

Creazione di una Connessione

Flussi e logiche relative alla creazione di nuove connessioni

- [Invio della richiesta di creazione](#)
- [Tipologie di Connessione](#)
- [Notifiche di connessione](#)

Invio della richiesta di creazione

La seguente documentazione fa riferimento alla versione 2 delle API di scrittura delle connessioni. La versione 1 è deprecata e non va utilizzata per nuove integrazioni.

In questo documento viene descritta l'API di creazione di una nuova connessione e gli eventuali errori che è possibile ricevere durante il processo..

API DI CREAZIONE

L'API di creazione di una nuova connessione è disponibile con metodo `POST` all'endpoint `/api/v2/connections`.

Header

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

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

Body

Il body della richiesta deve avere il seguente formato:

```
{
  "appId": "string",
  "featureCode": "string",
  "permission": "string",
  "recipientId": "string",
  "requesterId": "string",
  "serviceId": "string",
  "userRoles": [
    {
      "type": "TECH|PERSONAL",
      "userId": "string"
    }
  ]
}
```

```
}  
]  
}
```

- **appId:** identificativo dell'applicazione per la quale sta venendo effettuata la connessione (es: `EIP`)
- **featureCode:** identificativo della specifica feature dell'applicazione per la quale si sta effettuando la connessione. Se l'applicazione non ha feature secondarie, il campo va impostato a `null`. (es: `SDI`)
- **permission:** permessi da attribuire all'utente che sta creando la connessione. I valori accettati sono `READ`, `READ_WRITE`
- **recipientId:** identificativo univoco dell'item gestito all'interno dell'[anagrafica di TSDigital](#).
- **requesterId:** identificativo univoco dell'item gestore all'interno dell'[anagrafica di TSDigital](#).
- **serviceId:** identificativo univoco del servizio per il quale sta venendo effettuata la connessione. Ad ogni coppia `appId + featureCode` corrisponde uno ed un solo `serviceId` univoco. (es: `SDI-FLOW`)
- **userRoles:** array di utenze aggiuntive alle quali attribuire i ruoli risultanti dalla connessione. Se vuoto, i ruoli vengono assegnati alla sola utenza che ha effettuato la richiesta di connessione.
 - **type:** tipologia dell'utenza. `TECH` -> utenza applicativa, `PERSONAL` -> utenza regolare
 - **userId:** identificativo dell'utenza alla quale assegnare i ruoli (es: `test@mondora.com`, `12739bb6-9782-4d08-872e-98b8ea94e3ce`)

L'elenco completo delle applicazioni presenti su TS Digital e dei relativi `serviceId`, `appId`, e `featureCode` è consultabile invocando l'apposita API del servizio [services-subscription](#).

Risposte

L'operazione è avvenuta con successo se e solo se il codice HTTP della risposta è `202`. Ogni altro codice di risposta indica uno stato di errore.

HTTP 202

L'operazione è avvenuta con successo e il processo di creazione è stato preso in carico.

Body della risposta:

```
{}
```

HTTP 400

Uno o più parametri forniti nella richiesta sono errati, o mancano dei parametri obbligatori.

HTTP 401

Il token autorizzativo è scaduto, invalido o non è stato specificato.

HTTP 403

Il token autorizzativo fornito è valido, ma l'utente non ha i permessi necessari a creare una connessione per il gestore specificato.

HTTP 409

L'azienda gestita ha già un gestore per il servizio specificato. Non è permesso creare più connessioni per uno stesso servizio.

HTTP 500

Il server ha riscontrato un errore inaspettato nella creazione della richiesta di connessione

HTTP 502

Il server ha riscontrato un errore inaspettato nel comunicare con un servizio dal quale dipende per poter completare il processo (ad esempio, il servizio di auth non risulta essere disponibile)

Tutte le risposte d'errore condividono il seguente formato per il body di risposta:

```
{
  "code": "string",
  "message": "string",
  "status": "string",
  "subErrors": [
    {}
  ],
  "timestamp": "dd-MM-yyyy HH:mm:ss"
}
```

- **code:** corrisponde al codice d'errore HTTP ritornato (es: 409)
- **message:** messaggio d'errore (es: Impossibile creare una connessione gia' esistente)
- **status:** descrizione a parole del codice d'errore HTTP (es: Conflict)
- **subErrors:** eventuali errori innestati in quello ritornato
- **timestamp:** data ed ora di ritorno dell'errore

Tipologie di Connessione

A seconda delle condizioni di gestore e gestita, una connessione viene creata con una di 3 tipologie: **USER**, **AUTO** e **SYS**.

Connessioni USER

Una connessione di tipo USER è una connessione effettuata fra 2 item indipendenti all'interno di TS Digital. La connessione viene creata disattiva con status **UNVERIFIED**.

Per connessioni di questo tipo, l'azienda gestita riceve una notifica, visibile ad amministratori ed owner, la quale informa dell'avvenuta richiesta e propone la possibilità di accettarla o rifiutarla. Se la richiesta di connessione viene accettata, la connessione diventa immediatamente attiva e il suo status passa a **VALIDATED**. Se, in caso contrario, viene rifiutata, la connessione rimane inattiva e il suo stato passa a **REJECTED**. Una connessione rifiutata non può più essere attivata, ma rimarrà visibile fino all'arrivo di una nuova richiesta di connessione per lo stesso servizio.

Una volta che la connessione è in stato **VALIDATED** è possibile procedere con la sua certificazione tramite il caricamento di un Atto d'Affidamento. La certificazione non è necessaria per l'utilizzo della connessione, ma potrebbe essere necessaria per l'utilizzo di alcune funzionalità del servizio connesso (es: firma Teamsystem per il servizio di fatturazione).

Connessioni AUTO

Una connessione di tipo AUTO è una connessione nella quale l'item gestito non ha alcun utente owner (questi item sono noti anche come **aziende orfane**). L'item non deve inoltre aver mai avuto nessun altro gestore per lo specifico servizio connesso, o deve già esistere un AdA certificato sotto lo stesso appld per la coppia gestore/gestita.

La connessione viene creata già attiva ed in stato **VALIDATED**, è immediatamente utilizzabile ed è possibile procedere con la sua certificazione.

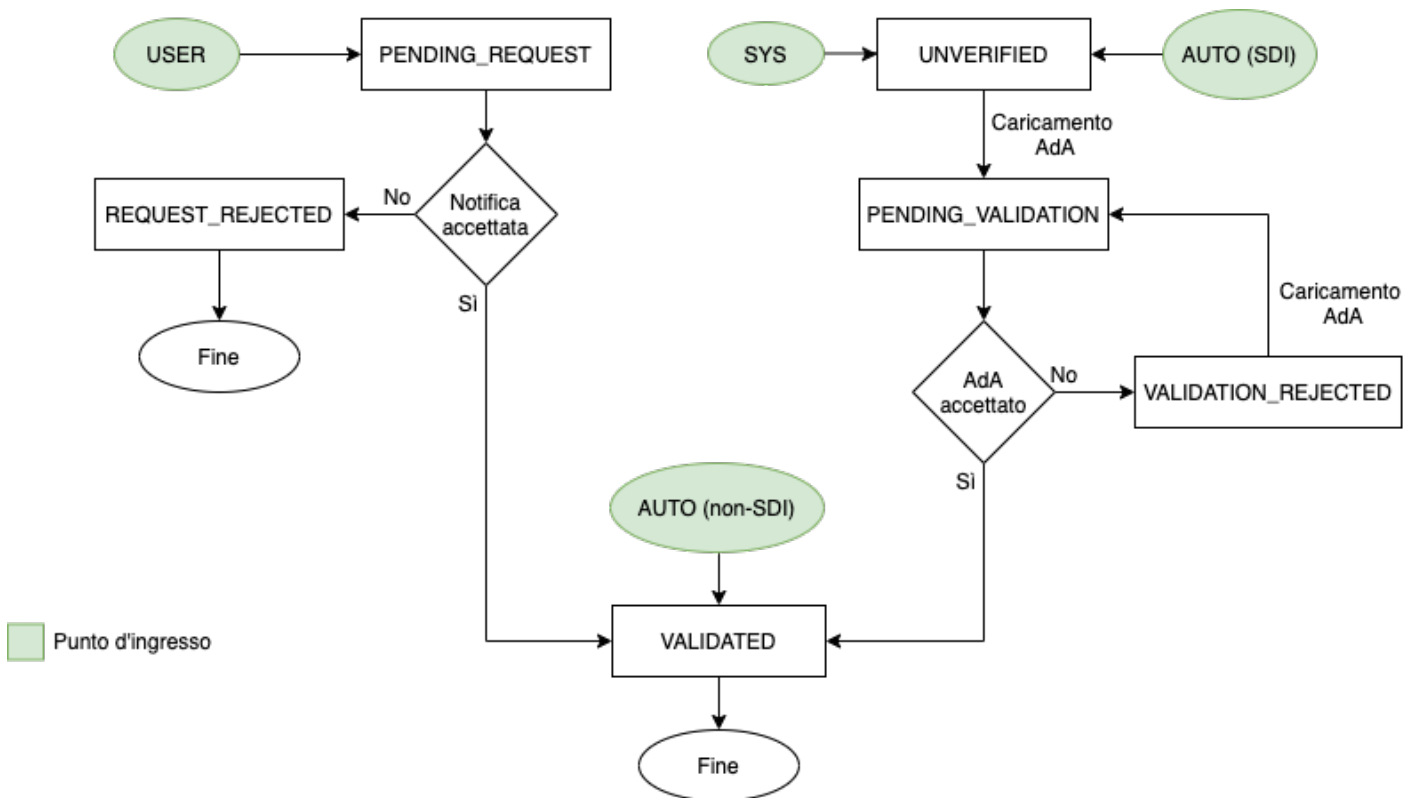
Connessioni SYS

Una connessione SYS è un caso particolare della connessione AUTO; si verifica nel caso in cui l'item gestito è un'azienda orfana che ha già avuto in passato un gestore (differente dall'attuale) per lo stesso servizio.

La connessione viene creata disattiva con status `UNVERIFIED`. Per attivare la connessione è necessario caricare un AdA ed attendere che venga certificato. Post certificazione la connessione viene attivata, il suo stato passa a `VALIDATED` e il suo `certificationStatus` passa a `CERTIFIED`.

Flussi di stato

Flusso di validazione



Il flusso di validazione influenza lo `status` di una connessione segue un flusso che ha due possibili stati finali: `VALIDATED` e `REQUEST_REJECTED`.

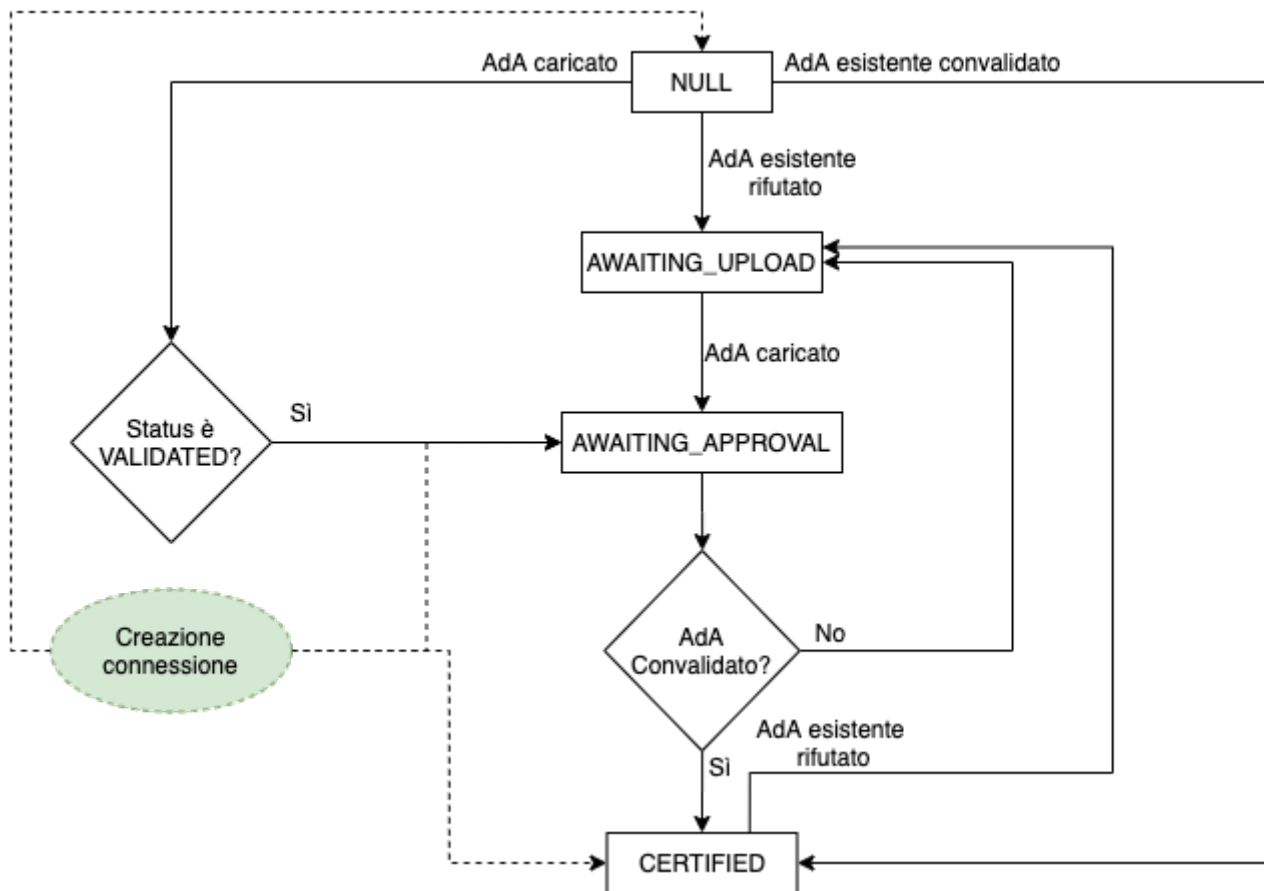
Le connessioni di tipologia `USER` vengono create con lo status `PENDING_REQUEST`. L'accettazione della notifica di richiesta connessione da parte della gestita porta lo status a `VALIDATED`, il suo rifiuto a `REQUEST_REJECTED`. Contestualmente al passaggio allo stato `VALIDATED`, la connessione viene anche attivata.

Le connessioni di tipo `AUTO` vengono create già attive, ma possono iniziare in uno di due stati a seconda del servizio a cui fanno riferimento:

- Le connessioni per `SDI` partono dallo stato `UNVERIFIED` e, per poter diventare `VALIDATED` devono necessariamente effettuare l'upload di un Atto D'Affidamento e seguire il relativo flusso di accettazione/rifiuto
- Le connessioni per ogni altro servizio partono direttamente dallo stato `VALIDATED`. Hanno comunque la possibilità di caricare un AdA e rientreranno nelle logiche del flusso di certificazione.

Le connessioni di tipo `SYS` partono dallo stato di `UNVERIFIED` ma, al contrario delle `AUTO`, nascono non attive. L'attivazione della connessione avviene alla fine del processo di convalida dell'AdA, il quale è quindi necessario per poterla utilizzare.

Flusso di Certificazione



Il flusso di certificazione influenza il `certificationStatus` della connessione, ed è interamente pilotato da eventi relativi all'AdA della connessione.

La condizione iniziale del `certificationStatus` dipende dalla tipologia di connessione considerata e dall'esistenza o meno di un AdA per una connessione esistente sotto lo stesso appld per la coppia gestore/gestita:

- Se la connessione è di tipo `SYS` partirà sempre dallo status `NULL`
- Se, per la coppia gestore/gestita, esiste già un AdA che si trovi almeno in `AWAITING_APPROVAL`, la nuova connessione viene creata allo stesso stato. Da quel momento in poi, modifiche a quell'AdA avranno effetto su tutte le connessioni ad esso collegate
- In ogni altro caso, lo stato di certificazione parte da `NULL`

Se una connessione ha stato di certificazione `NULL` ma il suo stato di validazione non è ancora `VALIDATED`, lo stato di certificazione rimarrà `NULL` fino al termine della convalida dell'AdA, quando passerà regolarmente allo stato di `CERTIFIED`.

Il flusso di certificazione è infinitamente ripetibile: in qualunque momento, se l'AdA dovesse venir invalidato (manualmente da backoffice, o come conseguenza di una rettifica di codice fiscale), lo

stato di certificazione verrà riportato ad `AWAITING_APPROVAL`.

Notifiche di connessione

Il servizio delle notifiche di connessione si occupa di raccogliere e gestire le notifiche generate dalle connessioni di tipologia `USER`.

Tipologie di notifiche

Il servizio delle notifiche di connessione gestisce due categorie di notifiche: `INFO` e `REQUEST`.

Notifiche INFO

Le notifiche di tipo `INFO` sono notifiche puramente informative. Vengono ricevute, possono venir segnate come lette o non lette, ma non hanno altre azioni collegate ad esse. Un esempio di notifica di tipo `INFO` è la notifica di connessione accettata che viene mandata all'item gestore nel caso in cui l'item gestito decida di accettare la richiesta.

Notifiche REQUEST

Le notifiche di tipo `REQUEST` sono le notifiche utilizzate per chiedere l'accettazione/rifiuto di una connessione di tipo `USER`. Un'interfaccia grafica che voglia gestire queste notifiche deve quindi presentare all'utente un bottone di accettazione e uno di rifiuto.

Notification Read

Servizio di lettura delle notifiche, offre una API per recuperare tutte le notifiche di un dato item e una per recuperare una specifica notifica dato il suo ID.

GET /api/v2/notifications/{notificationId}

API che permette di recuperare i dati di una specifica notifica, dato il suo ID univoco

Header

Il servizio richiede gli [header standard di TS Digital](#).

Path Parameters

- **notificationId:** identificativo univoco della notifica da recuperare

Risposta

In caso di risposta positiva l'API ritorna un codice HTTP `200`

```
{
  "notification": {
    "id": "string",
    "correlationId": "string",
    "requesterId": "string",
    "requesterName": "string",
    "recipientId": "string",
    "recipientName": "string",
    "name": "string",
    "type": "INFO",
    "createdAt": 0,
    "createdBy": "string",
    "note": "string",
    "resourceId": "string",
    "readStatus": true,
    "accepted": true,
    "rejected": true
  }
}
```

La chiave top level `notification` contiene tutte le informazioni recuperate sulla notifica:

- **id:** identificativo univoco della notifica
- **correlationId:** identificativo UUIDV4 utilizzato per correlare fra loro due o più notifiche (es: la notifica ricevuta dalla gestita per ottenere l'approvazione della connessione e la relativa notifica di feedback ritornata al gestore hanno lo stesso `correlationId`)
- **requesterId:** identificativo dell'item che ha originato la notifica
- **requesterName:** nome/ragione sociale dell'item che ha originato la notifica
- **recipientId:** identificativo dell'item che ha ricevuto la notifica
- **recipientName:** nome/ragione sociale dell'item che ha ricevuto la notifica
- **name:** nome identificativo della notifica (es: `CONNECTION_REMOVED_NOTIFICATION_REQUEST`)
- **type:** tipologia della notifica. Valori possibili: `INFO`, `REQUEST`
- **createdAt:** data di creazione della notifica, espressa come unix timestamp (risoluzione in millisecondi)
- **createdBy:** identificativo dell'utente che ha causato la creazione della notifica
- **note:** note aggiuntive sulla notifica
- **resourceId:** identificativo della connessione cui la notifica fa riferimento
- **readStatus:** booleano che indica se la notifica è stata marcata come letta o meno
- **accepted:** booleano che indica se la notifica è stata accettata
- **rejected:** booleano che indica se la notifica è stata rifiutata

GET /api/v2/notifications

API che permette di recuperare i dati di tutte le notifiche collegate ad uno specifico item

Header

Il servizio richiede gli [header standard di TS Digital](#).

Query Parameters

- **recipientId**: identificativo univoco dell'item per il quale si vogliono ottenere le notifiche ricevute
- **page**: numero di pagina da recuperare (0...N)
- **size**: numero di notifiche per pagina

Risposta

In caso di risposta positiva l'API ritorna un codice HTTP `200`

```
{
  "notifications": [
    {
      "id": "string",
      "correlationId": "string",
      "requesterId": "string",
      "requesterName": "string",
      "recipientId": "string",
      "recipientName": "string",
      "name": "string",
      "type": "INFO",
      "createdAt": 0,
      "createdBy": "string",
      "note": "string",
      "resourceId": "string",
      "readStatus": true,
      "accepted": true,
      "rejected": true
    }
  ],
  "totalItems": 0,
  "unreadNotifications": 0
}
```

La risposta contiene le seguenti informazioni:

- **notifications:** contiene un array contenente tutte le notifiche presenti nella pagina corrente. Per il significato dei termini, vedi la risposta di `/api/v2/notifications/{notificationId}`
- **totalItems:** totale delle notifiche disponibili per l'item specificato
- **unreadNotifications:** totale delle notifiche non lette disponibili per l'utente selezionato

Notification Write

Servizio di scrittura delle notifiche, fornisce le API per segnare come letta/non letta, accettare o rifiutare una notifica.

POST /api/v2/notifications/{id}/accept

API che permette di accettare una notifica di tipo `REQUEST`

Header

Il servizio richiede gli [header standard di TS Digital](#).

Path Parameters

- **id:** identificativo univoco della notifica da accettare

Body

```
{
  "note": "string"
}
```

- **note:** note aggiuntive sull'accettazione della notifica. Opzionale

Risposta

In caso di risposta positiva l'API ritorna un codice HTTP `200`

```
{
  "id": "string"
}
```

La risposta contiene le seguenti informazioni:

- **id:** identificativo della notifica accettata

POST /api/v2/notifications/{id}/read

API che permette di marcare una notifica come letta

Header

Il servizio richiede gli [header standard di TS Digital](#).

Path Parameters

- **id:** identificativo univoco della notifica da segnare come letta

Body

```
{
  "note": "string"
}
```

- **note:** note aggiuntive sulla lettura della notifica. Opzionale

Risposta

In caso di risposta positiva l'API ritorna un codice HTTP `200`

```
{
  "id": "string"
}
```

La risposta contiene le seguenti informazioni:

- **id:** identificativo della notifica segnata come letta

POST /api/v2/notifications/{id}/reject

API che permette di rifiutare una notifica di tipo `REQUEST`

Header

Il servizio richiede gli [header standard di TS Digital](#).

Path Parameters

- **id:** identificativo univoco della notifica da rifiutare

Body

```
{
  "note": "string"
}
```

- **note:** note aggiuntive sul rifiuto della notifica. Opzionale

Risposta

In caso di risposta positiva l'API ritorna un codice HTTP `200`

```
{
  "id": "string"
}
```

La risposta contiene le seguenti informazioni:

- **id:** identificativo della notifica rifiutata

POST /api/v2/notifications/{id}/unread

API che permette di segnare una notifica come non letta

Header

Il servizio richiede gli [header standard di TS Digital](#).

Path Parameters

- **id:** identificativo univoco della notifica da segnare come non letta

Body

```
{
  "note": "string"
}
```

- **note:** note aggiuntive sulla notifica. Opzionale

Risposta

In caso di risposta positiva l'API ritorna un codice HTTP `200`

```
{  
  "id": "string"  
}
```

La risposta contiene le seguenti informazioni:

- **id:** identificativo della notifica segnata come non letta