti inviati alla conservazione

Relativamente al deposito dei visualizzatori, ogni visualizzatore deve essere composto di un unico file eseguibile o installabile in un particolare ambiente operativo. La dimensione massima di un visualizzatore è fissata in:

50 MB (MegaByte)

5.3 Altre specifiche

5.3.1 Lunghezze massime valori dei parametri di conservazione

Si riportano le dimensioni massime (in caratteri) dei campi presenti nel file dei parametri di conservazione:

policy_id = 16

data_name = 80

data_hash = 64

index_name = 80

index_hash = 64

path = 256

5.3.2 Caratteri ammessi nei nomi dei file

I valori dei nomi dei file di dati e di indici che vengono spediti al sistema di conservazione possono contenere i seguenti caratteri:

- lettere stampabili maiuscole e minuscole (a...zA...Z)
- cifre numeriche decimali (0...9)
- i caratteri _.@- e lo spazio (blank)

5.3.3 Numero massimo di documenti estratti dalla funzione di ricerca

Il numero massimo di documenti estraibili con una ricerca è fissato in

200 occorrenze

Nel caso in cui il numero di documenti che soddisfano il criterio di ricerca ecceda tale valore massimo, verranno restituiti soltanto 200 risultati che identificano i 200 documenti più recenti che soddisfano il criterio di ricerca indicato. Per maggiori informazioni si veda 14 Funzioni di ricerca.

5.4 Per ottenere assistenza

Per qualsiasi problema è possibile contattare l'assistenza LegalDoc all'indirizzo di posta elettronica indicato nell'allegato tecnico dalla casella di posta certificata del Cliente.

In caso di problemi nell'utilizzo del servizio specificare nella richiesta di assistenza le seguenti informazioni:

- ambiente che si sta utilizzando (collaudo o produzione)
- data e ora in cui si è verificato il problema
- dati scambiati con LegalDoc (file dei parametri, di indici e di dati) e servizio invocato
- userid e identificativo della sessione di lavoro (ldSessionId)

- risposta del sistema (se non disponibile inviare una dettagliata descrizione dell'errore occorso)
- documento inviato nella richiesta (per richieste di conservazione e rettifica)
- token se previsto dal servizio utilizzato (es: rettifica ed esibizione)

6. Policy LegalDoc

Ogni cliente che utilizza LegalDoc deve aver preventivamente attivato una o più policy.

Una policy è un insieme di regole che definiscono il trattamento che il documento riceve all'interno al sistema di conservazione LegalDoc.

La policy da applicare ad un documento viene indicata nel file dei parametri tramite l'apposito tag <policy_id>.

Le regole associate alla policy definiscono:

- i mimetype assegnabili ad un file di dati nell'apposito tag <data_mimetype> presente nel file xml dei parametri (i valori possibili sono comunicati tramite l'apposito file di configurazione che si riceve all'atto dell'attivazione)
- il periodo di retention del documento
- le classi documentali assegnabili al documento attraverso l'attributo documentClass del tag <legaldocIndex> del file xml di indici
- il firmatario del documento (opzionale)
- l'invio via PEC del documento (opzionale)
- la cifratura del documento (opzionale)

7. Funzioni di autenticazione

7.1 Prerequisiti, url e invocazione del servizio

Per poter utilizzare i servizi messi a disposizione da LegalDoc è necessario iniziare una sessione di lavoro mediante autenticazione al sistema con userid e password ed ottenere il valore di **ldsessionid**, che rappresenta l'identificativo unico della sessione di lavoro, da inserire nell'omonimo header delle successive chiamate ai servizi di LegalDoc così come illustrato nei paragrafi successivi.

Ad ogni sessione di lavoro è anche associato un **pdv** (acronimo di Pacchetto Di Versamento, vedi 1.4 Termini e definizioni). Il pacchetto di versamento viene aperto al momento dell'inizio della sessione di lavoro e viene chiuso al momento dell'esecuzione della chiamata per terminare la sessione di lavoro.

L'URL da invocare per accedere ai servizi Legaldoc è:

{Legaldoc.URL}/session

Elenchiamo qui di seguito, per ognuno dei servizi di autenticazione, il relativo nome del web service e il metodo http da invocare:

Web service	Metodo HTTP	Descrizione
session	POST	Inizia una sessione di lavoro autenticando l'utente
session	DELETE	Termina la sessione di lavoro dell'utente
session	GET	Verifica la validità della sessione di lavoro

7.1.1 Input – Output di autenticazione

La modalità di inoltro dei dati varia da servizio a servizio, mentre per la ricezione dei dati di esito viene utilizzato sempre il formato XML.

Nei successivi capitoli si descriverà in dettaglio la modalità di scambio dei dati per ognuno dei servizi offerti.

7.2 Login

7.2.1 Input

E' sufficiente autenticarsi (utilizzando il metodo Http POST) attraverso un form di login https.

I campi di input devono avere nome `userid` e `password` ed il content-type deve essere `application/x-www-form-urlencoded`

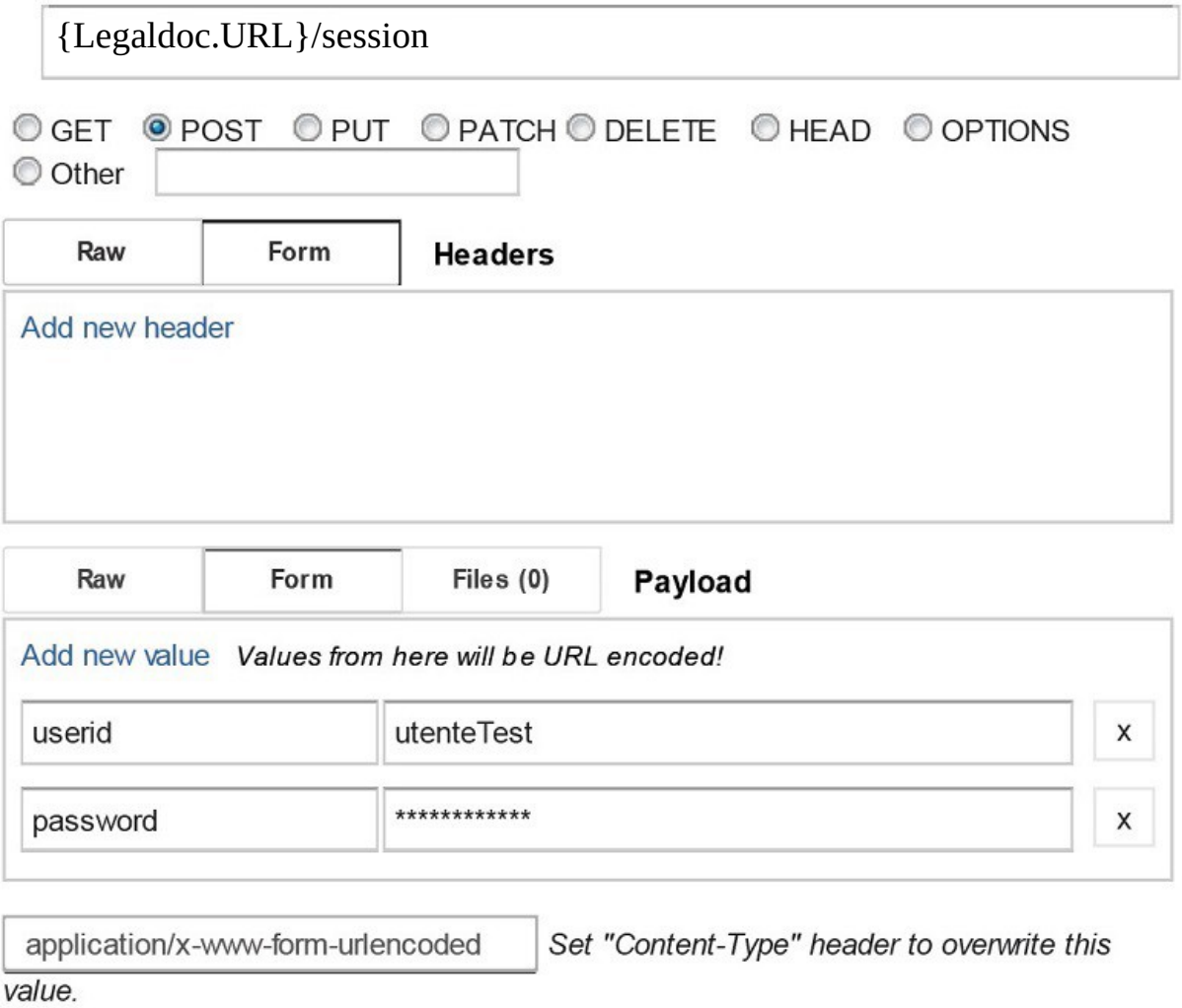

Figura 3: Esempio di utilizzo dei parametri per la creazione di una sessione di lavoro in un client per l'invocazione di servizi REST.

7.2.2 Output

Alla richiesta di login il sistema risponde con un file simile in caso di risposta positiva:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<loginResponse>
    <code>OK</code>
    <ldSessionId>1234</ldSessionId>
    <pdv>PDV587</pdv>
</loginResponse>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
`loginResponse`	Tag radice per la risposta ad una richiesta di login			S
`code`	Codice di risposta. OK se la richiesta ha avuto esito positivo			S
`ldSessionid`	E' l'identificativo univoco della sessione di lavoro			S
`pdv`	E' il valore dell'identificativo di versamento			S

Figura 4: Esempio di risposta con successo alla creazione di una sessione di lavoro.

mentre in caso di risposta negativa otterremo un file simile a questo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<error>
    <code>LD_AU100</code>
    <description>Credenziali non valide</description>
</error>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
`error`	Tag radice per la risposta ad una richiesta di login			S
`code`	Indica il codice di errore			S
`description`	Riporta una breve descrizione dell'errore			S

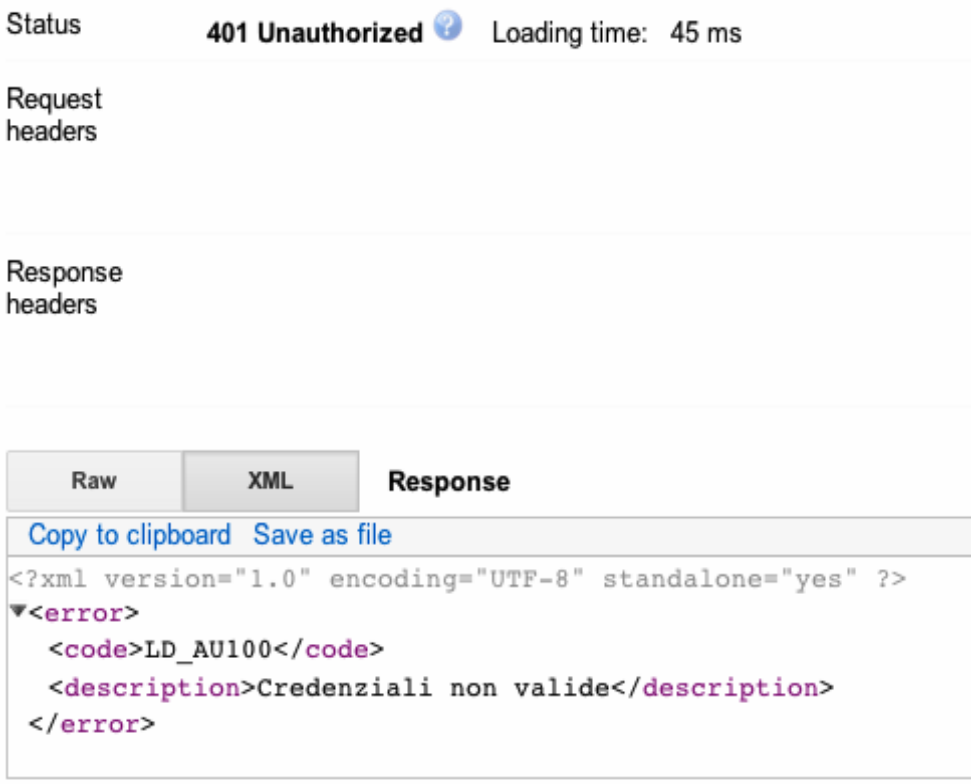

Figura 5: Esempio di risposta con fallimento alla creazione di una sessione di lavoro.

7.3 Logout

7.3.1 Input

Per effettuare la logout dal sistema si deve richiamare lo stesso url della login ma con metodo http DELETE e content type `application/x-www-form-urlencoded`

Inoltre bisogna aggiungere l'header con nome `ldSessionId` e valore corrisponde alla **ldsessionid** restituita dal servizio di login.

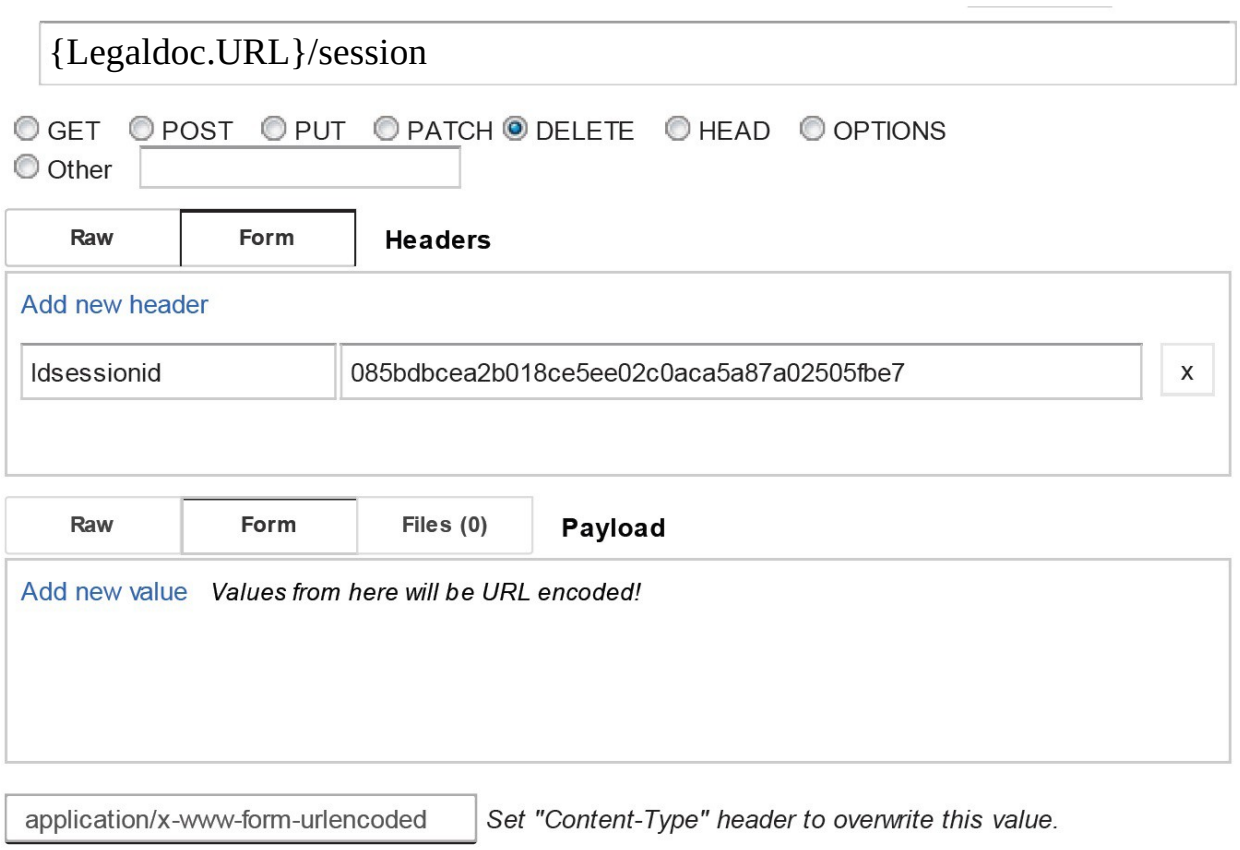

Figura 6: Esempio di utilizzo dei parametri per la chiusura di una sessione di lavoro in un client per l'invocazione di servizi REST.

7.3.2 Output

A seguito di una chiamata di logout con esito positivo otterremo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<logoutResponse>
    <code>OK</code>
</logoutResponse>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
`logoutResponse`	Tag radice per la risposta ad una richiesta di logout			S
`code`	Codice di risposta. Se la richiesta ha avuto esito positivo indica il codice OK			S

Status code **200 OK**

Time 733 ms

Headers
Date: Fri, 28 Sep 2012 12:14:16 GMT
Content-Encoding: gzip
X-Powered-By: AS Infocamere - newlegaldoc1
Transfer-Encoding: chunked
Connection: Keep-Alive
Server: WEB SERVER INFOCERT
Vary: Accept-Encoding,User-Agent
Content-Language: it
Content-Type: application/xml
Keep-Alive: timeout=15, max=100

Body **Raw response** **XML**

```
<?xml version="1.0" encoding="UTF-8"?>
<logoutResponse>
  <code>OK</code>
</logoutResponse>
```

Figura 7: Esempio di risposta con successo alla chiusura di una sessione di lavoro.

In caso di risposta negativa, LegalDoc restituirà un file xml contenente una risposta simile a:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<error>
    <code>AU_105</code>
    <description>LdSessionId non valida</description>
</error>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
error	Tag radice per la risposta ad una richiesta di logout			S
code	Indica il codice di errore			S
description	Riporta una breve descrizione dell'errore			S

7.4 Verifica dello stato della sessione

LegalDoc mette a disposizione anche un metodo per la verifica dello stato della sessione.

7.4.1 Input

Per richiamare questo servizio si deve utilizzare il metodo Http GET con content-type `application/x-www-form-urlencoded`

Inoltre bisogna aggiungere nell'header un campo con nome **ldSessionId** e il valore restituito dal servizio di login.

Figura 8: Esempio di utilizzo dei parametri per la verifica di una sessione di lavoro in un client per l'invocazione di servizi REST.

7.4.2 Output

La risposta in caso positivo è simile alla seguente:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<checkSessionResponse>
    <code>OK</code>
    <userId>pippo</userId>
</checkSessionResponse>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
`checkSessionResponse`	Tag radice per la risposta ad una richiesta di verifica della connessione			S
`code`	Codice di risposta. Se la richiesta ha avuto esito positivo indica il codice OK			S
`userId`	La user LegalDoc associata alla sessione			S

Status code **200 OK**

Time 36 ms

Headers
Date: Fri, 28 Sep 2012 13:54:53 GMT
Content-Encoding: gzip
X-Powered-By: AS Infocamere - newlegaldoc1
Transfer-Encoding: chunked
Connection: Keep-Alive
Server: WEB SERVER INFOCERT
Vary: Accept-Encoding,User-Agent
Content-Language: it
Content-Type: application/xml
Keep-Alive: timeout=15, max=100

Body **Raw response** **XML**

```
<?xml version="1.0" encoding="UTF-8"?>
▼<checkSessionResponse>
   <code>OK</code>
   <userId>userTest</userId>
 </checkSessionResponse>
```

Figura 9: Esempio di risposta con successo alla verifica di una sessione di lavoro.

Mentre in caso di risposta negativa avremo un tracciato simile a questo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<error>
    <code>LD_AU105</code>
    <description>LDSessionId non valida</description>
</error>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
`erro`	Tag radice per la risposta ad una richiesta di verifica della connessione			S
`code`	Indica il codice di errore			S
`description`	Riporta una breve descrizione dell'errore			S

Status **401 Unauthorized** Loading time: 287 ms

Request headers

Response headers

Raw | XML | **Response**

Copy to clipboard Save as file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
▼<error>
  <code>LD_AU105</code>
  <description>LDSessionId non valida</description>
 </error>
```

Figura 10: Esempio di risposta con fallimento alla verifica di una sessione di lavoro.

8. Funzione di conservazione

In questo capitolo vengono descritti i passi che un cliente deve fare per richiedere la conservazione di un documento in LegalDoc.

8.1 Prerequisiti, url e invocazione del servizio

Per poter utilizzare il servizio di conservazione è necessario autenticarsi al sistema, ed ottenere il **ldsessionid**, che rappresenta l'identificativo unico della sessione di lavoro.

Per richiamare questo servizio si deve utilizzare il metodo Http POST con content-type `multipart/form-data`

I parametri da assegnare ai 3 file di upload sono rispettivamente (case-sensitive):

- **PARAMFILE** per il file di parametri
- **INDEXFILE** per il file di indice
- **DATAFILE** per il file di dati

Inoltre bisogna aggiungere nell'header del pacchetto http un campo con nome **ldsessionid** e il valore restituito dal servizio di login.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/document/

Web service	Metodo HTTP	Descrizione
{bucket}/document	POST	Conserva un documento in LegalDoc

8.2 Input – output del processo di conservazione

8.2.1 Input

La richiesta di conservazione ha bisogno di passare informazioni sia tramite l'url, che tramite l'upload dei file. In particolare nell'url deve essere riportato il bucket, mentre nel pacchetto multipart da comporre vanno inseriti i tre file:

- **file_dei_parametri.xml**: *il file dei parametri di conservazione*, file in formato XML . Viene spiegato in dettaglio nel capitolo 9 Il file dei parametri di conservazione
- **file_di_indice_del_documento.xml**: *il file di indice*, file in formato XML. Viene spiegato in dettaglio nel capitolo 10 Il file di indice
- **file_di_dati.***: *il file di dati del documento*, file con uno dei formati tra quelli definiti per il cliente.

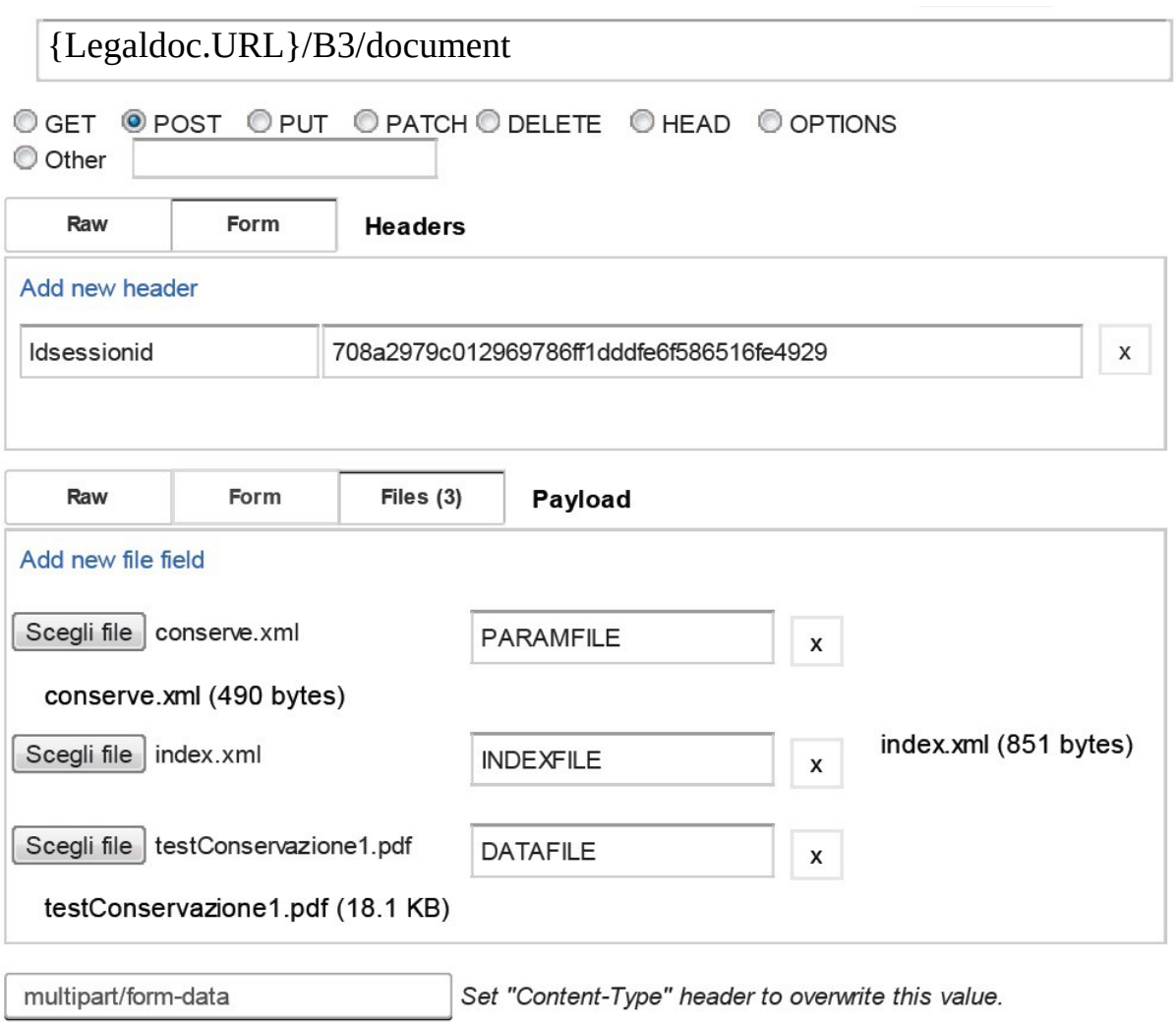

Figura 11: Esempio di utilizzo dei parametri per la conservazione di un documento in un client per l'invocazione di servizi REST.

8.2.2 Output

Alla richiesta di conservazione di un documento il sistema risponde con un codice http che indica il risultato dell'operazione: ad esempio a fronte di una conservazione conclusasi correttamente il sistema ritorna il codice HTTP 201 (created) ed il file IdC (Indice di Conservazione) firmato dal responsabile della conservazione e marcato temporalmente.

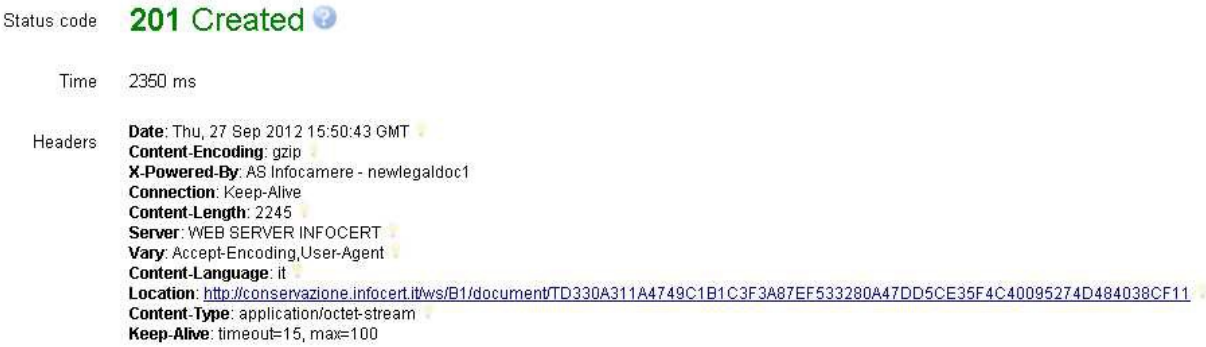

Figura 12: Esempio di risposta con successo alla conservazione di un documento.

Il file IdC attesta l'avvenuta conservazione del documento.

Se invece fossero stati caricati parametri errati, il sistema potrebbe rispondere con xml nel formato:

```
<error>
   <code>LD_XXNNN</code>
   <description>Breve descrizione del problema.</description>
</error>
```

Il tag `<code>` indica il codice associato all'errore, mentre il tag `<description>` restituisce una breve descrizione del problema.

Figura 13: Esempio di risposta con fallimento alla conservazione di un documento.

Per una completa descrizione di tutti i codici si rimanda a [4].

Riportiamo un esempio di IdC:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<IdC>
  <SelfDescription>
    <ID>TD330A311A4749C1B1C3F3A87EF533280A47DD5CE35F4C40095274D484038CF11</ID>
     <CreatingApplication>
       <Name>Legaldoc</Name>
       <Version>1.0</Version>
       <Producer>Infocert</Producer>
     </CreatingApplication>
  </SelfDescription>
  <VdC>
    <ID>TD330A311A4749C1B1C3F3A87EF533280A47DD5CE35F4C40095274D484038CF11</ID>
    <MoreInfo>
      <EmbeddMetadata>
        <additionalInfo key="token">
            TD330A311A4749C1B1C3F3A87EF533280A47DD5CE35F4C40095274D484038CF11
         </additionalInfo>
        <additionalInfo key="bucket">B1</additionalInfo>
```

```
        <additionalInfo key="policy">P1</additionalInfo>
        <additionalInfo key="operation">C</additionalInfo>
        <additionalInfo key="IDPdV">
            PDV1c85f515533a0da59d4ee162b2df6ca66408e488
        </additionalInfo>

      </EmbeddMetadata>
    </MoreInfo>
  </VdC>
  <FileGroup>
    <File>
          <ID>1</ID>
          <Hash>02de3f5576b2e9b3f1eb45cc7d629e782c1078aeb6db4b5b7f8924449405efc6</Hash>
          <MoreInfo>
                <EmbeddMetadata>conserve.xml</EmbeddMetadata>
          </MoreInfo>
    </File>
    <File>
          <ID>2</ID>
          <Hash>717dd3622f71f029e5c05b01a5ffc412fbb6ce646195b9b0ededb870c92c7dcd</Hash>
          <MoreInfo>
                <EmbeddMetadata>index.xml</EmbeddMetadata>
          </MoreInfo>
    </File>
    <File>
          <ID>3</ID>
          <Hash>ea3b1238a38f72b330aac53364bd0a0481946b93fc757dde7314ce3319f1840e</Hash>
          <MoreInfo>
                <EmbeddMetadata>testDataFile.pdf</EmbeddMetadata>
          </MoreInfo>
    </File>
  </FileGroup>
  <Process>
    <TimeReference>
      <TimeInfo>2013-05-15T16:09:37+0200</TimeInfo>
    </TimeReference>
  </Process>
</IdC>
```


In particolare, ecco il significato dei tag presenti:

TAG	*Significato*	*Attributo*	*Significato*
IdC	Radice del file xml		
SelfDescription	Informazioni relative all'indice di conservazione stesso		
ID	Identificativo univoco (token) assegnato al documento dal sistema di conservazione LegalDoc		
CreatingApplication	Informazioni sull'applicazione che ha generato l'IdC		
Name	Nome dell'applicazione che ha generato l'IdC		
Version	Versione dell'applicazione che ha generato l'IdC		
Producer	Nome del produttore dell'applicazione che ha generato l'IdC		
VdC	Informazioni relative al volume di conservazione		
ID	Identificativo univoco (token) assegnato al documento dal sistema di conservazione LegalDoc		
MoreInfo	Informazioni ulteriori relative al VdC		
EmbeddedMetadata	Le informazioni dell'elemento <MoreInfo> strutturate nel formato XML		
additionalInfo	Contenitore delle informazioni ulteriori	key="token"	Contiene il token del documento assegnatogli da LegalDoc
additionalInfo	Contenitore delle informazioni ulteriori	key="bucket"	ID del bucket a cui il documento appartiene così come indicato nell'url del servizio di conservazione

TAG	*Significato*	*Attributo*	*Significato*
`additionalInfo`	Contenitore delle informazioni ulteriori	`key="policy"`	ID della policy associata al documento così come indicata nel file dei parametri
`additionalInfo`	Contenitore delle informazioni ulteriori	`key="operation"`	Tipo di operazione richiesta al sistema di conservazione LegalDoc (conservazione, rettifica, cancellazione)
`additionalInfo`	Contenitore delle informazioni ulteriori	`key="IDPdV"`	ID del Pacchetto di Versamento
`FileGroup`	Elemento di aggregazione di più file oggetto di conservazione		
`File`	Informazioni relative al file oggetto di conservazione		
`ID`	Identificatore univoco del file descritto		
`Hash`	Informazioni sull'impronta del file cui l'elemento si riferisce		
`MoreInfo`	Informazioni ulteriori relative al file cui l'elemento si riferisce		
`EmbeddMetadata`	Nome del file cui l'elemento si riferisce		
`Process`	Informazioni relative alle modalità di svolgimento del processo di conservazione		
`TimeReference`	Informazioni relative alla data e ora di realizzazione dell'indice di conservazione		
`TimeInfo`	Generico riferimento temporale nel formato UNI ISO 8601:2010		

9. Il file dei parametri di conservazione

Il file dei parametri di conservazione è un file in formato XML simile al seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<parameters>
  <policy_id>P1</policy_id>
  <index_file>
    <index_name>index.xml</index_name>
    <index_hash>f2b24e7f9caa38329b954bd12ed924289620ddbaa2473464653670223767 3f8b</index_hash>
    <index_mimetype>text/xml;1.0</index_mimetype>
  </index_file>
  <data_file>
    <data_name>testDataFile.pdf</data_name>
    <data_hash>37fac9829dbd79a7fa8d792ceceda509649807238839c0bbd6184227cc2ba145</data_hash>
    <data_mimetype>application/pdf;1.4</data_mimetype>
  </data_file>
  <path>/fatture/2012</path>
  <encrypted_by_owner>S</encrypted_by_owner>
</parameters>
```

Legaldoc attribuisce di default a questo file il nome conserve.xml.

Di seguito sono riportati i tag necessari ed il loro significato:

TAG	*Significato*	*Attributo*	*Significato*	*Obbligatorietà*
`parameters`	Tag radice per la richiesta di conservazione			S
`policy_id`	Il codice della Policy con cui si vuole conservare il documento			S
`index_file`	Tag contenente le informazioni sul file di indici che si vuole conservare			S
`index_name`	Il nome del file di indice che si vuole conservare. E' il nome con cui verrà conservato il file in LegaldDoc			S
`index_hash`	E' l'hash calcolato con metodo SHA-256 del file di indice			S
`index_mimetype`	Indica il mimetype del file di indice che si vuole conservare			S
`data_file`	Tag contenente le informazioni sul file di dati che si vuole conservare			S
`data_name`	Il nome del file di dati che si vuole conservare. E' il nome con cui verrà conservato il file in LegaldDoc			S
`data_hash`	E' l'hash calcolato con metodo SHA-256 del file di dati			S
`data_mimetype`	Indica il mimetype del file di dati che si vuole conservare			S
`path`	E' il percorso logico dove si vuole salvare il proprio file di dati			S

TAG	*Significato*	*Attributo*	*Significato*	*Obbligatorietà*
`encrypted_by_owner`	Indica se il documento è stato cifrato dal Cliente prima dell'invio in conservazione. Valori ammessi "S" (è cifrato) o "N". Se valorizzato a "S" si solleva InfoCert da qualsiasi controllo sulla visualizzazione [2]			N (se non indicato, il valore di default è "N")

Controlli sul file dei parametri di conservazione

LegalDoc esegue una serie di controlli sul contenuto del file xml dei parametri di conservazione. Di seguito si riporta l'elenco dei controlli effettuati.

1. **Tracciato del file dei parametri**: il file di indice viene validato con l'xsd riportato in appendice A
2. **Policy**: si verifica l'esistenza della policy dichiarata ed il suo stato di attivazione
3. **MimeType del file degli indici**: si verifica la congruenza del mimetype indicato per il file di indici rispetto alla policy dichiarata
4. **MimeType del file di dati**: si verifica la congruenza del mimetype indicato per il file di dati rispetto alla policy dichiarata
5. **Univocità del nome del file di dati all'interno del path**: viene eseguito il controllo che non esista un documento con lo stesso nome all'interno del path indicato

[2] Se impostato a S, il Cliente si impegna a mantenere gli strumenti necessari per la visualizzazione del documento

10. Il file di indice

In LegalDoc è disponibile una funzionalità di indicizzazione del parco documentale conservato che facilita l'utente nella ricerca dei propri documenti. E' consentito definire quali sono i criteri di ricerca, per una più puntuale estrazione dei risultati.

10.1 Conservazione del file di indici per la ricerca

Gli indici devono essere memorizzati contestualmente alla conservazione del documento. Non è possibile inviare gli indici dopo che il documento è stato accettato dal sistema di conservazione, né è possibile modificarli a meno di non effettuare una operazione di rettifica del documento.

Un esempio di file di indice è il seguente:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<legaldocIndex documentClass="ae_fata" label="Fatture emesse">
  <field name="denominazione_s">Denominazione Azienda</field>
  <field name="codice_fiscale_s" label="Codice fiscale">SNITMS82T27G224S</field>
  <field name="partita_iva_s">01032450072</field>
  <field name="via_s">Corso Stati Uniti</field>
  <field name="localita_s">Padova</field>
  <field name="provincia_s">PD</field>
  <field name="email_destinatario_em">assistenza.legaldoc@legalmail.it</field>
  <field name="totale_importo_d">550</field>
  <field name="numero_registrazione_i">1</field>
  <field name="__numero_documento_l">2</field>
  <field name="__serie_s">S1</field>
  <field name="__data_inizio_numerazione_dt">01-01-2012</field>
  <field name="__data_documento_dt">15-05-2012</field>
  <field name="data_di_presa_in_carico_dt">09-05-2012 23:39:00</field>
  <field name="__anno_fiscale_i">2012</field>
  <field name="note_s">note</field>
  <field name="boolean_index_b">true</field>
  <field name="double_index_d">.254</field>
  <field name="float_index_f">2.34</field>
  <field name="location_index_p">-12.524, 35.245</field>
  <field name="__indice_fascicolo_s">pratica-01</field>
</legaldocIndex>
```

TAG	*Significato*	*Attributo*	*Significato*	*Obbligatorietà*
`legaldocIndex`	Tag radice del file di indice	`documentClass, label`	`documentClass:` (attributo obbligatorio) identifica la classe documentale descritta dai campi di ricerca. Tale valore viene fornito al Cliente in fase di contrattuale. `label:` (attributo facoltativo) nome logico della classe documentale	S
`field`	Descrive un campo di ricerca	`name, label`	`name:` (attributo obbligatorio) nome del campo di ricerca, Obbligatorio. `label:` (attributo facoltativo) nome logico del campo di ricerca	N

I campi che necessitano del controllo di obbligatorietà devono essere concordati in fase contrattuale. Per questi, deve essere inserito nell'attributo `name` il nome fornito da InfoCert.

Il nome degli altri indici può essere liberamente inserito dal Cliente. Sarà poi onere del Cliente utilizzare in fase di ricerca lo stesso nome inserito in fase di conservazione. La struttura del nome ha delle limitazioni espresse nel paragrafo relativo ai controlli sul file di indice. **Prestare massima cura nella scelta di questo campo in quanto non è modificabile una volta accettato dal sistema.**

LegalDoc permette di memorizzare un nome logico per la classe documentale e per i campi di ricerca allo scopo di poterlo utilizzare all'interno di interfacce utente. Il sistema memorizza la label di un campo alla prima operazione di conservazione ed è quindi possibile ometterla nei successivi invii. Se non viene mai specificata, la label viene valorizzata con il nome espresso dal campo `name`.

10.2 Regole per la conservazione del file di indici per la ricerca

Sono in vigore le seguenti regole sul formato degli indici:

- I nomi dei campi sono case-sentive se non diversamente specificato.
- I valori degli campi non sono case-sentive se non diversamente specificato.
- L'attributo `name` del TAG `field` può contenere lettere, numeri e i caratteri _ e -. Altri caratteri non sono ammessi. In particolare non sono ammessi spazi.
- Gli attributi `label` del TAG `legaldocIndex` e del TAG `field` possono contenere lettere, numeri, spazi, punti, apostrofo e i caratteri _ e -. Altri caratteri non sono ammessi.

- Il Cliente non può definire nuovi campi di ricerca il cui `name` inizi con il prefisso __ (doppio underscore) in quanto questa nomenclatura è riservata ai seguenti campi di ricerca, che comportano da parte del software particolari operazioni:
    - `__anno_fiscale_i`
    - `__data_documento_dt`
    - `__data_inizio_numerazione_dt`
    - `__serie_s`
    - `__numero_documento_l`
    - `__progr_inizio_l`
    - `__progr_fine_l`
    - `__indice_fascicolo_s`

      su questo indice Legaldoc non effettua alcun controllo. E' l'indice suggerito per avere un legame logico tra più documenti, ad esempio tra i documenti appartenenti ad una stessa pratica

- Tutti i campi di ricerca, tanto quelli richiesti da InfoCert quanto quelli definiti dal Cliente, dovranno necessariamente avere un suffisso tra i seguenti per indicare il tipo di dato contenuto:

Suffisso	***Tipo di campo***
_s	stringa
_i	intero
_l	long
_f	float
_d	double
_b	boolean
_dt	data
_em	e-mail
_p	location

- Ad eccezione del suffisso `_s` che indica un generico campo stringa, tutti gli altri suffissi attivano automaticamente dei controlli sul formato del valore del campo di ricerca. Segue l'elenco dettagliato dei controlli sul formato applicati ai vari campi in base all'attributo `name` del TAG `field` e il corrispondente formato da rispettare per la loro valorizzazione:

Tipo di campo	*Nome del campo*	*Valorizzazione del campo*

Data	Campo con suffisso `_dt`	I formati ammessi sono due:
		`dd-MM-yyyy`
		`dd-MM-yyyy HH:mm:ss`
		Esempio:
		`<field name="data_di_presa_in_carico_dt">09-05-2012 23:39:00</field>`
		`<field name="__data_documento_dt" label="data documento">01-01-2012</field>`
		`<field name="__data_inizio_numerazione_dt" label="data inizio numerazione">01-01-2012</field>`

Anno fiscale	Campo `__anno_fiscale_i`	Il formato ammesso è `yyyy` Esempio: `<field name="__anno_fiscale_i">2012</field>`
Serie di numerazione	Campo `__serie_s`	La lunghezza massima consentita per il campo `__serie_s` è di 256 caratteri. Il valore di tale tag NON può essere la stringa riservata ad Infocert "`__ldoc_default_sequence_name`". Inoltre, sono ammessi solo lettere, numeri e i caratteri / \ ' & : + ( ) @ Esempio: `<field name="__serie_s">S/1</field>`
E-mail	Campo con il suffisso `_em`	Il valore del campo deve essere un indirizzo e-mail valido. Esempio: `<field name="email_destinatario_em">assistenza.legaldoc@legalmail.it</field>`
Intero	Campo con il suffisso `_i`	Il valore del campo deve essere un numero intero compreso tra `-2147483648` e `2147483647` Esempio: `<field name="numero_registrazione_i">1</field>`

Long	Campo con il suffisso `_l`	Il valore del campo deve essere un numero intero compreso tra `-9223372036854775808` e `9223372036854775807`
		Esempio:
		`<field name="totale_importo_d">550</field>`
		`<field name="__numero_documento_l">1425</field>`
		`<field name="__progr_inizio_l">325</field>`
		`<field name="__progr_fine_l">330</field>`

Boolean	Campo con il suffisso `_b`	Il valore del campo deve essere di tipo boolean ed i valori consentiti sono '`true`' o '`false`' Esempio: `<field name="boolean_index_b">true</field>`

Tipo di campo	*Nome del campo*	*Valorizzazione del campo*
Float	Campo con il suffisso `_f`	Il valore del campo deve essere di tipo float (un numero decimale, 32 bit IEEE Floating Point) Esempio: `<field name="float_index_f">2.34</field>`
Double	Campo con il suffisso `_d`	Il valore del campo deve essere di tipo double (un numero decimale, 64 bit IEEE Floating Point) Esempio: `<field name="double_index_d">.254</field>`
Location	Campo con il suffisso `_p`	Il valore del campo deve rispettare il formato `double,double` Esempio: `<field name="location_index_p">-12.524,35.245</field>`

Il mancato rispetto di tali regole pregiudica il buon esito del processo di conservazione.

10.3 Controlli sul file di indice

LegalDoc esegue una serie di controlli sul contenuto del file xml di indice. Di seguito si riporta l'elenco dei controlli effettuati.

1. **Tracciato del file indice**: il file di indice viene validato con l'xsd riportato in Appendice A: XSD file utilizzati in LegalDoc
2. **Classe documentale e policy**: si verifica la correttezza della classe documentale designata e l'autorizzazione all'invio di documenti di tale classe per il bucket e la policy forniti
3. **Campi di ricerca obbligatori**: si verifica l'esistenza, la valorizzazione e la non duplicazione dei campi obbligatori. Gli indici di ricerca obbligatori sono configurabili per ciascuna classe documentale in fase di attivazione. Esiste un insieme minimo di campi obbligatori comuni a ciascuna classe documentale:

Nome campo	*Classi documentali fiscali*	*Classi documentali non fiscali*
`__data_documento_dt`	obbligatorio	obbligatorio
`__anno_fiscale_i`	obbligatorio	facoltativo

4. **Correttezza del tipo di dato**: i campi con controllo sul tipo di dato devono essere valorizzati. In caso il campo non sia obbligatorio e non si intende valorizzarlo, si deve omettere il tag. Non sono ammessi campi obbligatori o campi con controllo sul formato con tag vuoto. I controlli verificano che il contenuto di un certo campo individuato in base al nome rispetti le regole indicate nel paragrafo 10.2 Regole per la conservazione del file di indici per la ricerca

5. **Controllo di numerazione nella richiesta di conservazione**: per alcune classi documentali come i documenti fiscali (fatture, libri, registri...) è previsto un controllo di sequenza che si attiva se vengono utilizzati i nomi indici qui indicati.

   Sono previste due modalità di controllo: semplice e per range:
   - controllo di numerazione semplice, richiesto ad esempio nel caso delle fatture, gestito tramite il campo

   `__numero_documento_l`

   Il controllo verifica che il valore di questo indice sia in sequenza rispetto a quello dell'ultimo documento caricato

   - controllo di numerazione per range, utilizzato ad esempio per libri e registri, gestito tramite i campi

   `__progr_inizio_l`: progressivo iniziale

   `__progr_fine_l`: progressivo finale
   Il controllo verifica che:
   - il valore di __progr_inizio_l sia minore o uguale al valore di __progr_fine_l
   - il valore di __progr_inizio_l sia in sequenza rispetto al valore di __progr_fine_l dell'ultimo documento caricato

   Per entrambe le modalità di controllo è possibile gestire numerazioni parallele differenziate per uno o per entrambi i seguenti indici:
   - `__data_inizio_numerazione_dt`: data di inizio dell'esercizio fiscale di competenza del documento

   - `__serie_s`: serie di numerazione.

   È quindi, ad esempio, accettata la conservazione contemporanea dei seguenti documenti:

`__numero_documento_l`	__data_inizio_numerazione_dt	`__serie_s`
1	01/01/15	
1	01/01/15	S1
1	01/01/15	S2
1	01/03/15	

6. **Controllo di numerazione nella richiesta di rettifica**: diversamente dal controllo eseguito in fase di conservazione, in una richiesta di rettifica non vengono effettuati i controlli di numerazione. Viene solo controllata l'appartenenza alla stessa classe documentale del documento da rettificare.

11. Richiesta di esibizione

L'esibizione è il servizio LegalDoc che permette di estrarre dal sistema un documento per cui sia stata completata correttamente la procedura di conservazione, di rettifica o di cancellazione. Un cliente deve eseguire tale operazione quando vuole rendere disponibile un documento archiviato a norma e le informazioni che qualificano il processo di conservazione o rettifica.

L'esibizione può restituire il documento in due modalità differenti: in un unico pacchetto zip, oppure un file alla volta. Questa ultima modalità deve essere compatibile con il client di esibizione del cliente.

11.1 Prerequisiti, url e invocazione del servizio

Per richiamare questo servizio si deve utilizzare il metodo Http GET ed indicare la modalità con la quale si vuole avere il documento in risposta scegliendo tra:

```
application/x-zip-compressed (si otterrà uno zip contenente tutti i file del documento)
multipart/mixed (si otterranno i file separatamente in un unico "pacchetto" multipart)
```

Inoltre bisogna aggiungere nell'header un campo con nome **ldsessionid** e il valore restituito dal servizio di login.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/document/**{token}**

dove token è l'identificativo univoco del documento da esibire restituito da LegalDoc a fronte di una chiamata di conservazione o rettifica (vedi 8.2.2 per la descrizione dell'output del servizio di conservazione).

Web service	Metodo HTTP	Descrizione
{bucket}/document/{token}	GET	Esibisce un documento conservato o rettificato in LegalDoc

11.2 Input – output del processo di esibizione

11.2.1 Input

La richiesta di esibizione avviene tramite chiamata in GET all'url indicato. Di seguito un esempio di invocazione per l'esibizione di un file zip:

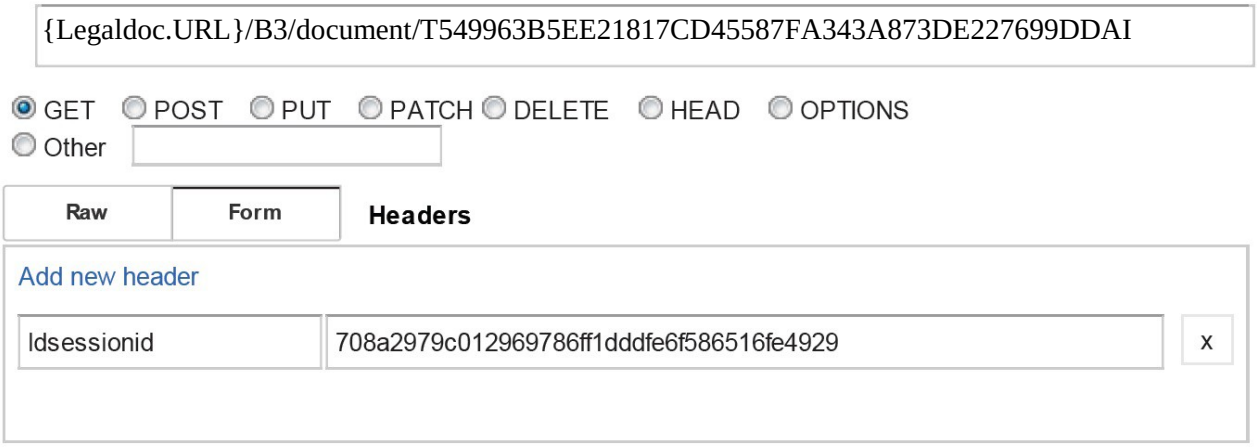

Figura 14: Esempio di utilizzo dei parametri per l'esibizione di un documento in formato zip in un client per l'invocazione di servizi REST.

Come si evince dalla figura, in questo caso le informazioni necessarie sono passate attraverso l'url di invocazione.

Per la richiesta in multipart bisogna aggiungere l'header `Accept: multipart/mixed`

Figura 15: Esempio di utilizzo dei parametri per l'esibizione di un documento in formato multipart in un client per l'invocazione di servizi REST.

11.2.2 Output

Alla richiesta di esibizione di un documento il sistema in base alla tipologia di risorse attese risponde:

- o con uno zip del file
- o con un pacchetto http multipart, formato da tutti i file del documento

In entrambi i casi, i file restituiti sono, nell'ordine,

- file IdC

- file di parametri
- file di indici
- file di dati

In caso di risposta positiva, il codice HTTP in entrambi i casi è 200 OK e, nel caso di richiesta di risposta in formato zip avrà il `content-type = application/zip` come mostrato nell'esempio seguente:

Status code **200 OK**

Time 436 ms

Headers
Date: Thu, 27 Sep 2012 16:05:27 GMT
Content-Encoding: gzip
X-Powered-By: AS Infocamere - newlegaldoc1
Transfer-Encoding: chunked
Content-Disposition: inline; filename=TD330A311A4749C1B1C3F3A87EF533280A47DD5CE35F4C40095274D484038CF11.zip
Connection: Keep-Alive
Server: WEB SERVER INFOCERT
Vary: Accept-Encoding,User-Agent
Content-Language: it
Content-Type: application/zip
Keep-Alive: timeout=15, max=100

Figura 16: Esempio di risposta con successo all'esibizione di un documento in formato zip.

mentre nel caso di richiesta di formato multipart, la risposta avrà il `content-type = multipart/mixed` come si può vedere dal tracciato della chiamata di esempio seguente:

Status code **200 OK**

Time 346 ms

Headers
Date: Fri, 28 Sep 2012 14:05:58 GMT
Content-Encoding: gzip
X-Powered-By: AS Infocamere - newlegaldoc1
Transfer-Encoding: chunked
Connection: Keep-Alive
Server: WEB SERVER INFOCERT
Vary: Accept-Encoding,User-Agent
Content-Language: it
Content-Type: multipart/mixed; boundary=Boundary_1_1394791734_1348841158718
Keep-Alive: timeout=15, max=100
MIME-Version: 1.0

Figura 17: Esempio di risposta con successo all'esibizione di un documento in formato multipart.

12. Funzione di Rettifica

12.1 Che cosa si intende per Rettifica di un documento

Il servizio di rettifica di un documento permette agli utenti LegalDoc di "rettificare" un documento precedentemente conservato, ovvero di conservare un documento che sostituisce logicamente il documento originale.

Il processo può essere schematizzato come segue:

- un GD manda in conservazione un documento (che in questo paragrafo chiameremo "documento originale")
- il documento originale, una volta terminata la sua elaborazione da parte del servizio di conservazione di LegalDoc, risulta conservato
- il GD decide che devono essere modificate alcune informazioni presenti nel documento originale o addirittura il documento stesso
- il GD utilizza quindi il servizio di rettifica per espletare la funzione

Il processo di rettifica può essere schematizzato come segue:

- il GD prepara il documento che deve sostituire il documento originale (che in questo paragrafo chiameremo "documento rettificante")
- il GD invoca il servizio di rettifica fornendo il documento rettificante (formato sempre da file di indici e file di dati) e il Token LegalDoc del documento originale direttamente nell'url che richiamerà
- LegalDoc conserva il documento rettificante e marca nei propri archivi il documento originale come "Rettificato"

Si vuole evidenziare che il documento originale rettificato non viene mai eliminato dagli archivi LegalDoc, ma gli viene semplicemente attribuito lo stato logico di rettificato.

Non è possibile rettificare un documento già rettificato (è possibile eventualmente rettificare il relativo documento rettificante) o già cancellato.

12.2 Esibizione di un documento rettificato e di un documento rettificante

Nel caso in cui si richieda l'esibizione del docume nto originale rettificato, il documento viene esibito normalmente e nella risposta viene indicato il suo stato di rettificato. Inoltre, nella risposta che si ottiene alla richiesta di esibizione del documento rettificato, si ha anche a disposizione, grazie all'header `X-Document-Changed` presente nella risposta, il token identificativo del documento che ha rettificato il documento esibito.

Nel caso si richieda l'esibizione del documento rettificante, nel file IdC di risposta è indicato il suo stato di rettificante e gli estremi del documento originale che esso rettifica.

12.3 Input – output del processo di rettifica

La modalità di fruizione del servizio di rettifica ricalca quello di conservazione (vedi 8 Funzione di conservazione) a meno di alcuni parametri che vengono di seguito elencati.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/document/**{token}**

dove token è l'identificativo univoco del documento da rettificare, restituito da LegalDoc a fronte di una chiamata di conservazione.

Web service	Metodo HTTP	Descrizione
{bucket}/document/{token}	PUT	Rettifica un documento conservato in LegalDoc

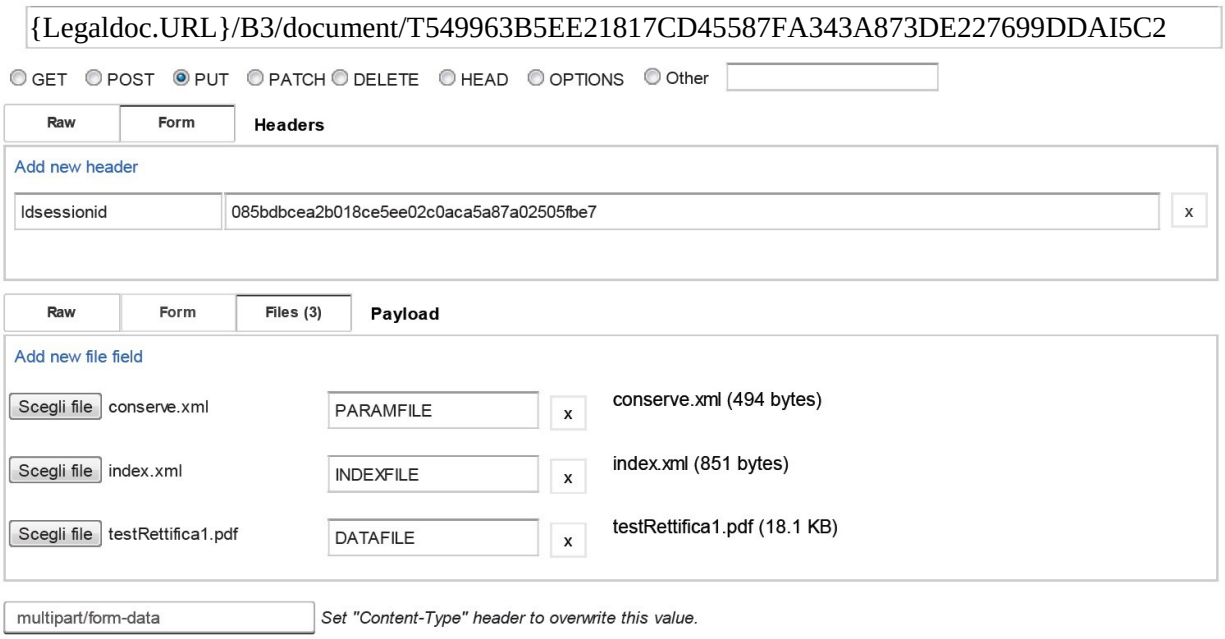

Figura 18: Esempio di utilizzo dei parametri per la rettifica di un documento in un client per l'invocazione di servizi REST.

13. Funzione di Cancellazione

13.1 Che cosa si intende per cancellazione di un documento

Il servizio di cancellazione di un documento permette agli utenti LegalDoc di "cancellare" un documento precedentemente conservato. La cancellazione è logica, dagli archivi LegalDoc non viene mai cancellato nulla fisicamente.

Inoltre non è possibile cancellare un documento rettificato (in questo caso è possibile cancellare il relativo documento rettificante) o già cancellato.

13.2 Esibizione di un documento cancellato e di un documento cancellante

Per l'esibizione di un documento cancellato e di un documento cancellante valgono le medesime considerazioni fatte per i documenti rettificato e rettificante al paragrafo 12.2 cui si rimanda.

13.3 Input – output del processo di cancellazione

La modalità di fruizione del servizio di cancellazione ricalca quello di conservazione (vedi 8 Funzione di conservazione) a meno di alcuni parametri che vengono di seguito elencati.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/document/**{token}**

dove token è l'identificativo univoco del documento da rettificare, restituito da LegalDoc a fronte di una chiamata di conservazione.

La richiesta di cancellazione, diversamente da quelle di conservazione e rettifica, inoltre, NON ha i file di indici e di dati ma soltanto il file dei parametri come quello di esempio presentato di seguito:

```
<?xml version="1.0" encoding="utf-8"?>
<parameters>
    <policy_id>P1</policy_id>
    <path>/test</path>
</parameters>
```

con i tag che hanno il medesimo significato già descritto in 9 Il file dei parametri di conservazione.

In risposta all'invocazione del servizio di cancellazione si otterrà il file IdC che attesta il momento dell'avvenuta cancellazione del documento indicato, attravero il token, nell'URL del servizio di cancellazione stesso.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/document/**{token}**

dove token è l'identificativo univoco del documento da cancellare, restituito da LegalDoc a fronte di una chiamata di conservazione.

Web service	Metodo HTTP	Descrizione
{bucket}/document/{token}	DELETE	Cancella un documento conservato in LegalDoc

Figura 19: Esempio di utilizzo dei parametri per la cancellazione di un documento in un client per l'invocazione di servizi REST.

14. Funzioni di ricerca

LegalDoc offre ai propri utilizzatori funzionalità avanzate di ricerca sul parco documentale conservato.

Ai servizi di conservazione e rettifica il cliente fornisce un file di indici con cui il sistema conserva i documenti. Grazie a tali indici è possibile operare ricerche specializzate.

La ricerca avviene fornendo in input una richiesta strutturata (query) ed ottenendo come risultato, ad ogni esito positivo, i documenti che corrispondono ai criteri di selezione specificati.

La fruizione di questo servizio avviene sempre tramite invocazione di web services REST in maniera del tutto analoga agli altri servizi LegalDoc.

Il flusso è il seguente:

1. all'atto della conservazione/rettifica di un documento deve essere allegato un file XML contenente gli indici di ricerca (nelle modalità espresse nel capitolo 10 Il file di indice)
2. il sistema LegalDoc conserva il documento indicizzandolo secondo quanto espresso in 1
3. il cliente utilizza il web service di ricerca per cercare i documenti che soddisfano determinati criteri di ricerca (che devono essere un sottoinsieme anche completo degli indici inviati all'atto della conservazione)
4. il web service invocato al punto precedente restituisce una lista di token di documenti in LegalDoc che soddisfano il criterio di ricerca fornito

14.1 Prerequisiti, url e invocazione del servizio

Per poter effettuare una ricerca, è necessario essere in possesso di una **ldsessionid** valida che può essere ottenuta effettuando la **login** al sistema.

L'url da invocare è il seguente:

{LegalDoc.URL}/{bucket}**/search/standard**

La chiamata deve utilizzare il protocollo **HTTPS**, il metodo HTTP POST ed il pacchetto di invio deve contenere:

- Header:

  ldsessionid valorizzato con l'id di sessione ottenuto in fase di login

  content-type valorizzato come "**application/xml**"

- Body:

  l'**xml della richiesta**

14.1.1 Input

Il file di ricerca da associare ad ogni chiamata consente di specificare:

- i campi (field) desiderati
- un filtro con i valori ammessi per determinati campi

- il numero massimo di documenti restituiti

Un file xml di ricerca è analogo al seguente:

```
<search>
    <request>
            <select>
                    <field name="cognome_s" />
                    [..]
            </select>
            <filter documentClass="FATTATT025">
                    <field name="cognome_s">Rossi</field>
                    <field name="anno_nascita_i">[1970 TO 1990]</field>
                    <field name="nome_s">Paol*</field>
                    [..]
            </filter>
            <limit documents="20" />
    </request>
</search>
```

TAG	*Significato*	*Attributo*	*Significato*	*Obbligatorietà*
`search`	Tag radice per la ricerca standard			S
`request`	Identifica il corpo della richiesta			S
`select`	Contiene la lista dei campi visualizzati in output per ogni documento			N (se non presente restituisce tutti gli indici)
`select → field`	Tag rappresentante l'indice di ricerca	`name`	Nome dell'indice, *Obbligatorio*	N
`filter`	E' il tag che indica la presenza di un filtro	`documentClass`	Classe documentale dei documenti da ricercare, *Obbligatorio*	S
`filter → field`	Tag rappresentante l'indice di ricerca	`name`	Nome dell'indice, *Obbligatorio*	N
limit	Consente di specificare il numero massimo di risultati	`documents`	Numero massimo di risultati, *Obbligatorio*	N (se non indicato il valore di default è 10, se indicato dev'essere minore di 200; se è maggiore di 200 verranno restituiti 200 risultati come indicato in 5.3.3)

Nella ricerca standard le condizioni specificate come campi `<field>` all'interno del tag `<filter>` sono implicitamente valutate in AND, ovvero necessitano che tutte siano soddisfatte contemporaneamente.

Ogni condizione supporta:

- match *esatto*: condizione soddisfatta per valore identico al criterio inserito

  Esempio: `<field name="cognome_s">Rossi</field>`

- match *per intervallo*: permette la ricerca di tutti i documenti che hanno indice compreso fra un valore minimo ed uno massimo

  Esempi e sintassi:

  - `<field name="anno_nascita_i">`[start TO end]`</field>` include anche gli estremi start ed end
  - `<field name="anno_nascita_i">`(start TO end)`</field>` esclude gli estremi dall'intervallo

- match *per pattern*: permette la ricerca di tutti i documenti che hanno l'indice che inizia con un pattern (stringa alfanumerica) predefinito

  Esempio: `<field name="nome_s">Paol*</field>`

14.1.2 Output

Alla richiesta di ricerca il sistema risponde con un file simile:

```
<?xml version="1.0" encoding="UTF-8"?>
<response start="0" totalDocuments="9777">
    <documents bucket="B1">
            <document          insertDate="2013-05-07T20:30:29Z"          fileName="file_6.ext"
path="/data/folder1/subfolder1/" pdv="dummy_pdv" documentClass="ae_gen" token="T5">
                    <field name="negative_l">-2</field>
                    <field name="text_s">mutuo</field>
                    <field name="bool_b">false</field>
            </document>
            <document          insertDate="2013-05-07T20:30:29Z"          fileName="file_12.ext"
path="/data/folder1/subfolder2/" pdv="dummy_pdv" documentClass="ae_fatp" token="T11">
                    <field name="negative_l">-100</field>
                    <field name="text_s">affitto</field>
            </document>
    </documents>
</response>
```

TAG	*Significato*	*Attributo*	*Significato*	*Presenza*
`response`	Tag radice per la risposta ad una ricerca full	`totalDocuments`, `start`	`totalDocuments` indica il numero totale di documenti che soddisfano le condizioni indicate nella richiesta. `start` è l'indice del primo documento, necessario alla paginazione	S
`documents`	Tag raccoglitore dei documenti per un determinato bucket	`bucket`	Indica il bucket su cui si è eseguita la ricerca	S
`documents → document`	E' il singolo documento trovato dalla ricerca	`fileName`, `path`, `pdv`, `documentClass`, `token`	`fileName` è il nome del file associato al documento. `path` è la cartella logica dove risiede il documento. `pdv` è il pacchetto di versamento. `documentClass` è la classe documentale. `token` è l'identificativo univoco del documento.	N
`documents → document → field`	Tag che rappresenta un campo del documento. E' tra quelli richiesti nella `select → field`	`name`	Nome dell'indice	N

15. Gestione dei visualizzatori

I documenti informatici sono memorizzati sotto forma di evidenza informatica su adeguati supporti. Le evidenze informatiche dei documenti possono essere strutturate in diversi modi, detti formati informatici (o mime/type). Affinché sia possibile utilizzare queste informazioni sono necessari degli appositi programmi che permettano l'interpretazione di una evidenza informatica di un documento, visualizzandolo a terminale e/o stampandolo su supporto cartaceo. Tali programmi, detti visualizzatori (o viewer), devono essere mantenuti all'interno del sistema di conservazione.

Nel servizio LegalDoc i formati informatici dei documenti sono divisi in due categorie:

- formati predefiniti da InfoCert in [1] . Per ognuno di questi InfoCert mette a disposizione un visualizzatore in grado di interpretare il relativo formato
- altri formati; per ognuno di questi il visualizzatore deve essere fornito dal Cliente e caricato a sistema nelle modalità descritte in questo capitolo

15.1 Regole relative al caricamento dei visualizzatori per i formati predefiniti

I visualizzatori dei formati predefiniti da InfoCert richiesti dal Cliente sono automaticamente assegnati all'atto della attivazione della propria area di conservazione, e verranno forniti da InfoCert al Cliente all'atto di attivazione del prodotto.. Tutti i documenti inviati in conservazione saranno associati al visualizzatore configurato per il particolare formato.

Si ricorda che, come espresso in [2], è responsabilità del Cliente controllare che i visualizzatori proposti da InfoCert siano effettivamente adeguati ai documenti del Cliente. In caso questo non sia, il Cliente può provvedere alla loro sostituzione **inviando i propri nelle modalità descritte nel paragrafo 15.4.**. L'utilizzo di un visualizzatore da parte del sistema avviene a partire dalla data di attivazione indicata al momento del caricamento del viewer e può essere retroattiva.

Ogni visualizzatore proposto da InfoCert è corredato di una scheda tecnica che definisce l'ambiente operativo nel quale il visualizzatore è utilizzabile.

15.2 Regole relative al caricamenti dei visualizzatori per formati aggiuntivi

I visualizzatori di formati aggiuntivi ai predefiniti devono essere inviati dal Cliente **prima** di iniziare la conservazione dei documenti. Tuttavia, il sistema accetta documenti in conservazione anche se il relativo visualizzatore non è caricato.

Il caricamento di un visualizzatore per un particolare mime/type va effettuato una sola volta. Ulteriori caricamenti per lo stesso mime/type verranno identificati come aggiornamenti di versione del visualizzatore nelle modalità descritte nel paragrafo successivo.

I formati aggiuntivi devono essere concordati tra il Cliente e InfoCert in fase contrattuale in [1]. Non è possibile caricare visualizzatori per formati non preventivamente concordati e configurati nel sistema.

15.3 Aggiornamento dei visualizzatori a nuove versioni

I visualizzatori dei formati predefiniti verranno aggiornati a successive versioni da InfoCert ove questo si renda necessario per adeguamenti tecnologici della piattaforma. L'aggiornamento avviene in maniera trasparente al Cliente previa comunicazione. I documenti inviati dal Cliente dopo l'avvenuto aggiornamento saranno associati alla nuova versione del visualizzatore. I documenti inviati dal Cliente prima dell'aggiornamento rimangono associati alla versione del visualizzatore operativa al momento del caricamento.

I visualizzatori forniti dal Cliente verranno aggiornati dal Cliente secondo le sue necessità utilizzando lo stesso meccanismo di caricamento. Il sistema LegalDoc identifica se per un particolare mime/type si sta procedendo al primo caricamento o ad un successivo upgrade del visualizzatore.

15.4 Caricamento di visualizzatori mediante servizio REST

Il caricamento di un visualizzatore mediante servizio REST, dal punto di vista operativo, è analogo al servizio di conservazione e viene qui di seguito descritto.

15.4.1 Prerequisiti, url e invocazione del servizio

Per poter utilizzare il servizio di caricamento di un visualizzatore è necessario autenticarsi al sistema, ed ottenere il **ldsessionid**, che rappresenta l'identificativo unico della sessione di lavoro.

Per richiamare questo servizio si deve utilizzare il metodo Http POST con content-type `multipart/form-data`

I parametri da assegnare ai 3 file di upload sono rispettivamente (case-sensitive):

- **PARAMFILE** per il file di parametri
- **VIEWERFILE** per il file del viewer
- **TSFILE** per il file della scheda tecnica

Inoltre bisogna aggiungere nell'header del pacchetto http un campo con nome **ldsessionid** e il valore restituito dal servizio di login.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/viewer/

15.4.2 Input

La richiesta di upload viewer ha bisogno di passare informazioni sia tramite l'url, che tramite l'upload dei file. In particolare nell'url deve essere riportato il bucket, mentre nel pacchetto multipart da comporre vanno inseriti i tre file:

- **file_dei_parametri.xml**: *il file dei parametri di upload viewer*, file in formato XML . Viene spiegato in dettaglio nel paragrafo 15.4.4

- **file_di_dati.***: *il file del viewer*,
- **file_della_scheda_tecnica.***: *il file della Scheda Tecnica*, . Descrive in breve le caratteristiche tecniche del viewer. Il formato di questo file non viene dichiarato. Per i Viewer messi a disposione da InfoCert è un file txt.

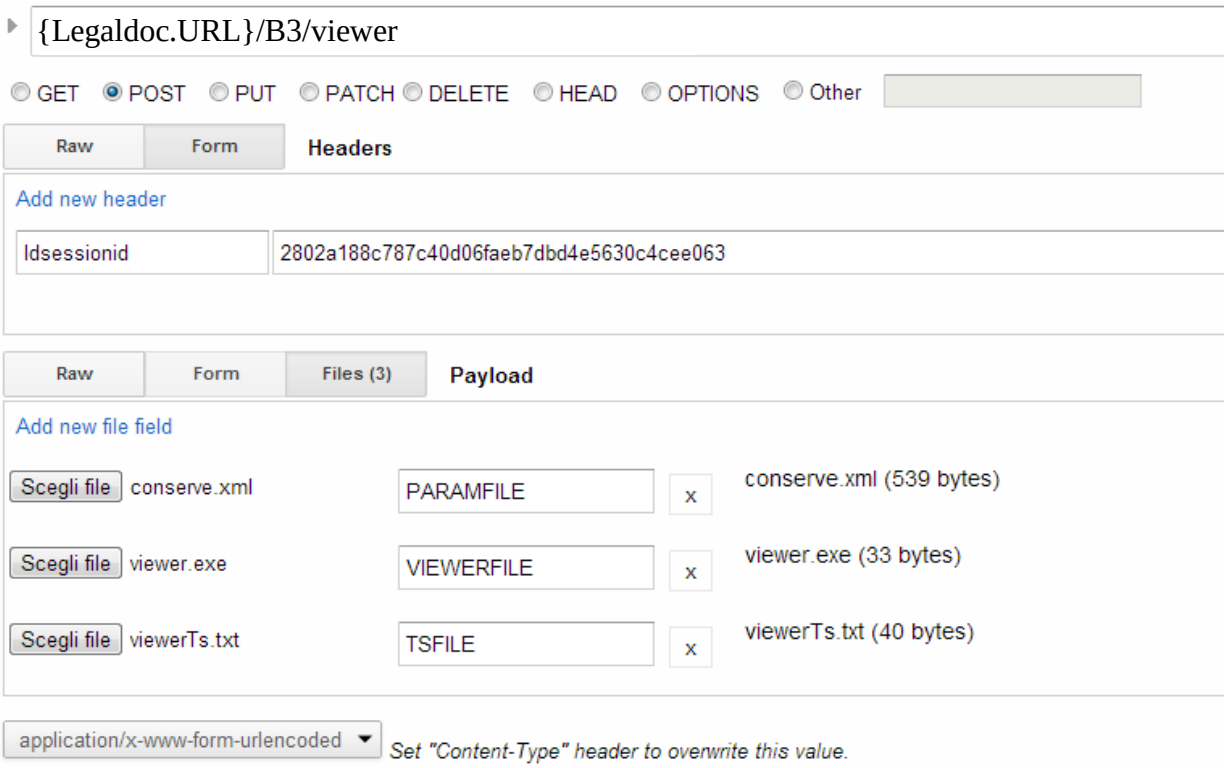

Figura 20: Esempio di utilizzo dei parametri per il caricamento di un viewer in un client per l'invocazione di servizi REST.

15.4.3 Output

Alla richiesta di upload di un viewer il sistema risponde con un codice http che indica il risultato dell'operazione: ad esempio a fronte di un upload conclusosi correttamente il sistema ritorna il codice HTTP 201 (created) ed il file IdC (Indice di Conservazione) firmato dal responsabile della conservazione e marcato temporalmente.

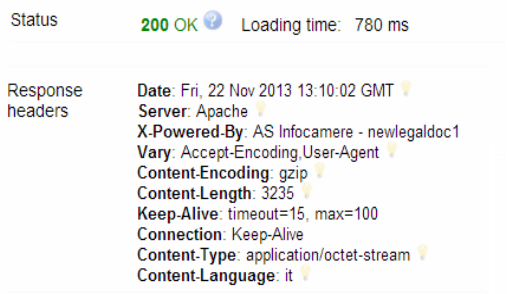

Figura 21: Esempio di risposta con successo al caricamento di un viewer.

Il file IdC attesta l'avvenuta conservazione del viewer.

Se invece fossero stati caricati parametri errati, il sistema potrebbe rispondere con xml nel formato:

```
<error>
   <code>LD_XXNNN</code>
   <description>Breve descrizione del problema.</description>
</error>
```

Il tag `<code>` indica il codice associato all'errore, mentre il tag `<description>` restituisce una breve descrizione del problema.

Figura 22: Esempio di risposta con fallimento al caricamento di un viewer.

Per una completa descrizione di tutti i codici si rimanda a [4].

Riportiamo un esempio di IdC:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<IdC>
  <SelfDescription>
    <ID>V1</ID>
    <CreatingApplication>
      <Name>Legaldoc</Name>
      <Version>1.0</Version>
      <Producer>Infocert</Producer>
    </CreatingApplication>
  </SelfDescription>
  <VdC>
    <ID>V1</ID>
    <MoreInfo>
      <EmbeddMetadata>
        <additionalInfo key="viewerCode">
            V1
        </additionalInfo>
        <additionalInfo key="bucket">B1</additionalInfo>
      </EmbeddMetadata>
```

```
      </MoreInfo>
    </VdC>
    <FileGroup>
      <File>
              <ID>1</ID>
              <Hash>02de3f5576b2e9b3f1eb45cc7d629e782c1078aeb6db4b5b7f8924449405efc6</Hash>
              <MoreInfo>
                      <EmbeddMetadata>ts.file</EmbeddMetadata>
              </MoreInfo>
      </File>
    </FileGroup>
    <Process>
      <TimeReference>
        <TimeInfo>2013-05-15T16:09:37+0200</TimeInfo>
      </TimeReference>
    </Process>
  </IdC>
```


In particolare, ecco il significato dei tag presenti:

TAG	*Significato*	*Attributo*	*Significato*
IdC	Radice del file xml		
SelfDescription	Informazioni relative all'indice di conservazione stesso		
ID	Identificativo univoco assegnato al viewer dal sistema di conservazione LegalDoc		
CreatingApplication	Informazioni sull'applicazione che ha generato l'IdC		
Name	Nome dell'applicazione che ha generato l'IdC		
Version	Versione dell'applicazione che ha generato l'IdC		
Producer	Nome del produttore dell'applicazione che ha generato l'IdC		
VdC	Informazioni relative al volume di conservazione		
ID	Identificativo univoco assegnato al viewer dal sistema di conservazione LegalDoc		
MoreInfo	Informazioni ulteriori relative al VdC		
EmbeddedMetadata	Le informazioni dell'elemento <MoreInfo> strutturate nel formato XML		
additionalInfo	Contenitore delle informazioni ulteriori	key="viewerCode"	Contiene il codice del viewer assegnatogli da LegalDoc
additionalInfo	Contenitore delle informazioni ulteriori	key="bucket"	ID del bucket a cui il viewer appartiene così come indicato nell'url del servizio di uplaod
FileGroup	Elemento di aggregazione di più file oggetto di conservazione		

TAG	*Significato*	*Attributo*	*Significato*
File	Informazioni relative al file oggetto di conservazione		
ID	Identificatore univoco del file descritto		
Hash	Informazioni sull'impronta del file cui l'elemento si riferisce		
MoreInfo	Informazioni ulteriori relative al file cui l'elemento si riferisce		
EmbeddMetadata	Nome del file cui l'elemento si riferisce		
Process	Informazioni relative alle modalità di svolgimento del processo di conservazione		
TimeReference	Informazioni relative alla data e ora di realizzazione dell'indice di conservazione		
TimeInfo	Generico riferimento temporale nel formato UNI ISO 8601:2010		

15.4.4 Il file dei parametri di upload

Il file dei parametri di upload è un file in formato XML simile al seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<viewerParameters>
  <mimetype>application/pdf;NA</mimetype>
  <description>viewer di test</description>
  <viewer_file>
   <viewer_name>viewer-01.exe</viewer_name>
   <viewer_hash>37fac9829dbd79a7fa8d792ceceda509649807238839c0bbd6184227cc2ba145</viewer_hash>
  </viewer_file>
  <ts_file>
   <ts_name>schedaTecnica.txt;1.0</ts_name>
   <ts_hash>7c5eb867859f83dc0720c09c321e4bff33ea4d6dc68aa229aab44dae76bd0bec</ts_hash>
  </ts_file>
  <activationTime>2013-10-28</activationTime>
</viewerParameters>
```

Di seguito sono riportati i tag necessari ed il loro significato:

TAG	*Significato*	*Attributo*	*Significato*	*Obbligatorietà*
viewerParameters	Tag radice per la richiesta di upload			S
mimetype	Indica il mime type di cui il viewer permette la visualizzazione			S
description	Descizione del viewer			S
viewer_file	Tag contenente le informazioni sul file viewer di cui si vuole fare upload			S
viewer_name	Il nome del file viewer di cui si vuole fare upload. E' il nome con cui verrà conservato il viewer in LegaldDoc			S
viewer_hash	E' l'hash calcolato con metodo SHA-256 del file viewer			S
ts_file	Tag contenente le informazioni sul file TS che si vuole conservare			S
ts_name	Il nome del file TS di cui si vuole fare l'upload. E' il nome con cui verrà conservato il file in LegaldDoc			S
ts_hash	E' l'hash calcolato con metodo SHA-256 del file TS			S
activationTime	La data di attivazione del viewer nel formato YYYY-MM-DD			S

15.4.5 Controlli sul file dei parametri di upload

LegalDoc esegue una serie di controlli sul contenuto del file xml dei parametri di conservazione. Di seguito si riporta l'elenco dei controlli effettuati.

1. **Tracciato del file dei parametri**: il file di indice viene validato con l'xsd riportato in appendice A
2. **MimeType**: si verifica se il mimetype indicato è presente in almeno una delle policy dichiarate per il bucket

15.4.6 Download dei visualizzatori

Il download dei viewer è il servizio LegalDoc che permette di estrarre dal sistema i visualizzatori necessari per aprire un documento per cui sia stata completata correttamente la procedura di conservazione, di rettifica o di cancellazione.

Il download restituisce l'insieme dei viewer in un unico pacchetto zip.

15.4.7 Prerequisiti, url e invocazione del servizio

Per richiamare questo servizio si deve utilizzare il metodo Http GET ed indicare la modalità con la quale si vogliono avere i viewer in risposta ossia:

```
application/x-zip-compressed (si otterrà uno zip contenente tutti i viewer del documento)
```

Inoltre bisogna aggiungere nell'header un campo con nome **ldsessionid** e il valore restituito dal servizio di login.

L'URL a cui risponde il servizio è:

{Legaldoc.URL}/{bucket}/viewer/{token}

dove token è l'identificativo univoco del documento di cui si vogliono ottenere i viewer.

15.4.8 Input

La richiesta di download avviene tramite chiamata in GET all'url indicato. Di seguito un esempio di invocazione per l'esibizione di un file zip:

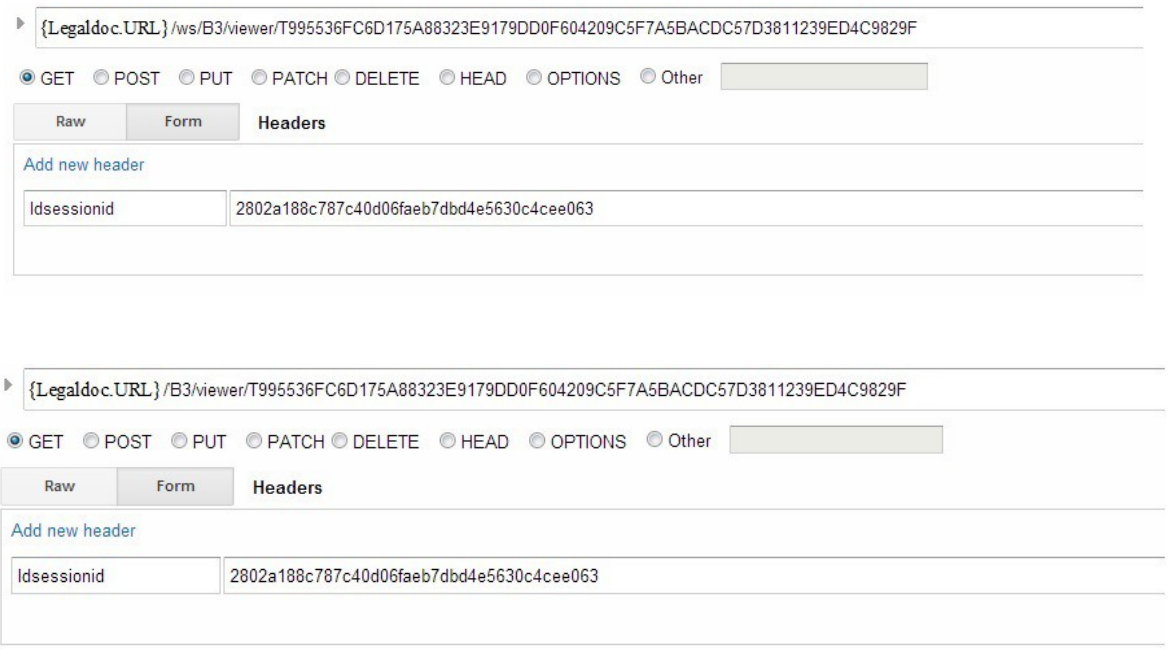

Figura 23: Esempio di utilizzo dei parametri per il download dei viewer in formato zip in un client per l'invocazione di servizi REST.

Come si evince dalla figura, le informazioni necessarie sono passate attraverso l'url di invocazione.

15.4.9 Output

Alla richiesta di download dei viewer di un documento il sistema risponde:

- con uno zip dei viewer

In caso di risposta positiva, il codice HTTP in è 200 OK e avrà il `content-type = application/zip` come mostrato nell'esempio seguente:

Status	200 OK Loading time: 8190 ms
Response headers	**Date**: Fri, 22 Nov 2013 14:10:15 GMT **Server**: Apache **X-Powered-By**: AS Infocamere - newlegaldoc1 **content-disposition**: inline; filename=DocBridgeView_Setup2.exe.zip **Vary**: Accept-Encoding,User-Agent **Content-Encoding**: gzip **Keep-Alive**: timeout=15, max=100 **Connection**: Keep-Alive **Transfer-Encoding**: chunked **Content-Type**: application/zip **Content-Language**: it

Figura 24: Esempio di risposta con successo al download dei viewer in formato zip.

16. Invocazione Web Services LegalDoc

16.1 Servizi esposti

16.2 Composizione dell'URL da invocare per accedere a LegalDoc

Gli url per l'ambiente di produzione `{Legaldoc.URL}` e di collaudo `{LegaldocCL.URL}` sono elencati nel documento Scheda dati tecnici per l'attivazione di LegalDoc (rif. [1]).

L'URL da invocare per accedere ai servizi LegalDoc è da comporre come elencato in tabella (nelle parentesi graffe i parametri che il cliente deve inserire):

Servizio	Metodo HTTP	URL	Parametro1	/Parametro2	/Parametro3	Content-type della richiesta
Login	POST	{Legaldoc.URL}	session			application/x-www-form-urlencoded
Logout	DELETE	{Legaldoc.URL}	session			application/x-www-form-urlencoded
Conservazione	POST	{Legaldoc.URL}	/{bucket}	/document		multipart/form-data
Esibizione	GET	{Legaldoc.URL}	/{bucket}	/document	/{token}	application/x-zip-compressed multipart/mixed
Rettifica	PUT	{Legaldoc.URL}	/{bucket}	/document	/{token}	multipart/form-data
Cancellazione	DELETE	{Legaldoc.URL}	/{bucket}	/document	/{token}	multipart/form-data
Ricerca	POST	{Legaldoc.URL}	/search	/standard		application/xml
Upload viewer	POST	{Legaldoc.URL}	/{bucket}	/viewer		multipart/form-data
Download viewer	GET	{Legaldoc.URL}	/{bucket}	/viewer	/{token}	application/x-zip-compressed

16.3 Esempi di invocazione dei servizi in Java

16.3.1 Librerie

Sono necessarie le seguenti librerie (dependency Maven):

```
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpcore</artifactId>
    <version>4.1.4</version>
</dependency>
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpmime</artifactId>
    <version>4.1.3</version>
</dependency>
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.1.3</version>
</dependency>
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.0.1</version>
</dependency>
```

16.3.2 Esempio metodo chiamata ws generico

Viene un fornito un generico metodo Java per la chiamata ad un ws REST a titolo puramente esemplificativo:

```
public HttpResponse callWS(String url, HttpMethod method, HttpEntity entity) {
            return callWS(url, method, null, null, entity);
}
public HttpResponse callWS(String url, HttpMethod method, String mediaType, String acceptType,
HttpEntity entity) {
    try{
            HttpRequestBase baseRequest = null;
            switch(method) {
            case GET:
                    baseRequest = new HttpGet(url);
                    break;
            case POST:
                    baseRequest = new HttpPost(url);
                    if(entity != null)
                            ((HttpPost)baseRequest).setEntity(entity);
                    break;
```

```
            case PUT:
                baseRequest = new HttpPut(url);
                if(entity != null)
                    ((HttpPut)baseRequest).setEntity(entity);
                break;
            case DELETE:
                baseRequest = new HttpDelete(url);
                break;
            }
            if(mediaType != null) baseRequest.setHeader("Content-Type", mediaType);
            if (acceptType != null) baseRequest.setHeader("Accept", acceptType);
            baseRequest.addHeader("ldSessionId", ldSessionId);

            //da settare per l'esibizione multipart
            //baseRequest.addHeader("Accept", "multipart/mixed");

            HttpClient httpclient = new DefaultHttpClient();
            return httpclient.execute(baseRequest);
        }
        catch(Exception ex) {//gestire eccezione}
    }
```

16.3.3 Esempio metodo chiamata ws login

Si riporta un esempio di utilizzo del servizio di login in LegalDoc da un software scritto in linguaggio Java.

```
    public void login(String resourceUrl, userid, password) throws Exception {
        try {
            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
            nameValuePairs.add(new BasicNameValuePair("userid", userid));
            nameValuePairs.add(new BasicNameValuePair("password", password));
            HttpResponse response = callWS(resourceUrl, HttpMethod.POST, new
UrlEncodedFormEntity(nameValuePairs));
            HttpEntity resEntity = response.getEntity();
            String resultXml = EntityUtils.toString(resEntity);
        }
        catch (Exception ex) {//gestire eccezione}
    }
```

16.3.4 Esempio di logout

Si riporta un esempio di utilizzo del servizio di logout in LegalDoc da un software scritto in linguaggio Java.

```
    public void logout(String resourceUrl) {
        callWS(resourceUrl, HttpMethod.DELETE, null);
    }
```

16.3.5 Esempio di conservazione

Si riporta un esempio di utilizzo del servizio di conservazione in LegalDoc da un software scritto in linguaggio Java.

```
public void conserve(String conserveResourceUrl,
    InputStream parametersStream, String parametersFileName,
    InputStream indexStream, String indexMimeType, String indexFileName,
    InputStream dataStream, String dataMimeType, String dataFileName) {
    try {
            MultipartEntity requestEntity = new MultipartEntity();
            requestEntity.addPart("PARAMFILE",      new      InputStreamBody(parametersStream,
"conserve.xml"));
            requestEntity.addPart("INDEXFILE", new InputStreamBody(indexStream, indexMimeType,
indexFileName));
            requestEntity.addPart("DATAFILE",  new  InputStreamBody(dataStream,  dataMimeType,
dataFileName));
    // creazione request
            HttpResponse      response      =      this.restInvoker.callWS(conserveResourceUrl,
HttpMethod.POST,      requestEntity);
            if (response.getStatusLine().getStatusCode() == 201) {
                    // chiamata conserve ok
                    logger.debug("Documento conservato.");
            } else
                    logger.debug("Documento non conservato.");
            indexStream.close();
            dataStream.close();
            parametersStream.close();
    }
    catch (Exception ex) {//gestire eccezione}
}
```

16.3.6 Esempio di esibizione zip

Si riporta un esempio di utilizzo del servizio di esibizione in LegalDoc da un software scritto in linguaggio Java.

```
public void retrieve(String retrieveResourceUrl) {
    try {
            HttpResponse     response     =     callWS(retrieveResourceUrl,     HttpMethod.GET,
"application/x-zip-compressed", null, null);
            if (response.getStatusLine().getStatusCode() == 200) {
            // chiamata retrieve ok
                    logger.debug("Documento estratto.");
            } else
                    logger.debug("Documento non estratto.");
    }
    catch (Exception ex) {//gestire eccezione}
}
```

16.3.7 Esempio di esibizione multipart

Si riporta un esempio di utilizzo del servizio di esibizione in LegalDoc da un software scritto in linguaggio Java.

```
public void retrieve(String retrieveResourceUrl) {
    try {
        //dopo aver aggiunto l'header "Accept" = "multipart/mixed"
        HttpResponse response = callWS(retrieveResourceUrl, HttpMethod.GET,
"application/x-zip-compressed", null, null);
        if (response.getStatusLine().getStatusCode() == 200) {
        // chiamata retrieve ok
            logger.debug("Documento estratto.");
        } else
            logger.debug("Documento non estratto.");
    }
    catch (Exception ex) {//gestire eccezione}
}
```

16.3.8 Esempio di ricerca

Si riporta un esempio di utilizzo del servizio di ricerca in LegalDoc da un software scritto in linguaggio Java.

```
public void search(String searchResourceUrl) {
    try {
        StringEntity entity = new StringEntity("xml input ricerca", "UTF-8");
        entity.setContentType("application/xml");
        HttpResponse response = callWS(searchResourceUrl, HttpMethod.POST,
        "application/xml", entity);
        if (response.getStatusLine().getStatusCode() == 200) {
            // chiamata search ok
            logger.debug("Documento estratto.");
        } else
            logger.debug("Documento non estratto.");
    }
    catch (Exception ex) {//gestire eccezione}
}
```

17. Appendice A: XSD file utilizzati in LegalDoc

17.1 XSD del file dei parametri di conservazione

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="parameters">
        <xs:annotation>
            <xs:documentation>Elemento di root per una chiamata al servizio di
conservazione</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:all>
                <xs:element name="policy_id">
                    <xs:annotation>
                        <xs:documentation>Id della policy</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:pattern value="P\d+"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="index_file">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:annotation>
                                <xs:documentation>Sezione del file di
indici</xs:documentation>
                            </xs:annotation>
                            <xs:element name="index_name">
                                <xs:annotation>
                                    <xs:documentation>Nome del file
di indici</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                    <xs:restriction
base="xs:string">
                                        <xs:pattern value="[_\
[\]\w\.@\s-]{3,80}"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="index_hash">
                                <xs:annotation>
                                    <xs:documentation>Hash del file
di indici</xs:documentation>
                                </xs:annotation>
```

```
                                                        <xs:simpleType>
                                                                <xs:restriction
base="xs:string">
                                                                        <xs:pattern   value="[A-
Fa-f0-9]{64}"/>
                                                                <xs:restriction>
                                                        <xs:simpleType>
                                                <xs:element>
                                                <xs:element name="index_mimetype">
                                                        <xs:annotation>
                                                                <xs:documentation>Mimetype  del
file di indici</xs:documentation>
                                                        <xs:annotation>
                                                        <xs:simpleType>
                                                                <xs:restriction
base="xs:string">
                                                                        <xs:pattern
value="[\p{L}\-\p{Nd}\.\|\s\+/;]+"/>
                                                                <xs:restriction>
                                                        <xs:simpleType>
                                                <xs:element>
                                        <xs:sequence>
                                <xs:complexType>
                        <xs:element>
                        <xs:element name="data_file">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:annotation>
                                                        <xs:documentation>Sezione   del   file
inviato in conservazione</xs:documentation>
                                                <xs:annotation>
                                                <xs:element name="data_name">
                                                        <xs:annotation>
                                                                <xs:documentation>Nome del file
inviato in conservazione</xs:documentation>
                                                        <xs:annotation>
                                                        <xs:simpleType>
                                                                <xs:restriction
base="xs:string">
                                                                        <xs:pattern   value="[_\
[\]\w\.@\s-]{3,80}"/>
                                                                <xs:restriction>
                                                        <xs:simpleType>
                                                <xs:element>
                                                <xs:element name="data_hash">
                                                        <xs:annotation>
                                                                <xs:documentation>Hash del file
inviato in conservazione</xs:documentation>
                                                        <xs:annotation>
                                                        <xs:simpleType>
                                                                <xs:restriction
base="xs:string">
                                                                        <xs:pattern   value="[A-
Fa-f0-9]{64}"/>
```

```
                                                                    </xs:restriction>
                                                            </xs:simpleType>
                                                    </xs:element>
                                                    <xs:element name="data_mimetype">
                                                            <xs:annotation>
                                                                    <xs:documentation>Mimetype  del
file inviato in conservazione</xs:documentation>
                                                            </xs:annotation>
                                                            <xs:simpleType>
                                                                    <xs:restriction
base="xs:string">
                                                                            <xs:pattern
value="[\p{L}\-\p{Nd}\.\|\s\+/;]+"/>
                                                                    </xs:restriction>
                                                            </xs:simpleType>
                                                    </xs:element>
                                            </xs:sequence>
                                    </xs:complexType>
                            </xs:element>
                            <xs:element name="path">
                                    <xs:annotation>
                                            <xs:documentation>Percorso   di   conservazione   del
file</xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                    <xs:pattern value="(/|(/[_\w' \.\-@]+)+)"/>
                                            </xs:restriction>
                                    </xs:simpleType>
                            </xs:element>
                    </xs:all>
            </xs:complexType>
    </xs:element>
</xs:schema>
```

17.2 XSD del file di indice

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema attributeFormDefault="unqualified" elementFormDefault="qualified" version="1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="legaldocIndex">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="field" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                            <xsd:attribute name="label">
                                    <xsd:simpleType>
                                            <xsd:restriction base="xsd:string">
```

```
                                                <xsd:pattern
                                                        value="[A-Za-z\d\s\-_\.']{1,512}" />
                                        </xsd:restriction>
                                </xsd:simpleType>
                        </xsd:attribute>
                        <xsd:attribute name="name" use="required">
                                <xsd:simpleType>
                                        <xsd:restriction base="xsd:string">
                                                <xsd:pattern value="[A-Za-z\d_-]{1,256}" />
                                        </xsd:restriction>
                                </xsd:simpleType>
                        </xsd:attribute>
                </xsd:extension>
            </xsd:simpleContent>
           </xsd:complexType>
         </xsd:element>
       </xsd:sequence>
       <xsd:attribute name="documentClass" use="required">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:pattern value="[A-Za-z\d_-]{1,256}" />
                </xsd:restriction>
            </xsd:simpleType>
       </xsd:attribute>
       <xsd:attribute name="label">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:pattern value="[A-Za-z\d\s\-_\.']{1,256}" />
                </xsd:restriction>
            </xsd:simpleType>
       </xsd:attribute>
     </xsd:complexType>
   </xsd:element>
 </xsd:schema>
```

17.3 XSD del file IdC

Questo XSD rispetta l'XSD descritto nel DPCM del 3 dicembre 2013

```
 <xsd:schema xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
<xsd:element name='IdC'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='SelfDescription'/>
      <xsd:element ref='VdC'/>
      <xsd:element ref='FileGroup' maxOccurs='unbounded'/>
      <xsd:element ref='Process'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='SelfDescription'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='ID'/>
      <xsd:element ref='CreatingApplication'/>
```

```
      <xsd:element ref='SourceIdC' minOccurs='0' maxOccurs='unbounded'/>
      <xsd:element ref='MoreInfo' minOccurs='0' maxOccurs='1'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='VdC'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='ID'/>
      <xsd:element ref='SourceVdC' minOccurs='0' maxOccurs='unbounded'/>
      <xsd:element ref='VdCGroup' minOccurs='0' maxOccurs='1'/>
      <xsd:element name="MoreInfo">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="EmbeddMetadata">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element maxOccurs="unbounded" ref="additionalInfo"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="additionalInfo">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="key" use="required" type="xsd:string"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name='FileGroup'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='Label' minOccurs='0' maxOccurs='1'/>
      <xsd:element ref='File' maxOccurs='unbounded'/>
      <xsd:element ref='MoreInfo' minOccurs='0' maxOccurs='1'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='Process'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='Agent' maxOccurs='unbounded'/>
      <xsd:element ref='TimeReference'/>
      <xsd:element ref='LawAndRegulations' minOccurs='0' maxOccurs='1'/>
      <xsd:element ref='MoreInfo' minOccurs='0' maxOccurs='1'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='ID'>
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base='xsd:string'>
        <xsd:attribute name='scheme' type='xsd:string' default='local'/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name='CreatingApplication'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='Name'/>
      <xsd:element ref='Version'/>
      <xsd:element ref='Producer'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='SourceIdC'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='ID'/>
      <xsd:element ref='Path' minOccurs='0' maxOccurs='1'/>
      <xsd:element ref='Hash'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='MoreInfo'>
  <xsd:complexType>
```

```
    <xsd:choice>
      <xsd:element ref='EmbeddMetadata'/>
      <xsd:element ref='ExternalMetadata'/>
    </xsd:choice>
    <xsd:attribute name='XMLScheme' type='xsd:NMTOKEN' use='required'/>
  </xsd:complexType>
</xsd:element>
<xsd:element name='SourceVdC'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='ID'/>
      <xsd:element ref='IdC_ID'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='VdCGroup'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='Label'/>
      <xsd:element ref='ID' minOccurs='0' maxOccurs='1'/>
      <xsd:element ref='Description' minOccurs='0' maxOccurs='1'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='Label' type='xsd:string'></xsd:element>
<xsd:element name='File'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='ID'/>
      <xsd:element ref='Path' minOccurs='0' maxOccurs='1'/>
      <xsd:element ref='Hash'/>
      <xsd:element ref='PreviousHash' minOccurs='0' maxOccurs='1'/>
      <xsd:element name="MoreInfo">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element type="xsd:string" name="EmbeddMetadata"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name='encoding' use='required'>
      <xsd:simpleType>
        <xsd:restriction base='xsd:string'>
          <xsd:enumeration value='7bit'/>
          <xsd:enumeration value='8bit'/>
          <xsd:enumeration value='binary'/>
          <xsd:enumeration value='base64'/>
          <xsd:enumeration value='quotedprintable'/>
          <xsd:enumeration value='xtoken'/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name='extension' type='xsd:NMTOKEN' use='optional'/>
    <xsd:attribute name='format' type='xsd:string' use='required'/>
  </xsd:complexType>
</xsd:element>
<xsd:element name='Agent'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='AgentName'/>
      <xsd:element ref='Agent_ID' minOccurs='0' maxOccurs='unbounded'/>
      <xsd:element ref='MoreInfo' minOccurs='0' maxOccurs='1'/>
    </xsd:sequence>
    <xsd:attribute name='type' use='required'>
      <xsd:simpleType>
        <xsd:restriction base='xsd:string'>
          <xsd:enumeration value='person'/>
          <xsd:enumeration value='organization'/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name='role' use='required'>
      <xsd:simpleType>
        <xsd:restriction base='xsd:string'>
          <xsd:enumeration value='Delegate'/>
          <xsd:enumeration value='PreservationManager'/>
          <xsd:enumeration value='Operator'/>
          <xsd:enumeration value='PublicOfficer'/>
          <xsd:enumeration value='OtherRole'/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name='otherRole' type='xsd:string' use='optional'/>
  </xsd:complexType>
```

```
  </xsd:element>
  <xsd:element name='TimeReference'>
    <xsd:complexType>
      <xsd:choice>
        <xsd:element ref='DetachedTimeStamp'/>
        <xsd:element ref='AttachedTimeStamp'/>
        <xsd:element ref='TimeInfo'/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name='LawAndRegulations'>
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base='xsd:string'>
          <xsd:attribute name='language' type='xsd:NMTOKEN' default='it'/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name='Name' type='xsd:string'></xsd:element>
  <xsd:element name='Version' type='xsd:string'></xsd:element>
  <xsd:element name='Producer' type='xsd:string'></xsd:element>
  <xsd:element name='Path' type='xsd:string'></xsd:element>
  <xsd:element name='Hash'>
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base='xsd:string'>
          <xsd:attribute name='canonicalXML' use='optional'>
            <xsd:simpleType>
              <xsd:restriction base='xsd:string'>
                <xsd:enumeration value='yes'/>
                <xsd:enumeration value='no'/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:attribute>
          <xsd:attribute name='function' type='xsd:NMTOKEN' use='required'/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name='EmbeddMetadata' type='xsd:string'></xsd:element>
  <xsd:element name='ExternalMetadata'>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref='ID'/>
        <xsd:element ref='Path' minOccurs='0' maxOccurs='1'/>
        <xsd:element ref='Hash'/>
        <xsd:element ref='PreviousHash' minOccurs='0' maxOccurs='1'/>
        <xsd:element ref='MoreInfo' minOccurs='0' maxOccurs='1'/>
      </xsd:sequence>
      <xsd:attribute name='encoding' use='optional' default='binary'>
        <xsd:simpleType>
          <xsd:restriction base='xsd:string'>
            <xsd:enumeration value='7bit'/>
            <xsd:enumeration value='8bit'/>
            <xsd:enumeration value='binary'/>
            <xsd:enumeration value='base64'/>
            <xsd:enumeration value='quotedprintable'/>
            <xsd:enumeration value='xtoken'/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:attribute name='extension' type='xsd:NMTOKEN' use='optional'/>
      <xsd:attribute name='format' type='xsd:string' use='required'/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name='IdC_ID'>
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base='xsd:string'>
          <xsd:attribute name='shceme' type='xsd:string' default='local'/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name='Description'>
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base='xsd:string'>
          <xsd:attribute name='language' type='xsd:NMTOKEN' default='it'/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
```

```
<xsd:element name='PreviousHash'>
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base='xsd:string'>
        <xsd:attribute name='canonicalXML' use='optional'>
          <xsd:simpleType>
            <xsd:restriction base='xsd:string'>


            <xsd:enumeration value='yes'/>
              <xsd:enumeration value='no'/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name='function' type='xsd:NMTOKEN' use='required'/>
        <xsd:attribute name='relatedIdC' type='xsd:NMTOKEN' use='required'/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name='AgentName'>
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref='NameAndSurname'/>
      <xsd:element ref='FormalName'/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name='Agent_ID'>
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base='xsd:string'>
        <xsd:attribute name='scheme' use='required'>
          <xsd:simpleType>
            <xsd:restriction base='xsd:string'>
              <xsd:enumeration value='TaxCode'/>
              <xsd:enumeration value='VATRegistrationNumber'/>
              <xsd:enumeration value='NationalHealthCareAuthority'/>
              <xsd:enumeration value='OtherScheme'/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name='otherScheme' type='xsd:string' use='optional'/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name='AttachedTimeStamp'>
  <xsd:complexType>
    <xsd:attribute name='normal' type='xsd:NMTOKEN' use='required'/>
  </xsd:complexType>
</xsd:element>
<xsd:element name='DetachedTimeStamp'>
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base='xsd:string'>
        <xsd:attribute name='normal' type='xsd:NMTOKEN' use='required'/>
        <xsd:attribute name='encoding' use='optional' default='binary'>
          <xsd:simpleType>
            <xsd:restriction base='xsd:string'>
              <xsd:enumeration value='7bit'/>
              <xsd:enumeration value='8bit'/>
              <xsd:enumeration value='binary'/>
              <xsd:enumeration value='base64'/>
              <xsd:enumeration value='quotedprintable'/>
              <xsd:enumeration value='xtoken'/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name='format' type='xsd:string' use='required'/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name='TimeInfo' type='xsd:string'></xsd:element>
<xsd:element name='NameAndSurname'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref='FirstName'/>
      <xsd:element ref='LastName'/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name='FirstName' type='xsd:string'></xsd:element>
```

```
  <xsd:element name='LastName' type='xsd:string'></xsd:element>
  <xsd:element name='FormalName' type='xsd:string'></xsd:element>
</xsd:schema>
```