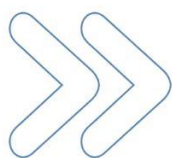


*L-Care ProxySign*

Specifiche tecniche



## Sommario

1 . Versione documento.....	3
2 . REST API.....	5
2.1 . Metodi e path.....	5
2.2 . Richiesta di Firma Remota o Automatica.....	9
2.2.1 . Risultato.....	10
2.3 . Richiesta OTP.....	11
2.3.1 . Risultato.....	11
2.4 . Recupero certificato.....	11
2.5 . Verifica di Firma e/o Marca.....	11
2.5.1 . Risultato.....	12
2.5.2 . Dettaglio della verifica.....	12
<b>2.5.2.1 . Codice e descrizione degli errori.....</b>	<b>17</b>
2.5.3 . Risultato in caso di mancato avviamento della verifica.....	19
2.5.4 . Estrazione del documento originale.....	20
2.5.5 . Verifica ed estrazione del documento originale.....	20
2.6 . Marca.....	21
2.6.1 . Risultato.....	21
2.7 . Richiesta di Firma Locale (con SmartCard o BusinessKey).....	22
2.7.1 . Descrizione del processo.....	22
<b>2.7.1.1 . Modalità 1: Web Applica .....</b>	<b>22</b>
<b>2.7.1.2 . Modalità 2: Applicazione Locale.....</b>	<b>22</b>
2.7.2 . Apertura di una transaction.....	23
2.7.3 . Recupero HTML.....	24
2.7.4 . Download file firmato.....	24
2.8 . Firma grafometrica.....	25
2.8.1 . Descrizione del processo.....	25
<b>2.8.2 . Avvio del workflow.....</b>	<b>25</b>
<b>2.8.2.1 . Servizio REST.....</b>	<b>25</b>
<b>2.8.2.2 . Spooler di stampa PostScript.....</b>	<b>26</b>
<b>2.8.3 . Stato del Workflow.....</b>	<b>26</b>
<b>2.8.4 . Risultato del Workflow (pdf/xml).....</b>	<b>27</b>
<b>2.8.5 . Annullamento del Workflow.....</b>	<b>27</b>
<b>2.8.6 . Caricamento documento per mul .....</b>	<b>27</b>
<b>2.8.7 . Stato del documento.....</b>	<b>28</b>
<b>2.8.8 . Workflow XML.....</b>	<b>28</b>
<b>2.8.9 . JSON dei valori.....</b>	<b>33</b>
2.9 . Descrizione dei Parametri per le varie tipologie di firma.....	34
2.10 . Risposta in caso di errore.....	38
2.10.1 . Codici di errore.....	38
3 . Esempi di integrazione.....	40
4 . Ambiente di collaudo per system integrator.....	43
5 . Contatti.....	44

## 2 REST API

Viene messo a disposizione un endpoint REST tramite cui è possibile sottoporre al sistema i documenti da processare con il sistema di firma automatica o remota.

### 2.1 Metodi e path

Di seguito l'elenco dei metodi REST disponibili, fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**

Descrizione	Path	Metodo	Content-Type in input	Parametri obbligatori	Content-Type in output
Firma Remota CAdES, vedi § 9	<b>/rcontext</b> /sign/cades/ <b>alias</b>	POST	multipart/form-data	pin, otp, contentToSign- <b>n</b>	application/pkcs7-mime
Firma Remota con Marca Temporale CAdES-T, vedi § 9	<b>/rcontext</b> /sign/cades-t/ <b>alias</b>	POST	multipart/form-data	pin, otp, contentToSign- <b>n</b>	application/pkcs7-mime
Firma Remota PAdES § 9	<b>/rcontext</b> /sign/pades/ <b>alias</b>	POST	multipart/form-data	pin, otp, contentToSign- <b>n</b>	application/pdf
Firma Remota con Marca Temporale PAdES-T § 9	<b>/rcontext</b> /sign/pades-t/ <b>alias</b>	POST	multipart/form-data	pin, otp, contentToSign- <b>n</b>	application/pdf
Firma Automatica CAdES § 9	<b>/acontext</b> /sign/cades/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	application/pkcs7-mime
Firma Automatica con Marca Temporale CAdES-T § 9	<b>/acontext</b> /sign/cades-t/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	application/pkcs7-mime
Firma Automatica PAdES § 9	<b>/acontext</b> /sign/pades/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	application/pdf

Firma Automatica con Marca Temporale PAdES-T § 9	<b>/acontext</b> /sign/pades-t/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	application/pdf
Firma Automatica XAdES Enveloped § 9	<b>/acontext</b> /sign/xades-enveloped/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	text/xml
Firma Automatica XAdES Enveloping § 9	<b>/acontext</b> /sign/xades-enveloping/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	text/xml
Firma Automatica XAdES Detached § 9	<b>/acontext</b> /sign/xades-detached/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	text/xml
Firma Automatica con Marca Temporale XAdES-T Enveloped § 9	<b>/acontext</b> /sign/xades-t-enveloped/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	text/xml
Firma Automatica con Marca Temporale XAdES-T Enveloping § 9	<b>/acontext</b> /sign/xades-t-enveloping/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	text/xml
Firma Automatica con Marca Temporale XAdES-T Detached § 9	<b>/acontext</b> /sign/xades-t-detached/ <b>alias</b>	POST	multipart/form-data	pin, contentToSign- <b>n</b>	text/xml
Richiesta OTP § 11	<b>/rcontext</b> /request-otp/ <b>alias</b>	GET	N/A	N/A	text/xml

Verifica di Firma e/o Marca § 11	<b>/vcontext</b>	POST	multipart/form-data	contentToVerify	text/xml
Verifica di Firma e/o Marca Detached § 11	<b>/vcontext</b>	POST	multipart/form-data	contentToVerify, originalFileToVerify	text/xml
Marca TSD § 21	<b>/tcontext/tsd</b>	POST	multipart/form-data	timestamp_username, timestamp_password, contentToTimestamp-0	application/octet-stream
Marca M7M § 21	<b>/tcontext/m/m</b>	POST	multipart/form-data	timestamp_username, timestamp_password, contentToTimestamp-0	application/octet-stream
Marca TSR § 21	<b>/tcontext/tsr</b>	POST	multipart/form-data	timestamp_username, timestamp_password, contentToTimestamp-0	application/octet-stream
Avvio Firma Locale CAdES § 22	<b>/lcontext/sign/cades</b>	POST	multipart/form-data	contentToSign- <b>n</b>	text/html
Avvio Firma Locale CAdES-T § 22	<b>/lcontext/sign/cades-t</b>	POST	multipart/form-data	contentToSign- <b>n</b>	text/html
Avvio Firma Locale PAdES § 22	<b>/lcontext/sign/pades</b>	POST	multipart/form-data	contentToSign- <b>n</b>	text/html
Avvio Firma Locale PAdES-T § 22	<b>/lcontext/sign/pades-t</b>	POST	multipart/form-data	contentToSign- <b>n</b>	text/html
Download HTML Firma Locale	<b>/lcontext/get-transaction-id/html</b>	GET	N/A	N/A	text/html
Download file firmato Firma Locale	<b>/lcontext/get-signed-file/transaction-id</b>	GET	N/A	N/A	application/octet-stream

Richiesta certificato di Firma Remota § 9	<b>/rcontext</b> /alias/ <b>alias</b>	GET	N/A	N/A	application/x-x509-user-cert
Richiesta certificato di Firma Automatica § 9	<b>/rcontext</b> /alias/ <b>alias</b>	GET	N/A	N/A	application/x-x509-user-cert
Estrazione documento originale firmato CADES/CADES-T § 17	<b>/vcontext</b> /extract	POST	multipart/form-data	contentToVerify	application/octet-stream
Verifica ed estrazione documento originale firmato § 17	<b>/vcontext</b> /verifyExtractor	POST	multipart/form-data	contentToVerify	text/xml

## 2.2 Richiesta di Firma Remota o Automatica

Di seguito un esempio di richiesta REST per eseguire una firma remota di un documento (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /rcontext/sign/format/alias HTTP/1.1
Host: hostname: port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary

--boundary
Content-Disposition: form-data; name="pin"

PIN
--boundary
Content-Disposition: form-data; name="otp"

OTP
--boundary
Content-Disposition: form-data; name="contentToSign-n"; filename="filename"
Content-Type: content_type

binary_content
--boundary--
```

Alla stregua dei campi `pin` e `otp` riportati nell'esempio di firma remota, i seguenti campi sono specificabili come parti `form-data` a seconda della modalità di firma (per il loro significato si faccia riferimento al § 2.9):

### Tipo di chiamata

Firma Remota CAdES, CAdES-T, XAdES  
Enveloped, XAdES Enveloping, XAdES  
Detached, XAdES-T Enveloped, XAdES-T  
Enveloping, XAdES-T Detached

Firma Automatica CAdES, CAdES-T, XAdES  
Enveloped, XAdES Enveloping, XAdES  
Detached, XAdES-T Enveloped, XAdES-T  
Enveloping, XAdES-T Detached

Firma Remota PAdES, PAdES-T

### Parti form-data accettate

**PIN**  
**OTP**  
**LANGUAGE**  
**timestamp\_username** (CAdES-T, XAdES-T)  
**timestamp\_password** (CAdES-T, XAdES-T)  
**xpath** (XAdES, XAdES-T)

**PIN**  
**LANGUAGE**  
**timestamp\_username** (CAdES-T, XAdES-T)  
**timestamp\_password** (CAdES-T, XAdES-T)  
**xpath** (XAdES, XAdES-T)

**PIN**  
**OTP**  
**LANGUAGE**  
**box\_signature\_page**  
**box\_signature\_llx**  
**box\_signature\_lly**  
**box\_signature\_urx**  
**box\_signature\_ury**  
**box\_signature\_lbl\_reason**  
**box\_signature\_reason**  
**box\_signature\_lbl\_date**  
**box\_signature\_format\_date**  
**box\_signature\_lbl\_signedby**  
**box\_signature\_font**  
**box\_signature\_font\_size**  
**box\_signature\_font\_style**  
**box\_signature\_image**  
**timestamp\_username** (PAdES-T)  
**timestamp\_password** (PAdES-T)

Firma Automatica PAdES, PAdES-T

**PIN**  
**LANGUAGE**  
**box\_signature\_page**  
**box\_signature\_llx**  
**box\_signature\_lly**  
**box\_signature\_urx**  
**box\_signature\_ury**  
**box\_signature\_lbl\_reason**  
**box\_signature\_reason**  
**box\_signature\_lbl\_date**  
**box\_signature\_format\_date**  
**box\_signature\_lbl\_signedby**  
**box\_signature\_font**  
**box\_signature\_font\_size**  
**box\_signature\_font\_style**  
**box\_signature\_image**  
**timestamp\_username** (PAdES-T)  
**timestamp\_password** (PAdES-T)

Firma Remota CAdES, CAdES-T

**nest**

### 2.2.1 Risultato

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

In caso di successo, lo status code restituito è 200 ed il contenuto è un documento diverso a seconda del metodo invocato e della molteplicità degli elementi sottomessi. Un resoconto delle modalità di restituzione del dato è fornito di seguito:

<b>Tipo di chiamata</b>	<b>Contenuto restituito</b>	<b>MIME Media Type</b>
CAdES o CAdES-T singolo documento	Documento firmato	application/pkcs7-mime
PAdES o PAdES-T singolo documento	Documento firmato	application/pdf
XAdES Enveloped, XAdES Enveloping, XAdES-T Enveloped o XAdES-T Enveloping singolo documento	Documento firmato	text/xml
XAdES Detached o XAdES-T Detached singolo documento	Firma XAdES detached	text/xml
CAdES o CAdES-T documenti multipli	ZIP contenente documenti firmati	application/zip contenente application/pkcs7-mime
PAdES o PAdES-T documenti multipli	ZIP contenente documenti firmati	application/zip contenente application/pdf
XAdES Enveloped, XAdES Enveloping, XAdES-T Enveloped o XAdES-T Enveloping documenti multipli	ZIP contenente documenti firmati	text/xml



XAdES Detached o XAdES-T Detached documenti multipli	ZIP contenente firme XAdES detached	application/zip contenente text/ xml
---	--	---

## 2.3 Richiesta OTP

Di seguito un esempio di richiesta REST per richiedere l'invio di un OTP (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
GET /rcontext/request-otp/alias/LANGUAGE HTTP/1.1  
Host: hostname: port
```

### 2.3.1 Risultato

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).  
In caso di successo, lo status code restituito è 200 ed il contenuto è un documento XML come quello riportato di seguito:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<response>  
  <status>OK</status>  
</response>
```

## 2.4 Recupero certificato

Sia il servizio di firma automatica che di firma remota hanno a disposizione una chiamata per il recupero del certificato associato ad un alias (passato come parametro). Di seguito due esempi di richieste REST per recuperare il certificato (fare riferimento al § 2.8.1 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
GET /rcontext/alias/alias HTTP/1.1  
Host: hostname: port
```

```
GET /accontext/alias/alias HTTP/1.1  
Host: hostname: port
```

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10). In caso di successo, lo status code restituito è 200 ed il contenuto è un file **.crt** contenente il certificato richiesto.

## 2.5 Verifica di Firma e/o Marca

Di seguito un esempio di richiesta REST per eseguire una verifica di un documento (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /vcontext HTTP/1.1  
Host: hostname: port  
Content-Length: length  
Content-Type: multipart/form-data; boundary=boundary  
  
--boundary  
Content-Disposition: form-data; name="language"
```

**LANGUAGE****-- boundary**Content-Disposition: form-data; name="contentToVerify"; filename="**filename**"Content-Type: **content\_type****-- boundary**Content-Disposition: form-data; name="originalFileToVerify"; filename="**filename**"Content-Type: **content\_type****binary\_content****-- boundary--**

## 2.5.1 Risultato

Il servizio restituisce come risultato i seguenti risultati:

**Esito**

Successo, verifica eseguita correttamente e documento valido

Verifica fallita, verifica eseguita ma con esito negativo

Problemi ad avviare la verifica

**Risultato restituito**

HTTP status 200, content-type text/xml, e contenuto XML come descritto al § 2.5.2

HTTP status 200, content-type text/xml, e contenuto XML come descritto al § 2.5.2

HTTP status 500, content-type text/xml, e contenuto XML come descritto al § 2.5.3

## 2.5.2 Dettaglio della verifica

Quando il servizio è in grado di verificare il documento fornito, sia in caso di successo che di fallimento, il servizio restituisce un documento XML conforme alla seguente Document Type Declaration (DTD):

```
<! ELEMENT deSign (signedData?, timeStamp?)+ >
<!-- ATTLIST deSign
  release CDATA #IMPLIED
  releaseDate CDATA #IMPLIED
  controlloCACRL CDATA #IMPLIED
  controlloCRLPKCS7 CDATA #IMPLIED
  controlloCRLTS CDATA #IMPLIED
  controlloValidita CDATA #IMPLIED
  controlloSigningCertificateAttr CDATA #IMPLIED
  controlloSHA1Agosto2010 CDATA #IMPLIED
  controlloSHA1Luglio2011 CDATA #IMPLIED
  controlloFirmeAnnidiate CDATA #IMPLIED
  verificaTS101903 CDATA #IMPLIED
  serverRelease CDATA #IMPLIED
  downloadCRL CDATA #IMPLIED
  useProxy CDATA #IMPLIED
  useSOCKS CDATA #IMPLIED
-->
<!-- ELEMENT signedData ((signer+, signedData?) |
  (errorCode?, errorMessage?, status))>
<!-- ATTLIST signedData
  filename CDATA #IMPLIED
  filetype CDATA #IMPLIED
  detachedDocument CDATA #IMPLIED
  content CDATA #IMPLIED
-->
```

```

    encapsulated CDATA #IMPLIED
    pkcs7 CDATA #IMPLIED
    tsr CDATA #IMPLIED
>

<! ELEMENT timeStamp (serial?, subject?, issuer?, policyInformationList?,
    certNotBefore?,
    certNotAfter?, certificate?, timeStampSerial?, timeStampDate?,
    timeStampImprintAlgorithm?, timeStampImprint?,
    signedAttributes?, digestAlgorithm?,
    (crlThisUpdate | ocspThisUpdate)?,
    (crlExpired | ocspExpired)?,
    caCertRevoked?,
    (crlRevocationDate | ocspRevocationDate)?,
    (crlHoldDate | ocspHoldDate)?,
    (crlInvalidSince | ocspInvalidSince)?,
    caCertExpired?, expiredCertsOnCRL?,
    verificationTime?, certExpired?,
    errorCode?, errorMessage?, status)
>

<! ATTLIST timeStamp
    filename CDATA #IMPLIED
    filetype CDATA #IMPLIED
    detachedDocument CDATA #IMPLIED
    cadesCompliant CDATA #IMPLIED
>

<! ELEMENT signer (serial?, subject?, issuer?, directoryAttributes?,
    policyInformationList?, qcStatements?, certNotBefore?, certNotAfter?,
    certificate?,
    signedAttributes?, signingTime?,
    signatureTimeStamp?, digestAlgorithm?,
    (crlThisUpdate | ocspThisUpdate)?,
    (crlExpired | ocspExpired)?,
    caCertRevoked?,
    (crlRevocationDate | ocspRevocationDate)?,
    (crlHoldDate | ocspHoldDate)?,
    (crlInvalidSince | ocspInvalidSince)?,
    caCertExpired?, expiredCertsOnCRL?,
    signingTimeWithinSignedAttrs?,
    verificationTime?, certExpired?,
    errorCode?, errorMessage?, status, countersigner*)
>

<! ATTLIST signer
    cadesCompliant CDATA #IMPLIED
>

<! ELEMENT countersigner (serial?, subject?, issuer?, directoryAttributes?,
    policyInformationList?, qcStatements?, certNotBefore?, certNotAfter?,
    certificate?,
    signedAttributes?, signingTime?,
    signatureTimeStamp?, digestAlgorithm?,
    (crlThisUpdate | ocspThisUpdate)?,
    (crlExpired | ocspExpired)?,
    caCertRevoked?,
    (crlRevocationDate | ocspRevocationDate)?,
    (crlHoldDate | ocspHoldDate)?,
    (crlInvalidSince | ocspInvalidSince)?,
    caCertExpired?, expiredCertsOnCRL?,
    verificationTime?, certExpired?,
    errorCode?, errorMessage?, status, countersigner*)
>

<! ATTLIST countersigner
    cadesCompliant CDATA #IMPLIED
>

<! ELEMENT signatureTimeStamp (serial?, subject?, issuer?,

```

```

    policyInformationList?, certNotBefore?, certNotAfter?, certificate?,
    timeStampSerial?, timeStampDate?, timeStampImprintAlgorithm?,
    timeStampImprint?,
    signedAttributes?, digestAlgorithm?,
    (crlThisUpdate | ocspThisUpdate)?,
    (crlExpired | ocspExpired)?,
    caCertRevoked?,
    (crlRevocationDate | ocspRevocationDate)?,
    (crlHoldDate | ocspHoldDate)?,
    (crlInvalidSince | ocspInvalidSince)?,
    caCertExpired?, expiredCertsOnCRL?,
    certExpired?,
    errorCode?, errorMessage?, status)
>

<!-- ATTLIST signatureTimeStamp
      cadesCompliant CDATA #IMPLIED -->
>

<!-- ELEMENT subject (C | O | OU | L | SUR | SER | GIVEN | DNQUALIF | TITLE |
CN | EMAIL | DOM | PSEUD | DESCR | PROV | oid)+ -->

<!-- ELEMENT issuer (C | O | OU | L | SUR | SER | GIVEN | DNQUALIF | TITLE |
CN | EMAIL | DOM | PSEUD | DESCR | PROV | oid)+ -->

<!-- ELEMENT policyInformationList (policyInformation)+ -->

<!-- ELEMENT policyInformation (policyID, policyQualifierList)+ -->

<!-- ELEMENT policyQualifierList (policyQualifier)+ -->

<!-- ELEMENT policyQualifier (policyQualifierID, cpsUri?, explicitText)+ -->

<!-- ELEMENT qcStatements (qcCompliance | qcLimitValue | qcRetensionPeriod |
qcSSCD)+ -->

<!-- ELEMENT directoryAttributes (gender | placeOfBirth | dateOfBirth |
countryOfCitizenship | countryOfResidence)+ -->

<!-- ELEMENT signedAttributes (messageDigest | signingTime |
signingCertificate | signingCertificateV2)* -->

<!-- ELEMENT SUR ( #PCDATA ) -->
<!-- ELEMENT GIVEN ( #PCDATA ) -->
<!-- ELEMENT SER ( #PCDATA ) -->
<!-- ELEMENT DNQUALIF ( #PCDATA ) -->
<!-- ELEMENT DOM ( #PCDATA ) -->
<!-- ELEMENT PSEUD ( #PCDATA ) -->
<!-- ELEMENT PROV ( #PCDATA ) -->

<!-- ELEMENT serial ( #PCDATA ) -->

<!-- ELEMENT messageDigest ( #PCDATA ) -->
<!-- empty -->

<!-- ELEMENT signingCertificate ( #PCDATA ) -->
<!-- empty -->

<!-- ELEMENT signingCertificateV2 ( #PCDATA ) -->
<!-- OID of digest algorithm -->
<!-- represented in "dot notation" -->

<!-- ELEMENT timeStampSerial ( #PCDATA ) -->

<!-- ELEMENT timeStampImprint ( #PCDATA ) -->
<!-- Base64 of DER representation of timestamp imprint -->

<!-- ELEMENT timeStampImprintAlgorithm ( #PCDATA ) -->
<!-- OID of imprint algorithm -->
<!-- represented in "dot notation" -->

```

```

<| ELEMENT digestAlgorithm ( #PCDATA) >
<!-- OID of digest algorithm -->
<!-- represented in "dot notation" -->

<| ELEMENT gender ( #PCDATA) >

<| ELEMENT placeOfBirth ( #PCDATA) >

<| ELEMENT dateOfBirth ( #PCDATA) >

<| ELEMENT countryOfCitizenship ( #PCDATA) >

<| ELEMENT countryOfResidence ( #PCDATA) >

<| ELEMENT policyID ( #PCDATA) >

<| ELEMENT policyQualifierID ( #PCDATA) >

<| ELEMENT cpsUri ( #PCDATA) >

<| ELEMENT explicitText ( #PCDATA) >

<| ELEMENT signingTimeWithinSignedAttrs EMPTY>

<| ELEMENT qcCompliance EMPTY>

<| ELEMENT qcSSCD EMPTY>

<| ELEMENT qcLimitValue ( #PCDATA) >

<| ELEMENT qcRetensionPeriod ( #PCDATA) >

<| ELEMENT certificate ( #PCDATA) >
<!-- Base64 of DER representation of signer certificate -->

<| ELEMENT certNotBefore ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT certNotAfter ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT crlThisUpdate ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT ocspThisUpdate ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT crlExpired ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT ocspExpired ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT crlRevocationDate ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT ocspRevocationDate ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT crlHoldDate ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT ocspHoldDate ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT crlInvalidSince ( #PCDATA) >
<!-- in UTC format -->

<| ELEMENT ocspInvalidSince ( #PCDATA) >

```

```

<!-- in UTC format -->

<ELEMENT signingTime ( #PCDATA ) >
<!-- in UTC format -->

<ELEMENT verificationTime ( #PCDATA ) >
<!-- in UTC format -->

<ELEMENT timeStampDate ( #PCDATA ) >
<!-- in GeneralizedTime format -->

<ELEMENT caCertRevoked ( #PCDATA ) >
<!-- in UTC format -->

<ELEMENT caCertExpired ( #PCDATA ) >
<!-- in UTC format -->

<ELEMENT certExpired ( #PCDATA ) >
<!-- in UTC format -->

<ELEMENT expiredCertsOnCRL ( #PCDATA ) >
<!-- in GeneralizedTime format -->

<ELEMENT status ( #PCDATA ) >
<!-- Could be -->
<!-- OK or -->
<!-- KO -->

<ELEMENT errorCode ( #PCDATA ) >
<!-- See error table -->

<ELEMENT errorMessage ( #PCDATA ) >
<!-- See error table -->

<ELEMENT C ( #PCDATA ) >
<!-- country -->

<ELEMENT O ( #PCDATA ) >
<!-- organization -->

<ELEMENT OU ( #PCDATA ) >
<!-- organizationUnit -->

<ELEMENT L ( #PCDATA ) >
<!-- locality -->

<ELEMENT TITLE ( #PCDATA ) >
<!-- title -->

<ELEMENT CN ( #PCDATA ) >
<!-- commonName -->

<ELEMENT DESCR ( #PCDATA ) >
<!-- descr -->

<ELEMENT EMAIL ( #PCDATA ) >
<!-- email -->

<ELEMENT oid ( #PCDATA ) >
<!-- other OIDs -->
<!-- the attribute value is the OID -->
<!-- represented in "dot notation" -->

<ATTLIST oid
    oidValue CDATA #REQUIRED
>

<ELEMENT originalFileExtracted ( #PCDATA ) >
<!-- Base64 of original file -->

```

### *2.5.2.1 Codice e descrizione degli errori*

Gli errori contenuti nel rapporto XML descritto nel DTD al paragrafo precedente negli elementi `errorCode` ed `errorMessage` sono riportati qui di seguito, riportati rispettivamente nelle colonne Codice e Descrizione:

<b>Codice</b>	<b>Descrizione</b>
00001400	Errore interno
00001401	Manca l'argomento, provare deSign -h
00001402	Non e' possibile leggere il file
00001403	Il file non contiene una busta PKCS#7
00001404	La busta PKCS#7 non contiene un SignedData
00001405	Certificato del Firmatario non trovato
00001406	Certificato del Firmatario non verificato
00001407	Certificato del Firmatario scaduto
00001408	Certificato del Firmatario revocato
00001409	Chiave non abilitata alla firma
0000140A	Certificato di CA non trovato
0000140B	Certificato di CA non valido
0000140C	Certificato di CA non verificato
0000140F	Chiave della CA non abilitata alla firma
00001410	Chiave della CA non abilitata alla firma CRL
00001411	Impossibile scaricare la CRL
00001412	Firma sulla CRL non verificata
0000141B	Certificato di Marca scaduto
0000141D	Impossibile leggere la cache delle CRL
0000141E	Impossibile leggere la lista dei Certificati
0000141F	Impossibile aggiornare la cache delle CRL
00001420	Detached verify non permessa su file m7m
00001421	Impossibile salvare la lista dei Certificati
00001422	Algoritmo di digest non implementato
00001423	Impossibile salvare la cache delle CRL
00001424	CRL scaduta:
00001425	Certificato di Marca revocato
00001426	Certificato del Firmatario non ancora valido
00001427	Certificato di Marca sospeso

00001429	Certificato del Firmatario sospeso
0000142B	Impossibile leggere la configurazione del proxy
0000142C	Configurazione del proxy: manca proxyAddress
0000142D	Configurazione del proxy: manca proxyPort
0000142E	Configurazione del proxy: manca proxyPasswd
0000142F	Marca Temporale non valida
00001430	La busta PKCS#7 non contiene una Marca Temporale
00001431	Il file non contiene un TimeStamp Token
00001432	Impossibile leggere la configurazione SOCKS
00001433	Configurazione SOCKS: manca SOCKSAddress
00001434	Configurazione SOCKS: manca SOCKSPort
00001435	Configurazione SOCKS: manca SOCKSPasswd
00001436	deSignServer non e' raggiungibile
00001440	File di input con formato non conosciuto
00001441	Certificato del Firmatario non conforme allo standard X.509
00001442	Digest in authenticatedAttrs non corretto
00001443	Chiave non abilitata alla marcatura temporale
00001444	Stato del certificato alla data di firma non disponibile
00001445	Formato dell'attributo SigningCertificate non valido
00001446	Attributo SigningCertificate non presente
00001447	L'attributo SigningCertificate non identifica il firmatario
00001448	Impossibile inizializzare la lista dei Certificati
00001449	Documento originale non trovato
0000144A	Data firma con formato non corretto
0000144B	Impossibile aggiungere alla lista dei certificati
0000144F	Il formato del Punto di Distribuzione non è riconosciuto
00001450	Firma non verificata
00001451	Marca Temporale non verificata
00001452	Digest calcolato su un documento vuoto
00001454	File non conforme allo standard XML
00001455	Il file non contiene nessuna XML Signature
00001456	Reference con URI non gestita
00001457	File non conforme allo standard PDF 1.4
00001458	Il file non contiene nessuna PDF Signature



00001459	Stato del certificato di marcatura non disponibile
0000145A	Marca Temporale scaduta
0000145B	Marca Temporale senza obbligo di conservazione
0000145C	Formato dell'attributo SigningCertificateV2 non valido
0000145D	L'attributo SigningCertificateV2 non identifica il firmatario
0000145E	Impossibile accedere allo stato OCSP
00001463	Servizio OCSP: dati scaduti
00001464	Algoritmo di digest non dichiarato
00001467	Firma non valida in quanto apposta dopo il 30 giugno 2011
00001468	Firma non valida in quanto basata sull'algoritmo SHA1
00001469	Marca Temporale non conforme alle regole tecniche DigitPA
0000146B	Almeno una delle CRL utilizzate nella verifica e' scaduta
0000146C	Busta crittografica non conforme alla normativa vigente
0000146D	Firma calcolata solo su una parte del documento
0000146E	Busta crittografica non conforme alla normativa precedente l'11 giugno 2014
0000146F	Il file non contiene un certificato X.509
00001470	SubFilter PDF non gestito
00001471	Algoritmo di Firma non implementato
00001472	Formato del DistributionPoint non gestito
00001474	Impossibile aprire il certificato X.509
00001478	Il certificato di CA non è ritenuto affidabile
00001479	ArchiveTimeStamp precedente alla data di firma certificata
0000147C	DocumentTimeStamp precedente alla data di firma certificata
00001480	Attributo ATSTHashIndex non valido
00001481	Formato dell'attributo ATSTHashIndex non valido
00001483	SigningCertificate e SigningCertificateV2 entrambi presenti
00001489	il file non contiene un Associated Signature Container
0000148A	il digest in ASiCManifest non è corretto
0000148B	il file mimetype non è presente
0000148C	file mimetype non valido
0000148D	l'Associated Signature Container non contiene nessuna firma/marca
0000148E	l'Associated Signature Container non contiene nessuna firma/marca
0000148F	l'Associated Signature Container non contiene nessuna firma/marca
00001490	SigRef in ASiCManifest non corretto

Risultato in caso di mancato avviamento della verifica

In caso di fallimento, lo status code restituito è 500 ed il contenuto è un documento XML come quello riportato di seguito:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <status>K0</status>
  <error>
    <error-code>error_code</error-code>
    <error-description>error_description</error-description>
  </error>
</response>
```

Dove i parametri espressi in **grassetto corsivo rosso** sono:

Parametro	Descrizione
error_code	Codice dell'errore restituito è ERROR-VERIFY
error_description	Descrizione dell'errore

### 2.5.3 Estrazione del documento originale

Di seguito un esempio di richiesta REST per estrarre da un documento firmato il file originale prima della firma (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /vcontext/extract HTTP/1.1
Host: hostname: port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary

--boundary
Content-Disposition: form-data; name="language"

LANGUAGE
--boundary
Content-Disposition: form-data; name="contentToVerify"; filename="filename"
Content-Type: content_type

binary_content
--boundary--
```

Il servizio restituisce come risultato i seguenti risultati:

Esito	Risultato restituito
Successo, estrazione eseguita correttamente	HTTP status 200, documento estratto
Estrazione fallita	HTTP status 500, content-type text/xml, e contenuto XML come descritto al § 2.5.3

### 2.5.4 Verifica ed estrazione del documento originale

Di seguito un esempio di richiesta REST per la verifica e l'estrazione da un documento firmato del file originale prima della firma (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /vcontext/verifyExtractor HTTP/1.1
Host: hostname: port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary

--boundary
Content-Disposition: form-data; name="language"

LANGUAGE
--boundary
Content-Disposition: form-data; name="contentToVerify"; filename="filename"
Content-Type: content_type

binary_content
--boundary--
```

Il servizio restituisce come risultato i seguenti risultati:

Esito	Risultato restituito
Successo, verifica eseguita correttamente e documento valido	HTTP status 200, content-type text/xml, content-disposition attachment; filename='fileorig. <b>ESTENSIONE_FILE_ORIGINALE</b> ', e contenuto XML come descritto al § 2.5.2
Verifica fallita, verifica eseguita ma con esito negativo	HTTP status 200, content-type text/xml, e contenuto XML come descritto al § 2.5.2
Problemi ad avviare la verifica	HTTP status 500, content-type text/xml, e contenuto XML come descritto al § 2.5.3

## 2.6 Marca

Di seguito un esempio di richiesta REST per eseguire marca di un documento (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /tcontext/tformat HTTP/1.1
Host: hostname: port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary

--boundary
Content-Disposition: form-data; name="timestamp_username"

TIMESTAMP_USERNAME
--boundary
Content-Disposition: form-data; name="timestamp_password"

TIMESTAMP_PASSWORD
--boundary
Content-Disposition: form-data; name="language"

LANGUAGE
```

```
-- boundary
Content-Disposition: form-data; name="contentToTimestamp-0"; filename="filename"
Content-Type: content_type

binary_content
-- boundary --
```

## 2.6.1 Risultato

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

In caso di successo, lo status code restituito è 200 ed il contenuto è un documento diverso a seconda della modalità invocata. Un resoconto delle modalità di restituzione del dato è fornito di seguito:

Modalità	Contenuto restituito	MIME Media Type
tsd	Busta TSD contenente documento e marca temporale	application/timestamped-data
m7m	Busta M7M contenente documento e marca temporale	application/x-m7m
tsr	Marca temporale detached	application/timestamp-reply

## 2.7 Richiesta di Firma Locale (con SmartCard o BusinessKey)

### 2.7.1 Descrizione del processo

Il processo di firma locale consente di firmare documenti tramite ProxySign avvalendosi dell'applicazione locale Dike, consentendo quindi l'utilizzo di SmartCard e Token USB collegati al PC.

Il processo di firma locale ha 2 differenti modalità

#### 2.7.1.1 Modalità 1: Web Application

La modalità 1 viene utilizzata prevalentemente negli scenari dove l'applicativo che invoca la firma è una web application ed il client accede da un PC mediante un browser. Sul PC è già installato Dike. I passaggi sono i seguenti:

Apertura di una *transaction*, mediante una chiamata POST verso ProxySign al path:  
**/lcontext/start-transaction/format**

con l'ottenimento del *transaction-id*.

Prelevamento da parte del browser sul PC del codice HTML da eseguire in un IFRAME o in una pagina nuova:  
**/lcontext/get-html/transaction-id/LANGUAGE**

Prelevamento del file firmato o dell'esito del processo, mediante una chiamata GET verso ProxySign al path:  
**/lcontext/get-signed-file/transaction-id/LANGUAGE**

Alternativamente al prelevamento del file con la chiamata GET è possibile richiedere a ProxySign che nel codice HTML ottenuto con il punto precedente sia incluso il codice Javascript per effettuare lo scaricamento del file firmato alla fine del processo (parametro *client\_download*).

### 2.7.1.2 Modalità 2: Applicazione Locale

La modalità 2 viene utilizzata prevalentemente negli scenari dove l'applicativo che invoca la firma è installato localmente sul PC del client. Sul PC del client è già installato Dike. I passaggi sono i seguenti:

Apertura di una *transaction*, mediante una chiamata POST verso ProxySign al path:  
**/lcontext/start-transaction/format**

Per questa modalità occorre abilitare il parametro *return\_html*, che consentirà la ricezione del *transaction-id* insieme al codice HTML da visualizzare, ad esempio, in un browser embedded.

Prelevamento del file firmato o dell'esito del processo, mediante una chiamata GET verso ProxySign al path:

**/lcontext/get-signed-file/transaction-id/LANGUAGE**

Alternativamente al prelevamento del file con la chiamata GET è possibile richiedere a ProxySign che nel codice HTML ottenuto con il punto precedente sia incluso lo il codice Javascript per effettuare lo scaricamento del file firmato alla fine del processo (parametro *client\_download*).

## 2.7.2 Apertura di una transaction

Di seguito un esempio di richiesta REST per eseguire l'apertura di una *transaction* (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /lcontext/start-transaction/format HTTP/1.1
Host: hostname:port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary

--boundary
Content-Disposition: form-data; name="client_download"

CLIENT_DOWNLOAD
--boundary
Content-Disposition: form-data; name="return_html"

RETURN_HTML
--boundary
Content-Disposition: form-data; name="language"

LANGUAGE
--boundary
Content-Disposition: form-data; name="contentToSign-n"; filename="filename"
Content-Type: content_type

binary_content
--boundary--
```

Nel caso di firma PADES o PADES-T è possibile specificare ulteriori parametri (per il loro significato si faccia riferimento al § 2.9), di seguito un esempio di richiesta:

```
POST /lcontext/start-transaction/format HTTP/1.1
Host: hostname:port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary
```

```
-- boundary
Content-Disposition: form-data; name="client_download"

CLIENT_DOWNLOAD
-- boundary
Content-Disposition: form-data; name="return_html"

RETURN_HTML
-- boundary
Content-Disposition: form-data; name="language"

LANGUAGE
-- boundary
Content-Disposition: form-data; name="box_signature_page"

BOX_SIGNATURE_PAGE
-- boundary
Content-Disposition: form-data; name="box_signature_llx"

BOX_SIGNATURE_L LX
-- boundary
Content-Disposition: form-data; name="box_signature_lly"

BOX_SIGNATURE_L LY
-- boundary
Content-Disposition: form-data; name="box_signature_urx"

BOX_SIGNATURE_URX
-- boundary
Content-Disposition: form-data; name="box_signature_ury"

BOX_SIGNATURE_URY
-- boundary
Content-Disposition: form-data; name="box_signature_reason"

BOX_SIGNATURE_REASON
-- boundary
Content-Disposition: form-data; name="box_signature_image"

BOX_SIGNATURE_IMAGE
-- boundary
Content-Disposition: form-data; name="contentToSign-n"; filename="filename"
Content-Type: content_type

binary_content
-- boundary --
```

In caso di successo, lo status code restituito è 200 ed il contenuto sarà una stringa rappresentante il *transaction-id* se non è stato passato il parametro *return\_html* altrimenti il contenuto restituito sarà un JSON così composto:

```
{
  "transaction-id" : "<TRANSACTION_ID>",
  "html" : "<HTML PER AVVIARE LA FIRMA LOCALE>"
}
```

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### 2.7.3 Recupero HTML

Di seguito un esempio di richiesta REST per il recupero del HTML necessario all'avvio della firma Locale (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
GET /lcontext/get-html/transaction-id/LANGUAGE HTTP/1.1  
Host: hostname: port
```

In caso di successo, lo status code restituito è 200 ed il contenuto è una stringa rappresentante il codice HTML per avviare la firma.

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

## 2.7.4 Download file firmato

Di seguito un esempio di richiesta REST per avviare il download del file firmato (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
GET /lcontext/get-signed-file/transaction-id/LANGUAGE HTTP/1.1  
Host: hostname: port
```

In caso di successo, lo status code restituito è 200 ed il contenuto è il documento firmato.

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

## 2.8 Firma grafometrica

### 2.8.1 Descrizione del processo

Il processo di firma grafometrica di ProxySign consente ad un'applicazione integrata di scatenare, su una postazione target (es. un PC o un tablet), un workflow che può comprendere la visualizzazione, la compilazione e la firma grafometrica di un documento PDF.

### 2.8.2 Avvio del workflow

ProxySign mette a disposizione due tipi di interfaccia per avviare un workflow di firma grafometrica:

- Servizio REST
- Spooler di stampa PostScript

#### 2.8.2.1 Servizio REST

Di seguito un esempio di richiesta REST per avviare il workflow su una postazione target (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /gcontext/rest/request/startworkflow HTTP/1.1  
Host: hostname: port  
Content-Length: length  
Content-Type: multipart/form-data; boundary=boundary  
  
-- boundary
```

```
Content-Disposition: form-data; name="pdf_file"
Content-Type: application/pdf
```

```
pdf_file
--boundary
```

```
Content-Disposition: form-data; name="workflow_file"
Content-Type: text/xml
```

```
workflow_file
```

```
--boundary
```

```
Content-Disposition: form-data; name="json_file"
Content-Type: application/json
```

```
json_file
```

```
--boundary
```

```
Content-Disposition: form-data; name="recipient"
```

```
recipient
```

```
--boundary
```

```
Content-Disposition: form-data; name="workflow_alias"
```

```
workflow_alias
```

```
--boundary
```

```
Content-Disposition: form-data; name="document_id"
```

```
gdocument_id
```

```
--boundary
```

```
Content-Disposition: form-data; name="expire"
```

```
expire
```

In caso di successo, lo status code restituito è 200 ed il contenuto è il transaction ID, successivamente riportato come **gtransaction\_id**. Per gestire la lavorazione occorre invocare i metodi descritti al § 2.8.3, § 2.8.4 e § 2.8.5.

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### *2.8.2.2 Spooler di stampa PostScript*

All'interno di ProxySign possono essere definiti un certo numero di workflow, descritti mediante file XML (vedi § 2.8.8). Per ogni workflow definito nel sistema, viene creato uno spooler di stampa PostScript (es. driver HP Color LaserJet 4550 PS), in grado di ricevere job di stampa mediante protocollo IPP.

Quando un job di stampa viene ricevuto, viene avviato un workflow di firma grafometrica usando il titolo del job di stampa, solitamente popolato con il nome del file stampato.

Il titolo del job di stampa deve avere il seguente formato:

```
nomeworkflow_target_documentu.id.pdf
```

dove:

<b>Parametro</b>	<b>Descrizione</b>
------------------	--------------------

<b>nomeworkflow</b>	è l'identificativo del workflow da avviare
---------------------	--

<b>target</b>	è l'identificativo della macchina (PC o tablet) su cui avviare il workflow
---------------	--



***documentuid*** è l'identificativo univoco del documento, utilizzato per localizzare il documento contenente il risultato delle operazioni

In caso di fallimento il processo di stampa restituisce un errore IPP `client-error-bad-request`.

In caso di successo ProxySign mette a disposizione una cartella condivisa CIFS da cui poter prelevare i risultati. I risultati saranno archiviati con il seguente nome file:

***documentuid***.pdf

dove:

Parametro	Descrizione
<b><i>documentuid</i></b>	è l'identificativo univoco del documento come specificato nel nome del file che avvia il processo

### 2.8.3 Stato del Workflow

Di seguito un esempio di richiesta REST per recuperare lo stato del workflow avviato come descritto nel § 2.8.2 (fare riferimento al § 2.9 per il significato dei parametri espressi in ***grassetto corsivo rosso***):

```
GET /gcontext/rest/request/status/gtransaction_id HTTP/1.1  
Host: hostname: port
```

In caso di successo, lo status code restituito è 200 ed il contenuto è il seguente documento XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<status>  
  <code>gstatus</code>  
  <description>gstatus_description</description>  
</status>
```

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### 2.8.4 Risultato del Workflow (pdf/xml)

Di seguito un esempio di richiesta REST per recuperare il risultato di un workflow avviato come descritto nel § 2.8.2 (fare riferimento al § 2.9 per il significato dei parametri espressi in ***grassetto corsivo rosso***):

```
GET /gcontext/rest/request/getfile/filetype/gtransaction_id HTTP/1.1  
Host: hostname: port
```

In caso di successo, lo status code restituito è 200 ed il contenuto è un documento XML contenente le scelte effettuate durante il Workflow nel caso di `filetype=xml`, altrimenti è il documento PDF completato e firmato.

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### 2.8.5 Annullamento del Workflow

Di seguito un esempio di richiesta REST per annullare l'esecuzione di un workflow avviato come descritto nel § 2.8.2 (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /gcontext/rest/request/stopworkflow/gtransaction_id HTTP/1.1
Host: hostname:port
```

In caso di successo, lo status code restituito è 200.

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### 2.8.6 Caricamento documento per multi-workflow

Con questa chiamata è possibile caricare un documento ed ottenere un **gdocument\_id** mediante il quale è possibile lanciare più workflow in sequenza dove ognuno viene eseguito utilizzando come documento quello risultante dal workflow lanciato precedentemente. Su uno stesso documento non possono essere lanciati contemporaneamente più di un workflow. Di seguito un esempio di richiesta REST per caricare un documento (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
POST /gcontext/rest/request/uploaddocument HTTP/1.1
Host: hostname:port
Content-Length: length
Content-Type: multipart/form-data; boundary=boundary
```

```
--boundary
Content-Disposition: form-data; name="document"
Content-Type: application/pdf
```

```
document
--boundary
Content-Disposition: form-data; name="expire"
```

```
expire
```

In caso di successo, lo status code restituito è 200 ed il contenuto è il document ID, successivamente riportato come **gdocument\_id**.

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### 2.8.7 Stato del documento

Di seguito un esempio di richiesta REST per recuperare lo stato di un documento caricato come descritto nel § 2.8.6 (fare riferimento al § 2.9 per il significato dei parametri espressi in **grassetto corsivo rosso**):

```
GET /gcontext/rest/request/documentsstatus/gdocument_id HTTP/1.1
Host: hostname:port
```

In caso di successo, lo status code restituito è 200 ed il contenuto è il seguente documento XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<status>
  <docid>gdocument_id</docid>
  <filename>gfilename</filename>
  <transactionid>gtransaction_id</transactionid>
  <workflow>workflow_file</workflow>
  <alias>workflow_alias</alias>
  <recipient>recipient</recipient>
  <status_code>gstatus</status_code>
  <status_code_description>gstatus_description</status_code_description>
  <error-code>error_code</error-code>
  <error-description>error_description</error-description>
</status>
```

In caso di errore lo status code restituito è 500 (per maggiori dettagli fare riferimento al § 2.10).

### 2.8.8 Workflow XML

Per descrivere il workflow da attivare per il processo di firma grafometrica è necessario sottomettere un documento XML.

Un modello di riferimento per XML che descrive il workflow è riportato di seguito:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!--
  Elemento:      workflows
  Descrizione:   elemento radice
-->
<workflows>

  <!--
    Elemento:      step (all'interno di un tag workflow possono esserci più step)
    Descrizione:   elemento del workflow

    Attributi:
      target = la piattaforma per cui il workflow è riservato, corrisponde al nome del
               dispositivo (es. STU-430, STU-530, etc.). La parola chiave "all" definisce
               l'applicabilità a tutti i dispositivi
  -->
  <workflow target="xxxx">

    <!--
      Elemento:      step (all'interno di un tag workflow possono esserci più step)
      Descrizione:   elemento del workflow

      Attributi:
        type = tipo di elemento, può essere "question", "signature", "pdf", "pdffinline", "acquisition"
               question = una domanda posta all'utente
               signature = una richiesta di firma posta all'utente
               pdf = la visualizzazione del PDF all'utente
               pdffinline = La visualizzazione del PDF all'interno del device
               acquisition = avvia l'acquisizione di immagine utilizzando la fotocamera del tablet

        id = identificativo dello step, può essere alfanumerico, serve
             ad identificare lo step univocamente all'interno del workflow
    -->
    <step type="xxxx" id="xxxx">

      <!--
        Elemento:      text (all'interno di un tag step possono esserci più text, uno per lingua)
        Descrizione:   testo da visualizzare verso l'utente

        Attributi:
          language = lingua da utilizzare (es. it, en, de), corrisponde al codice
                    di lingua utilizzato per invocare il servizio

          position = posizione del testo, può essere:
                    top-left, top-center, top-right, middle-left, center,
                    middle-right, bottom-left, bottom-center, bottom-right, (x,y)
      -->
```

```

    Contenuto:    testo da mostrare

-->
<text language="xx" position="xxxx">xxxx</text>

<!--
Elemento: input (1 o più occorrenze)
Descrizione: elemento interattivo da visualizzare verso l'utente
Attributi:
    type    = tipo di elemento interattivo. I valori possibili sono:
              button    = bottone
              checkbox  = casella di controllo

              width = larghezza dell'elemento in pixel [opzionale]
-->
<input type="xxxx" width="xxxx">

<!--
Elemento:    label (1 o più occorrenze con lingua diversa)
Descrizione: testo da inserire nell'elemento interattivo

Attributi:
    language = lingua da utilizzare (es. it, en, de), corrisponde al codice
              di lingua utilizzato per invocare il servizio -->
<label language="xx">xxxx</label>

<!--
Elemento:    position (1 o più occorrenze con target diverso)
Descrizione: posizione dell'elemento interattivo

Attributi:
    target = la piattaforma per cui la posizione è riservata, corrisponde al nome del
            dispositivo (es. STU-430, STU-530, etc.) [opzionale]

Contenuto:    può essere uno dei seguenti valori:
              top-left, top-center, top-right, middle-left, center, middle-right,
              bottom-left, bottom-center, bottom-right, (x,y)
-->
<position target="xxxx">xxxx</position>

<!--
Elemento:    item (1 o più occorrenze con form_id diverso)
Descrizione: elemento del form contenuto nel PDF da associare al presente elemento interattivo

Attributi:
    type      = può essere "checkbox", "radio" o "textbox"
    form_id   = identificativo dell'item che deve corrispondere
               all'identificativo AcroForm
    value     = per checkbox e radio, valore da inserire nell'AcroForm in caso di
               abilitazione dell'elemento. Se l'elemento AcroForm è checkbox o radio
               value può assumere i valori di "checked" o "unchecked"
-->
<item type="xxxx" form_id="xxxx">xxxx</item>

<!--
Elemento:    select (unica occorrenza)
Descrizione: zona da visualizzare e contornare quando l'elemento corrente ha il focus

Attributi:
    page      = numero di pagina del PDF da visualizzare
    x         = coordinata x nel sistema di riferimento del PDF, espressa in pt,
               dell'angolo in basso a sinistra del rettangolo da evidenziare
    y         = coordinata y nel sistema di riferimento del PDF, espressa in pt,
               dell'angolo in basso a sinistra del rettangolo da evidenziare
    width     = larghezza del rettangolo da evidenziare (in pt)
    height    = altezza del rettangolo da evidenziare (in pt)
    zoom      = tipo di zoom da applicare, i valori selezionabili sono:
               fit-page = fa entrare l'intera pagina nello schermo
               fit-width = fa entrare l'intera pagina in larghezza nello schermo
               fit-height = fa entrare l'intera pagina in altezza nello schermo
-->
<select page="xxxx" x="xxxx" y="xxxx" width="xxxx" height="xxxx" zoom="xxxx"/>

<!--
Elemento:    action
Descrizione: azione da eseguire alla pressione dell'elemento interattivo

Attributi:
    type      = tipo di azione selezionabile tra i seguenti valori:
               goto      = vai allo step indicato da target
               abort     = annulla il processo

```

```

        finalize = concludi il processo ed invia i dati al server
        up       = scorri verso l'alto (in visualizzazione pdf)
        down     = scorri verso il basso (in visualizzazione pdf)
        pageup   = scorri verso l'alto di una pagina (in visualizzazione pdf)
        pagedown = scorri verso il basso di una pagina (in visualizzazione pdf)
        sign     = avvia il processo di firma grafometrica su tablet

        target = identificativo dello step cui muovere la procedura (valido solo per
        action = "goto")

-->
<action type="xxxx" target="xxxx" />
</input>

<!--
Elemento: signature (unica occorrenza, solo se step type = "signature")
Descrizione: dettagli sulla firma da raccogliere

Attributi:
        type           = tipo di firma, ammesso solo "graphometric"
        form_id        = identificativo AcroForm del campo firma che deve contenere
                        il rendering grafico della firma acquisita
        page           = la pagina dove deve essere creato il campo firma [opzionale]
        x              = coordinata x nel sistema di riferimento del PDF, espressa in pt,
                        dell'angolo in basso a sinistra del campo di firma da creare
                        [opzionale]
        y              = coordinata y nel sistema di riferimento del PDF, espressa in pt,
                        dell'angolo in basso a sinistra del campo di firma da creare
                        [opzionale]
        width          = larghezza del campo di firma da creare (in pt) [opzionale]
        height         = altezza del campo di firma da creare (in pt) [opzionale]
        who            = firmatario (es. nome, cognome, titolo e affiliazione) [opzionale]
        why            = motivo della firma (es. "Per accettare") [opzionale]
        additional_data = dati aggiuntivi da includere nei dati di firma

-->
<signature type="xxxx" form_id="xxxx" who="xxxx" why="xxxx" additionaldata="xxxx" />

<!--
Elemento: start (unica occorrenza, solo se step type = "pdf")
Descrizione: indica il punto del PDF da cui parte la visualizzazione

Attributi:
        page = numero della pagina da cui far partire la visualizzazione del PDF
        offset = coordinata y espressa in pt nel sistema di riferimento della pagina PDF
                da cui far partire la visualizzazione del PDF

-->
<start page="xxxx" offset="xxxx" />

<!--
Elemento: end (unica occorrenza, solo se step type = "pdf")
Descrizione: indica il punto del PDF in cui finisce la visualizzazione

Attributi:
        page = numero della pagina in cui finisce la visualizzazione del PDF
        offset = coordinata y espressa in pt nel sistema di riferimento della pagina PDF
                in cui termina la visualizzazione del PDF

-->
<end page="xxxx" offset="xxxx" />

<!--
Elemento: zoom (unica occorrenza, solo se step type = "pdf")
Descrizione: indica il tipo di zoom da applicare alla visualizzazione del PDF

Contenuto:
        fit-page      = fa entrare l'intera pagina nello schermo
        fit-width     = fa entrare l'intera pagina in larghezza nello schermo
        fit-height    = fa entrare l'intera pagina in altezza nello schermo
        numero intero = scala applicata espressa in punti percentuali

-->
<zoom>xxxx</zoom>

<!--
Elemento: imagefilename (unica occorrenza, solo se step type = "acquisition")
Descrizione: indica il nome con cui verrà rinominato il file immagine acquisito

Contenuto:
        nomeFile = nomeFile.png

-->

```

```

<imagefilename>xxxx</imagefilename>

<!--
  Elemento:    imagedescription (unica occorrenza, solo se step type = "acquisition")
  Descrizione: descrizione dell'immagine acquisita

  Contenuto:    testo da mostrare

-->
<imagedescription>xxxx</imagedescription>

<!--
  Elemento:    imagerequired (unica occorrenza, solo se step type = "acquisition")
  Descrizione: indica se l'acquisizione dell'immagine è obbligatoria o meno

  Contenuto:
                può essere uno dei seguenti valori: true (acquisizione obbligatoria),
                false (acquisizione opzionale)

-->
< imagerequired>xxxx</imagerequired>

</step>

</workflow>

</workflows>

```

Come implementazione di base del modello riportato sopra, viene fornito un esempio di descrizione di un workflow. Il PDF collegato a questo workflow possiede una checkbox (identificativo AcroForm “conferma”) ed un campo firma (identificativo AcroForm “firma”). Il workflow mira ad eseguire le seguenti azioni, consentendo in ogni momento l’annullamento del processo:

- acquisire dall’utente finale la risposta ad un singolo quesito descritto nel PDF
- in caso di accettazione segnare la casella corrispondente
- far visualizzare il PDF compilato
- attendere che l’utente selezioni l’azione di firma
- raccogliere la firma grafometrica
- finalizzare il documento PDF compilato e firmato

Di seguito viene riportato il flusso di lavoro ipotizzato sotto forma di diagramma:



Di seguito viene riportato il flusso di lavoro descritto sotto forma di XML:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<workflows>
  <workflow>
    <step id="1" type="question">
      <text language="it" position="center">Consenti il trattamento del dato?</text>
      <input type="button">
        <label language="it">No</label>
        <position>bottom-left</position>
        <action type="abort"/>
      </input>
      <input type="button">
        <label language="it">Sì</label>
        <position>bottom-right</position>
        <item type="checkbox" form_id="conferma">checked</item>
        <action type="goto" target="2"/>
      </input>
    </step>
    <step id="2" type="pdf">
      <input type="button">
        <label language="it">Annulla</label>
        <position>bottom-left</position>
        <action type="abort"/>
      </input>
      <input type="button">
        <label language="it">Firma</label>
        <position>bottom-right</position>
        <action type="goto" target="3"/>
      </input>
    </step>
    <step id="3" type="signature">
      <text language="it" position="center">Raccogli la firma</text>
      <input type="signature">
        <position>center</position>
      </input>
      <input type="button">
        <label language="it">Annulla</label>
        <position>bottom-left</position>
        <action type="abort"/>
      </input>
      <input type="button">
        <label language="it">Accetta</label>
        <position>bottom-right</position>
        <action type="goto" target="4"/>
      </input>
    </step>
    <step id="4" type="pdf">
      <text language="it" position="center">Finalizzazione PDF firmato</text>
    </step>
  </workflow>
</workflows>

```

```

</input>
</step>
<step id="3" type="signature">
  <signature type="graphometric" form_id="firma" why="Per accettazione" />
  <input type="button">
    <label language="it">Annulla</label>
    <position>bottom-left</position>
    <action type="abort"/>
  </input>
  <input type="button">
    <label language="it">Accetta</label>
    <position>bottom-right</position>
    <action type="finalize"/>
  </input>
</step>
</workflow>
</workflows>

```

### 2.8.9 JSON dei valori

Durante la chiamata di avvio del workflow è possibile sottomettere un file JSON con i valori da assegnare agli acrofield del pdf in modo da poter precompilare il documento da processare. Il JSON è composto da una sezione *values* dove per ogni campo acrofield del pdf viene specificato il valore da assegnare e da una sezione *signatures* dove invece per ogni campo firma del pdf viene specificato il nome e cognome di chi firma e il motivo della firma. Un modello di riferimento per il JSON è riportato di seguito:

```

{
  "values": {
    "<ID CAMPO ACROFIELD 1>" : "<VALORE DA ASSEGNARE>",
    "<ID CAMPO ACROFIELD 2>" : "<VALORE DA ASSEGNARE>",
    ...
  },
  "signatures": {
    "<ID CAMPO FIRMA 1>": {
      "who": "<NOME e COGNOME DEL FIRMATARIO>",
      "why": "<MOTIVO DELLA FIRMA>"
    },
    "<ID CAMPO FIRMA 2>": {
      "who": "<NOME e COGNOME DEL FIRMATARIO>",
      "why": "<MOTIVO DELLA FIRMA>"
    },
    ...
  }
}

```

Di seguito un esempio di file JSON:

```

{
  "values": {
    "txtNomeCognome" : "Mario Rossi",
    "txtDatadiNascita" : "18/07/1970",
    "txtLuogodiNascita" : "Roma"
  },
  "signatures": {
    "sgtCampoFirma1": {
      "who": "Mario Rossi",
      "why": "Accettazione contratto"
    }
  }
}

```

## 2.9 Descrizione dei Parametri per le varie tipologie di firma

I parametri specificati nei precedenti paragrafi in **grassetto corsivo rosso** devono essere sostituiti con valori reali in runtime. Il significato ed i valori che questi assumono è descritto di seguito:



Parametro	Descrizione
rcontext	Path dell'endpoint in cui è esposto il servizio di firma proxysign per la Firma Remota (default: <b>remote</b> ), questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
accontext	Path dell'endpoint in cui è esposto il servizio di firma proxysign per la Firma Automatica (default: <b>auto</b> ), questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
vcontext	Path dell'endpoint in cui è esposto il servizio di firma proxysign per la Verifica di Firma e/o Marca (default: <b>verify</b> ), questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
tcontext	Path dell'endpoint in cui è esposto il servizio di firma proxysign per la Marca Temporale (default: <b>timestamp</b> ), questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
lcontent	Path dell'endpoint in cui è esposto il servizio di firma proxysign per la Firma Locale (default: <b>local</b> ), questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
gcontext	Path dell'endpoint in cui è esposto il servizio di firma proxysign per il servizio di firma grafometrica (default: <b>grapho</b> ), questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
format	<p>Formato di imbustamento della firma, selezionabile fra i seguenti valori (case sensitive):</p> <ul style="list-style-type: none"> <li>• cades, per restituire una busta di firma conforme allo standard CAdES</li> <li>• pades, per restituire un documento firmato secondo lo standard PAdES</li> <li>• xades-enveloped, per restituire un documento firmato secondo lo standard XAdES con firma enveloped</li> <li>• xades-enveloping, per restituire un documento firmato secondo lo standard XAdES con firma enveloping</li> <li>• xades-detached, per restituire un documento firmato secondo lo standard XAdES con firma detached</li> <li>• cades-t, per restituire una busta di firma conforme allo standard CAdES-T (firma e marca)</li> <li>• pades-t, per restituire un documento firmato secondo lo standard PAdES-T (firma e marca)</li> <li>• xades-t-enveloped, per restituire un documento firmato secondo lo standard XAdES-T enveloped (firma e marca)</li> </ul>

	<ul style="list-style-type: none"> <li>• xades-t-enveloping, per restituire un documento firmato secondo lo standard XAdES-T enveloping (firma e marca)</li> <li>• xades-t-detached, per restituire un documento firmato secondo lo standard XAdES-T detached (firma e marca)</li> </ul>
alias	Alias del profilo da attivare nel processo di firma, solitamente corrispondente al codice fiscale del firmatario
hostname	Nome dell'host dove è esposto il servizio di firma proxysign, questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
port	Porta dove il servizio di firma proxysign è in ascolto, questo dato viene comunicato congiuntamente alla presente documentazione ed è specifico dell'istanza utilizzata
length	Lunghezza totale del body del messaggio REST
boundary	Sequenza di caratteri che identifica il separatore tra i contenuti facenti parte del body codificato in multipart/form-data, solitamente una sequenza randomica di caratteri stampabili (vedi RFC822)
PIN	PIN di firma, necessario a sbloccare il processo di firma automatica o remota
OTP	OTP di firma, necessario a sbloccare il processo di firma remota (il processo di firma automatica non richiede OTP pertanto tale parametro verrà ignorato in tale evenienza)
LANGUAGE	Sigla del linguaggio in cui si vuole che vengano restituiti i messaggi di errore. Le sigla delle lingue attualmente supportate sono: it, en, de. Nel caso in cui tale parametro non venisse specificato viene utilizzato il linguaggio impostato come default sul server di ProxySign.
n	Numero di serie del contenuto da firmare. Il primo numero è 0.
content_type	Tipo di contenuto espresso come MIME Media Type (mimetype, vedi RFC2046)
binary_content	Contenuto da sottoporre al processo di firma. Possono essere specificati più contenuti di firma, ciascuno dei quali in un MIME part differente.
box_signature_page	Numero di pagina del PDF in cui si intende inserire il box di firma per le firme PAdES e PAdES-T
box_signature_llx	Ascissa (x) espressa in punti tipografici (pt, 1 pt = 0,35278 mm) dell'angolo in basso a sinistra del box di firma, con origine degli assi posta nell'angolo in basso a sinistra del foglio, per le firme PAdES e PAdES-T
box_signature_lly	Ordinata (y) espressa in punti tipografici (pt, 1 pt = 0,35278 mm) dell'angolo in basso a sinistra del box di firma, con origine degli assi posta nell'angolo in basso a sinistra del foglio, per le firme PAdES e PAdES-T
box_signature_urx	Ascissa (x) espressa in punti tipografici (pt, 1 pt = 0,35278 mm)

	dell'angolo in alto a destra del box di firma, con origine degli assi posta nell'angolo in basso a sinistra del foglio, per le firme PAdES e PAdES-T
box_signature_ury	Ordinata (y) espressa in punti tipografici (pt, 1 pt = 0,35278 mm) dell'angolo in alto a destra del box di firma, con origine degli assi posta nell'angolo in basso a sinistra del foglio, per le firme PAdES e PAdES-T
box_signature_lbl_reason	Etichetta (label) che precede la motivazione di firma (reason) che compare nel box di firma prima della motivazione stessa, per le firme PAdES e PAdES-T
box_signature_reason	Motivazione di firma (reason) che compare nel box di firma, per le firme PAdES e PAdES-T
box_signature_lbl_date	Etichetta (label) che precede la data riportata nel box di firma per le firme PAdES e PAdES-T
box_signature_format_date	Formato della data/ora riportata nel box di firma per le firme PAdES e PAdES-T. Segue le specifiche Java SimpleDateFormat.
box_signature_lbl_signedby	Etichetta (label) che precede il soggetto firmatario riportato nel box di firma per le firme PAdES e PAdES-T
box_signature_font	Tipo di carattere (font) utilizzato all'interno del box di firma per le firme PAdES e PAdES-T
box_signature_font_size	Dimensione in punti tipografici (pt) del tipo di carattere (font) utilizzato all'interno del box di firma per le firme PAdES e PAdES-T
box_signature_font_style	Stile del tipo di carattere (font) utilizzato all'interno del box di firma per le firme PAdES e PAdES-T. Lo stile specificato è un intero con i seguenti possibili valori: <ul style="list-style-type: none"> <li>• 1 = Grassetto</li> <li>• 2 = Grassetto Corsivo</li> <li>• 3 = Corsivo</li> <li>• 4 = Sottolineato</li> <li>• 5 = Standard (Default)</li> </ul>
box_signature_image	Immagine inserita nel box di firma raffigurante la firma autografa del soggetto firmatario per le firme PAdES e PAdES-T
tformat	Modalità di restituzione della marca temporale <ul style="list-style-type: none"> <li>• tsd, per restituire il documento marcato e la marca temporale in una busta TimeStampedData come descritto in RFC 5544</li> <li>• m7m, per restituire il documento marcato e la marca temporale in una busta MIME (secondo il formato M7M)</li> <li>• tsr, per restituire la marca <i>detached</i></li> </ul>
timestamp_username	Nome utente parte delle credenziali per accedere al servizio di Marca Temporale
timestamp_password	Password parte delle credenziali per accedere al servizio di Marca Temporale
xpath	L'espressione <i>xpath</i> da utilizzare per identificare il blocco XML da

	sottoporre al processo di firma XAdES a blocchi
pdf_file	Documento PDF da sottoporre al workflow di firma grafometrica
workflow_file	Descrizione del workflow di firma grafometrica da attivare, maggiori dettagli specificati al § 2.8.8
recipient	Nome della postazione target su cui attivare il workflow di firma grafometrica
gtransaction_id	L'identificativo del workflow di firma grafometrica riferito, maggiori dettagli specificati al § 2.8.2
workflow_alias	Il nome del workflow da invocare registrato all'interno della Dashboard (inserendo questo parametro viene ignorato il parametro <b>workflow_file</b> )
gdocument_id	Identificativo del documento sul quale verranno eseguite chiamate multiple di workflow, maggiori dettagli specificati al § 2.8.6
expire	Il timestamp di scadenza dei documenti processati, superato tale limite i file verranno cancellati e non sarà più possibile scaricarli
gstatus	Lo stato di un processo di workflow, I valori possibili sono: 1 (In elaborazione), 2 (Completato), 3 (Esecuzione fallita)
gstatus_description	La descrizione dello stato di un processo di workflow, I valori possibili sono: PROCESSING, COMPLETE, FAILED
document	Il documento pdf da caricare nel sistema e sul quale sarà possibile avviare una sequenza di workflow
gfilename	Il filename di un documento caricato mediante la procedura descritta a paragrafo § 2.8.6
json_file	Il file con i valori da utilizzare per valorizzare i campi acrofield del pdf nella firma grafometrica. Per maggiori dettagli fare riferimento a § 2.8.9
gfiletype	Il tipo di documento prodotto da un workflow. I valori possibili sono: xml per ottenere un file xml contenente le scelte effettuate nel workflow, pdf per ottenere il df completo e firmato.
CLIENT_DOWNLOAD	Se impostato a 1 alla fine del processo di firma locale il file firmato viene scaricato automaticamente dal client, se impostato a 0 o non passato il file non viene scaricato ma bisogna invocare la chiamata specifica per il download
RETURN_HTML	Se impostato a 1 oltre al transaction-id viene restituito HTML da lanciare sul client per invocare DIKE, se invece viene impostato a 0 o non passato la chiamata start-transaction della firma locale restituirà solo il transaction-id
nest	Se impostato a 1 la firma viene innestata all'eventuali altre firme già presenti nel documento. Se impostato a 0 o non passato la firma del documento viene inserita parallelamente a quelle eventualmente già presenti

## 2.10 Risposta in caso di errore

In caso di errore, lo status code restituito è 500 ed il contenuto è un documento XML come quello riportato di seguito:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <status>K0</status>
  <error>
    <error-code>error_code</error-code>
    <error-description>error_description</error-description>
    <!-- presente solo se error_code equivale a ERROR-SIGNATURE -->
    <error-code-signature>error_code_signature</error-code-signature>
    <proxysign-error-code>proxysign_error_code</proxysign-error-code>
    <proxysign-error-description>proxysign_error_description</proxysign-error-description>
  </error>
</response>
```

Dove i parametri espressi in **grassetto corsivo rosso** sono:

Parametro	Descrizione
error_code	Codice dell'errore. I valori possibili sono ERROR- UNKNOWN, ERROR- REQUEST e ERROR- SIGNATURE.
error_description	Descrizione dell'errore
error_code_signature	Presente solo quando error_code equivale a ERROR- SIGNATURE. Rappresenta il codice interno dell'errore di firma
proxysign_error_code	Codice dell'errore secondo la nuova numerazione.
proxysign_error_description	Descrizione dell'errore

### 2.10.1 Codici di errore

Di seguito la lista dei codici di errore:

Codice	Descrizione
PRS-0000	Errore sconosciuto
PRS-0001	Formato del pin errato, deve essere numerico e di 8 cifre
PRS-0002	L'alias fornito non ha un certificato valido
PRS-0003	L'alias fornito è bloccato per troppi tentativi di inserimento pin errati
PRS-0004	L'alias fornito non è abilitato all'invio dei codici OTP
PRS-0005	L'alias fornito ha il certificato scaduto
PRS-0006	OTP non specificato
PRS-0007	Parametri mancanti
PRS-0008	Errore di comunicazione con il server di firma
PRS-0009	Errore durante l'invio del codice OTP

PRS-0010	L'alias fornito non ha un nessun certificato in stato evaso
PRS-0011	L'alias o pin errati
PRS-0012	Tipologia di firma errata
PRS-0014	Non è stato fornito il pin
PRS-0015	Errore interno del server di firma
PRS-0016	L'OTP inserito non è corretto
PRS-0017	I parametri di accesso al server di firma non sono corretti
PRS-0018	Pin errato
PRS-0019	Alias non trovato
PRS-0020	Errore durante la ricerca dell'alias
PRS-0021	Errore durante la generazione della marca
PRS-0022	Tipologia di marca errata
PRS-0023	Errore durante il caricamento delle impostazioni
PRS-0024	Errore durante l'upload del file
PRS-0025	Errore di comunicazione con il server DIKES
PRS-0026	Sessione invalida
PRS-0027	Il content type del file inviato non è corretto
PRS-0028	Errore durante l'invio del workflow al client
PRS-0029	Il formato del workflow non è corretto
PRS-0030	Il formato del messaggio websocket è errato
PRS-0031	Sul client indicato è già in esecuzione un processo di firma
PRS-0032	Processo di firma annullato dal client
PRS-0033	Client non trovato
PRS-0034	Processo di firma grafometrica fallito
PRS-0035	La connessione websocket con il client è stata persa

### 3 Esempi di integrazione

Di seguito del codice Java di esempio per le varie chiamate al servizio ProxySign.

Dov'è presente <....> significa che va sostituito con i dati specifici per ogni utente, ad esempio:

<URL DEL SERVIZIO> <https://NOMESERVER.proxysign.it>:

Dipendeze Maven necessarie per tutti gli esempi di codice:

```
<dependency>
  <groupId>org.glassfish.jersey.core</groupId>
  <artifactId>jersey-client</artifactId>
  <version>2.17</version>
</dependency>
<dependency>
  <groupId>org.glassfish.jersey.media</groupId>
  <artifactId>jersey-media-multipart</artifactId>
  <version>2.7</version>
</dependency>
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>2.4</version>
</dependency>
```

Esempio chiamata per richiesta OTP:

<URL DEL SERVIZIO>/remote/request-otp/<ALIAS>

Esempio di chiamata per firma remota PADES:

```
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import org.apache.commons.io.FileUtils;
import org.glassfish.jersey.media.multipart.FormDataMultiPart;
import org.glassfish.jersey.media.multipart.MultiPartFeature;
import org.glassfish.jersey.media.multipart.file.FileDataBodyPart;

public class Test {

    public static void main(String[] args) {
        String alias = "<INSERIRE IL PROPRIO ALIAS>";
        String pin = "<INSERIRE IL PROPRIO PIN>";
        String otp = "<INSERIRE OTP RICEVUTO ESEGUENDO LA CHIAMATA request-otp VEDI ESEMPIO PRECEDENTE>";
        String filenameDaFirmare = "<PERCORSO COMPLETO DI NOME DEL FILE DA FIRMARE>";
        String filenameFirmato = "<PERCORSO COMPLETO DI NOME DOVE SALVARE IL FILE FIRMATO>";
        String urlServizio = "<URL DEL SERVIZIO>";
```

```

Client client = ClientBuilder.newBuilder().
    register(MultiPartFeature.class).build();

FormDataMultiPart form = new FormDataMultiPart();
form.field("pin", pin);
form.field("otp", otp);
form.bodyPart(new FileDataBodyPart("contentToSign-0", new
File(filenameDaFirmare)));

Response response = client.target(urlServizio).
    path("/remote/sign/pades/" + alias).
    request(MediaType.MULTIPART_FORM_DATA).
    post(Entity.entity(form, form.getMediaType()));

if (response.getStatus() == 200) {
    InputStream file = response.readEntity(InputStream.class);
    File targetFile = new File(filenameFirmato);
    try {
        FileUtils.copyInputStreamToFile(file, targetFile);
        System.out.print("File firmato");
    } catch (IOException e) {
        e.printStackTrace();
        System.out.print("Errore");
    }
} else {
    System.out.print("Errore: " + response.readEntity(String.class));
}
}
}

```

Esempio di chiamata per firma auto PAdES:

```

import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import javax.ws.rs.client.Client;
import javax.ws.rs.client.ClientBuilder;
import javax.ws.rs.client.Entity;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import org.apache.commons.io.FileUtils;
import org.glassfish.jersey.media.multipart.FormDataMultiPart;
import org.glassfish.jersey.media.multipart.MultiPartFeature;
import org.glassfish.jersey.media.multipart.file.FileDataBodyPart;

public class Test {

    public static void main(String[] args) {
        String alias = "<INSERIRE IL PROPRIO ALIAS>";
        String pin = "<INSERIRE IL PROPRIO PIN>";
        String filenameDaFirmare = "<PERCORSO COMPLETO DI NOME DEL FILE DA
FIRMARE>";
        String filenameFirmato = "<PERCORSO COMPLETO DI NOME DOVE SALVARE IL FILE
FIRMATO>";
        String urlServizio = "<URL DEL SERVIZIO>";

        Client client = ClientBuilder.newBuilder().
            register(MultiPartFeature.class).build();
    }
}

```



```
FormDataMultiPart form = new FormDataMultiPart();
form.field("pin", pin);
form.bodyPart(new FileDataBodyPart("contentToSign-0", new
File(filenameDaFirmare)));

Response response = client.target(urlServizio).
    path("/auto/sign/pades/" + alias).
    request(MediaType.MULTIPART_FORM_DATA).
    post(Entity.entity(form, form.getMediaType()));

if (response.getStatus() == 200) {
    InputStream file = response.readEntity(InputStream.class);
    File targetFile = new File(filenameFirmato);
    try {
        FileUtils.copyInputStreamToFile(file, targetFile);
        System.out.print("File firmato");
    } catch (IOException e) {
        e.printStackTrace();
        System.out.print("Errore");
    }
} else {
    System.out.print("Errore: " + response.readEntity(String.class));
}
}
```

## 4 Ambiente di collaudo per system integrator

Viene messo a disposizione un endpoint REST tramite cui è possibile sottoporre al sistema i documenti da processare con il sistema di firma automatica o remota e firma locale.

Endpoint: <https://test.proxysign.it/>

Questo endpoint consente alle aziende che vogliono integrare il sistema ProxySign di eseguire l'integrazione previo accordo con InfoCert per il rilascio di un certificato di firma Remota/automatica da utilizzare per eseguire le integrazioni applicative.

Abbiamo sviluppato anche una web application che in modalità grafica consente di utilizzare il certificato di firma rilasciato per firmare uno o più file e verificare file già firmati.

URL interfaccia grafica: <https://test.proxysign.it/proxysign/>

## 5 Contatti

Per qualunque chiarimento sulla presente documentazione, si prega di rivolgersi a:

Linkverse SRL  
helpdesk@linkverse.com  
Telefono: 06 90283935