

DOSIGN

Servizi Applicativi

Manuale Integrazione Fruitori

Versione 1.0

VERIFICHE E APPROVAZIONI

VERS.	REDAZIONE		CONTROLLO APPROVAZIONE		AUTORIZZAZIONE E EMISSIONE	
	NOME	DATA	NOME	DATA	NOME	DATA
1.0	COERO-BORGA CASTELLANETA	09/03/2022	MARENCHINO	09/03/2022	MARENCHINO	09/03/2022

STATO DELLE VARIAZIONI

VERSIONE	DATA VALIDITÀ	PARAGRAFO O PAGINA	DESCRIZIONE DELLA VARIAZIONE
1.0	09/03/2022		Versione iniziale

Sommario

Sommario	2
1 Introduzione	3
1.1 Scopo del documento	3
1.2 Prerequisito	3
1.3 Riferimenti	3
2 Principali scenari d'uso dei servizi	4
2.1 Verifica della Firma Digitale	5
2.2 Sigillo Elettronico	7
2.3 Sbustamento	8
2.4 Apposizione di Firma Digitale	9
2.4.1 Firma Remota	9
2.4.2 Firma Locale	11
2.5 Marcatura temporale	14

1 Introduzione

I servizi Dosign permettono di effettuare:

- la verifica di validità dei documenti firmati digitalmente secondo i profili previsti dalla normativa italiana;
- la verifica di validità dei documenti firmati digitalmente secondo le esigenze del business di Doqui Acta;
- la marcatura temporale dei documenti firmati digitalmente secondo i profili previsti dalla normativa italiana;
- l'apposizione di firme digitali per mezzo di credenziali di firma remota;
- l'apposizione di firme digitali per mezzo di certificati con smartcard e PIN di sblocco.

1.1 Scopo del documento

Il documento è indirizzato a tutti i possibili fruitori dei servizi applicativi del sistema Dosign.

Lo scopo è quello di fornire indicazioni per una integrazione rapida con Dosign e di essere d'aiuto in fase di stima di progetto.

1.2 Prerequisito

Per avviare la fase di sviluppo dell'integrazione, è necessario contattare il supporto tecnico Dosign tramite invio di un messaggio di posta all'indirizzo e-mail assistenza_dosign@csi.it.

1.3 Riferimenti

Si riportano di seguito i documenti relativi alle specifiche di dettaglio, disponibili in aggiunta al presente manuale ottenibili contattando gruppo_dosign@csi.it.

Numero	Nome file / link	Descrizione
[1]	http://tst-doquilab.csi.it/docs/dosign/	Descrizione dei servizi esposti (JavaDoc)

	<p style="text-align: center;">DOSIGN Manuale Integrazione Fruttori</p>	<p style="text-align: center;">DOSIGN-MANUALE- INTEGRAZIONE- FRUTTORI</p> <p style="text-align: right;">Pag. 4 di 14</p>
--	---	--

2 Principali scenari d'uso dei servizi

Di seguito sono illustrati, in modo schematico, alcuni degli scenari tipici di interazione di una applicazione client attraverso i servizi Dosign.

Questa non vuole essere una illustrazione esaustiva delle modalità in cui possono essere utilizzati ed orchestrati i servizi esposti, ma offrire una indicazione sul flusso logico degli scenari di utilizzo più comuni.

2.1 Verifica della Firma Digitale

È possibile verificare la validità di documenti sottoscritti in uno dei seguenti formati della busta crittografica:

- **PDF PKCS #7** (Public Key Cryptography Standards)
Formato utilizzato per inserire buste crittografiche all'interno di documenti PDF secondo lo standard ISO 32000-1. Tutti i documenti PDF sottoscritti mantengono l'estensione originaria PDF.
- **PADES-BES** (PDF Advanced Electronic Signature Basic Electronic Signature)
Formato utilizzato per inserire buste crittografiche all'interno di documenti PDF secondo il più recente standard CADES. È il formato utilizzato ad oggi.
Nella sottoscrizione di documenti PDF è possibile utilizzare features dello standard ISO 32000 per apporre firme visibili e invisibili valorizzando metadati del dizionario dati PDF utilizzati per caratterizzare le sottoscrizioni, le opzioni disponibili sono descritte in ISO 32000-1.
- **CAdES-BES** (CMS Advanced Electronic Signature Basic Electronic Signature)
Formato utilizzato per inserire all'interno di una busta crittografica documenti sottoscritti, di formati diversi, secondo lo standard CADES. È il formato da utilizzare ad oggi (ulteriore estensione P7M oltre all'estensione del documento).
Una busta crittografica CADES può a sua volta contenere altre buste crittografiche in questo caso viene applicata un'ulteriore estensione P7M. La busta CADES non necessariamente deve contenere il documento sottoscritto: in questo caso (firma detached) documento e busta sono file separati (il file della busta ha estensione P7S o P7M).
- **XAdES-BES** (XML Advanced Electronic Signature Basic Electronic Signature)
Formato utilizzato per la sottoscrizione in linguaggio XML secondo il più recente standard CADES.
- **PADES-T** (PADES with Time)
Formato del tutto equivalente al PAdES-BES ma che consente di aggiungere informazioni di tempo certificate alla sottoscrizione.
- **CAdES-T** (CAdES with Time)
Formato del tutto equivalente al CAdES-BES ma che consente di aggiungere informazioni di tempo certificate alla sottoscrizione.
- **XAdES-T** (XAdES with Time)
Formato del tutto equivalente al XAdES-BES ma che consente di aggiungere informazioni di tempo certificate alla sottoscrizione.
- **Time Stamped Data** secondo standard **RFC 5544**
Formato utilizzato per inserire all'interno di una busta crittografica documenti e dati di marcatura temporale in formato RFC 3161 secondo lo standard RFC 5544. I documenti a loro volta possono essere sottoscritti oppure no.
È il formato da utilizzare ad oggi in luogo del formato custom Infocert M7M (ulteriore estensione TSD oltre all'estensione del documento).

Dosign mette a disposizione alcuni metodi per la verifica della firma digitale. La prima distinzione che si può fare è tra metodi sincroni e asincroni. I primi, poi, si differenziano sulla base del tipo di file da verificare.

• Verifica Sincrona

- **verifyDocument** ([javaDoc](#))
Il metodo verifica la validità delle buste crittografiche inserite all'interno di documenti firmati digitalmente.
Si applica ai formati PDF PKCS #7, PAdES-BES, PAdES-T, PKCS #7, CAdES-BES, CAdES-T, Infocert M7M e Time Stamped Data (RFC 5544).
- **verifyDocumentParams** ([javaDoc](#))
Esattamente come il metodo precedente ma sono ammesse le seguenti direttive di verifica:
 - **verificationDate** Data di verifica dei certificati di sottoscrizione presenti nella busta crittografica;
 - **verificationType** Criterio di verifica del Certificate Status (NA);

- **verificationScope** Scopo del certificato (NA);
- **profileType** Profilo di verifica del Certificate Path (NA).
- **verifyDocumentXml** ([javaDoc](#))
Il metodo verifica la validità delle buste crittografiche XAdES-X utilizzate per la sottoscrizione in linguaggio XML.
Occorre specificare il documento XML (buffer) nel formato XAdES. Si applica ai formati XAdES-BES e XAdES-T.

Questi 3 metodi:

- ricevono in **input** una struttura dati costituita dai file che rappresentano il documento di cui si vuole effettuare la verifica della firma;
- restituiscono in **output** una struttura dati (**verifyReport**) che descrive la busta crittografica XAdES-X ovvero l'insieme di firme semplici, parallele e controfirme con le eventuali marche temporali.

Per ogni firma sono riportati i dati dei certificati X509 coinvolti insieme all'esito delle verifiche. Il **Signature** è la struttura dati che descrive la singola *electronic signature* che si distingue in firma, controfirma o marca temporale.

La proprietà **errorCode** indica se il processo si è concluso correttamente (e restituisce 0) oppure con fallimento. In questo caso viene restituito il numero corrispondente al passo di verifica che ha causato l'errore:

1. Verifica conformità e integrità busta crittografica.
2. Sbustamento busta crittografica.
3. Verifica consistenza firma.
4. Verifica validità certificato.
5. Verifica certification authority.
6. Verifica lista di revoca — CRL aggiornata non disponibile.
7. Verifica lista di revoca — certificato presente nella CRL.

Lo stato di conformità della busta crittografica ricevuta in input è indicato nella proprietà **conformitaParametriInput**. Il risultato è così definito:

0. NON OK
1. OK

• Verifica Asincrona

- **verifyAsyncDocument** ([javaDoc](#))
Il metodo verifica la validità delle buste crittografiche inserite all'interno di documenti firmati digitalmente.
Si applica ai formati PDF PKCS #7, PAdES-BES, PAdES-T, PKCS #7, CAdES-BES, CAdES-T, XAdES-BES, XAdES-T, Infocert M7M e Time Stamped Data (RFC 5544).
A differenza dell'equivalente metodo sincrono riceve in input un AsyncVerifyDto così composto:
 - signed → documento da verificare.
 - notifyUrl → url a cui notificare la fine del processo di verifica
 - tokenUuid → identificativo del job di verifica (generato dal client o dal servizio se null)Restituisce in output il tokenUuid.
- **getAsyncReport** ([javaDoc](#))
Il metodo recupera il report di validità delle buste crittografiche creato da verifyAsyncDocument.
Riceve in input un tokenUuid (identificativo del job di verifica creato dalla verifyAsyncDocument) e restituisce in output un AsyncReportDto contenente il risultato della verifica così composto:
 - report → VerifyReport,
 - status → SCHEDULED (da evadere) - READY (pronto) - EXPIRED (richiesta scaduta)

Vedere l'esempio al fondo del capitolo 2.2.

2.2 Sigillo Elettronico

Dosign mette a disposizione alcuni metodi per l'apposizione di un sigillo.

Si può fare una distinzione tra metodi sincroni e asincroni.

- **Apposizione Sincrona**

- **sigillo** ([javaDoc](#))

Metodo per l'apposizione del sigillo elettronico.

Riceve in input un SigilloSignatureDto (cioè i dati per l'apposizione del sigillo elettronico) e restituisce in output il byte array del file sigillato.

- **Apposizione Asincrona**

- **asyncSigillo** ([javaDoc](#))

Metodo per l'apposizione asincrona del sigillo elettronico.

Riceve in input un AsyncSigilloInDto così composto:

- data → documento da sigillare.
- notifyUrl → url a cui notificare la fine del processo di sigillatura
- tokenUid → identificativo del job di sigillatura (generato dal client o dal servizio se null)

Restituisce in output il tokenUid.

- **getAsyncSigilloOutDto** ([javaDoc](#))

Metodo per il recupero del documento con sigillo creato da asyncSigillo.

Riceve in input un tokenUid (identificativo del job di verifica creato creato nella asyncSigillo) e restituisce in output un AsyncSigilloOutDto contenente il risultato della firma così composto:

- data → byte array del file sigillato,
- status → SCHEDULED (da evadere) - READY (pronto) - EXPIRED (richiesta scaduta)

Esempio dell'applicazione di un sigillo seguito dalla verifica della sua validità:

```
Protected void testCOM3542() {
    try {
        String pdf = "D:\\Doc\\sign\\pkbox\\COM-3542\\bd7780dc-1582-11ec-a358-1d5908d385f3.pdf";
        File pdfFile = new File(pdf);
        byte[] byteArray = FileUtils.readFileToByteArray(pdfFile);
        SigilloSignatureDto sigilloDto = new SigilloSignatureDto();
        sigilloDto.setData(byteArray);
        sigilloDto.setDelegatedDomain("faCSI"); -- fisso
        sigilloDto.setDelegatedUser("applicativoX"); -- applicativo delegato
        sigilloDto.setDelegatedPassword("XYZ");
        sigilloDto.setType("PDF");
        sigilloDto.setUser("CF_DELEGANTE"); -- codice Fiscale del Delegante
        sigilloDto.setOtpPwd("dsign"); -- fisso
        sigilloDto.setTypeOtpAuth("faCSI"); -- fisso
        sigilloDto.setTypeHSM("COSIGN"); -- fisso
        byte[] firmato = dosignBean.sigillo(sigilloDto);
        SignedBuffer buffer = new SignedBuffer();
        buffer.setBuffer(firmato);
        VerifyReport result = dosignBean.verifyDocument(buffer);
        System.out.println(DosignUtil.objToJson(result));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

2.3 Sbustamento

È possibile effettuare lo sbustamento di un documento imbustato richiamando metodi diversi a seconda del formato:

- **extractDocumentFromEnvelope** ([javaDoc](#))

Il metodo consente lo sbustamento di un documento firmato digitalmente.

Si applica esclusivamente ai formati PKCS #7, CAdES-BES, CAdES-T, Infocert M7M e Time Stamped Data (RFC 5544).

Riceve in input il byte array del documento da sbustare e restituisce in output il byte array del documento imbustato.

- **extractDocumentFromXmlEnvelope** ([javaDoc](#))

Esattamente come il metodo precedente ma si applica esclusivamente al formato XAdES-X.

2.4 Apposizione di Firma Digitale

La firma digitale di documenti può avvenire con due tipi di credenziali:

- certificato di firma remota con PIN e OTP di sblocco
- certificato con smartcard e PIN di sblocco

Il documento sottoscritto può assumere uno dei seguenti formati della busta crittografica (CMS, Cryptographic Message Syntax):

- PAdES-BES (PDF Advanced Electronic Signature Basic Electronic Signature)
- PAdES-T (PAdES with Time)
- CAdES-BES (CMS Advanced Electronic Signature Basic Electronic Signature)
- CAdES-T (CAdES with Time)
- Time Stamped Data secondo standard RFC 5544

2.4.1 Firma Remota

In questa sezione sono descritti i metodi necessari per l'apposizione di firme digitali per mezzo di credenziali di firma remota.

I servizi esposti permettono di orchestrare operazioni di firma digitale remota per mezzo dei seguenti server di firma digitale:

- **PkBox Enterprise Remote** per la CA Infocert
- **Aruba Remote Signing Server** per la CA Aruba

La richiesta dell'invio dell'OTP di sblocco delle credenziali di firma remota si effettua richiamando il metodo:

- **pushOtp** ([javaDoc](#)).
Il metodo deve essere utilizzato per richiedere l'invio dell'OTP per mezzo di SMS quando previsto per la credenziale in uso.
Nel caso di CA Inforcert utilizzare lo *user* e la *password* assegnate.
Nel caso di CA Aruba utilizzare gli stessi *alias* e il *pin* utilizzati per la sottoscrizione.

Le operazioni sulle firme possono essere fatte richiamando questi metodi:

- **pdfsign** ([javaDoc](#))
Metodo per sottoscrivere in formato PADES un documento PDF anche se già firmato PADES. Il metodo può essere utilizzato solo quando il signed data (data) è un contenuto PDF.
- **sign** ([javaDoc](#))
Metodo per sottoscrivere in formato CADES un documento non altrimenti sottoscritto. Il metodo può essere utilizzato solo quando si tratta di sottoscrivere un contenuto PDF, PADES o signed data per prima sottoscrizione CADES. Se fosse utilizzato per signed data CADES al termine dell'apposizione si verrebbe a creare un contenuto con buste annidate.
- **addsign** ([javaDoc](#))
Metodo per sottoscrivere in formato CADES una busta crittografica CADES. Il metodo può essere utilizzato solo quando si tratta di sottoscrivere un contenuto a sua volta firmato CADES, dalla firma di questa busta si ottiene una ulteriore firma parallela. La busta crittografica CADES da sottoscrivere (envelope) può contenere il signed data oppure questo può essere fornito a parte (data default null) se si tratta di una detached signature.

È possibile effettuare operazioni di **firma multipla** aprendo una transazione all'inizio di un ciclo di firme e chiudendola al termine attraverso i seguenti metodi:

- **startTransaction** ([javaDoc](#))
Il metodo è da utilizzare all'inizio di un ciclo di operazioni di firma multipla. Restituisce il valore iniziale del OTP authenticated data da utilizzare per il numero massimo di operazioni specificato.

- **endTransaction** ([javaDoc](#))
Il metodo chiude la transazione OTP.

A seconda del server di firma digitale coinvolto occorre seguire una procedura leggermente diversa:

- **PkBox Enterprise Remote**

All'inizio del ciclo richiamare **startTransaction** indicando, oltre a ALIAS, PIN e OTP, il massimo numero di operazioni di firma della transazione nel parametro *maxTranSize*;

startTransaction restituisce l'OTP authenticated data da utilizzare per la prima operazione di firma multipla (*authData*).

Ogni operazione di firma restituisce un OTP authenticated data da utilizzare per la sottoscrizione successiva.

Nell'ultima operazione occorre passare il parametro *lastContent* valorizzato a *true*.

Se le operazioni di firma terminano prima del valore indicato in *maxTranSize* occorre richiamare **endTransaction**(alias, authData) per invalidare l'OTP authenticated data.

Es:

```
authData = startTransaction(alias, pin, otp, maxTranSize)
authData = sign(authData, false)
...
authData = sign(authData, false)
sign(authData, true)
```

- **Aruba Remote Signing Server**

All'inizio del ciclo richiamare **startTransaction** indicando ALIAS, PIN e OTP.

startTransaction restituisce l'OTP authenticated data (*authData*) da utilizzare nelle successive operazioni di firma.

Al termine del ciclo utilizzare **endTransaction** per chiudere la transazione.

Es:

```
authData = startTransaction(alias, pin, otp)
sign(alias, pin, authData)
...
sign(alias, pin, authData)
endTransaction(alias, pin, authData)
```

2.4.2 Firma Locale

In questa sezione sono descritti i metodi necessari per l'apposizione di firme digitali per mezzo di certificati con smartcard e PIN di sblocco.

I servizi si collocano come layer intermedio fra i servizi del server di firma digitale PkBox Enterprise ed i servizi dell'applet PkNet utilizzato dalla web client application per accedere alla smartcard. Sono pubblicati da DoSign per client applications che fanno uso di un applet per implementare una soluzione di firma digitale in modalità collaborativa ovvero con calcolo del digest del documento lato DoSign ed invio al browser del digest da firmare. L'applet accede alla smartcard per la selezione del certificato e la richiesta di sottoscrizione con chiave privata.

- **digest** ([javaDoc](#))

Metodo per il calcolo dell'impronta del documento da sottoscrivere in formato CADES.

Può essere utilizzato solo quando si tratta di prima apposizione ovvero il contenuto non è a sua volta una busta crittografica CADES. Se fosse utilizzato in questi casi al termine dell'apposizione si verrebbe a creare un contenuto con buste annidate.

Sono ammissibili le seguenti direttive di calcolo:

- [algorithm](#)
l'algoritmo di digest (hash function) utilizzato per il calcolo dell'impronta del documento
 - DOSIGN_MD2 ([1](#))
 - DOSIGN_MD5 ([2](#))
 - DOSIGN_SHA1 ([3](#))
 - DOSIGN_SHA224 ([4](#))
 - DOSIGN_SHA256 ([5](#)) ← DEFAULT
 - DOSIGN_SHA384 ([6](#))
 - DOSIGN_SHA512 ([7](#))
 - DOSIGN_RIPEMD128 ([8](#))
 - DOSIGN_RIPEMD160 ([9](#))
- [encoding](#)
l'encoding format da utilizzare per il digest calcolato (default [DOSIGN_BASE64](#)).

- **merge** ([javaDoc](#))

Metodo per il merge del signed data con la busta crittografica detached in formato CADES.

Può essere utilizzato solo quando si tratta di prima apposizione ovvero il signed data non è a sua volta una busta crittografica CADES.

Vengono uniti il signed data ([data](#)) con la detached signature ([envelope](#)).

Nel caso in cui sia richiesta l'apposizione di una marca temporale associata alla firma occorre fornire il digest del signed data ([digest](#)) così come calcolato in [digest](#).

Sono ammissibili le seguenti direttive di merge:

- [encoding](#)
l'encoding format da utilizzare per la busta crittografica CADES (default [DOSIGN_DER](#)).
- [timestamped](#)
marcatura temporale associata alla firma in accordo alla RFC 3161 (default false)
- [customerTsa](#)
identificativo della TSA da selezionare per l'emissione della marca temporale come definito nel server PkBox Enterprise (default null).
- [customerInformation](#)
identificativo dell'applicazione che richiede il servizio di sign (default null). Informazione utilizzata per esigenze di accounting.

- **pdfDigest** ([javaDoc](#))

Metodo per il calcolo dell'impronta del documento da sottoscrivere in formato PADES.

Può essere utilizzato solo quando si tratta di sottoscrivere un contenuto PDF anche se già firmato

PADES. Le direttive di calcolo reason, location, contact, date devono essere le stesse in input alla pdfMerge.

Il contenuto da sottoscrivere ([data](#)) è un documento PDF.

Il metodo può essere utilizzato solo quando si tratta di sottoscrivere un contenuto PDF anche se già firmato PADES.

Sono ammissibili le seguenti direttive di calcolo:

- [reason](#)
Reason della signature.
- [location](#)
Location della signature.
- [contact](#)
ContactInfo della signature.
- [digestDate](#)
Data e ora non certificata di generazione della firma, se null è il tempo corrente.
- [algorithm](#)
L'algoritmo di digest (hash function) utilizzato per il calcolo dell'impronta del documento (default [DOSIGN_SHA256](#)).
- [encoding](#)
L'encoding format da utilizzare per il digest calcolato (default [DOSIGN_BASE64](#)).

- **pdfMerge** ([javaDoc](#))

Metodo per il merge del documento PDF con la busta crittografica detached in formato PADES.

Il metodo può essere utilizzato solo quando il signed data è un documento PDF.

Vengono uniti il documento ([data](#)) con la detached signature ([envelope](#)).

Nel caso in cui sia richiesta l'apposizione di una marca temporale associata alla firma occorre fornire il digest del documento ([digest](#)) così come calcolato in [pdfDigest](#).

Sono ammissibili le seguenti direttive di merge:

- [reason](#)
Reason della signature come in pdfDigest.
- [location](#)
Location della signature come in pdfDigest.
- [contact](#)
ContactInfo della signature come in pdfDigest.
- [digestDate](#)
Data e ora non certificata come in pdfDigest.
- [timestamped](#)
Marcatura temporale associata alla firma in accordo alla RFC 3161 (default false)
- [customerTsa](#)
Identificativo della TSA da selezionare per l'emissione della marca temporale come definito nel server PkBox Enterprise (default null).
- [customerInformation](#)
Identificativo dell'applicazione che richiede il servizio di sign (default null). Informazione utilizzata per esigenze di accounting.

- **digestDetach** ([javaDoc](#))

Metodo per il calcolo dell'impronta di una busta crittografica CADES da sottoscrivere in formato CADES.

Il metodo può essere utilizzato solo quando si tratta di sottoscrivere un contenuto a sua volta firmato CADES. Insieme al digest viene restituita la busta crittografica detached della precedente sottoscrizione, dalla firma di questa busta si ottiene una ulteriore firma parallela.

Il contenuto da sottoscrivere ([data](#)) di cui deve essere calcolato il digest è il signed data.

Sono ammissibili le seguenti direttive di calcolo:

- [algorithm](#)
L'algoritmo di digest (hash function) utilizzato per il calcolo dell'impronta del documento

- (default [DOSIGN_SHA256](#)).
- [encoding](#)
L'encoding format da utilizzare per il digest calcolato e per la busta crittografica detached (default [DOSIGN_BASE64](#)).
- **detachdataMerge** ([javaDoc](#))
Metodo per il merge del signed data CADES con la busta crittografica detached in formato CADES. Il metodo può essere utilizzato solo quando si tratta di sottoscrivere un contenuto a sua volta firmato CADES. Al contenuto sottoscritto si aggiunge una ulteriore firma parallela. Vengono uniti il signed data CADES ([data](#)) con la detached signature ([envelope](#)). Nel caso in cui sia richiesta l'apposizione di una marca temporale associata alla firma occorre fornire il digest del signed data ([digest](#)) così come calcolato in [digestDetach](#). Sono ammissibili le seguenti direttive di merge:
 - [encoding](#)
L'encoding format da utilizzare per la busta crittografica CADES (default [DOSIGN_DER](#)).
 - [timestamped](#)
Marcatura temporale associata alla firma in accordo alla RFC 3161 (default false)
 - [customerTsa](#)
Identificativo della TSA da selezionare per l'emissione della marca temporale come definito nel server PkBox Enterprise (default null).
 - [customerInformation](#)
Identificativo dell'applicazione che richiede il servizio di sign (default null). Informazione utilizzata per esigenze di accounting.

2.5 Marcatura temporale

In questa sezione descritti i servizi di marcatura temporale dei documenti firmati digitalmente secondo i profili previsti dalla normativa italiana.

Il documento marcato temporalmente può assumere uno dei seguenti formati della busta crittografica (CMS, Cryptographic Message Syntax):

- PAdES-BES (PDF Advanced Electronic Signature Basic Electronic Signature)
 - PAdES-T (PAdES with Time)
 - CAdES-BES (CMS Advanced Electronic Signature Basic Electronic Signature)
 - CAdES-T (CAdES with Time)
 - Time Stamped Data secondo standard RFC 5544
- **createTimeStampedData** ([javaDoc](#))
Metodo per la marcatura temporale di documenti firmati digitalmente nel formato RFC 5544. Consente di creare buste crittografiche nel formato Time Stamped Data secondo lo standard RFC 5544 a partire da un qualsiasi contenuto ([data](#)). Il documento restituito è verificabile per mezzo del metodo [verifyTimeStampedData](#).
 - **verifyTimeStampedData** ([javaDoc](#))
Metodo per la verifica di validità delle buste crittografiche Time Stamped Data utilizzate per la marcatura temporale di documenti firmati digitalmente.
Occorre specificare il documento marcato temporalmente ([envelopeArray](#)) nel formato RFC 5544.
Si applica al formato Time Stamped Data secondo standard RFC 5544 sia per buste singole che annidate. Nel corso della medesima sessione di verifica è possibile richiedere di reiterare la verifica su signed data a loro volta firmati con sintassi CMS come il CADES, il XADES o il PAdES.