Certification Authority

Descrizione API Certification Authority TS

- Autenticazione
- Oggetto ritornato dalla API in caso di errore
- Regular expressions
- Emissione certificato
  - Emissione - Introduzione
  - Modalità di identificazione e documenti da caricare
  - Emissione - API ex-ante/direct
- Gestione certificato
  - Lista certificati
- Firma
  - Firma - Introduzione
  - Firma massiva
  - Firma con strong auth
- Identificazione e stacco certificato
  - API

Autenticazione

Per poter utilizzare le API della Certification Authority è necessario munirsi di apposito token TSID che abilita ad un flusso client-credentials.

Tale token ha al suo interno il claim **client_client_customerId** che individua il customer per il quale il token opera.

Utilizzare questo token equivale ad eseguire le operazioni nell'ambito di un determinato cliente.

Oggetto ritornato dalla API in caso di errore

Quando un qualsiasi errore si verifica in fase di interazione con le API, verrà ritornato un oggetto del tipo

```
{
    "type": "about:blank",
    "title": "Bad Request",
    "status": 400,
    "detail": "CA validation error",
    "instance": "/api/v1/ca-registry/validate",
    "traceId": "a10f77b67e485faa68c586859340f5b9",
    "code": "GE_001",
    "errors": {
        "owner.sex": "Not a valid choice"
    }
}
```

dove:

- **traceId** è un importante dato da comunicare per il troubleshooting. **Includerlo SEMPRE nelle richieste di assistenza.**
- **code** e **detail** forniscono un maggiore livello di dettaglio rispetto all'errore riscontrato
- in caso di problemi di validazione degli input, **errors** riporta l'elenco dei campi che hanno concorso all'errore con relativa causa

Regular expressions

Tutti i dati forniti ai vari metodi della CA, vengono preventivamente controllati sulla base di regular expressions di sistema.

Tali regular expressions possono essere scaricate da qui:

> Reg exps

Emissione certificato

Emissione - Introduzione

La Certification Authority permette di rilasciare un certificato di firma remota ad un titolare in modo che lui/lei possa usarlo per firmare digitalmente dei documenti.

Clienti

Un certificato è rilasciato ad un titolare **E** all'interno di un bucket cliente. Quindi un certificato è sempre associato ad un titolare **E** al cliente.

Tipi di certificato

Possono essere rilasciati 4 differenti tipologie di certificato:

- **AUTO**: è la tipologia di certificato utilizzato per le procedure di 'backoffice' in cui si firmano massivamente un elevato numero di documenti senza l'interazione del titolare. Di solito ha un limite d'uso.
- **SEAL**: è come **AUTO** ma per 'soggetti giuridici'.
- **FEQ**: è un certificato senza limiti d'uso che può essere usato solo con procedure aggiuntive di strong auth e per firmare un numero limitato di documenti per sessione di firma.
- **ONESHOT**: è come **FEQ** ma ha dei limiti molto stringenti di durata e/o numero di firme apponibili.

Identificazione

Per rilasciare un certificato, è importante identificare il futuro titolare.

Un certificato può essere rilasciato **dopo** o **prima** dell'identificazione:

- se un certificato è rilasciato dopo, parliamo di un processo di identificazione **ex-ante**
- al contrario di un processo di identificazione **ex-post**

Nel flusso **ex-post** il certificato viene rilasciato attivo ma sospeso e la Certification Authority attende che l'identificazione produca i dovuti documenti per riattivarlo.

Con le API correnti un certificato può essere rilasciato secondo i seguenti flussi:

- **Ex-ante and direct**: significa che il titolare deve essere identificato prima di richiedere il certificato e tutti i dati necessari al rilascio devono essere forniti all'API di emissione(ex. PIN ove necessario).

Procedure di identificazione

Le API correnti gestiscono 3 tipi di identificazione:

- **SPID**: tramite SPID
- **CIE**: tramite CIE
- **FEQ**: tramite una firma digitale valida
- **VIDEO**: tramite una piattaforma VIDEO di 3e parti

Modalità di identificazione e documenti da caricare

Come detto la Certification Authority accetta 3 modalità di identificazione:

- FEQ
- SPID
- CIE
- VIDEO

SPID

nel caso di identificazione SPID, dovranno essere forniti due documenti:

- la saml request
- la saml response

prodotte dall'identificazione SPID. Ovviamente dovranno essere documenti:

1. validi
2. con firma integra
3. relativi al titolare per il quale si sta facendo la richiesta
4. non scaduti

ATTENZIONE: per definizione lo SPID richiede che la persona da identificare sia in possesso di un codice fiscale italiano. In caso contrario l'emissione del certificato verrà bloccata.

CIE

nel caso di identificazione CIE, dovranno essere forniti due documenti:

- la saml request
- la saml response

prodotte dall'identificazione CIE. Ovviamente dovranno essere documenti:

1. validi
2. con firma integra
3. relativi al titolare per il quale si sta facendo la richiesta

4. non scaduti

ATTENZIONE: per definizione lo CIE richiede che la persona da identificare sia in possesso di un codice fiscale italiano. In caso contrario l'emissione del certificato verrà bloccata.

VIDEO

nel caso di identificazione VIDEO le API non richiedono alcun documento. O meglio, si aspettano che arrivi in seconda battuta uno zip contenente tutto ciò che la piattaforma di video riconoscimento ha prodotto.

FEQ

nel caso di identificazione FEQ dovrà essere fornito il pdf del modulo di richiesta opportunamente firmato dal titolare nei campi firma previsti con un certificato valido

EVIDENZE

inoltre le API richiedono, con l'eccezione del caso in cui si parli di identificazione FEQ, le evidenze che il titolare abbia accettato o meno le clausole riportate sul modulo di richiesta, di cui deve prendere obbligatoriamente visione.
Le evidenze hanno la seguente struttura:

```
{
    "owner#clause#signature#1":  true,
    "owner#clause#signature#2": true,
    "owner#clause#signature#3": true,
    "owner#clauses#1#yes": true,
    "owner#clauses#2#yes": false
}
```

Emissione - API ex-ante/direct

ex-ante/direct vuol dire, come già detto, che occorre fornire in fase di emissione tutte le info relative all'identificazione del titolare e tutti i dati necessari per emettere il certificato.

Chiamata di apertura richiesta di certificato

Per aprire una richiesta di certificato occorre chiamare il servizio:

> Apertura richiesta ex-ante

Nella chiamata occorre passare diverse informazioni:

```
{
  "profile": "string",
  "rao": {
    "certificateAlias": "string",
    "alias": "string"
  },
  "organization": {
    "name": "string",
    "unit": "string",
    "identifier": "string",
    "country": "string",
    "city": "string",
    "district": "string",
    "cap": "string",
    "address": "string",
    "email": "string",
    "processApplication": "string",
    "responsibleDocument": "string",
    "responsiblePosition": "string"
  },
```

```
  "owner": {
    "alias": "string",
    "firstName": "string",
    "lastName": "string",
    "sex": "string",
    "birthCity": "string",
    "birthCountry": "string",
    "birthDate": "2023-07-02",
    "birthDistrict": "string",
    "city": "string",
    "country": "string",
    "district": "string",
    "cap": "string",
    "address": "string",
    "email": "string",
    "phoneNumber": "string",
    "pec": "string",
    "fiscalCode": "string",
    "document": {
      "issuer": "string",
      "country": "string",
      "number": "string",
      "releaseDate": "2023-07-02",
      "expiryDate": "2023-07-02",
      "type": "IDC"
    }
  }
}
```

- **profile** è il nome del profilo di emissione. Contiene tutte le informazioni relative al tipo di certificato, all'identificazione richiesta, alla durata, alle possibili modalità di identificazione, al fatto che richieda o meno un pin etc...
- **rao** è il Registration Authority Officer, cioè la persona che si assume la responsabilità di emettere il certificato. Normalmente è un rappresentate del cliente. E' necessario fornire il suo alias(identificativo), mentre l'alias del certificato è opzionale. Ha senso solo se il RAO ha più di un certificato associato.
- **owner** è il titolare del certificato
- **organization** è l'organizzazione (azienda) per la quale il titolare lavora/opera. Obbligatorio nel caso di emissione di sigilli.

Il servizio esegue una validazione molto stringente dei valori forniti:

nome	obbligatorio	regex	condizioni	dipendenze
owner.firstName	x	[\w\s]{1,128}		
owner.lastName	x	[\w\s]{1,128}		
owner.sex	x	[FMfm]{1}		
owner.birthCity	x	ricavato dal servizio messo a disposizione dalla CA		
owner.birthCountry	x	iso alpha 2		
owner.birthdate	x	yyyy-MM-dd	>=18 && <=120 anni	
owner.birthDistrict	x	ricavato dal servizio messo a disposizione dalla CA		
owner.city	x	ricavato dal servizio messo a disposizione dalla CA		
owner.country	x	iso alpha 2		
owner.district	x	ricavato dal servizio messo a disposizione dalla CA		
owner.cap	x	[0-9]{3,10}		
owner.address	x	[\w\s]{1,128}		
owner.email	x	si applica reg exp relativa alla mail		
owner.pec		si applica reg exp relativa alla mail		
owner.phone	x	[+0-9]{8,15}		
owner.fiscalNumber		codice fiscale valido e ricavabile dai valori precedenti		
document.issuer	x	[\w\s]{1,128}		
document.country	x	iso alpha 2		
document.number	x	si applica reg exp relative al numero di documento		

document.releaseDate	x	yyyy-MM-dd	<=oggi	
document.expiryDate	x	yyyy-MM-dd	>oggi	
document.type	x	IDC|PAS|DLC		Se fiscalNumber non fornito il type potrà essere solo PAS
organization.name	x	[\w\s]{1,128}		obbligatorio solo per certificati di tipo SEAL
organization.unit		[\w\s]{1,128}		
organization.identifier	x	[\w\s]{1,128}		obbligatorio solo per certificati di tipo SEAL
organization.country	x	iso alpha 2		obbligatorio solo per certificati di tipo SEAL
organization.city	x	ricavato dal servizio messo a disposizione dalla CA		obbligatorio solo per certificati di tipo SEAL
organization.district	x	ricavato dal servizio messo a disposizione dalla CA		obbligatorio solo per certificati di tipo SEAL
organization.cap	x	[0-9]{3,10}		obbligatorio solo per certificati di tipo SEAL
organization.address	x	[\w\s]{1,128}		obbligatorio solo per certificati di tipo SEAL
organization.email	x	si applica reg exp relativa alla mail		obbligatorio solo per certificati di tipo SEAL
organization.processApplication	x	[\w\s]{1,128}		obbligatorio solo per certificati di tipo SEAL
organization.responsibleDocument		[\w\s]{1,128}		
organization.responsiblePosition		[\w\s]{1,128}		

Chiamata di completamento richiesta di certificato

Per completare una richiesta di certificato occorre chiamare il servizio:

> Completamento richiesta ex-ante

Il servizio può essere invocato solo con un token TSID di tipo API dello stesso cliente che ha aperto la richiesta.

Nella chiamata occorre passare diverse informazioni:

```
{
    "rao": {
        "certificatePin": "12345678"
    },
    "identificationProcess":"VIDEO",
    "identificationDocs": {

    },
    "extra": {
        "pin": "12345678"
    },
    "webhook": {
        "url": "callback_url"
    }
}
```

- **rao.certificatePin** è pin del certificato del RAO utilizzato per emettere il certificato
- **identificationProcess** indica il processo di identificazione utilizzato. E' un campo opzionale poichè nel caso di profili con un singolo processo associato questa informazione è ininfluente
- **extra.pin** è pin che si vuole associare al certificato che si sta emettendo. Ammesso che il certificato necessiti di un pin.
- **webhook.url** è l'url del webhook che il servizio invocherà per comunicare l'esito dell'operazione. Infatti il servizio in questione è asincrono.
- **identificationDocs** è la sezione che contiene i documenti di identificazione del titolare. Esistono due modi per fornire i documenti e si rimanda alla sezione relativa

Il servizio, in quanto, asincrono, ritorna un **202** per segnalare la presa in carico dell'operazione.

Ad emissione completata, il servizio avrà cura di:

- inviare al titolare (e se SEAL anche alla mail di organization) una mail con le indicazioni relative al certificato emesso, con in allegato il pdf del modulo di richiesta opportunamente firmato sia dal titolare che dal RAO
- inviare un webhook all'integratore con i dati del certificato
- inviare tutti i documenti prodotti in conservazione

Reg exp numero documento, in base al type

- IDC: ^([a-zA-Z]{2}[ ]?[\d]{5}[ ]?[a-zA-Z]{2}|[a-zA-Z]{2}[ ]?[\d]{7})$
- PAS: ^[a-zA-Z]{2}[\d]{7}$
- DLC: ^[U]1[A-Za-z0-9]{8}$ **OR** ^[A-Z]{2}\d{7}[A-Z]$

Reg exp mail

Data la mail user@domain

- user: (^[-!#$%&'+/=?^_`{}|~0-9A-Z]+(\.[-!#$%&'*+/=?^_`{}|~0-9A-Z]+)\Z|^"([\001-\010\013\014\016-\037!#-\[\]-\177]|\\[\001-\011\013\014\016-\177])*"\Z)
- domain: (?:A-Z0-9?\.)+(?:[A-Z]{2,6}|[A-Z0-9-]{2,}(?<!-))\Z

In risposta il servizio ritorna:

```
{
  "requestId": "64a0895e8620872ba1be33e7",
  "document":
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag0.......2SBV65FF6ic7HWtBZlDsR5T5FuYsFKZgHdlWwD/Rc8BLt/wGjSc
ohDQplbmRzdHJlYW0KZW5kb2JqCnN0YXJ0eHJlZgoyNzg1NTMKJSVFT0YK"
}
```

dove:

- **requestId** è l'id della richiesta appena aperta
- **document** è il pdf del modulo di richiesta opportunamente fillato in base 64. Tale modulo deve essere preso in visione dal
  titolare

Modalità di caricamento dei documenti di identificazione

In fase di chiamata di completamento

La modalità più semplice è rappresentata dal passaggio dei documenti in fase di chiamata di completamento all'interno del campo **identificationDocs**:

SPID

```
"identificationDocs": {
  "documents": [
    {
      "name": "evidences.json",
```

```
      "data":
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT....."
    },
    {
      "name" : "spid.request.xml",
      "data" :
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT....."
    },
    {
      "name" : "spid.response.xml",
      "data" :
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT....."
    }
  ]
}
```


CIE


```
"identificationDocs": {
  "documents": [
    {
      "name": "evidences.json",
      "data":
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT....."
    },
    {
      "name" : "cie.request.xml",
      "data" :
```

```
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT....."
    },
    {
      "name" : "cie.response.xml",
      "data" :
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT...."
    }
  ]
}
```


VIDEO


```
"identificationDocs": {
  "documents": [
    {
      "name": "evidences.json",
      "data":
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT...."
    }
  ]
}
```


FEQ


```
"identificationDocs": {
  "documents": [
    {
      "name": "request_module.pdf",
      "data":
"JVBERi0xLjYNJeLjz9MNCjQ5NiAwIG9iag08PC9MaW5lYXJpemVkIDEvTCAyNTA1NDQvTyA0OTgvRSAyMTAwNDkvTiAzL
1QgMjUwMTE2L0ggWyA3NzEgNDQ5XT4+DWVuZG9iag0gICAgICAgICAgICAgICAgDQo1NjIgMCBvYmoNPDwvRGVjb2RlUGFybXM
```

```
8PC9Db2x1bW5zIDUvUHJlZGljdG9yIDEyPj4vRmlsdGVyL0ZsYXRlRGVjb2RlL0lEWzwxM0VEQkU4Rjg5NjQ5NjRBODYyN
TcwOUIxMEQ1REJFRD48RkIxMTM4OUE0OTI1OEI0Mzg5QUYyNDYzRUY0QUUxRT...."
    }
  ]
}
```


Con chiamata separata

Un'altra possibilità è rappresentata dal servizio di caricamento:

> Caricamento documenti

Tale metodo ritorna in output una stringa **storageFolder** che indica il bucket dove i documenti sono stati caricati.

Quindi, in fase di chiamata di completamento, sarà sufficiente andare a fornire tale info all'interno di **identificationDocs**:

```
"identificationDocs": {
  "storageFolder": "nome"
}
```

Esempio di webhook finale con le informazioni di emissione


```
{
  "success": true,
  "traceId": "bc45872a9bb9c0c554b72b0c63a08686",
  "id": "651a7b31a1dbc2080aa58b67",
  "context": "CERTIFICATE_CREATION",
  "event": "EMITTED",
  "data": {
    "certificate": {
      "alias": "70724",
      "validFrom": "2023-10-02T08:11:00",
      "validTo": "2024-10-01T14:11:00",
      "profile": "AUTO",
      "signaturesNumber": 0
    },
    "owner": {
      "alias": "TINIT-FRMTTR76M06B715E",
      "firstName": "Ettore",
```

```
      "lastName": "Fieramosca",
      "birthDate": "1976-08-06",
      "birthDistrict": "CE",
      "birthCity": "Capua",
      "birthCountry": "IT",
      "sex": "M",
      "email": "a.mariano@teamsystem.com",
      "phoneNumber": "+393452495944"
    }
  }
}
```


I webhook emessi dal servizio saranno tutti caratterizzati dal context **CERTIFICATE_CREATION.** Il valore del campo **EVENT** permetterà di distinguere la tipologia di evento.

EMITTED per esempio sarà il valore del campo evento che segnalerà il completamento dell'emissione del certificato

Gestione certificato

Lista certificati

Si può richiedere la lista dei certificati con la seguente chiamata

> Lista certificati

Richiede obbligatoriamente il query param **ownerAlias** con l'alias del titolare di cui si vogliono recuperare i certificati. I certificati ritornati, quindi, saranno quelli relativi al titolare in questione e al customer presente nel token.

Il servizio permette di filtrare i risultati, che vengono ritornati sempre paginati, con i seguenti query param:

- alias: l'alias del certificato desiderato
- status: lo stato desiderato. Valori possibili: READY,REVOKED, SUSPENDED,EXPIRED
- profiles: lista in in CVS dei profili di interesse. Valori possibili: AUTO, SEAL, ONESHOT, FEQ
- withPem: true per richiedere lo scaricamento del PEM
- page: la pagina di interesse. Default a 0
- size: la page size voluta
- sort: modalità di sorting. Si deve valorizzare con "campo,desc" o "campo,asc" a seconda che si voglia un ordine discendente o ascendente sul campo indicato. I valori accettati per il campo sono:
  - validTo
  - id
  - alias
  - enrollDate
  - profile.type
  - profile.name
  - status
  - customer.name

La lista ritornata conterrà oggetti con il seguente formato:


```
{
  "alias": "49863",
  "profile": "AUTO",
  "serialNumber": "1533566",
  "customerName": "Pallante immobiliare",
  "phoneNumber": "+393208107602",
```

```
  "email": "a.mariano@teamsystem.com",
  "organization": {
    "name": "test_59b6ba1fef8e",
    "unit": "test_a38b0d639124",
    "identifier": "test_589e7de14fba",
    "country": "IT",
    "email": "test_3984f49473a6@example.com",
    "processApplication": "test_7c8bfb32f939",
    "responsibleDocument": "test_8b1297ba3c50",
    "responsiblePosition": "CEO"
  },
  "enabled": true,
  "status": "READY",
  "validFrom": "2023-06-30T12:11:00",
  "validTo": "2024-06-29T18:11:00",
  "hasPin": true
}
```

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

Identificazione e stacco certificato

API

Oltre alle API di "base" con le quali si può richiedere un certificato e gestirne il ciclo di vita, la Certification Authority fornisce anche una API per poter effettuare contestualmente l'identificazione e il rilascio di in certificato ad un titolare.

Metodo di innesco

Per avviare la sessione di identificazione e stacco occorre utilizzare il servizio:

> Identificazione e stacco

Il servizio può essere invocato solo con un token TSID e operando quindi all'interno di un determinato cliente.

I dati da passare sono del tipo:

```
{
  "profile": "IE-SPID",
  "identificationProcesses": ["VIDEO", "SPID"]
  "rao": {
    "certificatePin": "PinEsempio11",
    "alias": "RAOCST00A00A002A"
  },
  "owner": {
    "firstName": "Gaio Giulio",
    "lastName": "Cesare",
    "email": "a.mariano@teamsystem.com",
    "phoneNumber": "123456487",
    "fiscalCode": "CSRGGL44L13H501E",
    "sex"; "M"
  },
  "webhook": {
    "url": "webhookUrl"
  }
}
```

dove:

- **profile** è il nome del profilo da utilizzare per l'identificazione e l'enroll
- **identificationProcesses** è l'elenco dei processi di identificazione che voglio mettere a disposizione. La lista deve essere costituita dai processi di identificazione previsti in CA per quel dato profilo. Se il profilo ne prevede uno solo, il parametro è opzionale.
- **rao.alias** è il RAO che autorizza l'emissione del certificato
- **rao.certificateAlias** è l'alias del certificato del RAO da utilizzare. Non è obbligatorio ed ha senso solo nel caso in cui il RAO avesse più di un certificato
- **rao.certificatePin** è il pin del certificato da utilizzare
- **owner** sono i dati di base del titolare. Sono tutti obbligatori ad eccezione del codice fiscale
- **webhook.url** è l'url cui il servizio comunicherà gli esiti dell'operazione.

La risposta sarà semplicemente un json con il valore della sessione appena avviata.

In background il sistema invierà una mail all'indirizzo fornito in input per procedere con l'identificazione.
La mail conterrà un url relativo ad una UI che guiderà il futuro titolare attraverso gli step di identificazione e rilascio del certificato.

Webhook

Il sistema invierà una serie di webhook, tendenzialmente uno ad ogni cambio di stato della sessione.

Il webhook verrà inviato sempre in POST con una struttura minima del tipo:

```
{
  "success": true,
  "traceId": "c03b592fe3a3d05e6a60f1fcd809f0b7",
  "id": "651fb390d2cdf479208d7533",
  "context": "IDENTIFICATION_ENROLL",
  "event": "SESSION_OPENED",
  "data": {
    "intermediate": true
  }
}
```

  - **id** indica la sessione di riferimento. Serve per associare il webhook alla sessione cui si riferisce
  - **traceId** serve per troubleshooting
  - **success** indica se si tratta di un evento di errore (false) o meno (true)

- **context** è il contesto da cui arriva l'evento e sarà sempre **IDENTIFICATION_ENROLL**
- **event** è l'evento lanciato
- **data** conterrà dati dell'evento
  - **intermediate** sarà a true se si tratta di un evento intermedio

Di seguito la lista completa:

```
## EVENTO INIZIALE DI APERTURA SESSIONE
{
  "success": true,
  "traceId": "bf6c7c76e895faace1bb78e1c90a1d87",
  "id": "655caea6d70afb2cca0f13fa",
  "context": "IDENTIFICATION_ENROLL",
  "event": "SESSION_OPENED",
  "data": {
    "intermediate": true
  }
}



## EVENTO DI IDENTIFICAZIONE COMPLETATA CON ERRORE
{
  "success": false,
  "traceId": "999ea355bf8d1ab275e4808b56b3d0c5",
  "id": "651fb390d2cdf479208d7533",
  "context": "IDENTIFICATION_ENROLL",
  "event": "IDENTIFICATION",
  "data": {
    "intermediate": false,
    "code": "IDN_008",
    "detail": "External provider identification failed"
  }
}

## EVENTO DI IDENTIFICAZIONE COMPLETATA CON SUCCESSO
{
  "success": true,
  "traceId": "86f72deb36ab999dd31806598cc05a16",
  "id": "651fbc33d2cdf479208d7538",
  "context": "IDENTIFICATION_ENROLL",
```

```
  "event": "IDENTIFICATION",
  "data": {
    "intermediate": true
  }
}

## EVENTO DI COMPLETAMENTO RACCOLTA DATI DALL'UTENTE
{
  "success": true,
  "traceId": "4be434fa8b3599d21d8259e4af7c6e63",
  "id": "651fbc33d2cdf479208d7538",
  "context": "IDENTIFICATION_ENROLL",
  "event": "USER_SUBMITTED_DATA",
  "data": {
    "intermediate": true
  }
}

## EVENTO DI AVVIO RICHIESTA CERTIFICATO
{
  "success": true,
  "traceId": "3f9da6511cc1c56284907c409e74a2a0",
  "id": "651fbc33d2cdf479208d7538",
  "context": "IDENTIFICATION_ENROLL",
  "event": "CERT_OPENED",
  "data": {
    "intermediate": true
  }
}

## EVENTO DI COMPLETAMENTO CON SUCCESSO EMISSIONE CERTIFICATO
{
  "success": true,
  "traceId": "3f9da6511cc1c56284907c409e74a2a0",
  "id": "651fbc33d2cdf479208d7538",
  "context": "IDENTIFICATION_ENROLL",
  "event": "CERT_EMITTED",
  "data": {
    "intermediate": false,
    "certificate": {
```

```
      "alias": "72822",
      "validFrom": "2023-10-06T07:53:00",
      "validTo": "2028-10-05T07:53:00",
      "signaturesNumber": 0
    },
    "owner": {
      "alias": "TINIT-CSRGGL44L13H501E",
      "firstName": "Gaio Giulio",
      "lastName": "Cesare",
      "birthDate": "1944-07-13",
      "birthDistrict": "RM",
      "birthCity": "Roma",
      "birthCountry": "IT",
      "sex": "M",
      "email": "a.mariano@teamsystem.com",
      "phoneNumber": "+393452495944"
    }
  }
}

## EVENTO DI COMPLETAMENTO CON ERRORE EMISSIONE CERTIFICATO
{
  "success": false,
  "traceId": "3d7425144d2e69e359f10755d8d7e997",
  "id": "651fca40d2cdf479208d7539",
  "context": "IDENTIFICATION_ENROLL",
  "event": "CERT_EMITTED",
  "data": {
    "intermediate": false,
    "code": "CTH_008",
    "detail": "Rao certificate pin invalid"
  }
}
```


Cancellazione sessione

E' possibile cancellare una sessione di identificazione e stacco chiamando il seguente metodo:

> Cancellazione sessione

Ri-invio mail

E' possibile richiedere l'invio della mail di innesco della sessione chiamando il seguente metodo:

> Ri-invio mail