

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
}
```

Revision #22

Created 2021-06-04 10:04:59 UTC by Zittino

Updated 2024-04-09 13:16:03 UTC by Luca Zilembo