

# **Appliance Paper e-Sign® :**

## **interfaccia applicativa della procedura Lambda**

Oggetto: Alpha-PeS, manuale a uso del cliente per l'invocazione del servizio Lambda  
Versione: 2025-01-16 - 18:01  
Autore: Umberto Rustichelli  
Destinatari: Clienti dei sistemi alpha-PeS che utilizzano il servizio Lambda

---

### **Indice dei contenuti**

Appliance Paper e-Sign®:.....	1
interfaccia applicativa della procedura Lambda.....	1
Introduzione.....	1
Lambda Service: invocazione (modalità POST).....	2
Sovrascrittura di direttive della configurazione con parametri POST.....	2
Nota sulle procedure aggiuntive.....	3
Nota sulla procedura Lambda con XML in ingresso.....	3
Lambda Service: invocazione (modalità REST).....	4
Codici d'errore del servizio Lambda (in inglese).....	5

### **Introduzione**

Il presente documento illustra la API per invocare il servizio Lambda dell'appliance alpha-PeS. Le caratteristiche del servizio e le procedure di configurazione dell'appliance sono oggetto di documenti separati.

## Lambda Service: invocazione (modalità POST)

Il client che invoca la procedura è un software che deve autenticarsi tramite un certificato X509. Sull'appliance, a tale certificato corrisponderà un "profilo applicativo" autorizzato all'utilizzo di specifiche "configurazioni".

La URL del servizio è

**https://<indirizzo appliance>/Profiles/Lambda/lambda\_service.php**

e richiede i parametri POST

*oper* = "sign-data"  
*cfg* = il nome della configurazione preimpostata sull'appliance per il client  
*data* = contenuto del file PDF da firmare, codificato in Base64 senza ritorni a capo  
*fname* = nome del file PDF, verrà registrato nel Lambda Store (per limitazioni riguardo ai caratteri ammessi rivolgersi a GT50 S.r.l.)

Esiste inoltre il parametro *fdate* che **NON DEVE ESSERE INVIATO** tranne che in casi particolari da concordarsi con GT50 Srl. La sintassi è

*fdate* = YYYYMMDDHHMMSS (indica data e ora da registrare nel Lambda Store)

L'appliance restituirà, in caso di successo, il PAdES, e il MIME type sarà "application/PDF".

In caso di errore, invece, il MIME type sarà "text/plain" e viene restituito un breve contenuto testuale nella forma:

<codice numerico errore>:<messaggio errore>:

## Sovrascrittura di direttive della configurazione con parametri POST

I seguenti parametri possono essere inviati dal client affinché, per la singola operazione, sovrascrivano quelli presenti nella configurazione impostata sull'appliance:

- *signature\_widget\_0\_offset\_x* (intero)
- *signature\_widget\_0\_offset\_y* (intero)

*signature\_widget\_0\_offset\_x* e *signature\_widget\_0\_offset\_y* indicano la posizione del margine in basso a sinistra dell'evidenza grafica della firma PAdES (o "widget", che contiene il timbro), rispetto al vertice in basso a sinistra della pagina, espressi in 72esimi di pollice.

Nota: nella configurazione impostata sull'appliance, la direttiva che attiva la posizione custom del widget () deve valere 1, altrimenti questi due parametri sono ignorati.

## Nota sulle procedure aggiuntive

È possibile richiedere procedure aggiuntive attraverso il parametro POST

*extraproc*

Se il parametro vale "PDFpreproc", si richiede il posizionamento dell'evidenza grafica della firma in maniera dinamica, ovvero la posizione dell'immagine nel PDF firmato sarà determinata da una scritta ("stringa di ancoraggio") che si trova nel PDF originale.

La stringa di ancoraggio viene registrata nella configurazione dall'amministratore dell'appliance e deve essere tale che non ve ne siano più di una occorrenza nel documento. Per esempio si può utilizzare "timbro.digitale.qui".

Se la stringa non viene trovata, la procedura fallisce.

Si contatti GT50 S.r.l. per ulteriori informazioni in merito.

## Nota sulla procedura Lambda con XML in ingresso

Questa opzione è nata per clienti che già utilizzano il timbro digitale 2Dplus "classico" e che intendono passare al servizio Lambda inviando però lo stesso dato XML, quindi con variazioni minime nei confronti della propria operatività.

Dal punto di vista della API, ci sono i seguenti cambiamenti:

- la procedura richiede in ingresso un file XML invece che un PDF (nel campo/parametro *data*)
- nella chiamata cambia il valore di *oper*, che deve essere "sign-x2pdf-data"

L'appliance combina il contenuto XML con un foglio di stile (precaricato e indicato nella configurazione) per generare un PDF che viene quindi sottoposto alla normale procedura Lambda, ovvero viene firmato con l'apposizione del QRcode.

Si contatti GT50 S.r.l. per ulteriori informazioni in merito.

## Lambda Service: invocazione (modalità REST)

Per usufruire del servizio in modalità REST, la URL da invocare è:

**https://<indirizzo appliance>/Admin/Profiles/Lambda/lambda\_service.php/rest1**

L'autenticazione del client avviene, anche in questo caso, tramite certificato X509.

Il contenuto del body della richiesta è un JSON; i campi di seguito elencati hanno le stesse caratteristiche (tipo di caratteri, limiti di lunghezza) delle controparti in POST:

```
{
  "oper": "sign-data",
  "cfg": "<nome configurazione>",
  "data": "<PDF originale codificato in Base64>",
  "fname": "<nome del file>"
}
```

Il body della risposta è il JSON seguente:

```
{
  "code": "<codice errore>",
  "message": "<messaggio di errore o informativo>",
  "target": "lambda", [OPZIONALE]
  "version": "<versione della procedura Lambda>", [OPZIONALE]
  "doctype": "<MIME-type del contenuto di doc>", [OPZIONALE]
  "doc": "<PDF firmato (PAdES), codificato in Base64>" [OPZIONALE]
}
```

Dove

*code*: vale 0 in caso di successo dell'operazione

*doctype*: in caso di successo dell'operazione Lambda "classica", deve valere "application/pdf"

La risposta, laddove generata a livello applicativo, ha un codice di errore HTTP 200 o comunque "di successo", si rimanda tuttavia al valore di "code" per la verifica dell'effettivo esito dell'operazione di firma.

## Codici d'errore del servizio Lambda (in inglese)

La lista può subire variazioni, si riportano qui i codici principali.

3	appliance version not suitable for lambda services
5	appliance authentication filter is off
6	incongruent input or unsupported mix/procedure, e.g. (as of 2022.08.24) authentication via OAuth2 with REST, later also to indicate, in some non-Lambda services, that parameters are not found at all, for instance if the URL is for REST but no REST input could be found or parsed. Also used in some other, obscure cases
7	(by the time of writing, code 800+7 is used by Lambda Direct only
8	(by the time of writing, 8 is used by the remote signature API only
9	failed to recover signature certificate
10	oper not known
11	oper parameter not received or empty
12	cfg parameter not received or empty
13	data parameter not received or empty
14	fname parameter not received or empty (also see error 25)
15	bad cfg parameter
16	cannot read config
17	configuration lacks signature widget dimensions; in some other cases (example: remote signature API with PAdES): cannot produce [a valid] signature widget look
18	fdate has wrong syntax
19	the configuration has no Lambda extension
20	cannot save incoming data(?!)
21	user unknown or authentication failure
22	configuration (cfg) not in your possession
23	syntax error or bad format in some [other] user-provided parameter
24	some [other] input parameter is missing or empty
25	bad fname, possibly unsuitable for storage
27	there is an incompatibility between the parent configuration and Lambda (as of 2022.07.08, when the error has been introduced, it means that the signature type is not PAdES) For other services, it may mean that the signature type is incompatible with the requested operation
31	reload attempt or transaction re-use
32	transaction re-use
41	internal, cannot set up storage(?!). Disk full?
42	internal, cannot set up storage for SQ gen, X2PDF or some other procedure
43	cannot produce hash of data from user
44	internal, cannot gen random enc key
47	internal, cannot copy file
48	cannot complete X2PDF process (when input is XML, not PDF)
50	image for PAdES widget not found (as of 2024.09, this code is used by the remote signature API for PAdES)
51	image conversion to RGB failed for SQ code
52	SQ code image file generation/conversion failed
61	store comm mode not supported
71	extra-procedure error, config extension not found or cannot read/acquire it
72	extra-procedure error, error in main body of the procedure
73	unknown (extra/ancillary) procedure requested
74	communication error (not necessarily in communication with the Lambda server)

```
99    cannot tell length of file to serve(?!)  
  
100   generic/unspecified error in Lambda procedures (mostly related  
with/corresponding to 105 or 110)  
101   lambda-store: JSON support not available  
102   lambda-store: cannot read file to send to store(?!)  
105   lambda-store: error in connection to store  
110   lambda-store: error from store server  
115   lambda-store: cannot parse store reply (JSON expected)  
117   lambda-store: store returned error  
119   lambda-store: signer info not available  
  
120   lambda-store: generic error related to the Lambda store  
121   lambda-store: failed to obtain/produce the data required for the store  
(e.g., failed to produce the signature hash for Lambda upload v3)  
122   lambda-store: upload-to-store requested method is not supported/known  
  
130   mail delivery error  
131   unexpected, internal, post-generation: failed to serve content  
  
141   REST requested but not available on this platform  
  
999   unspecified error, likely internal, not well-caught, etc.  
  
600999 same as 999
```