Firma

Firma - Introduzione

La Certification Authority permette due possibili flussi di firma.

Massiva

Questa modalità è usata tipicamente da procedure massive di 'backoffice' quando occorre firmare un elevato numero di documenti senza l'interazione del titolare. Questo flusso non richiede alcun meccanismo di strong authentication.

Nei flussi massivi possono essere usati solo certificati **AUTO** e **SEAL**.

Con Strong Auth

Questa modalità è usata nelle procedure 'online' quando il titolare chiede di firmare pochi documenti.

In questi flussi possono essere usati solo certificati **FEQ** e **ONESHOT**.

La Certification Authority mette a disposizione due meccanismi di Strong Auth:

- OTP via SMS
- Chiamata telefonica (ingoing or outgoing)

Firma massiva

E' possibile eseguire la firma massiva di n hashes solo con certificati di tipo **SEAL** e **AUTO**.

Il numero massimo di hashes che si possono firmare per singola chiamata è definito in configurazione.

> Firma massiva

I dati da fornire sono:

- **l'alias dell'owner** per cui firmare in path param
- il body seguente

```
{
  "certificate": {
    "alias": "44039",
    "pin": "12345678"
  },
  "hashes": [
    "MGUifMonFmJHfa51PKL0jPxuP3bjwSQ13s42GMxQMwk="
  ]
}
```

dove:

- **certificate.alias** è il certificato da utilizzare. Il certificato DEVE essere un certificato che appartiene all'owner indicato **E** al customer relativo al token. Non è obbligatorio.
- **certificate.pin** è il pin del certificato da utilizzare, sempre che il certificato ne preveda uno.
- **hashes** è la collection di hashes in base64 da firmare. Gli hash devono essere in base64 e ottenuti con algoritmi consentiti dalla CA

In ritorno si ottiene la lista degli hash firmati.

Firma con strong auth

E' possibile eseguire la firma con strong auth di n hashes sono con certificati di tipo **FEQ** e **ONESHOT**.
A differenza della firma massiva, la firma con strong auth prevede una serie di chiamate poichè richiede l'apertura di una sessione di firma, l'autenticazione tramite strong auth e poi la possibilità di firmare n hashes, anche con chiamate successive.

Il numero massimo di hashes che si possono firmare in una sessione è definito in configurazione.

Apertura sessione di firma

> Apertura sessione

I dati da fornire sono:

- **l'alias dell'owner** per cui firmare in path param
- il body seguente

```
{
  "certificateAlias": "123",
  "strongAuth": "type",
  "maxSignatures": n,
  "webhook": {
    "url": "url"
  }
}
```

dove:

- **certificateAlias** è il certificato da utilizzare. Il certificato DEVE essere un certificato che appartiene all'owner indicato **E** al customer relativo al token. Non è obbligatorio.
- **strongAuth** è il tipo di strong authentication prevista. I valori possibili sono:
  - SMS_OTP: strong auth tramite otp inviato per sms
  - PHONE_CALL: strong auth tramite chiamata telefonica
- **maxSignatures** è il numero di firme che si vogliono effettuare nella sessione
- **webhookUrl** è l'url, opzionale, del webhook su cui si possono ricevere i cambi stato della firma. Tale webhook verrà utilizzato in base al tipo di strong auth: es. l'SMS_OTP non lo prevede, PHONE_CALL sì.

In ritorno si ottiene la sessione appena aperta con le seguenti info:


```
{
    "id": "64a19ba4ba848a71caae31d8",
    "certificateAlias": "44040",
    "signatureTimeout": "PT10M",
    "strongAuthTimeout": "PT5M",
    "strongAuth": "SMS_OTP",
    "maxSignatures": 20,
    "completedSignatures": 0,
    "status": {
        "value": "OPENED",
        "step": null,
        "remaining": null
    },
    "createdAt": "2023-07-02T15:45:40.025663395",
    "lastModifiedAt": "2023-07-02T15:45:40.025663395"
}
```


dove:

- **id** è l'id della sessione appena aperta. Da utilizzare nelle chiamate successive
- **certificateAlias** è l'alias del certificato che si userà per la firma
- **strongAuth** indica la tipologia di strong authentication che si userà
- **maxSignatures** è il numero di firme massimo che si può firmare in questa sessione
- **completedSignatures** indica le firme completate nella sessione. All'inizio sarò sempre zero
- **createdAt** e **lastModifiedAt** sono le date di inserimento e ultimo aggiornamento
- **signatureTimeout** indica il timeout della intera sessione di firma. E' un valore configurabile
- **strongAuthTimeout** indica il timeout della fase di strong authentication. E' un valore configurabile
- **status** indica lo stato della sessione. Appena creata sarò sempre in stato **OPENED**

A questo punto si procede in base alla tipologia di strong auth prevista dalla sessione.

1. SMS_OTP

Invio otp

Occorre inviare un OTP tramite il servizio

> Invio OTP API

fornendo in path param l'id di sessione ritornato al punto precedente.

L'invio dell'sms porta la sessione in stato:

```
"status": {
   "value": "STRONG_AUTH",
   "step": "SEND_SMS",
   "remaining": 2
}
```

dove:

- **value** indica che siamo passati nello stato di strong authentication
- **step** indica che ci troviamo nello step di sms inviato
- **reamining** indica quante volte ancora potremmo chiamare questo servizio per riinviare l'otp

Validazione otp

Occorre validare l'OTP

> Validazione OTP API

fornendo in path param l'id di sessione ritornato al punto precedente e l'otp inviato nel body.

La validazione dell'otp in caso di successo porta la sessione a **READY_TO_SIGN**, altrimenti lo stato diventa:

```
"status": {
   "value": "STRONG_AUTH",
   "step": "OTP_VALIDATION",
   "remaining": 2
}
```

dove:

- **value** indica che siamo passati nello stato di strong authentication
- **step** indica che ci troviamo nello step di validazione otp
- **reamining** indica quante volte ancora potremmo chiamare questo servizio per validare l'otp

Se la validazione dell'otp non viene completata nonostante i tentativi a disposizione, lo stato della sessione diventa:

```
"status": {
   "value": "FAILED",
   "step": "OTP_VALIDATION",
   "remaining": 0
}
```

2. PHONE_CALL

Avvio chiamata

Occorre avviare la chiamata tramite il servizio

> Avvio chiamata API

fornendo in path param l'id di sessione ritornato al punto precedente.

Dopo l'avvio della chiamata, i cambi di stato della sessione verranno comunicati all'integratore tramite webhook.

L'avvio della chiamata porterà la sessione in stato:

```
"status": {
   "value": "STRONG_AUTH",
   "step": "PHONE_CALL_OPENING"
}
```

dove:

- **value** indica che siamo passati nello stato di strong authentication
- **step** indica che ci troviamo nello step di sms inviato

Da questo momento in poi la sessione subirà delle modifiche in base all'andamento dell'operazione:

Se il chiamante esegue la chiamata al numero verde riportato lo stato diventerà:

```
"status": {
   "value": "STRONG_AUTH",
   "step": "PHONE_CALL_WAITING_PIN"
```

```
}
```

per indicare che il sistema attende l'immissione del pin di firma.

A pin inserito la sessione passerà in **READY_TO_SIGN** per indicare che si è pronti per la firma.

Firma hashes

Una volta completata la strong auth, la sessione si troverà in stato **READY_TO_SIGN**, cioè pronta per la firma.

Solo a questo punto sarò possibile chiamare il servizio:

> Firma API

con i seguenti valori:

- **id sessione** in path param
- il body seguente

```
{
  "certificatePin": "12345678",
  "hashes": [
    "myMXwslBoXkTDQ0olhq1QsiHRWWL4yj1V0IuoK+PYOg="
  ]
}
```

- **certificatePin** è il pin del certificato da utilizzare, sempre che il certificato ne preveda uno.
- **hashes** è la collection di hashes in base64 da firmare. Gli hash devono essere in base64 e ottenuti con algoritmi consentiti dalla CA

In ritorno si ottiene la lista degli hash firmati.

Recupero sessione

In qualsiasi momento è sempre possibile recuperare una data sessione tramite il servizio:

> Recupero sessione API