

Descrizione Api

- [ChangeLog](#)
- [Signer Read](#)
- [Signer Write](#)
- [Signature Read](#)
- [Signature Write](#)
- [Signature Config](#)
- [Digital Sign Api \(DEPRECATO\)](#)

ChangeLog

Versioni dei servizi attuali in PRODUZIONE

- signature-write ([Info versione corrente](#))
 - signature-read([Info versione corrente](#))
 - signer-write([Info versione corrente](#))
 - signer-read([Info versione corrente](#))
 - signature-config([Info versione corrente](#))
 - digital-sign-api ([Info versione corrente](#))
 - signature-authorizer-read([Info versione corrente](#))
 - signature-authorizer-write([Info versione corrente](#))
-

Versioni dei servizi attuali in TEST

- signature-write ([Info versione corrente](#))
 - signature-read([Info versione corrente](#))
 - signer-write([Info versione corrente](#))
 - signer-read([Info versione corrente](#))
 - signature-config([Info versione corrente](#))
 - digital-sign-api ([Info versione corrente](#))
 - signature-authorizer-read([Info versione corrente](#))
 - signature-authorizer-write([Info versione corrente](#))
-

Elenco rilascio nuove funzionalità in PRODUZIONE

- **nextRelease**
- **14/07/2021**
 - signature-write **v2.0.26**
 - signer-write **v2.0.17**
 - signature-authorizer-write **v1.2.2**
 - signer-read **v1.1.12**
 - signature-authorizer-read **v1.1.1**
 - signature-read **v2.0.15**

- **14/05/2021**

- signer-write v2.0.14

- **03/05/2021**

- signature-write v2.0.20
 - signature-read v2.0.13
 - signer-write v2.0.12
 - signer-read v1.1.11
 - signature-config v1.0.0
 - digital-sign-api v1.2.1
 - signature-authorizer-read v1.0.4
 - signature-authorizer-write v1.1.3
-
-

ChangeLog

14/07/2021 Rilasciato in produzione

- signature-write v2.0.26: fix per non inviare la mail in caso di signatureType **fes**
 - signature-authorizer-write v1.2.1/v1.2.2 :
 - signature-authorizer-read v1.1.0 :
 - signer-read v1.1.12 :
 - signature-read v2.0.15
-

19/06/2021

- signature-write v2.0.25: fix per invio del cambio stato di un documento tramite webhook
-

17/06/2021

- signer-write v2.0.17 : fix metadati per modulo FEA
-

16/06/2021

- signature-authorizer-write v1.2.0 : aggiunti permessi avanzati per gestione tipologie di documenti e firmatari
 - signature-authorizer-read v1.1.0 : aggiunti permessi avanzati per gestione tipologie di documenti e firmatari
-

15/06/2021

- signer-write `v2.0.16` : fix metadati per modulo FEA
 - signature-write `v2.0.24` : fix per invio del cambio stato di un documento tramite webhook
-

19/05/2021

- signature-write `v2.0.21/v2.0.22/v2.0.23` : aggiunto controllo sulla parte di upload async per evitare che vengano caricati pdf protetti da password, aggiunto campo `webHookUrl` nella request dell'upload async, aggiunta la possibilità durante l'upload async di creare sessioni di FES grafometriche con utilizzo di tavolette wacom
-

18/05/2021

- signer-write `v2.0.15` : aggiunta la possibilità di creare la FEA cartacea con in allegato le immagine dei documenti del firmatario
-

14/05/2021 Rilasciato in produzione

- signer-write `v2.0.13/v2.0.14` : fix aggiornamento certificati FEQ
-

03/05/2021 Rilasciato in produzione

30/04/2021

- digital-sign-api `v1.2.0/v1.2.1` : modificate le api di lettura e scrittura per i dati aggiuntivi FEA dov'è stato eliminato il foroCompetente, eliminate api non più utilizzabili
 - signature-authorizer-read `v1.0.4` : aggiunto /info
 - signature-authorizer-write `v1.1.3` : fix body request, aggiunta /info
 - signature-config `v1.0.0` : aggiunti log per i pacchetti disattivati
 - signature-write `v2.0.20` : disattivati i controlli sulla data di scadenza(**expiryDate**) sul validatore per processo async, disabilitato la parte di managerId per le connessioni, rimessa chiamata al vecchio servizio per gli adempimenti
 - signature-read `v2.0.13` : disattivato il controllo sull'obbligatorietà del campo **documentIntermediaryId** per la ricerca dei documenti
-

28/04/2021

- signature-write `v2.0.19` : riattivati i controlli sulla data di scadenza(**expiryDate**) sul validatore per processo async
-

21/04/2021

- signer-write `v2.0.12` : fix creazione FEA da gestore per una gestita, aggiunti log per consumi metering
-

15/04/2021

- signature-config `v0.0.9` : fix per la ricerca delle configurazione e del sigillo con gli uffici
-

14/04/2021

- signature-write `v2.0.18` : aggiunto controlli per l'upload async con sigillo personale
 - signature-config `v0.0.8` : eliminato foro competente nei dati aggiuntivi
 - signer-write `v2.0.11` : fix conteggio certificati, aggiornata creazione FEA senza foro competente
 - signer-read `v1.1.11` : eliminato foro competente per la rigenerazione del modulo FEA cartaceo
 - signature-read `v2.0.12` : aggiornata libreria per i permessi
-

13/04/2021

- signature-read `v2.0.11` : aggiunto filtro per managerId sui documenti eliminati
 - signature-config `v0.0.7` : fix lettura informazioni sigillo
-

31/03/2021

- signer-read `v1.1.10` : fix download modulo FEA cartaceo
-

25/03/2021

- signer-write `v2.0.8` : aggiornato modulo FEA
-

24/03/2021

- signature-config `v0.0.6` : fix api per l'aggiunta dei dati aggiuntivi
 - signature-write `v2.0.16` : aggiunto il campo `strictExpiryDate` per il processo async, fix processo async in caso di FDS
-

15/03/2021

- signature-config `v0.0.3` : fix aggiornamento pacchetto
- signer-write `v2.0.7` : aggiunta una chiamata per la lettura della config prima di allocare i consumi per i certificati

12/03/2021

- signature-config `v0.0.2` : aggiunte api per la scrittura e lettura degli extra data
- signature-write `v2.0.15` : cambiato endpoint per la lettura della configurazione del pacchetto durante l'upload di un documento

11/03/2021

- signer-write `v2.0.6` : fix consumi certificati di FEQ automatica e remota

09/03/2021

- signer-write `v2.0.5` : cambiato contratto per FEQ automatica/remota con l'aggiunta di numero di telefono e email

02/03/2021

- signer-write `v2.0.4` : fix create fea with paperSignature

25/02/2021

- signer-write `v2.0.2` / `v2.0.3` : fix su api v2 per mantenere la retro compatibilità:
 - create signer con il campo `fiscalCountry` non obbligatorio
 - create FEA con `use0tp`
 - assign e unAssign request

24/02/2021

- signer-write `v2.0.1` : fix su api v1 per mantenere la retro compatibilità

23/02/2021

- signer-write `v2.0.0` : introduzione api v2, modificate api per essere utilizzate anche per il servizio adempimenti
 - signer-read `v1.1.9` : modificate api per essere utilizzate anche per il servizio adempimenti
 - signature-write `v2.0.14` : modificate api per essere utilizzate anche per il servizio adempimenti
 - signature-read `v2.0.10` : reso obbligatorio il campo `documentIntermediaryId` nella ricerca
-
-

Signer Read

API di lettura per *Ts Digital Signature* per le operazioni riguardanti i firmatari e i certificati. [Swagger](#)

Header

Gli header richiesti dalla chiamata sono gli [header standard di TSDigital](#).

Il `Content-Type` deve essere `application/json`

In tutte le chiamate dove è richiesto `ServiceName` non impostare nulla a meno che si vuole utilizzare le api per il servizio adempimenti.

Descrizione API

SIGNER

Signer search

POST [/v1/rest/signers/search](#)

Permette di recuperare i firmatari di una data azienda

Query Params

- `page` numero di pagina
- `itemId` identificativo dell'azienda
- `size` numero di elementi per pagina
- `fullText` permette di fare una ricerca filtrata per alcuni dati dei firmatari
- `certs` permette di filtrare per tipologia di firma, i valori accettati sono **FRM**, **FSM**, **FEA**, **FES** (firma elettronica automatica, firma elettronica remota, firma elettronica avanzata, firma elettronica semplice)
- `strictCerts` permette di filtrare solamente le tipologie di firma attive
- `serviceName` **controllare le info su questo campo in cima al documento**

La ricerca `fullText` viene fatta su questi dati dei firmatari

- `firstName`
- `lastName`
- `fiscalCode`
- `ownerTaxId` codice fiscale dell'azienda su cui il firmatario è stato assegnato
- `ownerVatNumber` partita iva dell'azienda su cui il firmatario è stato assegnato
- `ownerDescription` ragione sociale dell'azienda su cui il firmatario è stato assegnato

Response

```
{
  "page":0,
  "itemsNumberPerPage":2,
  "totalItems":2,
  "totalPages":1,
  "signers":[
    {
      "id":219,
      "fiscalCode":"MNNRFL96D21I829W",
      "firstName":"Raffaele",
      "lastName":"Mainini",
      "sex":"M",
      "email":"rmainini@mondora.com",
      "cellNumber":"",
      "ownerDescription":null,
      "enabled":true,
      "certificates":[
        {
          "id":225,
          "type":"FSM",
          "status":"SUBMISSION_SUCCESS"
        },
        {
          "id":226,
          "type":"FRM",
          "certificateSn":"585",
          "certificateId":"10846",
          "enabled":true,
          "status":"FINAL_SUCCESS",
          "createBy":"rmainini@mondora.com"
        }
      ]
    }
  ]
}
```

```
        "id":60,
        "type":"FES",
        "enabled":true,
        "status":"ACTIVE"
    },
    {
        "id":81,
        "type":"FEA",
        "enabled":false,
        "documentHubId":"5e9847700e20480006394a98",
        "status":"INIT"
    }
],
"_links":{
    "self":{
        "href":"https://ts-signer-read-api-test.agyo.io/api/v1/rest/signers/219"
    }
}
},
{
    "id":247,
    "fiscalCode":"PNNGNN80A01I829P",
    "firstName":"Giovanni",
    "lastName":"Panni",
    "sex":"M",
    "email":"rmainini@mondora.com",
    "cellNumber":"",
    "ownerDescription":null,
    "enabled":true,
    "certificates":[

    ],
    "_links":{
        "self":{
            "href":"https://ts-signer-read-api-test.agyo.io/api/v1/rest/signers/247"
        }
    }
}
}
]
```

Retrive Signer

[GET] [/v1/rest/signers/{id}](#)

Permette di recuperare tutte le info riguardanti uno specifico firmatario

Info

- `id` identificativo del firmatario

Response

```
{
  "id":219,
  "fiscalCode":"MNNRFL96D21I829W",
  "firstName":"Raffaele",
  "lastName":"Mainini",
  "fiscalCountry":"IT",
  "sex":"M",
  "birthCity":"Sondrio",
  "birthCountry":"IT",
  "birthDate":"1996-04-21T00:00:00.000+0000",
  "birthDistrict":"S0",
  "city":"Sondrio",
  "country":"IT",
  "district":null,
  "cap":"23010",
  "address":"Via Fasulla",
  "email":"rmainini@mondora.com",
  "cellNumber":"",
  "ownerTaxId":null,
  "ownerVatNumber":null,
  "ownerDescription":null,
  "documentIssuer":"me",
  "documentNumber":"1234",
  "documentEndDate":"2020-04-30T00:00:00.000+0000",
  "documentType":"Carta d'identità",
  "_links":{
    "self":{
      "href":"https://ts-signer-read-api-test.agyo.io/api/v1/rest/signers/219"
```

```
}  
  
}  
  
}
```

Retrive If Signer Exist into Item

[GET] [/v1/rest/signers/searchByFiscalCode](#)

Permette di verificare se un determinato firmatario esiste già per un'azienda

Query Params

- `itemId` identificativo dell'azienda
- `fiscalCode` codice fiscale del firmatario che si sta controllando
- `serviceName` **controllare le info su questo campo in cima al documento**

UTILS

Download Filled/Signed Document

[GET] [/v1/rest/utls/download/document](#)

Permette di scaricare il contratto per i certificati FRM/FSM (Firma elettronica qualificata automatica/Firma elettronica qualificata remota) sia solamente pre compilato sia quello pre compilato e firmato

Query Params

- `signerId` identificativo del firmatario
- `certType` tipologia di certificato **FRM** o **FSM**
- `documentType` tipo documento **FILLED** o **SIGNED**

Generate and return base64 FEA precompiled module

[GET] [/v1/rest/utls/generate/fea/module](#)

Permette di generare il contratto del modulo di adesione FEA precompilato

Query Params

- `signerId` identificativo del firmatario
-

CERTIFICATE

Retrive Slot Usage

[GET] </v1/rest/certificates/slotUsage/{itemId}>

Permette di recuperare il conteggio degli slot per i certificati di un'azienda

Query Params

- `itemId` identificativo dell'azienda

Response

```
{
  "totalFRM":3,
  "freeFRM":2,
  "totalFSM":3,
  "freeFSM":3
}
```

Signer Write

API di scrittura per *Ts Digital Signature* per le operazioni riguardanti i firmatari e i certificati.

[Swagger](#)

Header

Gli header richiesti dalla chiamata sono gli [header standard di TSDigital](#).

Il `Content-Type` deve essere `application/json`

In tutte le chiamate dove è richiesto ServiceName non impostare nulla a meno che si vuole utilizzare le api per il servizio adempimenti.

Descrizione API

V2 - SIGNER

Create a new signer

[\[POST\] /v2/rest/signer](#)

Permette la creazione di un firmatario per una determinata azienda

*1 campo obbligatorio

*2 campo obbligatorio in caso si voglia richiedere un certificato di FEQ automatica o di FEQ remota

Info

- `firstName` *1, nome.
- `lastName` *1, cognome.
- `fiscalCode` *1, in caso si tratti di un firmatario italiano (dispone di codice fiscale, solamente questi firmatari possono richiedere un certificato di FEQ automatica o di FEQ remota). In caso il firmatario non abbia un codice fiscale il campo non deve essere compilato perchè verrà generato un UUID come id del firmatario. In caso di firmatario italiano viene fatto un controllo sulla lunghezza di 16 caratteri.
- `itemId` *1, codice fiscale dell'azienda.

- `fiscalCountry` domicilio fiscale, se valorizzato viene fatto un controllo che il valore sia [ISO 3166-1 alpha-2](#). Il valore di default da inserire è "IT", in caso il firmatario è sprovvisto di codice fiscale il valore non può essere "IT".
- `sex` *1, viene fatto un controllo che il valore sia "M" o "F".
- `birthCity` *1 città di nascita.
- `birthDate` *1, data di nascita.
- `birthCountry` se valorizzato viene fatto un controllo che il valore sia [ISO 3166-1 alpha-2](#).
- `birthDistrict` provincia di nascita, in caso di nascita all'estero il valore da passare è "EE". In caso di provincia italiana si deve usare la sigla da 2 cifre.
- `city` città di residenza.
- `country` nazione di residenza, se valorizzato viene fatto un controllo che il valore sia [ISO 3166-1 alpha-2](#).
- `district` -> **campo non usato, verrà dismesso.**
- `cap` cap di residenza, se valorizzato viene fatto un controllo che il valore sia di 5 caratteri e che contenga solo numeri.
- `address` indirizzo di residenza.
- `email` *1 indirizzo email.
- `prefixCellNumber` prefisso numero di telefono, se non valorizzato il valore di default sarà "+39", per richiedere una FEQ automatica o di FEQ remota il prefisso non può essere diverso da "+39". //TODO
- `cellNumber` *1, numero di telefono personale, solo numeri.
- `documentIssuer` *2, ente che ha rilasciato il documento.
- `documentNumber` *2, numero del documento.
- `documentEndDate` *2, data di scadenza del documento.
- `documentType` *2, può assumere questi determinati valori:
 - **"Carta d'identità"**
 - **"Patente di guida"**
 - **"Passaporto"**
 - **"Porto d'armi"**
 - **"Tessera di riconoscimento postale"**
 - **"Tessera rilasciata da PA"**
- `ownerTaxId` codice fiscale dell'azienda su cui il firmatario è stato assegnato .
- `ownerVatNumber` partita iva dell'azienda su cui il firmatario è stato assegnato .
- `ownerDescription` ragione sociale dell'azienda su cui il firmatario è stato assegnato .
- `serviceName` **controllare le info su questo campo in cima al documento.**

```
{
  "firstName": "string", *
  "lastName": "string", *
  "fiscalCode": "string", *
  "itemId": "string", *
  "fiscalCountry": "string", *
```

```
"sex": "string", *
"birthCity": "string", *
"birthCountry": "string",*
"birthDate": "2021-02-12T14:53:00.676Z", *
"birthDistrict": "string",
"city": "string",
"country": "string",
"district": "string",
"cap": "string",
"address": "string",
"email": "string", *
"documentIssuer": "string",
"documentNumber": "string",
"documentEndDate": "2021-02-12T14:53:00.676Z",
"documentType": "string",
"cellNumber": "string", *
"ownerTaxId": "string",
"ownerVatNumber": "string",
"ownerDescription": "string",
"serviceName": "A-FISCALI-IVA-FLOW" *
}
```

[Esempi FES](#)

[Esempi FEA](#)

[Esempi FEQ](#)

Response

```
{
  "signerId": 0,
  "itemId": "string",
  "fiscalCode": "string"
}
```

Disable an existent signer and disable all certificates

[\[DELETE\] /v2/rest/signer/{signerId}](#)

Permette di eliminare un determinato firmatario. Oltre a cancellare il firmatario verranno eliminate le extraSignature(FES/FEA) e i certificati collegati al firmatario(FRM/FSM). La cancellazione è solamente logica, verrà impostato `enabled=false`. In caso si provi a ricreare il firmatario con lo stesso codice fiscale verrà chiamata in automatico l'API **Update an existent signer**, il firmatario verrà riattivato e in caso aggiornate le properties

Info

- `signerId` identificativo del firmatario
- `serviceName` **controllare le info su questo campo in cima al documento**

Update an existent signer

[\[PATCH\]/v2/rest/signer/{signerId}](#)

Permette di aggiornare i dati di un firmatario

*1 campo obbligatorio

Info

- `itemId` *1 identificativo dell'azienda. Non può essere cambiato quindi deve avere lo stesso valore di quando il firmatario è stato creato
- `city` città di residenza.
- `country` nazione di residenza, se valorizzato viene fatto un controllo che il valore sia [ISO 3166-1 alpha-2](#).
- `district` **-> campo non usato, verrà dismesso.**
- `cap` cap di residenza, se valorizzato viene fatto un controllo che il valore sia di 5 caratteri e che contenga solo numeri.
- `address` indirizzo di residenza.
- `email` indirizzo email.
- `prefixCellNumber` prefisso numero di telefono, se non valorizzato il valore di default sarà "+39", per richiedere una FEQ automatica o di FEQ remota il prefisso non può essere diverso da "+39". //TODO
- `cellNumber` numero di telefono personale, solo numeri.
- `documentIssuer` ente che ha rilasciato il documento
- `documentNumber` numero del documento
- `documentEndDate` data di scadenza del documento
- `documentType` può assumere questi determinati valori:
 - **"Carta d'identità"**
 - **"Patente di guida"**
 - **"Passaporto"**
 - **"Porto d'armi"**
 - **"Tessera di riconoscimento postale"**

- **"Tessera rilasciata da PA"**
- `ownerTaxId` codice fiscale dell'azienda su cui il firmatario è stato assegnato
- `ownerVatNumber` partita iva dell'azienda su cui il firmatario è stato assegnato
- `ownerDescription` ragione sociale dell'azienda su cui il firmatario è stato assegnato
- `serviceName` **controllare le info su questo campo in cima al documento**

```
{
  "itemId": "string",
  "city": "string",
  "district": "string",
  "cap": "string",
  "address": "string",
  "email": "string",
  "documentIssuer": "string",
  "documentNumber": "string",
  "documentEndDate": "2021-02-12T15:30:23.713Z",
  "documentType": "string",
  "cellNumber": "string",
  "ownerTaxId": "string",
  "ownerVatNumber": "string",
  "ownerDescription": "string"
}
```

V2 - ENROLLMENT

Start enrollment for a signer

[\[POST\] /v2/rest/enrollment/start](#)

Permette di inizializzare una sessione di enrollment per creare un certificato.

*1 campo obbligatorio

Info

- `signerId` *1 identificativo del firmatario.
- `profile` *1 può assumere questi determinati valori:
 - **"teamsystemTriennale"**
 - **"teamsystemAuto"**
 - **"teamsystemOneshot"**

- `identificationProcesses` array che definisce la modalità di riconoscimento, può avere dentro questi valori:
 - **"FEQ"**
 - **"SPID"**
- `owner` *1 contiene le informazioni del firmatario.
- `owner.firstname` *1.
- `owner.lastName` *1.
- `owner.email` *1.
- `owner.phoneNumber` *1.
- `owner.fiscalCode`
- `webhook` *1 informazioni sul webhook da richiamare in fase di enrollment
- `webhook.url` *1 url da richiamare deve essere l'URL della `signer-write/v2/rest/enrollment/callback` a cui va aggiunto come parametro di query:
 - **"signerId"** *1 identificativo del firmatario

```
[
  {
    "signerId": "long",
    "profile": "string",
    "identificationProcesses": "string[]",
    "owner": {
      "firstName": "string",
      "lastName": "string",
      "email": "string",
      "phoneNumber": "string",
      "fiscalCode": "string"
    },
    "webhook": {
      "url": "string"
    }
  }
]
```

Response

```
[
  {
    "sessionId": "string"
  }
]
```

Delete existing enrollment session

[\[DELETE\]/v2/rest/enrollment/session/{sessionId}](#)

Permette di cancellare la sessione di enrollment

*1 campo obbligatorio

Info

- `sessionId` identificativo della sessione di enrollment

Resend enrollment email

[\[PUT\]/v2/rest/enrollment/session/{sessionId}/email/send](#)

Permette di reinviare la mail di inizio enrollment

*1 campo obbligatorio

Info

- `sessionId` identificativo della sessione di enrollment

V2 - EXTRA SIGNATURE

Activate FES

[\[POST\] /v2/rest/extraSignature/fes/{signerId}](#)

Permette di attivare la tipologia la FES (firma elettronica semplice) ad un determinato firmatario.

Info

- `signerId` identificativo del firmatario

Initialize FEA

Permette di effettuare l'adesione FEA (firma elettronica avanzata) per un determinato firmatario. Il processo di attivazione sarà completato solamente una volta firmato il documento che verrà in automatico generato e mandato in firma

Info

- `signerId` identificativo del firmatario
- `transmitterId` identificativo dell'azienda
- `imagesContent` base64 delle immagini o pdf del documento del firmatario e del codice fiscale.

Da inserire se non sono ancora stati conservati le copie di questi documenti.

I file dei documenti possono essere `.pdf.`, `.jpg`, `.jpeg`, `.jpe`, `.png`, `.tiff`, `.tif`

Ricordarsi di inserire il mime type del formato, esempio: ``

- `documentSignType` metodo con cui verrà firmato il documento. I valori possibili possono essere **FEA remota** o **FEA grafometrica**
- `mainSignerId` codice fiscale del firmatario possessore di un certificato di Firma elettronica qualificata automatica (FRM) che sigillerà il documento
- `use0tp` valore da mettere a **true** solamente se `documentSignType` è **FEA remota**. Questo permetterà di utilizzare come strumento di autenticazione del modulo di richiesta per la FEA tramite otp e non SecureCall
- `paperSignature` permette di avere un adesione FEA cartacea, significa che viene scaricato sul cliente l'obbligo di conservare una copia cartacea dell'adesione FEA del firmatario. Se impostato a **true** permette di ricevere come response della chiamata il base64 del modulo di richiesta per la FEA e verrà attivata automaticamente senza firmare il modulo. se si vuole usare `onlyUpload=true` il valore deve essere impostato a **false**
- `onlyUpload` se impostato a **true** permette di firmare l'adesione FEA insieme al caricamento del primo documento.

NOTA BENE, il primo documento che si caricherà per questo firmatario dovrà avere lo stesso `mainSignerId` e la stessa tipologia di firma e lo stesso strumento di autenticazione per essere valida. Inoltre fino a quando il firmatario non firmerà il primo documento non potrà sottoscriverne altri. Se invece viene impostato a **false** verrà creata una sessione solo per la firma dell'adesione FEA.

- `itemData` dati riguardanti all'azienda a cui fa capo il firmatario
- `itemData.id`
- `itemData.vatNumber`
- `itemData.city`
- `itemData.fullAddress`
- `itemData.description`
- `itemData.address`
- `itemData.itemCellNumber`
- `itemData.itemEmail`
- `itemData.policyNumber`
- `itemData.webSite`

```
{
  "transmitterId": "string",
  "imagesContent": [
    "string"
  ],
  "signerId": 0,
  "documentSignType": "string",
  "mainSignerId": "string",
  "itemData": {
    "id": "string",
    "vatNumber": "string",
    "city": "string",
    "fullAddress": "string",
    "description": "string",
    "address": "string",
    "itemCellNumber": "string",
    "itemEmail": "string",
    "policyNumber": "string",
    "webSite": "string"
  },
  "useOtp": true,
  "paperSignature": true
}
```

[Esempio](#)

Delete Extra Signature

[\[DELETE\] /v2/rest/extraSignature/{extraSignatureId}](#)

Permette di eliminare le extraSignature (FEA, FES) di un determinato firmatario.

Info

- `signerId` identificativo del firmatario
- `extraSignatureId` identificativo della FES o FEA collegate ad un firmatario

V2 - CERTIFICATE

Un assign certificate to signer

[\[POST\] /v2/rest/certificate/{certificateId}/unAssignTo/{signerId}](#)

Sostituisce le API v1 [/v1/rest/certificate/frm/unassign-certificate](#) e [/v1/rest/certificate/fsm/unassign-certificate](#)

Scollega un certificato da un firmatario

Info

- `signerId` identificativo del firmatario
- `certificateId` identificativo del certificato
- `serviceName` [controllare le info su questo campo in cima al documento](#)

Send submit request to Trust to start certificate validation

[\[POST\] /v2/rest/certificate/{certificateId}/send](#)

Sostituisce le API v1 [/v1/rest/certificate/fsm/ca/submit](#) e [/v1/rest/certificate/frm/ca/submit](#)

Permette di finalizzare le richieste dei certificati di Firma elettronica qualificata remota/Firma elettronica qualificata automatica(FSM/FRM) mandate alla CA

Info

- `signerId` identificativo del firmatario
- `certificateId` identificativo del certificato che viene staccato durante la richiesta di inizializzazione
- `file` base64 del file pdf di richiesta pre compilato e firmato pades
- `serviceName` [controllare le info su questo campo in cima al documento](#)

```
{
  "signerId": 0,
  "certificateId": 0,
  "file": "string",
  "serviceName": "SIGNATURE-FLOW"
}
```

Assign certificate to signer

[POST] /v2/rest/certificate/{certificateId}/assignTo/{signerId}

Sostituisce le API v1 /v1/rest/certificate/frm/assign-certificate e /v1/rest/certificate/fsm/assign-certificate

Permette di assegnare ad un firmatario un certificato già creato in precedenza da un'altra azienda.

Nome	Cognome	Codice fiscale	ItemId	CertificateId	
Utente1	Utente1	CF1	ITEM1	CERT1	
Utente1	Utente1	CF1	ITEM2		

L'API si occupa di collegare il certificato se il firmatario è lo stesso ma per diverse aziende

Nome	Cognome	Codice fiscale	ItemId	CertificateId	
Utente1	Utente1	CF1	ITEM1	CERT1	
Utente1	Utente1	CF1	ITEM2	CERT1	

Info

- `signerId` identificativo del firmatario
- `certificateId` identificativo del certificato
- `serviceName` controllare le info su questo campo in cima al documento
- `itemId` identificativo dell'azienda
- `name` ragione sociale dell'azienda

```
{
  "itemId": "string",
  "name": "string"
}
```

Initialize an remote Feq

[POST] /v2/rest/certificate/fsm

Sostituisce API v1 /v1/rest/certificate/fsm/initialize

Permette di inizializzare la richiesta per il certificato FSM (Firma elettronica qualificata remota) che poi sarà mandata alla CA.

Info

- `signerId` identificativo del firmatario
- `itemId` identificativo dell'azienda
- `identificationType` permette di specificare lo strumento di autenticazione che verrà utilizzato per il certificato di FEQ qualificata remota. I valori accettati sono **otp** o **SecureCall**. Se il parametro non viene specificato di default viene utilizzato **SecureCall**
- `serviceName` [controllare le info su questo campo in cima al documento](#)

```
{
  "signerId": 0,
  "serviceName": "SIGNATURE-FLOW",
  "itemId": "string",
  "identificationType": "string"
}
```

Initialize an automatic Feq

[\[POST\] /v2/rest/certificate/frm](#)

Sostituisce API v1 /v1/rest/certificate/frm/initialize

Permette di inizializzare la richiesta per il certificato FRM (Firma elettronica qualificata automatica) che poi sarà mandata alla CA.

Info

- `signerId` identificativo del firmatario
- `itemId` identificativo dell'azienda
- `pin` valore di 8 cifre numeriche che serviranno come pin del certificato
- `serviceName` [controllare le info su questo campo in cima al documento](#)

```
{
  "signerId": 0,
  "serviceName": "SIGNATURE-FLOW",
  "itemId": "string",
  "pin": "string"
}
```

Migrate certificate

[POST] /v2/rest/certificate/{certificateId}/migrate

Permette di migrare un certificato da CA Trust a CA Teamsystem.

*1 campo obbligatorio

Info

- `certificateId` *1 identificativo del certificato.
- `signerId` *1 identificativo del firmatario.
- `profile` *1 può assumere questi determinati valori:
 - **"teamsystemTriennale"**
 - **"teamsystemAuto"**
- `identificationProcesses` array che definisce la modalità di riconoscimento, può avere dentro questi valori:
 - **"FEQ"**
- `owner` *1 contiene le informazioni del firmatario.
- `owner.firstname` *1.
- `owner.lastName` *1.
- `owner.email` *1.
- `owner.phoneNumber` *1.
- `owner.fiscalCode`
- `webhook` *1 informazioni sul webhook da richiamare in fase di enrollment
- `webhook.url` *1 url da richiamare deve essere l'URL della signer-write/v2/rest/enrollment/callback a cui va aggiunto come parametro di query:
 - **"signerId"** *1 identificativo del firmatario

```
{
  "signerId": "long",
  "profile": "string",
  "identificationProcesses": "string[]",
  "owner": {
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "phoneNumber": "string",
    "fiscalCode": "string"
  },
  "webhook": {
    "url": "string"
  }
}
```

V1

Create a new signer

[\[POST\] /v1/rest/signer](#)

Permette la creazione di un firmatario per una determinata azienda

Da sostituire con la versione v2, tutte le info sulla request sono le medesime della v2

Disable an existent signer and disable all certificates

[\[DELETE\] /v1/rest/signer/{signerId}](#)

Da sostituire con la versione v2, tutte le info sulla request sono le medesime della v2

Update an existent signer

[\[PATCH\] /v1/rest/signer/{signerId}](#)

Da sostituire con la versione v2, tutte le info sulla request sono le medesime della v2

Initialize FEA

[\[POST\] /v1/rest/extra-signature/create/fea](#)

Da sostituire con la versione v2, tutte le info sulla request sono le medesime della v2

Activate FES

[\[POST\] /v1/rest/extra-signature/create/fes](#)

Da sostituire con la versione v2, tutte le info sulla request sono le medesime della v2

Delete Extra Signature

[\[DELETE\] /v1/rest/extra-signature/delete](#)

Da sostituire con la versione v2, tutte le info sulla request sono le medesime della v2

Initialize

[\[POST\] /v1/rest/certificate/frm/initialize](#)

[\[POST\] /v1/rest/certificate/fsm/initialize](#)

Permette di inizializzare la richiesta per i certificati FRM o FSM (Firma elettronica qualificata automatica, Firma elettronica qualificata remota) che poi sarà mandata alla CA.

Il parametro **identificationType** nella request permette di specificare lo strumento di autenticazione che vorremmo usare per il certificato di FEQ qualificata remota(FSM, **Valido esclusivamente per questa tipologia di certificato**).

Info

- `signerId` identificativo del firmatario
- `itemId` identificativo dell'azienda
- `pin` valore di 8 cifre numeriche che serviranno come pin del certificato
- `identificationType` permette di specificare lo strumento di autenticazione che verrà utilizzato per il certificato di FEQ qualificata remota. I valori accettati sono **otp** o **SecureCall**. Se il parametro non viene specificato di default viene utilizzato **SecureCall**
- `serviceName` il valore da inserire è **SIGNATURE-FLOW**
- `savePin` va impostato sempre a true

Richiesta FRM

```
{
  "signerId": 0,
  "serviceName": "SIGNATURE-FLOW",
  "itemId": "string",
  "pin": "string",
  "savePin": true
}
```

Richiesta FSM

```
{
  "signerId": 0,
  "serviceName": "SIGNATURE-FLOW",
  "itemId": "string",
  "identificationType": "string"
}
```

Submit

[\[POST\] /v1/rest/certificate/frm/ca/submit](#)

[\[POST\] /v1/rest/certificate/fsm/ca/submit](#)

Permette di finalizzare la richiesta del certificato mandando alla CA il documento fillato e firmato

```
{
  "file": "string", //base64 del contratto fillato e firmato
  "signerId": 0
}
```

AssignCertificate

[\[POST\] /v1/rest/certificate/frm/assign-certificate](#)

[\[POST\] /v1/rest/certificate/fsm/assign-certificate](#)

Permette di assegnare ad un firmatario un certificato già creato in precedenza da un'altra azienda.

Nome	Cognome	Codice fiscale	ItemId	CertificateId	
Utente1	Utente1	CF1	ITEM1	CERT1	
Utente1	Utente1	CF1	ITEM2		

L'api si occupa di collegare il certificato se il firmatario è lo stesso ma per diverse aziende

Nome	Cognome	Codice fiscale	ItemId	CertificateId	
------	---------	----------------	--------	---------------	--

Utente1	Utente1	CF1	ITEM1	CERT1	
Utente1	Utente1	CF1	ITEM2	CERT1	

```
{
  "certificateId": 0,
  "itemId": "string",
  "serviceName": "string", //SIGNATURE-FLOW
  "signerId": 0
}
```

UnAssignCertificate

[\[DELETE\] /v1/rest/certificate/frm/unassign-certificate](#)
[\[DELETE\] /v1/rest/certificate/fsm/unassign-certificate](#)

Scollega un certificato da un firmatario

```
{
  "certificateId": 0,
  "signerId": 0
}
```

Signature Read

API di lettura per *Ts Digital Signature* per le operazioni riguardanti i documenti. [Swagger](#)

Header

Gli header richiesti dalla chiamata sono gli [header standard di TSDigital](#).

Il `Content-Type` deve essere `application/json`

Descrizione API

DOCUMENTS

Search documents

[POST] [/v1/documents/search](#)

Permette di effettuare una ricerca filtrata per i documenti di una determinata azienda

Query params

- `page` numero di pagina
- `size` numero di elementi per pagina
- `sort` campo per cui ordinare i documenti

Request body

- `page` numero di pagina **//TODO FIX**
- `size` numero di elementi per pagina **//TODO FIX**
- `sort` campo per cui ordinare i documenti **//TODO FIX**
- `managerId` identificativo dello studio a cui è connessa l'azienda, permette di visualizzare i documenti che lo studio ha creato per conto di una determinata azienda
- `documentStatusId` stato del documento
- `documentTypeId` tipo documento
- `documentOwnerId`

- ownerCriteria.ownerType
- ownerCriteria.searchTextCriteria
- batchId
- multiDocumentSessionId id del documento multi sessione creato
- documentIntermediaryId identificativo azienda, **campo obbligatorio**
- cctStatusCode stato della conservazione del documento
- lastTimestampFrom timestamp da cui inizierà la ricerca dei documenti che sono stati aggiornati fino al momento della chiamata
- minCreationDate data di creazione del documento da cui iniziare la ricerca
- maxCreationDate data di creazione del documento da per finire la ricerca
- expiryDate data di scadenza
- signerTextFieldsSearchCriteria permette di effettuare una ricerca per i firmatario inseriti nei documenti, il campi in cui viene fatta la ricerca sono //chiedere a Nico

```
{ //TODO FIX
  "page": {
    "page": 0,
    "size": 0,
    "sort": [
      "string"
    ]
  },
  "request": {
    "managerId": "string",
    "documentStatusId": "string",
    "documentTypeId": "string",
    "documentOwnerId": "string",
    "ownerCriteria": {
      "ownerType": "string",
      "searchTextCriteria": "string"
    },
    "batchId": "string",
    "multiDocumentSessionId": "string",
    "documentIntermediaryId": "string",
    "cctStatusCode": "string",
    "lastTimestampFrom": 0,
    "minCreationDate": "2021-02-24T23:34:28.076Z",
    "maxCreationDate": "2021-02-24T23:34:28.076Z",
    "expiryDate": "2021-02-24T23:34:28.076Z",
    "signerTextFieldsSearchCriteria": "string"
  }
}
```

Retrieve detail of single document

[GET] [/v1/documents/{hubId}](#)

Permette di recuperare tutte le info di un determinato documento

Query params

- `hubId` identificativo del documento

Get document signature links

[GET] [/v1/documents/{hubId}/signatureLink](#)

Restituisce il link riguardanti una determinata sessione in stato `ALLA_FIRMA` divisi per firmatari

Query params

- `hubId` identificativo del documento

Response

```
[
  {
    "fiscalCode": "CSAMRC80A01I829M",
    "email": "rmainini@mondora.com",
    "url": "https://b2bstaticdev.blob.core.windows.net/static-apps/b2b-firma/actions/redirect.html?sign_token=Y2NjNDYxZTgtOWEyNy00OTdmLTkwYTQtZjd1Yzg1ODUzNmZl"
  },
  {
    "fiscalCode": "DBSNDR93T31B519E",
    "email": "rmainini@mondora.com",
    "url": "https://b2bstaticdev.blob.core.windows.net/static-apps/b2b-firma/actions/redirect.html?sign_token=MzFhZGE3NmItOTM2OS00YTFlWE4MmEtNzg5ZTM5NmExODhl"
  }
]
```

Get multiSessionDocument signature links

[GET] [/v1/documents/multiDocumentSession/{multiDocumentSessionId}/signatureLink](#)

Restituisce il link riguardanti una determinata sessione in stato **ALLA_FIRMA** divisi per firmatari per i documenti multi sessione

Query params

- **multiDocumentSessionId** identificativo del documento multi sessione

Response

```
[
  {
    "fiscalCode": "CSAMRC80A01I829M",
    "email": "rmainini@mondora.com",
    "url": "https://b2bstaticdev.blob.core.windows.net/static-apps/b2b-firma/actions/redirect.html?sign_token=Y2NjNDYxZTgtOWEyNy00OTdmLTkwYTQtZjd1Yzg1ODUzNmZl"
  },
  {
    "fiscalCode": "DBSNDR93T31B519E",
    "email": "rmainini@mondora.com",
    "url": "https://b2bstaticdev.blob.core.windows.net/static-apps/b2b-firma/actions/redirect.html?sign_token=MzFhZGE3NmItOTM2OS00YTFlWE4MmEtNzg5ZTM5NmExODhl"
  }
]
```

Download attachment as base64

[GET] [/v1/documents/{hubId}/attachment/{attachmentId}](#)

Permette il download in base64 di eventuali allegati del documento

Path params

- **hubId** identificativo del documento
- **attachmentId** identificativo dell'allegato, l'id dell'allegato si trova nel json del documento alla chiave **attachments[].id**

Download attachment

[\[GET\] /v1/documents/{hubId}/attachment/{attachmentId}/file](#)

Permette il download del file di eventuali allegati del documento

Path params

- `hubId` identificativo del documento
- `attachmentId` identificativo dell'allegato, l'id dell'allegato si trova nel json del documento alla chiave **attachments[].id**

Download base64 document

[\[GET\] /v1/documents/{hubId}/download](#)

Permette il download in base64 del documento

Path params

- `hubId` identificativo del documento

Query params

- `signed` se messo a true verrà scaricato il documento firmato sennò il documento originale

Download document

[\[GET\] /v1/documents/{hubId}/download/file](#)

Permette il download del file del documento

Path params

- `hubId` identificativo del documento

Query params

- `signed` se messo a true verrà scaricato il documento firmato sennò il documento originale

DOCUMENTS TYPES

Retrieve document type detail

[GET] [/v1/documentTypes/{id}](#)

Restituisce il dettaglio del singolo tipo di documento

Path params

- `id` identificativo del tipo di documento

Retrieve list of document types

[GET] [/v1/documentTypes](#)

Restituisce la lista dei tipi documento

DOCUMENTS STATUSES

Retrieve document status detail

[GET] [/v1/documentStatuses/{id}](#)

Restituisce il dettaglio del singolo stato documento

Path params

- `id` identificativo dello stato di un documento

Retrieve list of document types

[GET] [/v1/documentStatuses](#)

Restituisce la lista degli stati documento

ARCHIVED DOCUMENTS

Search archived documents

Permette di effettuare una ricerca filtrata per i documenti che sono stati cancellati di una determinata azienda

Query params

- `page` numero di pagina
- `size` numero di elementi per pagina
- `sort` campo per cui ordinare i documenti

Request body

- `page` numero di pagina **//TODO FIX**
- `size` numero di elementi per pagina **//TODO FIX**
- `sort` campo per cui ordinare i documenti **//TODO FIX**
- `managerId` identificativo dello studio a cui è connessa l'azienda, permette di visualizzare i documenti che lo studio ha creato per conto di una determinata azienda
- `documentStatusId` stato del documento
- `documentTypeId` tipo documento
- `documentOwnerId`
- `ownerCriteria.ownerType`
- `ownerCriteria.searchTextCriteria`
- `batchId`
- `multiDocumentSessionId` id del documento multi sessione creato
- `documentIntermediaryId` identificativo azienda, **campo obbligatorio**
- `cctStatusCode` stato della conservazione del documento
- `lastTimestampFrom` timestamp da cui inizierà la ricerca dei documenti che sono stati aggiornati fino al momento della chiamata
- `minCreationDate` data di creazione del documento da cui iniziare la ricerca
- `maxCreationDate` data di creazione del documento da per finire la ricerca
- `expiryDate` data di scadenza
- `signerTextFieldsSearchCriteria` permette di effettuare una ricerca per i firmatario inseriti nei documenti, il campi in cui viene fatta la ricerca sono **//chiedere a Nico**

```
{ //TODO FIX
  "page": {
    "page": 0,
    "size": 0,
    "sort": [
      "string"
    ]
  },
  "request": {
```

```
"managerId": "string",
"documentStatusId": "string",
"documentTypeId": "string",
"documentOwnerId": "string",
"ownerCriteria": {
  "ownerType": "string",
  "searchTextCriteria": "string"
},
"batchId": "string",
"multiDocumentSessionId": "string",
"documentIntermediaryId": "string",
"cctStatusCode": "string",
"lastTimestampFrom": 0,
"minCreationDate": "2021-02-24T23:34:28.076Z",
"maxCreationDate": "2021-02-24T23:34:28.076Z",
"expiryDate": "2021-02-24T23:34:28.076Z",
"signerTextFieldsSearchCriteria": "string"
}
}
```

Retrieve detail of single archived document

[GET] [/v1/archivedDocuments/{hubId}](#)

Permette di recuperare tutte le info di un determinato documento che è stato cancellato

Query params

- `hubId` identificativo del documento

TEMPLATES

Get templates list

[GET] [/v1/templates](#)

Permette di avere una lista dei template creati

Query params

- `page` numero di pagina
 - `size` numero di elementi per pagina
 - `sort` campo per cui ordinare i documenti
 - `ownerId` identificativo azienda
-

Retrieve detail of single template

[GET] [/v1/templates/{templateId}](#)

Permette di recuperare tutte le info di un determinato template

Path params

- `templateId` id del template
-

Download base64 template document

[GET] [/v1/templates/{templateId}/templateDocuments/{templateDocumentId}/download](#)

Permette il download in base64 di un determinato documento inserito nel template

Path params

- `templateId` id del template
 - `templateDocumentId` id del documento
-

Download template document

[GET]
[/v1/templates/{templateId}/templateDocuments/{templateDocumentId}/download/file](#)

Permette il download del file di un determinato documento inserito nel template

Path params

- `templateId` id del template
 - `templateDocumentId` id del documento
-
-

//TODO MUOVERE NELLA SEZIONE ESEMPI

Esempi di utilizzo

Fetch documenti incrementale

Per ottenere i documenti in maniera incrementale è necessario utilizzare l'endpoint **/v1/documents/search** specificando un filtro per **lastTimestampFrom** ed un ordinamento per **lastTimestamp**.

```
POST /api/v1/documents/search?page=0&size=100&sort=lastTimestamp,ASC
```

```
body:
```

```
{  
  "lastTimestampFrom" : 1587736390388  
}
```

Il timestamp è uno unix epoch in millisecondi e la prima volta che si effettua la chiamata non è necessario specificarlo.

Ad ogni chiamata è necessario salvarsi il valore più grande del campo **lastTimestamp** presente sui documenti ottenuti all'ultima pagina richiesta (se il sorting è ascendente, in caso di sorting discendente va preso il primo documento della prima pagina).

Di seguito un esempio:

```
--- chiamata 1 ---
```

```
POST /api/v1/documents/search?page=0&size=100&sort=lastTimestamp,ASC
```

```
body: { }
```

```
response:
```

```
{  
  "_embedded": {  
    "documentList": [  
      ... documenti ...  
    ],  
  },  
  "_links": {
```



```

    ... links hateoas ...
  },
  "page": {
    "size": 100,
    "totalElements": 150,
    "totalPages": 2,
    "number": 0
  }
}

--- chiamata 2 ---
POST /api/v1/documents/search?page=1&size=100&sort=lastTimestamp,ASC

body: { }

response:
{
  "_embedded": {
    "documentList": [
      ... primi 49 documenti ...,
      {
        ... campi 50° documento ...,
        "lastTimestamp": 1587736390388
      }
    ]
  },
  "_links": {
    ... links hateoas ...
  },
  "page": {
    "size": 100,
    "totalElements": 150,
    "totalPages": 2,
    "number": 1
  }
}

```

Finito il primo giro di richieste paginate sarà necessario salvare il campo `lastTimestamp` più grande trovato nei documenti ottenuti, in questo esempio il valore **1587736390388**.

Alla richiesta successiva sarà necessario impostare il suddetto valore come filtro **lastTimestampFrom** nel body della richiesta:

```
--- chiamata 1 ---
POST /api/v1/documents/search?page=0&size=100&sort=lastTimestamp,ASC

body:
{
  "lastTimestampFrom": 1587736390388
}

response:
{
  "_embedded": {
    "documentList": [
      ... 9 documenti ...
      {
        ... campi 10° documento ...
        "lastTimestamp": <nuovo valore da salvare>
      }
    ]
  },
  "_links": {
    ... links hateoas ...
  },
  "page": {
    "size": 100,
    "totalElements": 10,
    "totalPages": 1,
    "number": 0
  }
}
```

In questo modo ad ogni nuova chiamata saranno presenti solo documenti che hanno subito una variazione, quindi con un lastTimestamp maggiore a quello ottenuto dalla chiamata precedente.

Signature Write

API di scrittura per *Ts Digital Signature* per le operazioni riguardanti i documenti. [Swagger](#)

Header

Gli header richiesti dalla chiamata sono gli [header standard di TSDigital](#).

Il `Content-Type` deve essere `application/json`

Descrizione API

UPLOAD

Upload file for async sign

[POST] [/v1/rpc/upload/async](#)

Permette di caricare un documento per il [processo asincrono](#)

Request body

I campi contrassegnati da **asterisco (*)** sono da considerarsi **obbligatori**.

- `transmitterId` (*) id dell'azienda
- `managerId` id del commercialista dell'azienda
- `batchId`
- `fileName` (*) nome del file da caricare
- `fileContent` (*) base64 del file da caricare. il limite è di **15MB**
- `documentType` (*) tipo documento, consultare [qui](#) la lista dei tipi di documento disponibili
- `sessionName` nome personalizzato da assegnare alla sessione
- `notes` note aggiuntive sulla sessione
- `thirdPartyEmailAddresses` array di indirizzi email che riceveranno il documento firmato via mail
- `owner` (*) dati relativi all'intestatario del documento, può essere una **persona fisica** o **azienda**

- `owner.firstName`
- `owner.lastName`
- `owner.companyName` ragione sociale
- `owner.vat` partita iva
- `owner.fiscalCode` codice fiscale
- `owner.kind` tipo dell'intestatario. i valori possibili sono **company** o **person**
- `metadata` dati aggiuntivi per la conservazione del documento. **Chiedere info in più** perchè ogni tipo di documento ha dei metadati di conservazione diversi. Oltre ai metadati specifici per la singola classe documentale, per ogni classe documentale bisogna aggiungere questi campi obbligatori:
 - `SoggettoTitolare-CodiceFiscale`
 - `SoggettoTitolare-PartitaIva`
 - `SoggettoTitolare-Nome` e `SoggettoTitolare-Cognome` in caso di persona fisica altrimenti `SoggettoTitolare-Denominazione`
 - `SoggettoTitolare-IdPaese`
- `autoSendCCT` permette di mandare automaticamente il documento in conservazione una volta che il documento è stato firmato. Per far sì che la conservazione avvenga serve che l'azienda abbia il **servizio di cct attivo** e abbia inserito tutti i **metadata obbligatori** per lo specifico documento
- `signatureMode` (*) i valori che può assumere sono **pades** o **cados**
- `mainSignerId` identificativo di un firmatario che ha un certificato di **FEQ automatica** per sigillare il documento. Se si è scelto di utilizzare il **sigillo TS** bisogna impostarlo a **null**
- `expiryDate` (*) data di scadenza del documento, dopo 30 giorni dalla data di invio del documento alla firma per le sessioni **FEA** e **FES** e 60 giorni per le sessioni di **FEQ** il documento andrà automaticamente in stato scaduto se non verrà firmato
- `strictExpiryDate` se impostato a **true** ed è presente una data di scadenza il documento, se non verrà firmato entro quella data, passerà in stato **SCADUTO**. Se invece viene impostato a **false** non verrà fatto il check sulla data di scadenza inserita ma saranno validi i 30/60 giorni specificati sopra per la scadenza del documento
- `year` anno di riferimento del documento
- `signatureProcessName` (*) tipo di processo di firma. I valori possono essere:
 - **FEA remota**
 - **FEA grafometrica**
 - **FES remota**
 - **FES grafometrica**
 - **FEQ remota**
 - **FEQ con dispositivo fisico**
- `areSignersOrdered`: parametro che serve per specificare che i firmatari devono essere ordinati e devono firmare in modo consecutivo
- `attachments` eventuali allegati del documento. In questa fase, gli allegati verranno solo specificati mentre il caricamento verrà effettuato durante il processo di firma della sessione
- `attachments.description`: descrizione dell'allegato
- `attachments.signerFiscalCode`: codice fiscale del firmatario a cui richiedere l'allegato (deve essere presente tra i firmatari)

- `attachments.details`: array dei **file** che verranno richiesti nella sessione di firma. Indipendentemente dal numero di `attachments.details` specificati il risultato finale sarà un PDF con l'aggregazione dei details di un singolo allegato.
- `attachments.details.description`: descrizione del **file** da caricare
- `attachments.details.optional`: impostare a **true** se non si vuol rendere l'allegato obbligatorio

- `signers` (*) array di firmatari che devono firmare il documento
- `signers.fiscalCode`
- `signers.firstName`
- `signers.lastName`
- `signers.email` email del firmatario
- `signers.mobile` numero del **telefono cellulare** del firmatario, verrà utilizzato per la sessione di firma
- `signers.canEmailBeSentAgain` //TODO DA ELIMINARE

- `signers.signatureData` array delle area di firma del firmatario sul documento
- `signers.signatureData.signatureType` identifica il tipo di firma che verrà effettuata, **tutte le area di firma sul documento** devono avere lo stesso valore. I tipi disponibili sono:
 - **SecureCall** tipologia disponibile per **FEA remota** e **FES remota**
 - **otp** tipologia disponibile per **FEA remota** e **FES remota**
 - **BioSignature** tipologia che serve per la **FEA grafometrica in presenza**
 - **BioSignatureRemote** tipologia che serve per la **FEA grafometrica da remoto**
 - **fes_noservice** tipologia che serve per la **FES grafometrica in presenza**
 - **fes_noservice_remote** tipologia che serve per la **FES grafometrica da remoto**
 - **fes** tipologia che serve per la **FES grafometrica con ausilio di tavoletta grafometrica Wacom**
 - **FSM** tipologia che serve per la **FEQ remota**, il valore può essere oltre a FSM anche **FDS** se il certificato di firma qualificata remota è stato richiesto con il metodo di autenticazione **otp**
 - **FSC** tipologia che serve per la **FEQ Remota con dispositivo fisico**
- `signers.signatureData.fieldGroup` può essere valorizzato con un valore personalizzato. Se lo stesso **fieldGroup** verrà messo su più aree di firma il firmatario le firmerà contemporaneamente altrimenti verrà richiesta una firma per ogni singola aree di firma presente sul documento
- `signers.signatureData.shortDescription` descrizione corta dell'area sul documento. Limite caratteri tra 5 e 20
- `signers.signatureData.longDescription` descrizione lunga dell'area sul documento. Limite caratteri 100
- `signers.signatureData.optional` impostare a **true** se non si vuol rendere l'area di firma obbligatoria
- `signers.signatureData.textType` identifica il tipo di area sul documento. Se non impostato di default verrà messo come area di firma. **Il documento deve contenere almeno un area di firma per essere valido.** I tipi possono essere:
 - **signature** area di firma

- **checkbox** per inserire una checkbox, oltre al `textType` dovrà essere inserito come `longDescription` il titolo che si vuole avere la selezione o meno della checkbox
 - **radiobutton** per inserire un radio button, oltre al `textType` le varie scelte del radio button dovranno avere sia la `longDescription` che il `fieldGroup` uguale per ogni casella di scelta del radio button. La `shortDescription` identificherà le varie opzioni. Vedere esempio
 - **text** per inserire del testo, oltre al `textType` dovrà essere inserito come `longDescription` il titolo che si vuole avere per la sezione
 - `signers.signatureData.positioningStrategy` identifica il tipo di posizionamento che verrà usato per le varie aree sul documento. I valori possono essere:
 - **coords** vengono utilizzate delle coordinate per identificare il posizionamento
 - **textPattern** viene utilizzato un id all'interno del documento che farà da riferimento per il posizionamento
-
- `coordsRegion.page`
 - `coordsRegion.h`
 - `coordsRegion.w`
 - `coordsRegion.x`
 - `coordsRegion.y`
-
- `textPatternRegion.pattern` identificativo sul documento dell'area
 - `textPatternRegion.h`
 - `textPatternRegion.w`
 - `textPatternRegion.offsetX`
 - `textPatternRegion.offsetY`
-
- `multipleDocumentSession` se impostato a true il documento farà parte di una sessione multi documento (più documenti verranno firmati con un'unica sessione di firma)
 - `useSignatureTimestamp` permette di applicare una marca temporale al documento
 - `sendSignedDocumentToMail` se impostato a true i firmatari presenti nel documento alla fine della firma del documento riceveranno una mail con in allegato il documento firmato
 - `webHookUrl` permette di specificare un url dove verranno mandati gli eventi relativi all'aggiornamento di stato del documento.
Esempio del body del webhook che verrà mandato
 - `afterSignRedirectUrl` permette di specificare un url su cui l'utente verrà reindirizzato a processo di firma terminato.

```
{
  "hubId": "616ecd1edbd9bc7f725633da",
  "status": {
    "statusCode": "Firmato",
    "errorDescription": null,
    "debugDescription": null,
    "creationDate": "2021-10-19T13:51:58.696Z"
  }
}
```

```
}  
}
```

```
{  
  "transmitterId": "string",  
  "managerId": "string",  
  "batchId": "string",  
  "fileName": "string",  
  "sessionName": "string",  
  "notes": "string",  
  "thirdPartyEmailAddresses": [  
    "string",  
    ...  
    "string"  
  ]  
  "fileContent": "string",  
  "documentType": "string",  
  "owner": {  
    "firstName": "string",  
    "lastName": "string",  
    "companyName": "string",  
    "vat": "string",  
    "fiscalCode": "string",  
    "kind": "string"  
  },  
  "metadata": {  
    "cct": {  
      "additionalProp1": "string",  
      "additionalProp2": "string",  
      "additionalProp3": "string"  
    },  
    "document": {  
      "additionalProp1": "string",  
      "additionalProp2": "string",  
      "additionalProp3": "string"  
    }  
  },  
  "autoSendCCT": true,
```

```
"signatureMode": "string",
"mainSignerId": "string",
"expiryDate": "string",
"year": 0,
"signatureProcessName": "string",
"areSignersOrdered": true,
"webHookUrl": "string",
"afterSignRedirectUrl": "string",
"attachments": [
  {
    "description": "string",
    "signerFiscalCode": "string",
    "details": [
      {
        "description": "string",
        "optional": true
      }
    ]
  }
],
"signers": [
  {
    "fiscalCode": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "mobile": "string",
    "canEmailBeSentAgain": true,
    "signatureData": [
      {
        "signatureType": "string",
        "fieldGroup": "string",
        "shortDescription": "string",
        "longDescription": "string",
        "textType": "string",
        "positioningStrategy": "string",
        "coordsRegion": {
          "page": 0,
          "x": 0,
          "y": 0,
```



```
        "w": 0,
        "h": 0
    },
    "fieldNameRegion": {
        "name": "string",
        "offsetX": 0,
        "offsetY": 0,
        "w": 0,
        "h": 0
    },
    "textPatternRegion": {
        "pattern": "string",
        "number": 0,
        "offsetX": 0,
        "offsetY": 0,
        "w": 0,
        "h": 0
    },
    "optional": true
}
]
}
],
"commonSign": true,
"multipleDocumentSession": true,
"useSignatureTimestamp": true,
"sendSignedDocumentToMail": true
}
```

[Esempi](#)

Upload file and download signed document

[POST] </v1/rpc/upload/sync>

Permette il caricamento di un file seguendo il [processo sincrono](#)

Request body

- `transmitterId` id dell'azienda
- `managerId` id del commercialista dell'azienda
- `batchId`
- `fileName` nome del file da caricare
- `fileContent` base64 del file da caricare. il limite è di **30MB**
- `documentType` tipo documento, consultare [qui](#) la lista dei tipi disponibili
- `owner` dati relativi all'intestatario del documento, può essere una **persona fisica o azienda**
 - `owner.firstName`
 - `owner.lastName`
 - `owner.companyName` ragione sociale
 - `owner.vat` partita iva
 - `owner.fiscalCode` codice fiscale
 - `owner.kind` tipo dell'intestatario. i valori possibili sono **company** o **person**
 - `metadata` dati aggiuntivi per la conservazione del documento. **Chiedere info in più** perchè ogni tipo di documento ha dei metadati di conservazione diversi. Oltre ai metadati specifici per la singola classe documentale per ogni classe documentale bisogna aggiungere questi campi obbligatori:
 - `SoggettoTitolare-CodiceFiscale`
 - `SoggettoTitolare-PartitaIva`
 - `SoggettoTitolare-Nome` e `SoggettoTitolare-Cognome` in caso di persona fisica sennò `SoggettoTitolare-Denominazione`
 - `SoggettoTitolare-IdPaese`
- `autoSendCCT` permette di mandare automaticamente il documento in conservazione una volta che il documento è stato firmato, per far si che la conservazione avvenga serve che l'azienda abbia il **servizio di cct attivo** e abbia inserito tutti i **metadata obbligatori** per lo specifico documento
- `signatureMode` i valori che può assumere sono **pades** o **cados** o **xades**
- `signer` firmatario con un certificato di **FEQ automatica**
 - `signer.fiscalCode`
 - `signer.firstName`
 - `signer.lastName`
 - `signer.email` email del firmatario
 - `signer.mobile` numero del **telefono cellulare** del firmatario
 - `signer.canEmailBeSentAgain` //TODO DA ELIMINARE
 - `pin` deprecato
 - `useSignatureTimestamp` permette di applicare una marca temporale al documento
 - `withoutFile` se impostato a true nella response non verrà riportato il base64 del file firmato

```
{
  "transmitterId": "string",
  "managerId": "string",
```

```
"batchId": "string",
"fileName": "string",
"fileContent": "string",
"documentType": "string",
"owner": {
  "firstName": "string",
  "lastName": "string",
  "companyName": "string",
  "vat": "string",
  "fiscalCode": "string",
  "kind": "string"
},
"metadata": {
  "cct": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "document": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
},
"autoSendCCT": true,
"signatureMode": "string",
"signer": {
  "fiscalCode": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "mobile": "string",
  "canEmailBeSentAgain": true,
  "pin": "string"
},
"withoutFile": true,
"useSignatureTimestamp": false
}
```

V2 - UPLOAD

Upload file for async sign

[POST] [/v2/rpc/upload/async](#)

Permette di caricare un documento per il [processo asincrono](#)

Request body

- `transmitterId` id dell'azienda
- `managerId` id del commercialista dell'azienda
- `batchId`
- `fileName` nome del file da caricare
- `fileContent` base64 del file da caricare. il limite è di **15MB**
- `documentType` tipo documento, consultare [qui](#) la lista dei tipi disponibili
- `sessionName` nome personalizzato da assegnare alla sessione
- `notes` note aggiuntive sulla sessione
- `owner` dati relativi all'intestatario del documento, può essere una **persona fisica o azienda**
 - `owner.firstName`
 - `owner.lastName`
 - `owner.companyName` ragione sociale
 - `owner.vat` partita iva
 - `owner.fiscalCode` codice fiscale
 - `owner.kind` tipo dell'intestatario. i valori possibili sono **company** o **person**
 - `metadata` dati aggiuntivi per la conservazione del documento. **Chiedere info in più** perchè ogni tipo di documento ha dei metadati di conservazione diversi. Oltre ai metadati specifici per la singola classe documentale per ogni classe documentale bisogna aggiungere questi campi obbligatori:
 - `SoggettoTitolare-CodiceFiscale`
 - `SoggettoTitolare-PartitaIva`
 - `SoggettoTitolare-Nome` e `SoggettoTitolare-Cognome` in caso di persona fisica sennò `SoggettoTitolare-Denominazione`
 - `SoggettoTitolare-IdPaese`
- `autoSendCCT` permette di mandare automaticamente il documento in conservazione una volta che il documento è stato firmato, per far sì che la conservazione avvenga serve che l'azienda abbia il **servizio di cct attivo** e abbia inserito tutti i **metadata obbligatori** per lo specifico documento
- `signatureMode` i valori che può assumere sono **pades** o **cades**
- `mainSignerId` identificativo di un firmatario che ha un certificato di **FEQ automatica** per sigillare il documento. Se si scelto di utilizzare il **sigillo TS** bisogna impostarlo a **null**
- `expiryDate` data di scadenza del documento, dopo 30 giorno dalla data di invio del documento alla firma per le sessioni **FEA** e **FES** e 60 giorni per le sessioni di **FEQ** il documento andrà automaticamente in stato scaduto se non verrà firmato

- `strictExpiryDate` se impostato a **true** ed è presente una data di scadenza il documento se non verrà firmato entro quella data passerà in stato **SCADUTO** se invece viene impostato a false non verrà fatto il check sulla data di scadenza inserita ma saranno validi i 30/60 giorni specificati sopra per la scadenza del documento
 - `year` anno di riferimento del documento
 - `signatureProcessName` tipo di processo di firma. I valori possono essere:
 - **FEA remota**
 - **FEA grafometrica**
 - **FES remota**
 - **FES grafometrica**
 - **FEQ remota**
 - **FEQ con dispositivo fisico**
 - `areSignersOrdered`: parametro che serve per specificare che i firmatari devono essere ordinati e devono firmare in modo consecutivo
 - `attachments` eventuali allegati del documento, gli allegati qui verranno solo specificati mentre il caricamento verrà effettuato durante il processo di firma della sessione
 - `attachments.description`: descrizione dell'allegato
 - `attachments.signerFiscalCode`: codice fiscale del firmatario a cui richiedere l'allegato (deve essere presente tra i firmatari)
 - `attachments.details`: array dei **file** che verranno richiesti nella sessione di firma. Indipendentemente dal numero di `attachments.details` specificati il risultato finale sarà un PDF con l'aggregazione dei details di un singolo allegato.
 - `attachments.details.description`: descrizione del **file** da caricare
 - `attachments.details.optional`: impostare a true se non si vuol rendere l'allegato obbligatorio
-
- `signers` array di firmatari che devono firmare il documento
 - `signers.fiscalCode`
 - `signers.firstName`
 - `signers.lastName`
 - `signers.email` email del firmatario
 - `signers.mobile` numero del **telefono cellulare** del firmatario, verrà utilizzato per la sessione di firma
 - `signers.enrollmentInfo` oggetto che consente di passare le informazioni di un firmatario al quale si vuole attivare un certificato di firma qualificata per potergli poi far firmare un documento, al quale devono essere passate le seguenti property:
 - `signerId` identificativo del firmatario.
 - `profile` che può avere uno dei questi determinati valori:
 - **"teamsystemTriennale"**
 - **"teamsystemAuto"**
 - **"teamsystemOneshot"**
 - `identificationProcesses` array che definisce la modalità di riconoscimento, i valori ammessi sono:
 - **"FEQ"**
 - **"SPID"**

- `enrollmentOwner` informazioni del firmatario.
 - `enrollmentOwner.firstname`
 - `enrollmentOwner.lastName`
 - `enrollmentOwner.email`
 - `enrollmentOwner.phoneNumber`
 - `enrollmentOwner.fiscalCode` (facoltativo)
-
- `signers.signatureData` array delle aree di firma del firmatario sul documento
 - `signers.signatureData.signatureType` identifica il tipo di firma che verrà effettuata, **tutte le aree di firma sul documento** devono avere lo stesso valore. I tipi disponibili sono:
 - **SecureCall** tipologia disponibile per **FEA remota** e **FES remota**
 - **otp** tipologia disponibile per **FEA remota** e **FES remota**
 - **BioSignature** tipologia che serve per la **FEA grafometrica in presenza**
 - **BioSignatureRemote** tipologia che serve per la **FEA grafometrica da remoto**
 - **fes_noservice** tipologia che serve per la **FES grafometrica in presenza**
 - **fes_noservice_remote** tipologia che serve per la **FES grafometrica da remoto**
 - **fes** tipologia che serve per la **FES grafometrica con ausilio di tavoletta grafometrica Wacom**
 - **FSM** tipologia che serve per la **FEQ remota**, il valore può essere oltre a FSM anche **FDS** se il certificato di firma qualificata remota è stato richiesto con il metodo di autenticazione **otp**
 - **FSC** tipologia che serve per la **FEQ Remota con dispositivo fisico**
 - `signers.signatureData.fieldGroup` può essere valorizzato con un valore personalizzato, se lo stesso **fieldGroup** verrà messo su più aree di firma il firmatario le firmerà contemporaneamente senno verrà richiesta una firma per ogni singola aree di firma presente sul documento
 - `signers.signatureData.shortDescription` descrizione corta dell'area sul documento Limite caratteri tra 5 e 20
 - `signers.signatureData.longDescription` descrizione lunga dell'area sul documento. Limite caratteri 100
 - `signers.signatureData.optional` impostare a **true** se non si vuol rendere l'area di firma obbligatoria
 - `signers.signatureData.textType` identifica il tipo di area sul documento, se non impostato di default verrà messo come area di firma. Il documento deve contenere almeno un area di firma per essere valido. I tipi possono essere:
 - **signature** area di firma
 - **checkbox** per inserire una checkbox, oltre al `textType` dovrà essere inserito come `longDescription` il titolo che si vuole avere la selezione o meno della checkbox
 - **radiobutton** per inserire un radio button, oltre al `textType` le varie scelte del radio button dovranno avere sia la `longDescription` che il `fieldGroup` uguale per ogni casella di scelta del radio button. La `shortDescription` identificherà le varie opzioni. Vedere esempio
 - **text** per inserire del testo, oltre al `textType` dovrà essere inserito come `longDescription` il titolo che si vuole avere per la sezione
 - `signers.signatureData.positioningStrategy` identifica il tipo di posizionamento che verrà usato per le varie aree sul documento. I valori possono essere:

- **coords** vengono utilizzate delle coordinate per identificare il posizionamento
- **textPattern** viene utilizzato un id all'interno del documento che farà da riferimento per il posizionamento

- `coordsRegion.page`
- `coordsRegion.h`
- `coordsRegion.w`
- `coordsRegion.x`
- `coordsRegion.y`

- `textPatternRegion.pattern` identificativo sul documento dell'area
- `textPatternRegion.h`
- `textPatternRegion.w`
- `textPatternRegion.offsetX`
- `textPatternRegion.offsetY`

- `multipleDocumentSession` se impostato a true il documento farà parte di una sessione multi documento (più documenti verranno firmati con un'unica sessione di firma)
- `useSignatureTimestamp` permette di applicare una marca temporale al documento
- `sendSignedDocumentToMail` se impostato a true i firmatari presenti nel documento alla fine della firma del documento riceveranno una mail con in allegato il documento firmato
- `webHookUrl` permette di specificare un url dove verranno mandati gli eventi relativi all'aggiornamento di stato del documento.
- `afterSignRedirectUrl` permette di specificare un url su cui l'utente verrà reindirizzato a processo di firma terminato.

Esempio del body del webhook che verrà mandato

```
{
  "hubId": "616ecd1edbd9bc7f725633da",
  "status": {
    "statusCode": "Firmato",
    "errorDescription": null,
    "debugDescription": null,
    "creationDate": "2021-10-19T13:51:58.696Z"
  }
}
```

```
{
  "transmitterId": "string",
  "managerId": "string",
  "batchId": "string",
}
```

```
"fileName": "string",
"fileContent": "string",
"documentType": "string",
"sessionName": "string",
"notes": "string",
"owner": {
  "firstName": "string",
  "lastName": "string",
  "companyName": "string",
  "vat": "string",
  "fiscalCode": "string",
  "kind": "string"
},
"metadata": {
  "cct": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "document": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
},
"autoSendCCT": true,
"signatureMode": "string",
"mainSignerId": "string",
"expiryDate": "string",
"year": 0,
"signatureProcessName": "string",
"areSignersOrdered": true,
"webHookUrl": "string",
"afterSignRedirectUrl": "string",
"attachments": [
  {
    "description": "string",
    "signerFiscalCode": "string",
    "details": [
      {
```



```
        "description": "string",
        "optional": true
    }
]
}
],
"signers": [
{
    "fiscalCode": "string",
    "firstName": "string",
    "lastName": "string",
    "email": "string",
    "mobile": "string",
    "canEmailBeSentAgain": true,
    "signatureData": [
        {
            "signatureType": "string",
            "fieldGroup": "string",
            "shortDescription": "string",
            "longDescription": "string",
            "textType": "string",
            "positioningStrategy": "string",
            "coordsRegion": {
                "page": 0,
                "x": 0,
                "y": 0,
                "w": 0,
                "h": 0
            },
            "fieldNameRegion": {
                "name": "string",
                "offsetX": 0,
                "offsetY": 0,
                "w": 0,
                "h": 0
            },
            "textPatternRegion": {
                "pattern": "string",
                "number": 0,
                "offsetX": 0,
```

```

        "offsetY": 0,
        "w": 0,
        "h": 0
    },
    "optional": true
}
]
}
],
"commonSign": true,
"multipleDocumentSession": true,
"useSignatureTimestamp": true,
"sendSignedDocumentToMail": true
}

```

Esempi

Upload file and download signed document

[POST] [/v2/rpc/upload/sync](#)

Permette il caricamento di un file seguendo il [processo sincrono](#)

Request body

- `transmitterId` id dell'azienda
- `managerId` id del commercialista dell'azienda
- `batchId`
- `fileName` nome del file da caricare
- `fileContent` base64 del file da caricare. il limite è di **30MB**
- `documentType` tipo documento, consultare [qui](#) la lista dei tipi disponibili
- `owner` dati relativi all'intestatario del documento, può essere una **persona fisica** o **azienda**
 - `owner.firstName`
 - `owner.lastName`
 - `owner.companyName` ragione sociale
 - `owner.vat` partita iva
 - `owner.fiscalCode` codice fiscale
 - `owner.kind` tipo dell'intestatario. i valori possibili sono **company** o **person**

- `metadata` dati aggiuntivi per la conservazione del documento. **Chiedere info in più** perchè ogni tipo di documento ha dei metadati di conservazione diversi. Oltre ai metadati specifici per la singola classe documentale per ogni classe documentale bisogna aggiungere questi campi obbligatori:
 - `SoggettoTitolare-CodiceFiscale`
 - `SoggettoTitolare-PartitaIva`
 - `SoggettoTitolare-Nome` e `SoggettoTitolare-Cognome` in caso di persona fisica senno `SoggettoTitolare-Denominazione`
 - `SoggettoTitolare-IdPaese`
- `autoSendCCT` permette di mandare automaticamente il documento in conservazione una volta che il documento è stato firmato, per far si che la conservazione avvenga serve che l'azienda abbia il **servizio di cct attivo** e abbia inserito tutti i **metadata obbligatori** per lo specifico documento
- `signatureMode` i valori che può assumere sono **pades** o **cados** o **xades**
- `signer` firmatario con un certificato di **FEQ automatica**
- `signer.fiscalCode`
- `signer.firstName`
- `signer.lastName`
- `signer.email` email del firmatario
- `signer.mobile` numero del **telefono cellulare** del firmatario
- `signer.canEmailBeSentAgain` //TODO DA ELIMINARE
- `pin` deprecato
- `withoutFile` se impostato a true nella response non verrà riportato il base64 del file firmato

```
{
  "transmitterId": "string",
  "managerId": "string",
  "batchId": "string",
  "fileName": "string",
  "fileContent": "string",
  "documentType": "string",
  "owner": {
    "firstName": "string",
    "lastName": "string",
    "companyName": "string",
    "vat": "string",
    "fiscalCode": "string",
    "kind": "string"
  },
  "metadata": {
    "cct": {
```

```
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  },
  "document": {
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
  }
},
"autoSendCCT": true,
"signatureMode": "string",
"signer": {
  "fiscalCode": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "mobile": "string",
  "canEmailBeSentAgain": true,
  "pin": "string"
},
"withoutFile": true
}
```

Append document to multiDocumentSession

Desc

JOB

Start signature job

[POST] </v1/jobs/start>

Permette di creare la sessione di firma del [processo asincrono](#) per firmare il documento

Request body

- `hubId` identificativo del documento
- `sendEmail` **default = true**, permette di ricevere una mail con il link per la sessione di firma per ogni firmatario

[Esempio](#)

Start signature job for multiDocuments

[POST] [/v1/jobs/start/multidocument](#)

Permette di creare la sessione di firma del [processo asincrono](#) contemporaneamente su più documenti(documenti multi sessione)

Request body

- `multiDocumentSessionId` id del documento multi sessione
- `sendEmail` **default = true**, permette di ricevere una mail con il link per la sessione di firma per ogni firmatario

[Esempio](#)

V2 - JOB

Start signature job per CA Teamsystem

[POST] [/v2/jobs/start](#)

Permette di creare la sessione di firma del [processo asincrono](#) per firmare il documento

Request body

- `hubId` identificativo del documento
- `sendEmail` **default = true**, permette di ricevere una mail con il link per la sessione di firma per ogni firmatario

Esempio

Start signature job for multiDocuments

[POST] </v2/jobs/start/multidocument>

Permette di creare la sessione di firma del [processo asincrono](#) contemporaneamente su più documenti(documenti multi sessione)

Request body

- `multiDocumentSessionId` id del documento multi sessione
- `sendEmail` **default = true**, permette di ricevere una mail con il link per la sessione di firma per ogni firmatario

Esempio

Cancel signature job

[POST] </v2/jobs/cancel>

Permette di cancellare una sessione di firma

Request body

- `hubId` id del documento in sessione
- `multiDocumentSessionId` id della multi sessione

CCT

Send document to cct

[POST] </v1/cct/send>

Permette di inviare un documento in conservazione

Request body

- `hubId` identificativo del documento

Send documents to cct

[POST] [/v1/cct/send/multi](#)

Permette di inviare più documenti in conservazione per un massimo di 50 documenti per chiamata

Request body

- `hubIds` array di hubId

MAIL

Resend signature mail

[POST] [/v1/mail/resend](#)

Permette di rimandare la mail con il link per la sessione di firma per uno o più firmatari di un documento se non hanno ancora firmato

Request body

- `hubId` identificativo del documento
- `fiscalCodes` array dei codici fiscali dei firmatari presenti all'interno del documento che devono ricevere di nuovo la mail

DOCUMENT

Archive a document

[POST] [/v1/document/{hubId}/archive](#)

Permette di eliminare un documento. Il documento verrà spostato in una collection quindi per la ricerca eventuale del documento eliminato bisogna utilizzare [l'api apposita](#)

Path params

- `hubId` identificativo del documento
-

Archive a multiDocument session

[POST] [/v1/document/multiDocumentSession/{multiDocumentSessionId}/archive](#)

Permette di eliminare un documento multi sessione. Il documento verrà spostato in una collection quindi per la ricerca eventuale del documento eliminato bisogna utilizzare [l'api apposita](#)

Path params

- `multiDocumentSessionId` id del documento multi sessione

Unarchive a document

[POST] [/v1/document/{hubId}/unarchive](#)

Permette di ripristinare un documento eliminato. Solamente i documenti in stato Firmato o Caricato possono essere ripristinati

Path params

- `hubId` identificativo del documento

Unarchive a multiDocument session

[POST] [/v1/document/multiDocumentSession/{multiDocumentSessionId}/unarchive](#)

Permette di ripristinare un documento multi sessione eliminato. Solamente i documenti in stato **Firmato** o **Caricato** possono essere ripristinati

Path params

- `multiDocumentSessionId` id del documento multi sessione

Remove from multiDocument session

[PATCH] [/v1/document/{hubId}/removeFromMultiDocumentSession](#)

Permette di eliminare un documento da una sessione multi documento. Il documento nello specifico non verrà cancellato ma verrà considerato con un normale singolo documento //TODO

Path params

- `hubId` identificativo del documento

TEMPLATE

Titolo

Desc

Titolo

Desc

Titolo

Desc

Titolo

Desc

Titolo

Desc

Titolo

Desc

Cancel Jobs

POST /v1/jobs/cancel

Permette di cancellare un job precedentemente avviato. Questo comporta l'eliminazione o lo stralcio del documento. Lo stato del documento passerà in **ELIMINATO** se tutto il processo di firma non era ancora concluso altrimenti in **STRALCIATO**. I documenti già mandati in CCT non possono essere ne eliminati ne stralciati.

Signature Config

API di configurazione per *Ts Digital Signature*. [Swagger](#)

Header

Gli header richiesti dalla chiamata sono gli [header standard di TSDigital](#).

Il `Content-Type` deve essere `application/json`

Descrizione API

CONFIG

Get config for specify itemId

[GET] [/config/{itemId}](#)

Permette di recuperare le info sul pacchetto attivato per una specifica azienda. Se l'azienda ha 2 pacchetti, uno intestato a se stessa e uno al commercialista, verrà recuperato il pacchetto intestato al commercialista.

Path params

- `itemId` identificativo dell'azienda. Come valore accetta sia il vecchio valore dell'id (codice fiscale/p.iva) sia **uuid**

Response

```
{
  "itemId": "edf11f25-1a02-402d-9092-4d9d4de64b7d",
  "ownerId": "edf11f25-1a02-402d-9092-4d9d4de64b7d",
  "oldItemId": "MNNRFL96D21I829W",
  "oldOwnerId": "MNNRFL96D21I829W",
  "createdAt": "2021-03-10T13:35:37.087+00:00",
```

```
"updatedAt": "2021-03-10T13:35:37.087+00:00",
"active": true,
"fga": true,
"fra": false,
"frq": false,
"fmq": true,
"erp": false,
"signatureTimeStamp": false,
"readOnly": false
}
```

EXTRADATA

Get extra-data for specify item

[GET] [/registry/extra-data](#)

Permette di recuperare i dati aggiuntivi per una specifica azienda

Query params

- `itemId` identificativo dell'azienda

Response

```
{
  "id": "MNNRFL96D21I829W",
  "webSite": " https://agyo.elevio.help/it/articles/477",
  "policyNumber": "IFL0009971",
  "email": "aa@aa.com",
  "cellNumber": "12342"
}
```

Set extra-data for specify item

[POST] [/registry/extra-data](#)

Permette di impostare/aggiornare i dati aggiuntivi per una specifica azienda

Query params

- `itemId` identificativo dell'azienda

Request Body

- `webSite` default **https://agyo.elevio.help/it/articles/477**
- `cellNumber`
- `email`
- `policyNumber` default **IFL0009971**

SEAL

Get seal info for specify item

[GET] [/seal](#)

Permette di recuperare le informazioni sul sigillo per una specifica azienda

Query params

- `uuid` identificativo uuid dell'azienda

Response

```
{
  "personalSeal": false,
  "personalSealDefault": null
}
```

Set seal option for specify item

[POST] [/seal](#)

Permette di impostare/aggiornare i dati del sigillo per una specifica azienda

Request Body

- `uuid` identificativo uuid dell'azienda
- `personalSeal` se messo a **true** verrà utilizzato il certificato di firma di un firmatario per sigillare i documenti se invece messo a **false** verrà utilizzato il certificato di firma di

TeamSystem

- `personalSealDefault` in caso di `personalSeal` a true si può impostare il **cf** del firmatario titolare del certificato di firma qualificata automatica che sigillerà tutti i documenti

Digital Sign Api

(DEPRECATO)

API di configurazione per *Ts Digital Signature*. [Swagger](#)

Header

Gli header richiesti dalla chiamata sono gli [header standard di TSDigital](#).

Il `Content-Type` deve essere `application/json`

[Servizio deprecato in favore di Signature Config](#)

Descrizione API

EXTRADATA

Get extra-data for specify item (**deprecata**)

[GET] [/registry/extra-data](#)

Permette di recuperare i dati aggiuntivi per una specifica azienda

Query params

- `itemId` identificativo dell'azienda

Response

```
{
  "id": "MNNRFL96D21I829W",
  "webSite": " https://agyo.elevio.help/it/articles/477",
```

```
"policyNumber": "IFL0009971",  
"email": "aa@aa.com",  
"cellNumber": "12342"  
}
```

Set extra-data for specify item (**deprecata**)

[POST] [/registry/extra-data](#)

Permette di impostare/aggiornare i dati aggiuntivi per una specifica azienda

Query params

- `itemId` identificativo dell'azienda

Request Body

- `webSite` default **https://agyo.elevio.help/it/articles/477**
- `cellNumber`
- `email`
- `policyNumber` default **IFL0009971**