

Invoice

- [Intro](#)
- [Linee guida Generali API TS Digital](#)
- [Descrizione API](#)
 - [Upload Fatture](#)
 - [Simulazione invio a SDI](#)
 - [Caricamento notifiche](#)
 - [ItemOverview](#)
 - [Lettura Fatture](#)
 - [ItemId \(UUID\)](#)
 - [Migrazione da API REST v1 a v2](#)
- [Flussi](#)
 - [Flusso SDIPR](#)
 - [Flusso SDIPA](#)
 - [Flusso SELFINV](#)
 - [Flusso SELFSEND](#)
 - [Flusso IMPEXP](#)
 - [Flusso CSV](#)
- [Processo di Conservazione \(CCT\)](#)

Intro

Prerequisiti

- L'azienda deve avere un'anagrafica presente su TS Digital;
- L'azienda deve avere il servizio di fatturazione attivo;
- Il file XML deve essere già generato.

Le API effettuano dei controlli sulla struttura del file XML (come da specifiche tecniche dettate dall' AdE) e lo inviano allo SDI.

Header

Per l'utilizzo di qualsiasi API di TS Digital sarà necessario rispettare le [Linee Guida Generali](#)

Chiamate

Le API di Fatturazione si dividono in due tipologie:

Scrittura (b2bwrite)

Chiamate che consentono l'invio di documenti a TS Digital.

L'invio di ogni documento segue un processo ASincrono, quindi, viene caricato un documento e si riceve in risposta un **hubId** con il quale referenziarlo, il documento verrà processato successivamente in un secondo momento, il tempo di processamento differisce in base al traffico presente sui sistemi. Indicativamente può variare da pochi secondi a qualche ora.

Lettura (b2bread)

Chiamate che consentono la lettura dei documenti presenti in TS Digital.

E' possibile recuperare i dettagli di una singola fattura, la lista delle fatture, il download del file XML, ecc.

Info utili

Servizio SDI

Il servizio SDI è il servizio che permette l'invio di fatture elettroniche al sistema di interscambio. Fra le configurazioni presenti vi è quella di poter scegliere se firmare o meno il documento che viene inviato al sistema di interscambio (SDI).

Se si sceglie di NON firmare, potranno essere inviate solo fatture elettroniche SDIPR. per le SDIPA è obbligatoria la firma.

Servizio E-Ricevi

Il servizio E-Ricevi permette la ricezione delle fatture elettroniche dal servizio di interscambio (SDI) ma anche l'utilizzo dei seguenti flussi:

- SELFSEND
- STORE
- IMPEXP
- OCR

Se non si possiede il servizio sarà impossibile utilizzare i suddetti flussi. Mentre In caso di ricezione di una fattura passiva da SDI sarà possibile vederla ma non è possibile scaricarla o effettuarne l'anteprima.

A breve non sarà più possibile nemmeno vederla, verrà messa fra le fatture "parcheeggiate" e sarà possibile accedervi solo dopo aver acquistato il pacchetto

HubId

Ad ogni documento presente su TS Digital viene attribuito un codice univoco definito "HubId". Il riferimento ad una fattura avviene quindi tramite HubId.

Questo elemento verrà richiesto in alcune chiamate, specie in quelle di lettura.

L'hubId dovrà essere trattato come una stringa univoca da 1 a 36 caratteri. Esempio:

`630f6f88ea51031614a62651`

Gli hubId già forniti NON verranno modificati.

Uffici

Gli uffici sono sotto-insiemi di un'azienda. Per referenziare un ufficio bisogna aggiungere un suffisso all'ItemId dell'azienda.

Per referenziare l'ufficio 1 dell'azienda ABCDE ad esempio, occorre specificare ABCDE-**001**

Per referenziare tutti gli uffici di un'azienda si può utilizzare il suffisso **-ALL**

Cancellazione Fatture

Tutte le fatture scartate da TS Digital (non dallo SDI) saranno eliminate fisicamente dal DB dopo 120 gg.

Se si prova a visualizzare quella fattura sarà restituito errore “documento non esistente” e status code 404.

Linee guida Generali API TS Digital

Header

Ad ogni chiamata è obbligatorio inserire i parametri riportati e popolarli nel seguente modo:

Authorization: Bearer Token. Come si genera il token? Vedi nella sezione [info](#)

X-App-Name: codice identificativo rilasciato da Teamsystem al momento del censimento. Non utilizzare valori diversi dall'AppName assegnato in quanto vi sono dei controlli che bloccano tutte le chiamate con AppName non censiti.

X-User-Agent: nome dell'applicativo/software utilizzato.

X-App-Version: versione dell'applicativo/software utilizzato.

X-Request-ID: Identificativo univoco della singola richiesta nel formato cuuid. Esempio: se per ottenere la lista documenti devo chiamare Auth, Anagrafica e B2BRead ogni singola chiamata deve avere un valore diverso di X-Request-ID. NON utilizzare lo stesso valore per chiamate diverse.

X-Correlation-ID: Identificativo univoco del flusso in formato uuid-v4. Esempio: se per ottenere la lista documenti devo chiamare Auth, Anagrafica e B2BRead tutte le chiamate dovrebbero avere lo stesso X-Correlation-ID. In questo modo nei log è possibile tracciare e correlare tutte le richieste. NON utilizzare lo stesso valore per flussi diversi.

X-Item-ID: identificativo di chi effettua la richiesta (UUID). Come si recupera l'UUID? Vedi nella sezione [info](#)

X-User-ID: identificativo dell'utenza che sta effettuando la richiesta. (ID della Chiave Tecnica).

X-Manager-ID: parametro opzionale. Da popolare con UUID dello studio che opera per aziende gestite.

Chiavi Tecniche

Le credenziali (id e secret) da utilizzare nell'autenticazione possono essere generate dalla sezione applicativi del portale.

Per saperne di più, clicca [qui](#).

Token

Per generare il token invece, i passaggi sono i seguenti:

1. Generare il nonce, inserendo nel body della chiamata l'ID della chiave tecnica.
2. Generare il token, inserendo nel body della chiamata l'ID della chiave tecnica.

Calcolare il digest tramite la seguente formula: $sha256(sha256(ID + secret) + nonce)$

Chiamata per generare il nonce:

```
curl --location --request POST 'https://b2b-auth-service-test.agyo.io/api/v3/nonces' \
--header 'accept: application/json' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: TSxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: 1' \
--header 'X-Correlation-ID: 1' \
--header 'Content-Type: application/json' \
--data-raw '{
  "id": "string"
}'
```

Chiamata per generare il token:

```
curl --location --request POST 'https://b2b-auth-service-test.agyo.io/api/v3/tokens' \
--header 'accept: application/json' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: TSxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: 1' \
--header 'X-Correlation-ID: 1' \
--header 'Content-Type: application/json' \
--data-raw '{
  "id": "string",
  "digest": "string"
}'
```

Scadenza Token

Il token generato tramite le chiavi tecniche non ha scadenza.

Se vengono effettuate però modifiche sulla chiave tecnica il token potrebbe invalidarsi.

A quel punto è necessario fare un refresh della chiamata.

Per verificare che il token sia valido c'è una chiamata apposita: <https://b2b-auth-service-test.agyo.io/api/v3/verify>

UUID

Per recuperare l'UUID invece bisogna effettuare la chiamata sottostante ed inserire il CF/P.iva dell'azienda nel parametro "*vatNumber*".

```
curl --location --request GET 'https://registry-read-test.agyo.io/api/v3/items?identifier.vatNumber=48548641546&packageType=BASE&pagination.itemsPerPage=10&pagination.pageNumber=0&identifier.taxRegion=IT' \
--header 'accept: application/json;charset=utf-8' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: TSxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: 1' \
--header 'X-Correlation-ID: 1' \
--header 'Authorization: Bearer ' \
--data-raw ''
```

Nella response poi, il parametro da prendere in considerazione è "**id**".

Esempio:

[image-1649080956843.png](#)

Switchare da TEST a PRODUZIONE

Una volta effettuate tutte le prove sull'ambiente di test TS Digital si potrà passare all'ambiente di produzione.

Per effettuare questo switch bisogna:

- Eliminare il (-test) dall'endpoint delle chiamate.

Esempio <https://b2b-auth-service-test.agyo.io/api/v3/tokens> --> <https://b2b-auth-service-agyo.io/api/v3/tokens>

- Generare il token con le chiavi tecniche dell'ambiente di produzione.

Errori durante le chiamate

Possibili errori che possono essere restituiti dalla response:

- **400 Bad Request** - Probabilmente c'è un errore nella struttura della chiamata. Controllare che tutti i tag del body, gli header e l'endpoint siano impostati nel modo corretto.
- **401 Unauthorized** - Probabilmente c'è un errore nel token generato. Controllare che ID e Secret siano corretti e fare un refresh della chiamata per generare un nuovo token.
- **403 Forbidden** - Probabilmente non si hanno i permessi necessari per poter effettuare quella chiamata.
In questo caso bisogna verificare i permessi attribuiti alla chiave tecnica da portale, eventualmente aggiungerli e poi procedere con la generazione di un nuovo token.

A breve verranno introdotti limiti/controlli sul numero di chiamate effettuabili in un arco temporale (da definire). In caso di sfioramento dei tali limiti è necessario gestire il codice http 429 Too many Requests.

Descrizione API

Upload Fatture

Le tipologie di documenti inviabili sono definite "Flussi" sono visibili [qui](#). Ogni flusso ha i suoi stati.

L'upload di un xml non valido rispetto allo schema xsd avrà come risposta un errore 400 (la fattura non viene quindi salvata su TSDigital).

Per i flussi SDIPA, SDIPR e SELFINV verranno effettuati anche i controlli da parte dello SDI e il mancato superamento impedirà l'upload con un errore 400. Anche in questo caso la fattura non viene salvata su TSDigital.

Nell'ambiente di test è possibile inviare notifiche pilotando l'esito di risposta SDI. Vedi [Mock Sdi](#).

E' quindi importante gestire correttamente gli errori in risposta poiché in futuro questi controlli potrebbero essere estesi.

Upload Fattura

[POST/v2/invoices](#)

Esempio chiamata:

```
curl --location --request POST 'https://b2bwrite-api-test.agyo.io/api/v2/invoices' \
--header 'Authorization: Bearer' \
--header 'accept: application/json;charset=utf-8' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: VRxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: Codice Univoco (cuid)' \
--header 'X-Correlation-ID: Codice Univoco (cuid)' \
--header 'X-Item-ID: ItemID Azienda' \
--header 'X-User-ID: ID Chiave Tecnica' \
--header 'Content-Type: application/json;charset=utf-8' \
--data-raw '{
  "transmitterId": "ItemId Azienda",
  "senderId": "ItemId Azienda",
  "flowType": "SDIPR",
```

```
{
  "fileName": "nome file XML",
  "content": "file XML in base64"
}'
```

Esempio risposta upload OK (Status Code 201):

```
{
  "hubId": "5e980d7f9715a77a998d0047",
  "link": {
    "rel": "self",
    "href": "https://b2bread-api-test.agyo.io/api/v2/invoices/5e980d7f9715a77a998d0047"
  }
}
```

In caso di risposta positiva 201 potrebbe essere necessario fino a qualche secondo affinché la fattura sia disponibile in lettura, questo per via dei tempi di replica dell'informazione sulle diverse repliche del database. Per questioni di performance il server ritorna il 201 quando il dato è replicato sulla metà + 1 delle repliche, questo consente di garantire il salvataggio del dato e una minor attesa del client.

Una volta effettuato l'upload evitare di chiedere immediatamente lo stato ma attendere almeno 5 minuti prima di tentare aggiornamenti di stato. Il processo asincrono sebbene sia spesso immediato può richiedere più tempo in situazioni di alto carico. Chiamate a vuoto non fanno altro che alzare il carico

Esempio di risposta upload XML non valido, controllo effettuato per tutti i flussi (Status Code 400):

```
{
  "status": "BAD_REQUEST",
  "code": "400",
  "timestamp": "16-04-2020 07:48:51",
  "message": "SCARTATO - XML della fattura non conforme alla linea 14 e colonna 33. Motivo dello scarto: cvc-complex-type.2.4.a: contenuto non valido che inizia con l'elemento \"CessionarioCommittente\". È previsto un elemento \"{CedentePrestatore}\".",
  "subErrors": null
}
```

Esempio risposta mancato superamento controlli SDI, controlli effettuati solo per i flussi SDIPA, SDIPR, SELFINV (Status Code 400):

```
{
  "status": "BAD_REQUEST",
  "code": "400",
  "timestamp": "16-04-2020 07:54:45",
  "message": "Fattura non valida. Errori: [00400 - Sulla riga di dettaglio con aliquota IVA pari a zero deve essere presente il campo Natura]",
  "subErrors": null
}
```

Esempio risposta upload XML FPA12 ma indicando nel payload flusso SDIPR:

```
{
  "status": "BAD_REQUEST",
  "code": "400",
  "timestamp": "16-04-2020 08:01:05",
  "message": "Non è possibile utilizzare il formato trasmissione FPR12 quando il flusso è SDIPA",
  "subErrors": null
}
```

Controllo XML

[POST/v2/invoices/validate](#)

Questo endpoint permette di effettuare la validazione di un xml sia sulla base dello schema xsd sia sulla base dei controlli formali effettuati dal Sistema di Interscambio.

Payload di richiesta

```
{
  "content": "xml della fattura in base64"
}
```

Payload di risposta

```
{
  "sdiErrors": [
    "string"
  ],
  "xsdError": "string"
}
```

```
}
```

Se l'xml è corretto e valido rispetto ai controlli formali Sdi si otterrà in risposta un json in cui l'array `sdiErrors` è vuoto ed il campo `xsdError` è null.

```
{
  "sdiErrors": [],
  "xsdError": null
}
```

Ovviamente in caso fallisca la validazione xsd non sarà possibile effettuare i controlli formali, si avrà pertanto una risposta in cui sarà valorizzato solo il campo `xsdError`. Esempio:

```
{
  "sdiErrors": null,
  "xsdError": "XML della fattura non conforme alla linea 5 e colonna 28. Motivo dello
scarto: cvc-complex-type.2.4.a: contenuto non valido che inizia con l'elemento
\"CedentePrestatore\". È previsto un elemento \"{DatiTrasmissione}\"."
}
```

Di seguito un payload ottenuto in risposta in seguito al controllo di un xml valido rispetto allo schema xsd ma formalmente errato:

```
{
  "sdiErrors": [
    "00423 - Il valore del campo PrezzoTotale non risulta calcolato secondo le regole
definite nelle specifiche tecniche",
    "00422 - Il valore del campo ImponibileImporto non risulta calcolato secondo le regole
definite nelle specifiche tecniche",
    "00421 - Il valore del campo Imposta non risulta calcolato secondo le regole definite
nelle specifiche tecniche"
  ],
  "xsdError": null
}
```

Simulazione invio a SDI

SDI Mock

In ambiente di test è disponibile un Mock dei servizi SDI che permette la simulazione dei differenti tipi di notifiche.

le notifiche vengono consegnate dopo un determinato delay in caso di notifiche multiple, fra una notifica e l'altra vi è un attesa pari al NOTIFICATION_DELAY configurato sul server (ora a 3 minuti). I tipi di notifiche supportate sono i seguenti :

- ACCETTATO
- ACCETTATO_CON_RITARDO
- RIFIUTATA
- SCARTATO
- NESSUNA_RISPOSTA
- NON_RECAPITATO

Il tipo deve essere specificato in un tag <Causale>?</Causale> all'interno della fattura

Se non viene specificato nulla nel tag causale la fattura viene considerata come in stato ACCETTATO

Forward della fattura

Se si vuole che il mock una volta ricevuta la fattura faccia il forward e la consegna quindi all'azienda indicata come cessionario occorre specificare i seguenti codici destinatari:

Per le SDIPR:

AGY0FWD

Per le SDIPA

UFPRT0

Occorre specificare nel cessionario in fattura piva o cf del cessionario presente su digital

Forward della fattura con assegnazione su ufficio specifico

utilizzare:

AGY0FW + una lettera. ad ES. AGY0FWA , AGY0FWB

per fare in modo che la fattura venga assegnata all'ufficio corretto va poi associato l'ufficio al codice destinatario da portale.

(Attenzione. non è più fattibile assegnare un codice a piacere da portale)

Digital

UFFICI

Hai più sedi operative? Gestisci qui i tuoi uffici/sedi ottimizzando il tuo account TS Digital.

Ricerca per descrizione o indirizzo e premi invio

Codice ufficio fatturazione elettronica	Data Creazione	Azioni
	26/05/2021	

Caricamento notifiche

E' possibile allegare notifiche ad una fattura in due modi:

Contemporaneamente al caricamento fattura

<https://b2bwrite-api-test.agyo.io/api/swagger-ui/index.html?configUrl=/api/v3/api-docs/swagger-config#/InvoicesV2/uploadInvoice>

POST /v2/invoices

Il filename deve essere strutturato seguendo il naming SDI

```
...  
  
"attachments": [  
  {  
    "fileName": "string",  
    "content": "base64"  
  }  
],  
  
...
```

Dopo il caricamento fattura

<https://b2bwrite-api-test.agyo.io/api/swagger-ui/index.html?configUrl=/api/v3/api-docs/swagger-config#/InvoicesV2/addNotification>

PATCH /v2/invoices/{hubId}/addNotification

Attenzione che il metodo HTTP è PATCH in quando va a modificare un entità già presente

Il filename deve essere strutturato seguendo il naming SDI


```
{
  "notifications": [
    {
      "fileName": "string",
      "content": "base64"
    }
  ]
}
```

Conservazione Digitale (CCT)

Al caricamento di una fattura con già le notifiche allegate il processo segue il flusso normale, per quanto riguarda le notifiche aggiunte in un secondo momento verrà prevista la generazione automatica di un evento per tentarne la conservazione

ItemOverview

Questa API consente di verificare rapidamente la presenza di fatture ricevute, inviate e scartate per una o più aziende a partire da un dato timestamp.

Il comportamento è identico alla vecchia `companyOverview`, è stata aggiornata la logica relativa alla paginazione (adeguandola al `continuationToken`) e la nomenclatura dei campi per renderla coerente con `itemId`. In query string è possibile indicare la dimensione della pagina attraverso il parametro `size` (valore massimo ammesso 1000) è consigliato usare valori più bassi come 100 e sfruttare il meccanismo del `continuation token` per recuperare le pagine successive. questo permette migliori performance e riduce il rischio di rallentare il db.

I valori di *`lastTimestampActive`* e *`lastTimestampPassive`* nel payload di richiesta devono essere espressi in millisecondi e non possono essere antecedenti al 01/01/2019 nè successive al timestamp corrente.

Funzionamento

Aggiornamento fatture passive

Quando TSDigital assegna una fattura ricevuta dal Sistema di Interscambio viene prodotto un evento che, **quando verrà processato**, aggiornerà le informazioni del campo *`lastTimestampPassive`* con il timestamp di assegnazione della fattura.

Esempio:

- TSDigital assegna la fattura all'azienda con codice fiscale ABCDEF e `itemId` `adcf9209-fe9b-4b37-9b68-8238da8fd6bb` alle 15:30 del 15/04/2020 (UTC), corrispondente al timestamp `1586964600000`
- Viene prodotto l'evento
- TSDigital processa l'evento precedente alle 17:00 del 15/04/2020 (UTC), corrispondente al timestamp `1587081600000`
- Il valore di *`lastTimestampPassive`* per l'`itemId` `adcf9209-fe9b-4b37-9b68-8238da8fd6bb` sarà dunque `1586964600000` (orario di assegnazione della fattura) e non `1587081600000` (orario in cui è stato processato l'evento)

Aggiornamento fatture attive/scartate

Vale quanto detto in precedenza, con l'unica differenza che l'evento di aggiornamento dell'`ItemOverview` viene prodotto a partire da un cambio stato della fattura e produrrà un cambiamento per i valori di *`lastTimestampActive`* e *`lastTimestampRejected`*.

Funzionamento delle API

Per semplicità prendiamo in esame un payload di richiesta relativo ad un solo item così definito:

```
{
  "items": [
    {
      "itemId": "ABCDEF",
      "lastTimestampActive": 1583143700735, // 2 Marzo 2020, 10:08:20 UTC
      "lastTimestampPassive": 1583244701735 // 3 Marzo 2020, 14:11:51 UTC
    }
  ]
}
```

L'itemId *ABCDEF* su database contiene queste informazioni:

```
{
  "itemId": "ABCDEF",
  "taxId": "XYZ",
  "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC
  "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC
  "lastTimestampRejected": 1584378112720, // 16 Marzo 2020, 17:01:52 UTC
  [...]
}
```

I flag saranno calcolati così:

```
boolean active = lastTimestampActive > lastTimestampActiveFromRequest;
boolean rejected = lastTimestampRejected > lastTimestampActiveFromRequest;
boolean passive = lastTimestampPassive > lastTimestampPassiveFromRequest;

// Utilizzando i dati di esempio:

// 6 Aprile 2020, 07:59:01 UTC > 2 Marzo 2020, 10:08:20 UTC => true
boolean active = 1586159941109 > 1583143700735

// 16 Marzo 2020, 17:01:52 UTC > 2 Marzo 2020, 10:08:20 UTC => true
boolean rejected = 1584378112720 > 1583143700735

// 3 Marzo 2020, 16:00:00 UTC > 3 Marzo 2020, 14:11:51 UTC => true
boolean passive = 1583251200000 > 1583244701735
```

e produrranno quindi il seguente payload di risposta:

```
[...]
{
  "active": true,
  "passive": true,
  "rejected": true,
  "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC
  "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC
  "lastTimestampRejected": 1584378112720, // 16 Marzo 2020, 17:01:52 UTC
  "itemId": "ABCDEF",
  "taxId": "XYZ"
}
[...]
```

Utilizzo

Ad esclusione della prima volta in cui si utilizza l'API per garantire il corretto comportamento è **fondamentale** utilizzare come valori di *lastTimestampActive* e *lastTimestampPassive* in request quelli ottenuti dalla response precedente di *company overview*.

Effettuare una prima chiamata all'API utilizzando un valore valido a piacere per i campi *lastTimestampActive* e *lastTimestampPassive*, ad esempio 1577836800000 (1 Gennaio 2020, 00:00:00 UTC).

Ottenuta la prima response è opportuno memorizzare i valori di *lastTimestampActive* e *lastTimestampPassive* che verranno utilizzati per la successiva chiamata di *company Overview*.

Riprendendo l'esempio di prima ed ipotizzando che la situazione delle fatture sia rimasta invariata per l'itemId ABCDEF la nuova chiamata avrà come payload di request:

```
{
  "items": [
    {
      "itemId": "ABCDEF",
      "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC ottenuto dalla
      response precedente
      "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC ottenuto dalla
      response precedente
    }
  ]
}
```

e come response

```
[...]
{
  "active": false,
  "passive": false,
  "rejected": false,
  "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC
  "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC
  "lastTimestampRejected": 1584378112720, // 16 Marzo 2020, 17:01:52 UTC
  "itemId": "ABCDEF",
  "taxId": "XYZ"
}
[...]
```

Request/Response

Esempio di payload di richiesta con una post a <https://b2bread-api-test.agyo.io/api/v2/items/overview?size=1>

```
{
  "items": [
    {
      "itemId": "ABCDEF00G00H000L",
      "lastTimestampActive": 1583143700735,
      "lastTimestampPassive": 1583143700735
    },
    {
      "itemId": "6f51950a-c982-40b7-99a3-405dd1328684",
      "lastTimestampActive": 1583143700735,
      "lastTimestampPassive": 1583143700735
    }
  ]
}
```

Esempio di payload di risposta:

```
{
  "_embedded": {
    "itemOverviewList": [
      {
```

```

        "active": true,
        "passive": true,
        "rejected": true,
        "lastTimestampActive": 1586159941109,
        "lastTimestampPassive": 1583243470858,
        "lastTimestampRejected": 1584378112720,
        "itemId": "ABCDEF00G00H000L",
        "taxId": "ABCDEF00G00H000L"
    }
}
],
"_links": {
    "first": {
        "href": "https://b2bread-api-
test.agyo.io/api/v2/items/overview?size=1&sort=agyoCompanyId,desc"
    },
    "self": {
        "href": "https://b2bread-api-
test.agyo.io/api/v2/items/overview?size=1&sort=agyoCompanyId,desc"
    },
    "next": {
        "href": "https://b2bread-api-
test.agyo.io/api/v2/items/overview?continuationToken=33dbhiloNf1U8m9EPLLkKceN2NROiLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrY02zsulG3tD05ipcarnPv6gjQk6xqnd0x_BShZwfqTmItxsAlz&size=1
&sort=agyoCompanyId,desc"
    }
},
"page": {
    "size": 1,
    "hasNext": true,
    "continuationToken":
"33dbhiloNf1U8m9EPLLkKceN2NROiLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrY02zsulG3tD05ip
carnPv6gjQk6xqnd0x_BShZwfqTmItxsAlz"
}
}

```

Nel payload di richiesta sono presenti due item ed in queryString è stata richiesta la pagina di un elemento ci aspettiamo dunque la presenza di ulteriori pagine, come confermato dal payload di risposta (`page.hasNext=true`). Possiamo accedere alla pagina successiva utilizzando il link presente in `_links.next` (non costruire il link a mano ma utilizzare quello ottenuto in risposta). In questo caso effettueremo una post a `https://b2bread-api-test.agyo.io/api/v2/items/overview?continuationToken=33dbhiloNf1U8m9EPLLkKceN2NROiLWgzLN`

h6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrYO2zsuLG3tDO5ipcarnPv6gjQk6xqnd0x_BShZwfqTmltxsAlz&size=1&sort=agyoCompanyId,desc

Payload di risposta:

```
{
  "_embedded": {
    "itemOverviewList": [
      {
        "active": true,
        "passive": false,
        "rejected": false,
        "lastTimestampActive": 1583243700735,
        "lastTimestampPassive": 1563982802310,
        "itemId": "6f51950a-c982-40b7-99a3-405dd1328684",
        "taxId": "03200510166"
      }
    ]
  },
  "_links": {
    "self": {
      "href": "https://b2bread-api-test.agyo.io/api/v2/items/overview?continuationToken=33dbhiloNf1U8m9EPLLkKceN2NR0iLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrYO2zsuLG3tDO5ipcarnPv6gjQk6xqnd0x_BShZwfqTmltxsAlz&size=1&sort=agyoCompanyId,desc"
    }
  },
  "page": {
    "size": 1,
    "hasNext": false,
    "continuationToken": null
  }
}
```

Dal payload di risposta notiamo che non ci sono ulteriori pagine (page.haseNext=false).

Lettura Fatture

Le chiamate per la lettura delle fatture si dividono in due step.

Questo non vuol dire però che sono complementari, ovvero, le chiamate di lettura fatture (step 2) possono essere effettuate anche senza lo step 1.

Perché è importante però utilizzare lo step 1?

Lo step 1, ovvero la chiamata di [ItemOverview](#), è importante perché ci consente di verificare rapidamente se ci sono delle nuove fatture.

Questo permette di non sovraccaricare i nostri server perché se chiamando l'ItemOverview (step 1) ci restituisce che non ci sono delle nuove fatture è inutile procedere con lo step 2.

Dopo aver effettuato la chiamata di ItemOverview possiamo procedere a chiamare le api di lettura per ogni item che ha fatture.

Lista Fatture

[GET/v2/invoices](#)

Parametri da impostare:

- **ownerId:** da popolare con l'identificativo dell'azienda (UUID/CF/P.iva) Se l'azienda ha più di un ufficio e si vogliono recuperare le fatture di tutti gli uffici è possibile aggiungere al valore di ownerId il suffisso **-ALL**.
- **active:** da popolare con *true* o *false*. True se si vuole ottenere la lista delle fatture attive. False se si vuole ottenere la lista delle fatture passive;
- **lastTimestampFrom:** da popolare con la data espressa in millisecondi. Ad eccezione della prima volta che può essere popolato con un data a piacere le volte successive deve essere popolato con il *"lastTimestamp"* della fattura più recente ricevuta dalla response.

Come impostare la chiamata:

Eseguo la prima chiamata con data a piacere, ad esempio 24/01/2022

Dalla response ottengo tutte le fatture che hanno subito aggiornamenti successivi alla data impostata.

Le fatture vengono restituite dalla più recente alla meno recente.

Il giorno seguente ho bisogno di sapere se ci sono altri aggiornamenti sulle fatture?

Quello che bisogna fare è fare è impostare nella chiamata il timestamp dell'ultima fattura ottenuta dalla response precedente.

Esempio:

24/01/2022 ore 11:00 timestamp: 1643022047000

Response:

```
"...  
"currentStatusName": "INVIATO",  
"lastTimestamp": 1643047247000,  
"active": true,  
..."
```

25/01/2022 ore 11:00 timestamp: 1643047247000

In questo modo dalla response otterremo solo le fatture che hanno subito un aggiornamento e non nuovamente tutta la lista delle fatture.

Il comportamento ideale da seguire quindi è sincronizzare le fatture tramite questa chiamata e poi impostare delle query da parte vostra per ottenere specifici risultati.

E' possibile trovare la stessa fattura con *timestamp* diversi?

Sì è possibile perché il *timestamp* di una fattura viene modificato ogniqualvolta essa subisce un aggiornamento.

Continuation Token

Il "continuationToken" ci permette di vedere, qualora siano presenti, le altre fatture ottenute dalla response.

Il "size" della chiamata è impostato di default a 20 in modo da ottenere dalla response non più di 20 record (fatture).

Nel caso in cui siano presenti più di 20 fatture, in fondo alla response, troveremo due tag che ne danno conferma e sono "hasNext" popolato con "true" e "continuationToken" popolato con una stringa.

Esempio:

```
},  
  "page": {  
    "size": 20,  
    "hasNext": true,
```

```
"continuationToken":  
"6g5TtcULxhoHlBd9DlT0mnTlqtnDa5Hz8zlimF7R1ahgTkS2f0YE1z3HM162btsWr8iV0xi2Pbd02nHD7XlwFX3Wtt2cA  
5iTxLJ07-zpwUXV1WJ9HAd9ko3t22M3tjjc"  
}
```

Quindi per visualizzare le altre fatture presenti nella response non dobbiamo far altro che aggiungere alla chiamata il parametro “continuationToken” e popolarlo con la stringa indicata.

Esempio di chiamata:

```
curl --location --request GET 'https://b2bread-api-  
test.agyo.io/api/v2/invoices?ownerId=bf01b67b-86c6-4695-b8c3-  
2135aa0b8d6c&active=true&lastTimestampFrom=1546335945000' \  
--header 'User-Agent: Postman' \  
--header 'X-App-Name: VRxyz' \  
--header 'X-App-Version: 1.0' \  
--header 'Content-Type: application/json' \  
--header 'X-Request-ID: cuid(codice univoco)' \  
--header 'X-Correlation-ID: cuid(codice univoco)' \  
--header 'X-Item-ID: UUID (identificativo azienda)' \  
--header 'X-User-ID: ID Chiave Tecnica' \  
--header 'Accept-Language: it-IT' \  
--header 'Authorization: Bearer '
```

Dettaglio Fattura

[GET/v2/invoices/{hubId}](#)

Esempio chiamata:

```
curl --location --request GET 'https://b2bread-api-test.agyo.io/api/v2/invoices/{hubId}' \  
--header 'accept: application/hal+json' \  
--header 'User-Agent: Postman' \  
--header 'X-App-Name: VRxyz' \  
--header 'X-App-Version: 1.0' \  
--header 'X-Request-ID: cuid (codice univoco)' \  
--header 'X-Correlation-ID: cuid (codice univoco)' \  
--header 'X-Item-ID: Identificativo azienda' \  
--header 'X-User-ID: ID Chiave Tecnica' \  
--header 'Authorization: Bearer '
```

Download Fattura

[GET/v2/invoices/{hubId}/download](#)

Permette il download della fattura così come ha raggiunto l'ultimo stato. Questo significa che il download di una fattura inviata a SDI scaricherà una fattura firmata XADES (a meno che l'azienda abbia il flag di invio senza firma) con terzo intermediario TS (a meno che la fattura non sia già stata firmata in upload)

Mentre per la fattura ricevuta da SDI si riceverà una fattura non firmata, quindi in semplice formato xml. Se la fattura fa parte di un lotto si riceve il solo xml della fattura a cui si fa riferimento in request in quanto essa viene scorporata dal lotto per facilità d'uso.

Per i flussi che non passano per SDI si riceve l'xml della fattura relativa all'ultimo stato, quindi se previsto l'embedding degli allegati conterrà anche quelli.

Download del file in PDF: <https://b2bread-api-test.agyo.io/api/v2/invoices/{hubId}/download?format=PDF>

Download del file in ASSOSOFTWARE: <https://b2bread-api-test.agyo.io/api/v2/invoices/{hubId}/download?format=ASSOSW>

Download del file in XML: <https://b2bread-api-test.agyo.io/api/v2/invoices/{hubId}/download?format=XML>

Esempio chiamata:

```
curl --location --request GET 'https://b2bread-api-test.agyo.io/api/v2/invoices/{hubId}/download?format=XML' \
--header 'accept: application/hal+json' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: VRxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: cuid (codice univoco)' \
--header 'X-Correlation-ID: cuid (codice univoco)' \
--header 'X-Item-ID: UUID (identificativo azienda)' \
--header 'X-User-ID: ID Chiave Tecnica' \
--header 'Authorization: Bearer '
```

[GET/v2/invoices/{hubId}/download/original](#)

Permette il download della fattura originale, quindi per una fattura inviata a SDI permetterà di scaricare la fattura così come è stata caricata dall'utente.

Mentre per le passive ricevute da SDI permette di scaricare la fattura originale ricevuta, firmata XADES o PADES (p7m).

Per i flussi che non passano per SDI si riceve l'xml caricato dall'utente senza alcuna modifica.

Se la fattura in request fa riferimento ad un lotto si riceverà il lotto intero, in quanto la fattura si trova all'interno di quel file originale.

Download Notifica SDI

[GET/v2/invoices/{hubId}/downloadAllMessages](#)

Esempio Chiamata:

```
curl --location --request GET 'https://b2bread-api-
test.agyo.io/api/v2/invoices/{hubId}/downloadAllMessages' \
--header 'accept: application/hal+json' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: VRxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: cuuid' \
--header 'X-Correlation-ID: uuid-v4' \
--header 'X-Item-ID: ID Azienda' \
--header 'X-User-ID: ID Chiave Tecnica' \
--header 'Authorization: Bearer '
```

Notifica di Scarto

[GET/v2/invoices/{hubId}/download](#)

Parametri:

- **messageId**: il valore da inserire in questo parametro è recuperabile dalla response della chiamata di *Dettaglio Fattura*.
Il tag di riferimento è "*notificationId*" e si trova nella sezione "*messages*";
- **format**: scegliere il tipo di formato della notifica (Esempio "XML").

Esempio Chiamata:

```
curl --location --request GET 'https://b2bread-api-
test.agyo.io/api/v2/invoices/{hubId}/download?messageId=Ivwy1pPrTtV1xA&format=XML' \
--header 'Authorization: Bearer ' \
--header 'X-App-Name: VRxyz' \
--header 'X-App-Version: 1.0' \
--header 'User-Agent: Postman' \
--header 'X-Request-ID: cuuid' \
--header 'X-Correlation-ID: uuid-v4' \
--header 'X-Item-ID: ID Azienda' \
--header 'X-User-ID: ID Chiave Tecnica' \
--header 'Accept-Language: it-IT' \
--header 'Content-Type: application/json' \
--header 'Cookie:
visid_incap_2770748=A9AiR++qR4+3fodrMwxP8h4nWGIAAAAAQUIPAAAAABJFK5guyJfBL1cz/JUufW5;
visid_incap_2773288=C84tkoe0SrSMtGsu/hERxgKbb2IAAAAAQUIPAAAAAABCra5YhrEW739veNMLkYa'
```

Download Metadata

[GET/v2/invoices/{hubId}/download/metadata](#)

Esempio Chiamata:

```
curl --location --request GET 'https://b2bread-api-
test.agyo.io/api/v2/invoices/{hubId}/download/metadata' \
--header 'Authorization: Bearer ' \
--header 'accept: application/hal+json' \
--header 'User-Agent: Postman' \
--header 'X-App-Name: VRxyz' \
--header 'X-App-Version: 1.0' \
--header 'X-Request-ID: cuuid' \
--header 'X-Correlation-ID: uuid-v4' \
--header 'X-Item-ID: ID Azienda' \
--header 'X-User-ID: ID Chiave Tecnica' \
```

ItemId (UUID)

Le API v2 di lettura e scrittura possono essere utilizzate con l'itemId da 36 (o 40 nel caso sia un ufficio) caratteri. Per garantire la retrocompatibilità e non pregiudicare il funzionamento delle API v1 l'itemId verrà convertito in maniera trasparente da TSDigital nel vecchio identificativo.

Chiariamo con un esempio.

L'azienda XYZ ha come vecchio identificativo AAABBB86C13D205E, nuovo identificativo c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1

Payload di upload fattura

```
{
  "transmitterId": "c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1",
  "senderId": "c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1",
  [...]
}
```

Se su API v1 otterrò un errore, su API v2 invece andrà a buon fine (ovviamente a patto di avere la necessaria autorizzazione ad operare sull'azienda XYZ).

La fattura verrà assegnata da TSDigital utilizzando il vecchio identificativo AAABBB86C13D205E

Effettuando una lettura alle API v1 con parametro senderId=c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1 non otterrò nessun risultato.

Effettuando invece una lettura alle API v2 con parametro ownerId=c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1 il payload di risposta sarà:

```
{
  "_embedded": {
    "invoiceList": [
      {
        "transmitterId": "AAABBB86C13D205E",
        "senderId": "AAABBB86C13D205E",
        [...]
      }
    ]
  }
}
```

Identico comportamento si otterrà interrogando le API v2 con parametro
ownerId=AAABBB86C13D205E

Il comportamento sarà questo fino a quando tutti i client non saranno tutti aggiornati per utilizzare l'itemId, a quel punto verranno spente le API SOAP e le API v1. A quel punto procederemo ad effettuare un aggiornamento del database pertanto interrogando le api v2 con parametro ownerId=AAABBB86C13D205E non otterrò nessun risultato, mentre utilizzando il nuovo itemId ownerId=c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1 il payload sarà:

```
{
  "_embedded": {
    "invoiceList": [
      {
        "transmitterId": "c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1",
        "senderId": "c77f35bf-fbca-4111-8d92-8a9e3ef9f1f1",
        [...]
      }
    ]
  }
}
```

Per sender, transmitter ed in generale qualsiasi campo trattato ad oggi come un codice fiscale vale quanto detto per l'hubId. Non effettuare nessuna logica ma trattare il dato come una stringa di lunghezza variabile

Migrazione da API REST v1 a v2

Lettura Fatture

Differenze con le attuali V1

Le API v2 di lettura differiscono dalle v1 relativamente alla paginazione. Nella nuova versione infatti non è più presente il conteggio degli elementi totali e del numero di pagine ma un booleano che indica se è presente una pagina successiva.

V1 (deprecate)

```
"page": {  
  "size": 20,  
  "totalElements": 111,  
  "totalPages": 6,  
  "number": 0  
}
```

V2

```
"page": {  
  "size": 20,  
  "hasNext": true,  
  "continuationToken": "abc...xyz"  
}
```

Per ottenere la pagina successiva è sufficiente richiamare la url fornita nella risposta nell'oggetto `_links`:

```
"_links": {  
  "first": {  
    "href": "https://b2bread-api-test.agyo.io/api/v2/invoices?ownerId=ABCDEFGHI&size=20&sort=lastTimestamp,desc"
```



```
    },
    "self": {
      "href": "https://b2bread-api-
test.agyo.io/api/v2/invoices?ownerId=ABCDEFGHI&size=20&sort=lastTimestamp,desc"
    },
    "next": {
      "href": "https://b2bread-api-
test.agyo.io/api/v2/invoices?ownerId=ABCDEFGHI&continuationToken=5WvMRcmA_8WJtFSiemEregGggF3et
SD4QqPv1aSDXnwfjLKGZK0wq13sy5f19NQ8799jMkJveAlTa0mC9H4Pn3Wtt2cA5iTyce_5YgEik8cNNEvvUwfz43t22M
3tjjc&size=20&sort=lastTimestamp,desc"
    }
  }
}
```

Come è possibile osservare esaminando questi link, nella chiamata viene aggiunto il parametro relativo al continuationToken. Questo parametro consentirà al servizio di lettura di riprendere la lettura nel punto in cui si è arrivati nella lettura precedente

Un'ulteriore differenza consiste nell'obbligatorietà del campo OwnerId

Questo campo identifica il proprietario della fattura. Confrontandolo con le V1 rappresenta il sender per la ricerca delle fatture attive e il recipient per la ricerca delle passive.

Questo campo è quindi un riferimento all'azienda con la quale si sta operando. Ed è anch'esso un AgyoltemId.

Flussi

Descrizione dei flussi di fatturazione

Flusso SDIPR

Flusso Attivo

Descrizione

Il nome file viene generato dal backend di Agyo per le fatture NON firmate (il parametro in ingresso viene quindi ignorato), viene invece preso il nome file così com'è in caso di fatture già firmate, con l'unico vincolo che non contenga la piva di TS altrimenti creerebbe conflitti con la generazione dei progressivi.

Si accettano file firmati XAdES(.xml) e CAdES (.p7m)

Per poter effettuare l'invio il sender deve esistere su ts-digital e deve corrispondere a quello indicato in fattura come cedente. Il sender può inviare anche se non in stato "Certificato". Il sender deve essere VALIDATO.

In questo flusso è possibile anche caricare fatture semplificate FSM10

Servizi

Per quanto riguarda il flusso attivo per poter inviare una fattura a SDI è necessario possedere il servizio [SDI](#).

é possibile ottenere il servizio per estensione mediante una connessione.

Verrà sempre utilizzato prima un eventuale pacchetto del sender e se non presente quello del transmitter

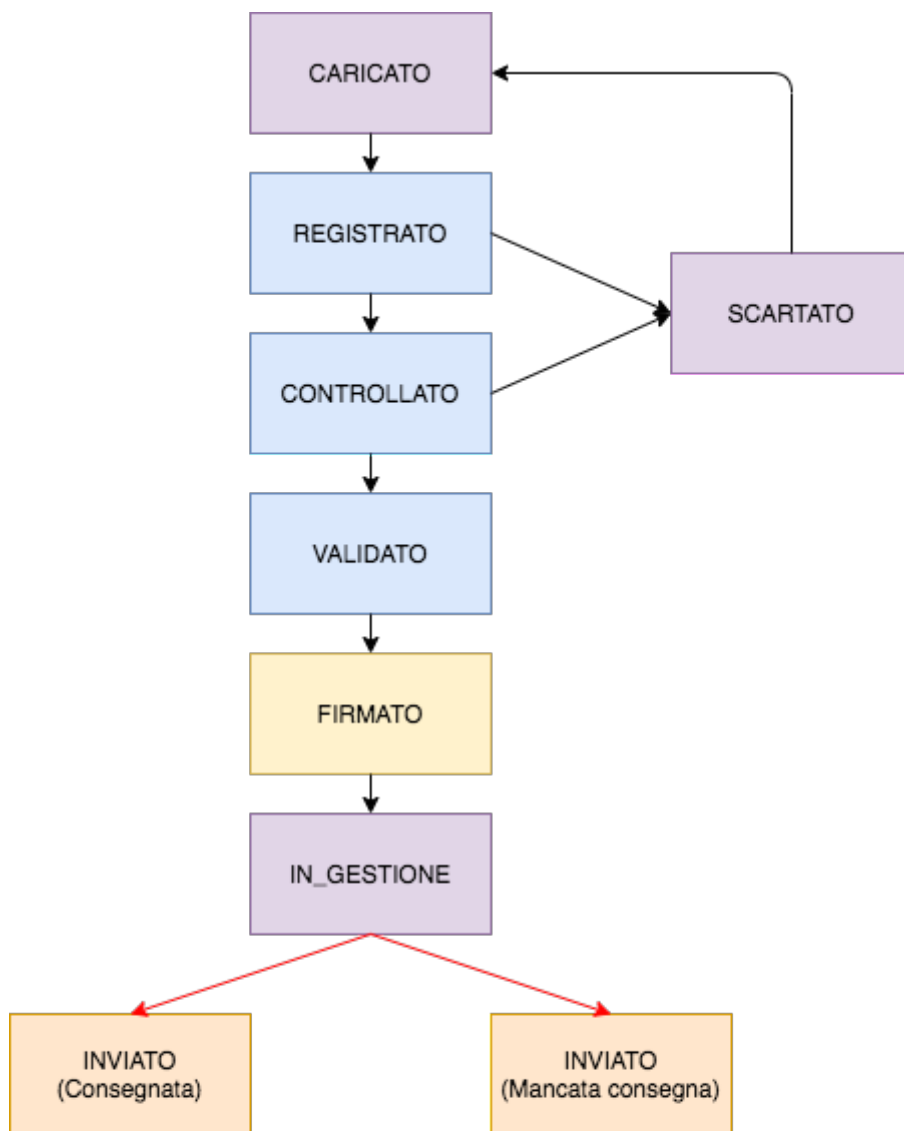
In caso il transmitter differisca dal sender viene verificata che ci sia una connessione se il pacchetto utilizzato è quello del transmitter.

Se il pacchetto utilizzato è quello del transmitter ed è necessario apporre firma TeamSystem viene verificato che la connessione sia Validata.

Motivi di scarto:

- Mittente non dichiarato
- Mittente non trovato in anagrafica
- Mittente non attivo
- Mittente non validato
- Owner dell'ufficio non trovato
- Owner dell'ufficio non attivo
- Owner dell'ufficio non validato
- Servizio SDI non attivo
- La fattura supera la dimensione massima di 5 MB
- [Lista di errori di validazione xml]
- Xml della fattura non conforme
- Errore nella gestione degli allegati
- La fattura non è formalmente corretta
- Estensione file non valida. Estensioni ammesse: .xml, .xml.p7m
- File con transmitter X e progressivo Y già inviato a SdI
- Nome file non valido
- Il sender non corrisponde al CedentePrestatore indicato in fattura
- Il flusso specifico X non coincide con il formato della fattura Y
- Impossibile determinare l'encoding della fattura
- Impossibile inviare una fattura firmata con encoding diverso da UTF-8

Rappresentazione degli stati



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Stato opzionale: si verifica solo in alcune condizioni

Freccia rossa: indica notifica da SDi

Flusso Passivo

Descrizione

Il flusso passivo rappresenta la ricezione di una fattura SDIPR da parte di SDI.

Una volta processata la fattura viene verrà assegnata di default all'ufficio principale (-000). in sua mancanza all'azienda. In caso si utilizzino i gov code all'ufficio configurato.

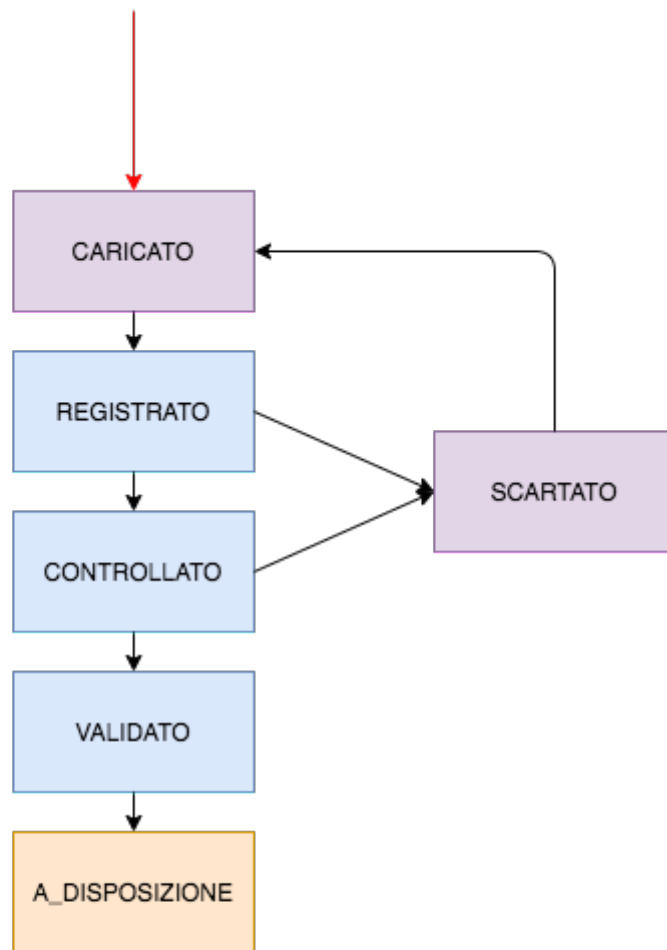
Servizi

Per poter ricevere è necessario avere il servizio [ERICEVI](#) arrivo. In caso di esaurimento del pacchetto le fatture verranno mantenute per 120 giorni ma non saranno assegnate alle aziende.

Sarà possibile riassegnare le fatture solo dopo aver rinnovato il pacchetto.

Al momento la riassegnazione deve esser richiesta all'assistenza. è in sviluppo uno strumento per automatizzare questo processo

Rappresentazione degli stati



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Stato opzionale: si verifica solo in alcune condizioni

Freccia rossa: indica notifica da SDi

Flusso SDIPA

Flusso Attivo

Descrizione

Il nome file viene generato dal backend di Agyo per le fatture NON firmate (il parametro in ingresso viene quindi ignorato), viene invece preso il nome file così com'è in caso di fatture già firmate, con l'unico vincolo che non contenga la piva di TS altrimenti creerebbe conflitti con la generazione dei progressivi.

Si accettano file firmati XAdES(.xml) e CAdES (.p7m)

Per poter effettuare l'invio il sender deve esistere su ts-digital e deve corrispondere a quello indicato in fattura come cedente. Il sender può inviare anche se non in stato "Certificato". Il sender deve essere VALIDATO

Servizi

Per quanto riguarda il flusso arrivo poter inviare una fattura a SDI è necessario possedere il servizio [SDI](#).

Per quanto riguarda il flusso arrivo poter inviare una fattura a SDI è necessario possedere il servizio [SDI](#).

é possibile ottenere il servizio per estensione mediante una connessione.

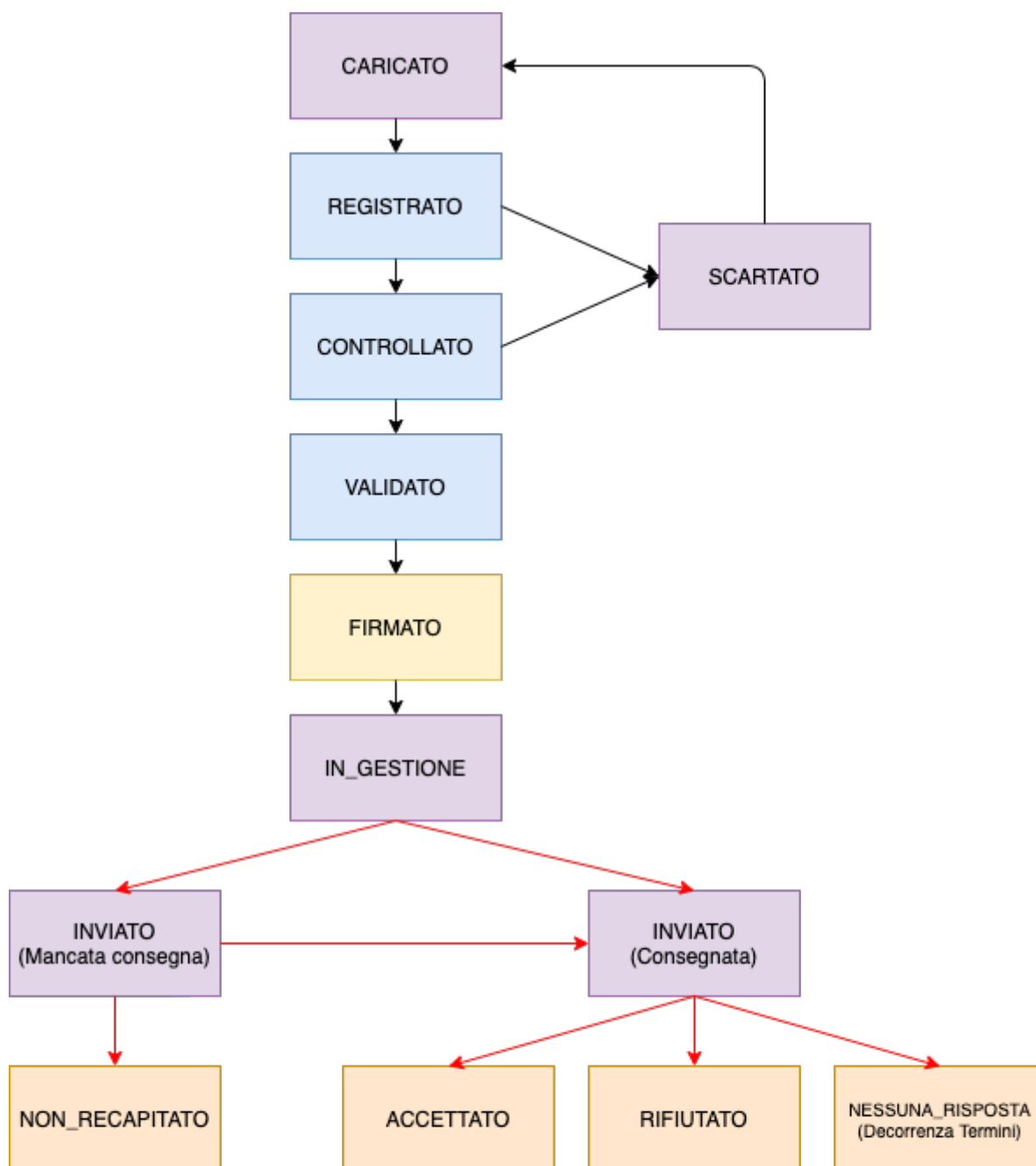
Verrà sempre utilizzato prima un eventuale pacchetto del sender e se non presente quello del transmitter

In caso il transmitter differisca dal sender viene verificata che ci sia una connessione se il pacchetto utilizzato è quello del transmitter. in caso contrario non viene verificata la connessione.

Motivi di scarto

- Mittente non dichiarato
- Mittente non trovato in anagrafica
- Mittente non attivo
- Mittente non validato
- Owner dell'ufficio non trovato
- Owner dell'ufficio non attivo
- Owner dell'ufficio non validato
- Servizio SDI non attivo
- Il destinatario deve essere codice IPA o destinatario SDI valido
- Destinatario non trovato nell'elenco dei codici IPA
- La fattura supera la dimensione massima di 5 MB
- [Lista di errori di validazione xml]
- Xml della fattura non conforme
- Errore nella gestione degli allegati
- La fattura non è formalmente corretta
- Estensione file non valida. Estensioni ammesse: .xml, .xml.p7m
- File con transmitter X e progressivo Y già inviato a SdI
- Nome file non valido
- Il sender non corrisponde al CedentePrestatore indicato in fattura
- Il flusso specifico X non coincide con il formato della fattura Y
- Impossibile determinare l'encoding della fattura
- Impossibile inviare una fattura firmata con encoding diverso da UTF-8

Rappresentazione degli stati



Stato finale : La fattura non può più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fatture

Stato opzionale: si verifica solo in alcune condizioni

Freccia rossa: indica notifica da SDI

Flusso Passivo

Descrizione

Il flusso passivo rappresenta la ricezione di una fattura SDIPR da parte di SDI.

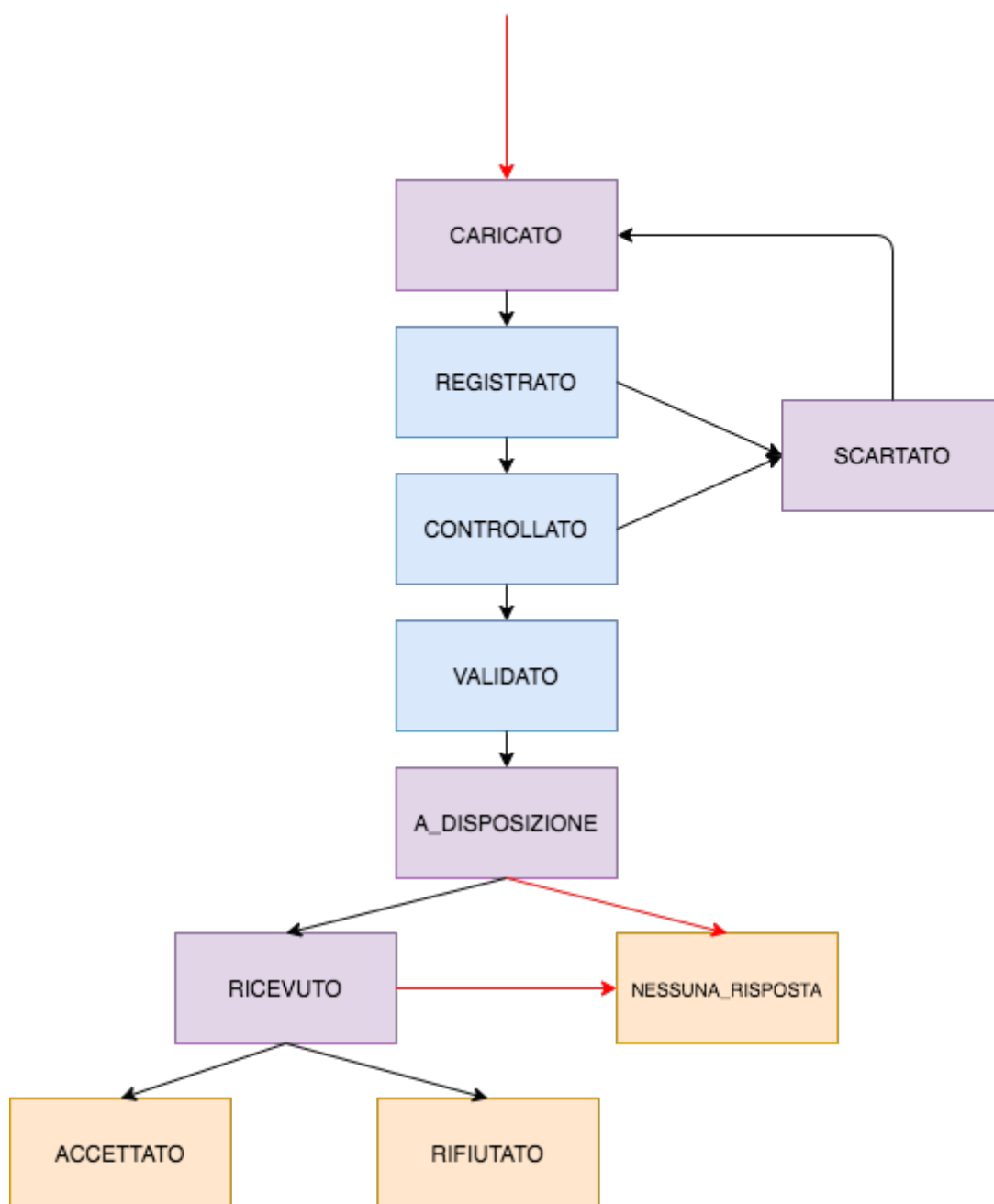
Una volta processata la fattura viene verrà assegnata di default all'ufficio principale (-000). in sua mancanza all'azienda. In caso si utilizzino i gov code all'ufficio configurato.

Servizi

Per poter ricevere non è necessario avere il servizio [ERICEVI](#) ma per poter accedere ai dettagli e scaricarla è obbligatorio.

Questo comportamento sta per variare con l'introduzione del flag di readonly sui pacchetti

Rappresentazione degli stati



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Stato opzionale: si verifica solo in alcune condizioni

Freccia rossa: indica notifica da SDI

Flusso SELFINV

Descrizione

I nome file viene generato dai nostri sistemi

i campi senderId, recipientId e transmitterId sono campi tecnici mirati all'instradamento della fattura e non si riferiscono al contenuto della fattura. essendo un autofattura. il sender e recipient sono per definizione gli stessi (se differiscono la fattura viene scartata).

Vengono firmate sole le fatture destinate alle pubbliche amministrazioni

Non è possibile inviare fatture già firmate

Servizi

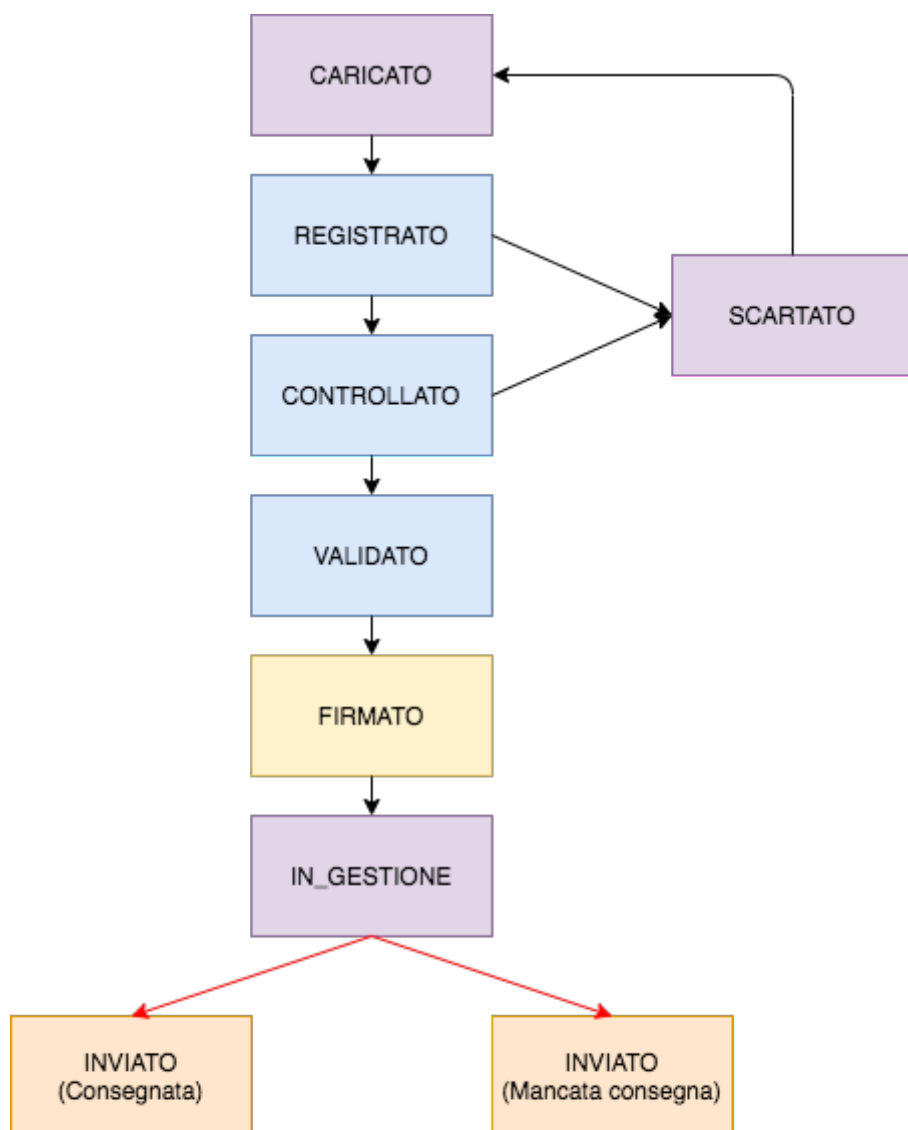
Per quanto riguarda il flusso arrivo poter inviare una fattura a SDI è necessario possedere il servizio [SDI](#).

Motivi di scarto:

- Mittente non dichiarato
- Mittente non trovato in anagrafica
- Mittente non attivo
- Mittente non validato
- Owner dell'ufficio non trovato
- Owner dell'ufficio non attivo
- Owner dell'ufficio non validato
- Servizio SDI non attivo
- La fattura supera la dimensione massima di 5 MB
- [Lista di errori di validazione xml]
- Xml della fattura non conforme
- Errore nella gestione degli allegati

- La fattura non è formalmente corretta
- Estensione file non valida. Estensioni ammesse: .xml, .xml.p7m
- File con transmitter X e progressivo Y già inviato a Sdl
- Nome file non valido
- Il sender non corrisponde al CedentePrestatore indicato in fattura
- Il flusso specifico X non coincide con il formato della fattura Y
- Impossibile determinare l'encoding della fattura
- Impossibile inviare una fattura firmata con encoding diverso da UTF-8

Rappresentazione degli stati



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Stato opzionale: si verifica solo in alcune condizioni

Freccia rossa: indica notifica da SDi

Flusso SELFSEND

Dettagli

Per poter eseguire le logiche di SELFSEND relative alla fattura occorre specificare un campo in extradata : '**aziendaDiRiferimento**'. Questo campo deve contenere il riferimento anagrafico [ItemId](#) dell'azienda di riferimento e può essere un ufficio.

Nel caso si operi come studio essa sarà necessaria per capire a quale azienda gestita dallo studio fa riferimento la fattura che si sta caricando. In questo caso verrà verificato che vi sia una connessione di tipo VALIDATO fra l'azienda di riferimento il transmitterId per appurare che il transmitter abbia i diritti ad operare su tale azienda.

L'azienda di riferimento coinciderà con il transmitterId nel caso si operi come azienda.

In particolare se l'azienda di riferimento coincide con il cedente prestatore la fattura risulterà come **attiva**. In caso coincidesse con il cessionario risulterà **passiva**. Questa comparazione viene effettuata prendendo in considerazione sia la piva che il cf. in questo modo viene gestito anche il caso particolare dei gruppi IVA (piva presente in fattura ma non esistente su Agyo)

Se l'azienda di riferimento non viene specificata oppure se non coincide ne con il cedente ne con il cessionario la fattura verrà scartata.

I parametri **transmitterId**, **senderId** e **recipientId** devono essere impostati con l'identificativo dell'azienda, o dello studio, che sta tendando l'invio. In caso uno dei parametri differisca si riceve un errore in fase di caricamento.

SenderId e **recipientId** verranno poi sovrascritti prendendoli dalla fattura, in particolare in caso di fattura passiva il senderId viene impostato come facciamo per le fatture passive ricevute da SDI (se c'è CF, altrimenti Piva). E' fondamentale che il sender(per le attive) o il recipient (per le passive) siano su Agyo.

In caso di scarto, **SenderId** e **recipientId**, e **ownerId** (**Sender e recipient non sarebbero più necessari**) non vengono modificati e rimangono uguali al **transmitterId** (per questioni di visibilità) viene inoltre messo il flag di active: false (per permetterne la visibilità al transmitter su console [scelta LV 08/02/2019]).

In caso di assegnazione l'ownerId viene ri-assegnato con l'itemId dell'azienda di riferimento. in questo modo la visibilità passa all'azienda di riferimento. Fintanto che vi sarà connessione fra i due,

potrà vederla ancora anche il transmitter (quindi lo studio che l'ha inviata).

Il metering assegnerà il consumo consapevole del fatto che la fattura è di tipo SELFSEND

Eventuali allegati non vengono mergiati nella fattura. (ci aspettiamo siano già allegati in quanto già in inviati a SDI)

Check duplicati

In caso venga trovato un duplicato la fattura verrà scartata[09].

Il check duplicato avviene verificando:

- DataDocumento
- NumeroDocumento
- OwnerId (viene cercata corrispondenza fra l'ownerId salvato su db con l'azienda di riferimento, l'ownerId se ufficio viene ANCHE valutato come se fosse solo azienda: ovvero senza -xxx)
- TipoDocumento
- ImportoTotaleDocumento se presente in entrambe le fatture che si stanno paragonando, in mancanza di uno o più di tale valore si verifica sul campo ImportoPagamento. Se non è presente alcun importo in nessuna delle due la fatture, oppure è presente solo uno dei due valori di riferimento, l'importo non parteciperà alla determinazione del duplicato.
- Cedente (check prima su IdFiscale iva e poi su CF)
- Cessionario (check prima su IdFiscale iva e poi su CF)

In particolare prima estraiamo da db per i primi 4 valori (OwnerId, data documento, numero documento, tipo documento) e successivamente valutiamo gli altri parametri

Il check viene fatto filtrando (non considerando) le fatture in stato SCARTATO, CARICATO o RIFIUTATO.

Caso particolare di Passaggio Interno

Si verifica nel caso in cui l'autofattura (cedente e cessionario sono lo stesso soggetto) e va registrata sia tra le fatture attive che tra quelle passive. Per gestire questo caso. nel momento in cui si genera un duplicato verrà verificata la coincidenza fra cedente e cessionario, se coincidono e la fattura duplicata è solo una verrà inserita una fattura passiva (la prima va automaticamente in attiva perchè viene prima fatto il check sul cedente). se le fatture duplicate sono invece 2 avverrà uno scarto per fattura duplicata[10].

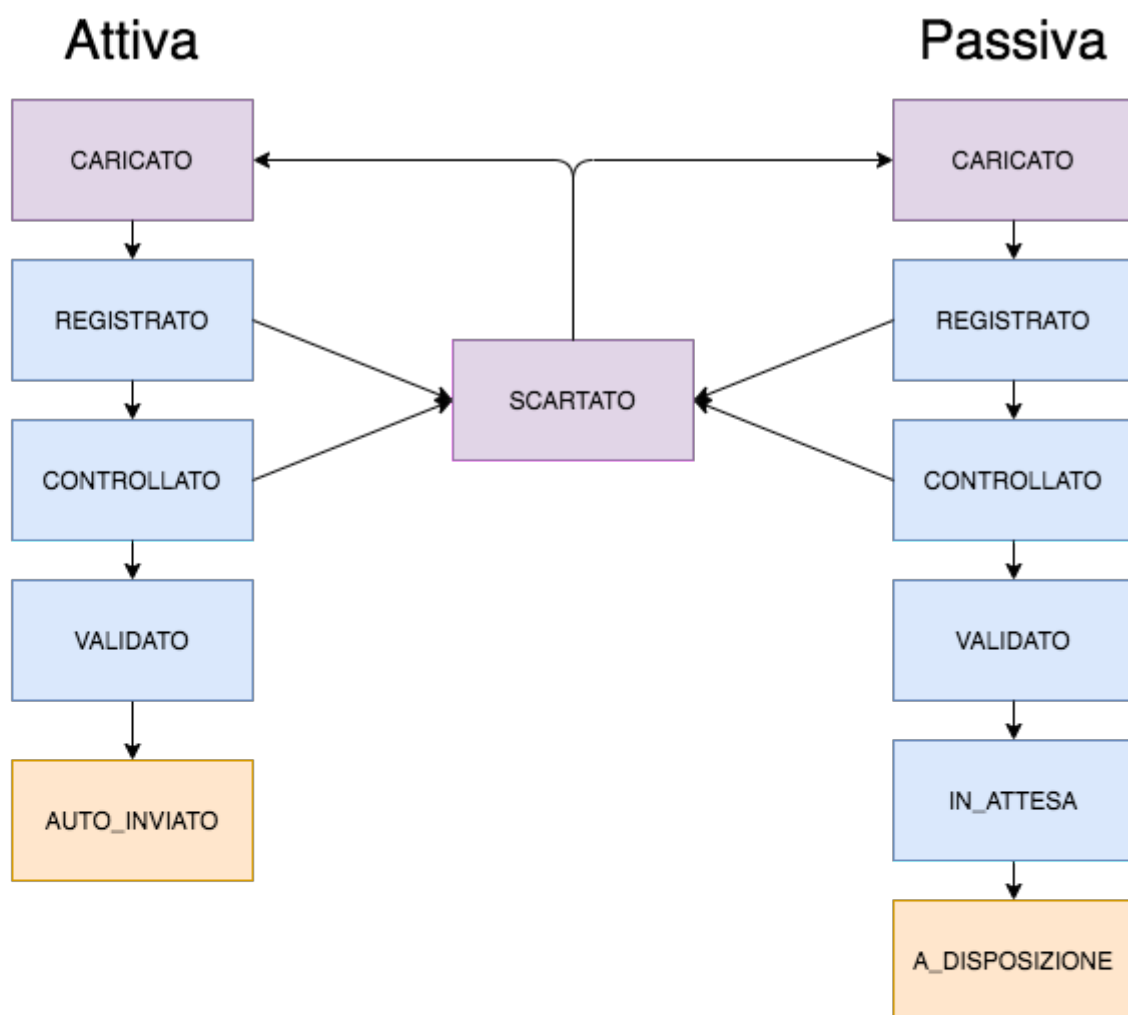
ATTENZIONE: Il check viene usando l'azienda di riferimento come parametro di ricerca perchè sarà quello che diventerà ownerId del documento al termine della procedura

Questo check si applica anche al suo sotto-flusso IMPEXP e OCR

Servizi

Il transmitter necessità del pacchetto [ERICEVI](#) (ottenibile tramite connessione o estensione)

Rappresentazione degli stati



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Stato opzionale: si verifica solo in alcune condizioni

Freccia rossa: indica notifica da SDi

Flusso IMPEXP

Descrizione

Il flusso IMPEXP è al momento veicolabile solo attraverso il flusso [CSV](#).

Ogni flusso IMPEXP può essere rappresentato nel flusso CSV da 2 diversi record group

Necessita del campo in extradata: **aziendaDiRiferimento**

Fare attenzione all'ordine dei campi

Record Group 1

```
id
senderId
recipientId
pivaCedente
cfCedente
ragSocialeCedente
pivaCessionario
cfCessionario
ragSocialeCessionario
dataDoc //formato ISO 8601:2004 -> 2020-01-25
numDoc
tipoDocumento
totale
```

Record Group 2

```
ImponibileImporto
aliquotaIva
imposta
esigibilitaIva
natura
```

Gli stati che può assumere sono

CARICATO

REGISTRATO

CONTROLLATO

AUTO_INVIATO

in caso di problemi lo stato sarà SCARTATO con relativa motivazione.

Flusso CSV

Descrizione

Il servizio permette di caricare un file CSV, trasformare ognuna delle sue righe o gruppo di righe (RecordGroup) in un xml e di sottomettere quest'ultimo ad un flusso pre-esistente. (al momento solo IMPEXP)

Per ogni flusso dovrà essere concordata la struttura del file CSV in modo da poter creare il corretto mapping fra RecordGroup e xml.

Una volta generato l'xml, ogni RecordGroup seguirà le regole per il flusso che è stato specificato in Extradata.

Al momento non è prevista la risottomissione.

Flussi attualmente veicolabili

Al momento è veicolabile solo il flusso IMPEXP.

Regole di struttura del file

L'unità base del csv è una riga chiamata anche Record. Ogni riga deve contenere un header obbligatorio alla prima colonna (indice 0): il recordType

è possibile raggruppare più record, il gruppo assumerà il nome di RecordGroup

Un RecordGroup permette di raggruppare più linee del CSV al fine di utilizzarle per generare uno stesso xml. Sono utili nel caso si voglia generare un xml con campi ripetuti.(vedi esempio)

Headers obbligatori del record group:

- id : riferimento univoco per riga all'interno del file (può essere anche un progressivo)

- senderId: id anagrafico di Agyo del mittente per la riga CSV (compilare con valore specifico del flusso utilizzato)
- recipientId: id anagrafico di Agyo del destinatario per la riga CSV (compilare con valore specifico del flusso utilizzato)

Questi campi obbligatori non hanno una posizione obbligatoria all'interno del file ma devono essere esplicitati in fase di condivisione del nuovo flusso che si sta definendo.

Come transmitter verrà utilizzato quello specificato per l'intero CSV.

Il file deve essere costruito secondo le regole definite dalla RFC4180 (<https://tools.ietf.org/html/rfc4180>) senza specificare il nome degli headers.

Il carattere di separazione è rappresentato dalla virgola : ,

Le informazioni inserite nei campi extradata della richiesta webService saranno utilizzate per ogni documento estratto dal csv. Bisognerà creare quindi il file tenendo in considerazione questa cosa.

Dati Invio

FlowType : CSV

I parametri **transmitterId**, **senderId** e **recipientId** devono essere impostati con l'identificativo dell'azienda, o dello studio, che sta tendando l'invio. In caso uno dei parametri differisca si riceve un errore in fase di caricamento.

Valori obbligatori in extraData:

- flowType: flusso da utilizzare per ogni recordGroup all'interno del CSV. ogni recordGroup del CSV è sottoposta allo stesso tipo di flusso.

Non è possibile avere CSV con all'interno tipi di flusso differenti.

Servizi

Il pacchetto necessario per l'invio dipende dal flusso dichiarato nell'extraData 'flowtype'

Lettura stato di elaborazione del CSV

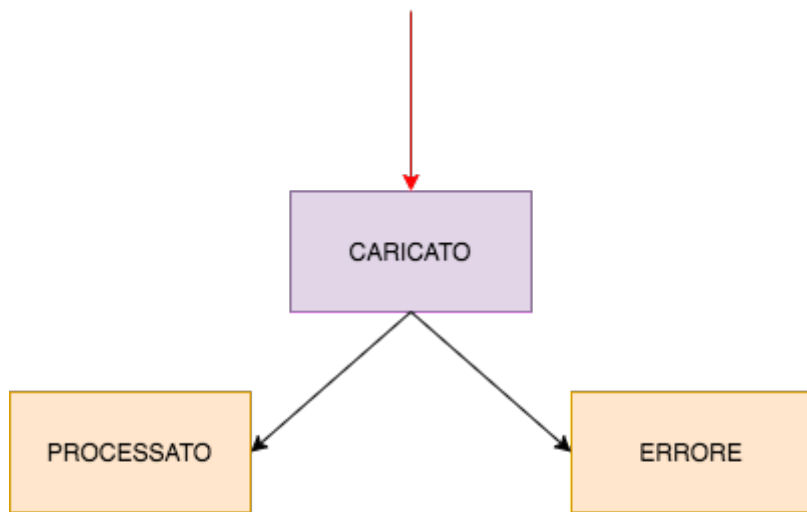
Verrà fornita un api di lettura dedicata che mediante l'id fornito in fase di caricamento fornirà informazioni relative al processamento del file stesso, se il documento ha terminato l'esecuzione verrà restituito per ogni riga presente:

- IdRecordGroup
- StatoRecordGroup
- descrizione
- hubId (potrà essere utilizzato per verificare lo stato del documento sulle api attuali di lettura di fatturazione)

Motivi di scarto del CSV

- Non hai specificato il flowtype in extradata (c'è il controllo ma in verità viene già bloccato un fase di upload)
- Flowtype in extradata non supportato per il parsing del csv.
- Dimensione massima superata.
- Errore nel parsing del csv, csv malformato.
- Id duplicato all'interno del csv

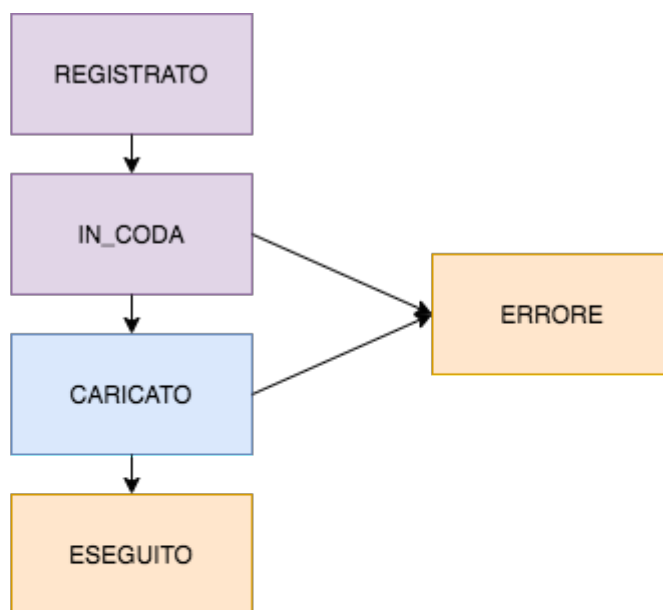
Rappresentazione degli stati del flusso CSV



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Rappresentazione degli stati del RecordGroupCsv



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Esempio

Il caricamento del seguente file CSV:

```
1,10,46546747471,46546747471,46546747471,46546747471,TEAMSYSTEM,0,MRCRRT72B14A145A,MERCANDELLI
ROBERTO,2019-01-01,1,TD01,162
2,160,N1,0,
2,2,N2,0,
```

Produce il seguente XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="FatturaPrivati_v1.2.xsl"?>
<p:FatturaElettronica xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:p="http://ivaservizi.agenziaentrate.gov.it/docs/xsd/fatture/v1.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" versione="FPR12">
  <FatturaElettronicaHeader>
    <DatiTrasmissione>
      <IdTrasmittente>
        <IdPaese>IT</IdPaese>
        <IdCodice>01641790702</IdCodice>
      </IdTrasmittente>
      <ProgressivoInvio>0</ProgressivoInvio>
      <FormatoTrasmissione>FPR12</FormatoTrasmissione>
    </DatiTrasmissione>
    <CedentePrestatore>
      <DatiAnagrafici>
        <IdFiscaleIVA>
          <IdPaese>IT</IdPaese>
          <IdCodice>46546747471</IdCodice>
        </IdFiscaleIVA>
        <CodiceFiscale>46546747471</CodiceFiscale>
        <Anagrafica>
          <Denominazione>TEAMSYSTEM</Denominazione>
        </Anagrafica>
        <RegimeFiscale>RF01</RegimeFiscale>
      </DatiAnagrafici>
    </CedentePrestatore>
    <CessionarioCommittente>
```

```
<DatiAnagrafici>
  <IdFiscaleIVA>
    <IdPaese>IT</IdPaese>
    <IdCodice>0</IdCodice>
  </IdFiscaleIVA>
  <CodiceFiscale>MRCRRT72B14A145A</CodiceFiscale>
  <Anagrafica>
    <Denominazione>MERCANDELLIROBERTO</Denominazione>
  </Anagrafica>
</DatiAnagrafici>
</CessionarioCommittente>
</FatturaElettronicaHeader>
<FatturaElettronicaBody>
  <DatiGenerali>
    <DatiGeneraliDocumento>
      <TipoDocumento>TD01</TipoDocumento>
      <Divisa>EUR</Divisa>
      <Data>2019-01-01</Data>
      <Numero>1</Numero>
      <ImportoTotaleDocumento>162</ImportoTotaleDocumento>
    </DatiGeneraliDocumento>
  </DatiGenerali>
  <DatiBeniServizi>
    <DatiRiepilogo>
      <Natura>N1</Natura>
      <ImponibileImporto>160</ImponibileImporto>
    </DatiRiepilogo>
    <DatiRiepilogo>
      <Natura>N2</Natura>
      <ImponibileImporto>2</ImponibileImporto>
    </DatiRiepilogo>
  </DatiBeniServizi>
</FatturaElettronicaBody>
</p:FatturaElettronica>
```


Processo di Conservazione (CCT)

Flussi conservabili

Sono conservabili le fatture:

- SDIPR (attive/passive)
- LOTTI SDIPR passivi
- SDIPR semplificate(attive/passive)
- SDIPA (attive/passive)
- SELFINV (attive)
- SELFSEND SDIPA (attive/passive)
- SELFSEND SDIPR (attive/passive)

Processo di conservazione

è un processo asincrono, il trigger viene scatenato dopo che la fattura viene elaborata e raggiunge il suo stato finale di successo. Se il flusso richiede l'invio a SDI il tentativo di invio in conservazione avverrà al cambio di stato tramite le notifiche SDI.

Fatture e notifiche vengono conservate separatamente ed ognuna ha un suo stato di conservazione.

Fatture e notifiche verranno automaticamente inviate in conservazione se il pagante della fattura ha il servizio di conservazione abilitato.

NON ANCORA IN PRODUZIONE: Solo da fronted ts-digital è al momento possibile disabilitare l'invio con granularità sul singolo flusso

Stati

IN_CARICO

Presente solo in alcuni casi per retrocompatibilità, significa che TS-Digital ha in carico l'invio in conservazione.

Non essendoci più la modalità con la quale è possibile dichiarare la volontà di conservare la fattura durante l'invio. questo stato andrà a sparire. (come dichiarato sopra, le fatture verranno conservate automaticamente se presente il servizio di conservazione correttamente attivato)

Ricevuto

Significa che il file ha iniziato il processo di conservazione. Lo stato non è uppercase per questioni di retrocompatibilità

Conservato in PdA

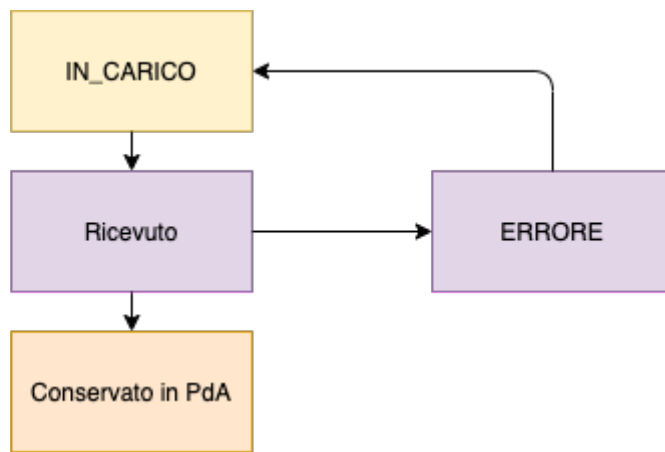
Significa che il file ha terminato il processo di conservazione digitale. Lo stato non è uppercase per questioni di retrocompatibilità

IN_CARICO

Presente solo in alcuni casi, significa che TS-Digital ha in carico l'invio in conservazione

ERRORE

Può verificarsi nel caso in cui durante il processo di conservazione si verificano errori che necessitano di un intervento dell'utente (problemi legati alla configurazione del servizio) oppure errori che possono essere risolti da un intervento di backoffice. Esistono ad oggi alcuni processi in mano al backoffice per risolvere alcune tipologie di ERRORE automaticamente.



Stato finale : La fattura non pò più assumere altri stati

Stato transitorio: la fattura potrà assumere altri stati

Stato tecnico: non si troverà mai come stato corrente delle fattura

Stato opzionale: si verifica solo in alcune condizioni