Stamp Integration APIs

Flow

When a third party application (called here POS) wants to use Stamp to send tax free invoices to Otello the flow consists of these steps:

1) The invoice is created by the merchant in the POS
2) The POS makes a call to Stamp APIs to get an access token (Get access token)
3) The POS makes a call to Stamp APIs and upload the invoice (Upload invoice draft)
4) The merchant will click a button **_Complete invoice using Stamp_** in the POS which will open Stamp Merchant Portal

5) The merchant will search the customer to invoice and save the tax free invoice in Stamp which will send it to Otello (all the invoices fields will be pre-filled with the data uploaded by the POS)

Fattura

Numero fattura: 09038/B
Data emissione: 10/25/2018

Federico Degrandis
ABCD45678

Federico Merchant
07647380968
via Pinco Pallo, 10/A
00100 Rome - IT

Codice	Descrizione	Qtà.	Prezzo IVA inclusa	IVA Aliquota	Netto
T888B20961	T-shirt PRADA Cotone Grigio	1	€747.00	22%	€612.30
T8B201261	T-shirt PRADA Cotone Rosso	1	€1,499.00	10%	€1,362.73

Subtotale	€1,975.03
Sconto netto	€0.00
IVA 0%	€0.00
Totale	€1,975.03

Tax free invoice status

After a tax free has been created, the customer has 3 months to exit EU and get his invoice approved. The status will be pending till the customer does not go through one of EU customs. The EU custom can approve or reject a tax free invoice.

These are the possible status for a tax free invoice:

- Pending
- Approved
- Rejected
- Cancelled

Use Get invoice status endpoint to get the status of a specific invoice.

Rejected

When a tax free invoice is rejected, we send an email to the merchant to notify him so he can create the debit note (for the VAT only). After the debit note has been created, the [Send debit note number](#) endpoint have to be called.

Cancelled

If a mistake has been made during the creation of the invoice or the customer wants to give back the goods he bought, the tax free invoice can be cancelled creating a credit note in the POS which will need to call [Cancel invoice](#) endpoint specifying the credit note number.

Open Stamp Merchant Portal

After the invoice has been upload, [Upload invoice draft](#) return a value called **DraftId** which need to be passed when the merchant wants to open the Merchant Portal and complete the tax free invoice. The ***Complete invoice using Stamp*** button should open the following link https://{MerchantPortalUrl}/draft-invoice/{draftId}

Data available

After a tax free invoice has been created, the POS can get the following data from Stamp APIs:

- Invoice & Invoice note pdf: use respectively [Get invoice pdf](#) or [Get invoice note pdf](#) endpoints. The **invoice pdf will contain all the data needed by the EU Custom authorities**.
- Customer data: use [Get customer data](#) endpoint
- Otello communication id (CRF0): this is the unique id which identifies the communication of the tax free invoice to Otello. Use [Get Otello communication id](#) endpoint to get it.

APIs

Get access token

Url: https://{MerchantApiUrl}/api/auth/sign
Method: POST

Headers

- Content-Type: application/json
- Accept: application/json

Input

```
{
    "Email": "user@email.com",
    "Password":"password"
}
```

Email and **Password** are provided by Stamp at the time of configuring the integration between the POS and Stamp for a specific merchant.

Output

```
{
    "AccessToken":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Ik9EazRRVFpGUXpVNU1FRkZNa0p
DTVRVek5EazVPRFV6T1VSR01VUkVORVk0UTBVME5UazNNZyJ9.eyJodHRwczovL2NsYWltcy
5zdGFtcGl0LmNvL3JvbGUiOiJtZXJjaGFudCIsImh0dHBzOi8vY2xhaW1zLnN0YW1waXQuY2
8vaWQiOiI0MThkZDNmNC0xMzdjLTQ0ZTMtODU1YS1hYzIyMWE2OGUzMTkiLCJodHRwczovL2
NsYWltcy5zdGFtcGl0LmNvL21lcmNoYW50SWQiOiJkNDlmNjNlMC01ZGVmLTRiMDQtOTVjOS
1iZmI4ODMxZmE2ZTMiLCJodHRwczovL2NsYWltcy5zdGFtcGl0LmNvL25hbWUiOiJBbW1pbm
1zdHJhemlvbmUiLCJodHRwczovL2NsYWltcy5zdGFtcGl0LmNvL3Blcm1pc3Npb25zIjoiYW
RtaW5pc3RyYXRvciIsImlzcyI6Imh0dHBzOi8vc3RhbXAtZGVtby5ldS5hdXRoMC5jb20vIi
wic3ViIjoiYXV0aDB8NTljYTQ3Yjg3YjM0MzUwNzQ5NGU2YzM5IiwiYXVkIjpbImh0dHBzOi
8vbWVyY2hhbnRhcGktZGVtby5zdGFtcGl0LmNvIiwiaHR0cHM6Ly9zdGFtcC1kZW1vLmV1Lm
F1dGgwLmNvbS91c2VyaW5mbyJdLCJpYXQiOjE1NDAzMTI0ODksImV4cCI6MTU0MDM5ODg4OS
wiYXpwIjoiSHlrdTRMc2tVd0RrSUo3V1h6OVNFeWNNZlp4VHV2S24iLCJzY29wZSI6Im9wZW
5pZCBwcm9maWxlIGVtYWlsIGFkZHJlc3MgcGhvbmUiLCJndHkiOiJwYXNzd29yZCJ9.z-NmD
_utz2WDFvyCz4uGs_6xZ9H9JNuzSYwXh90dXMeCsTUtH-lbaZWKJL9w5RXPzv6TUtLF_rx0w
UEKelrQz4krUlbmUAkcTYEIz5ehhVlV96umoLgmm-lQ_DmdBp7pxAiJ3oHpHuc3fQD-Kr07r
m3gQoUxphkR1Jmv9hHgTIKRrA4jr5xV5CXBlMR3U2FAQrATyig9D4CQnYyWerHATvse08csc
QugBdWxOKAIGM_OnqoiIa5qsEkfi818K_aQ1wuJvBAb_0kIIlejeCddVImRm3QXFDpALSt10
HUHYZbb0HGK8ucOMf5QZSUOuAP65gzjhXqzBaSRfjAauX-YAA",
    "ExpiresIn": 86400
}
```

Upload invoice draft

Url: https://{InvoicingApiUrl}/api/i/invoices/draft
Method: POST

Headers

- Authorization: Bearer {AccessToken}
- Content-Type: application/json

Input

```json
{
   "MerchantInvoiceId": "STAMP1234",
   "CreatedOn":"2018-12-25T10:30:00",
   "Items":[
      {
         "ItemCode": "GIA223",
         "Description": "Giacca in pelle verde",
         "Quantity": 1,
         "UnitPriceWithVat": 1500.99,
         "VatRate":22
      },
      {
         "ItemCode": null,
         "Description":"Camicia a quadri rossa",
         "Quantity":5,
         "UnitPriceWithVat":120.20,
         "VatRate":22
      }
   ],
   "Discount":30.66
}
```

- **ItemCode** can be null or empty.
- **Description** must be minimum 5 chars.
- Numbers are formatted using **. (dot)** instead of **, (comma)**.
- *CreatedOn* accepts the following format **dd/MM/yyyy hh:mm:ss**.
- **Discount** is the total discount to apply

Output

```json
{
   "DraftId": "464e2397-3662-4541-a8a9-a512a6803b6c",
   "InvoiceId":"de912414-114b-4426-bac2-df74143b78a9"
}
```

- **DraftId**: temporary id of the uploaded invoice
- **InvoiceId**: id of the invoice which will be created

Get invoice pdf

Url: https://{InvoicingApiUrl}/api/invoices/{invoiceId}/pdf
Method: GET

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/json

Input

- **InvoiceId**: use the field **InvoiceId** returned by Upload invoice draft call

Output

```
{
    "FileUri":
"https://stampstorageint.blob.core.windows.net/cus-invoices-37bf22ad-e70
7-4e8d-b51f-5cc1ec1cef23/Invoice
312-OTELLO.pdf?sv=2017-04-17&amp;sr=b&amp;sig=07OiKGD5vIP0D6CvM%2B77ZPaN
ENiYd%2Fy9FM0ApaL4%2BIw%3D&amp;st=2018-08-27T00%3A57%3A55Z&amp;se=2018-0
8-28T00%3A57%3A55Z&amp;sp=r"
}
```

- **FileUri**: the link from which the pdf can be downloaded

Get invoice note pdf

Url: https://{InvoicingApiUrl}/api/invoices/{invoiceId}/note/pdf
Method: GET

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/json

Input

- **InvoiceId**: use the field **InvoiceId** returned by Upload invoice draft call

Output

```
{
    "FileUri":
"https://stampstorageint.blob.core.windows.net/cus-invoices-37bf22ad-e70
7-4e8d-b51f-5cc1ec1cef23/Invoice
312-OTELLO.pdf?sv=2017-04-17&amp;sr=b&amp;sig=07OiKGD5vIP0D6CvM%2B77ZPaN
ENiYd%2Fy9FM0ApaL4%2BIw%3D&amp;st=2018-08-27T00%3A57%3A55Z&amp;se=2018-0
8-28T00%3A57%3A55Z&amp;sp=r"
}
```

- **FileUri**: the link from which the pdf can be downloaded

Get customer data

Url: https://{InvoicingApiUrl}/api/i/invoices/{invoiceId}/customer
Method: GET

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/xml

Input

- **InvoiceId**: use the field **InvoiceId** returned by Upload invoice draft call

Output

```
{
    "DateOfBirth": "1990-03-01T00:00:00",
    "DocumentType": 0,
    "IdentityDocumentIssuingCountryIso2Code": "CH",
    "Fullname": "Luca Rossi",
    "IdentityDocumentId": "YA12345678"
}
```

- **DateOfBirth**: date is formatted as **yyyy-MM-ddThh:mm:ss** (the time part will be always 00:00:00).
- **DocumentType**: the value can be **0** (Passport) or **1** (Identity Card).

Get invoice status

Url: https://{InvoicingApiUrl}/api/i/invoices/{invoiceId}/status
Method: GET

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/json

Input

- **InvoiceId**: use the field **InvoiceId** returned by [Upload invoice draft](#) call

Output

```
{
    "Status": 0
}
```

- **Status**:can be **0** (Pending), **1** (Approved), **2** (Rejected), **3** (Canceled)

Get Otello communication id

Url: https://{InvoicingApiUrl}/api/invoices/{invoiceId}/crf0
Method: GET

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/json

Input

- **InvoiceId**: use the field **InvoiceId** returned by [Upload invoice draft](#) call

Output

```
{
    "OtelloCommunicationId": "XXXXXXXXX"
}
```

Send debit note number

Url: https://{InvoicingApiUrl}/api/invoices/{invoiceId}/note/debit/number
Method: POST

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/json

Input

- **InvoiceId**: use the field **InvoiceId** returned by Upload invoice draft call

```
{
    "InvoiceNoteNumber": "..."
}
```

- **InvoiceNoteNumber**: the number of the debit note created by the merchant using the POS

Output

If the call is successful the endpoint will return HTTP status code 200 (OK) without any data.

Cancel invoice

Url: https://{InvoicingApiUrl}/api/invoices/{invoiceId}/cancel
Method: POST

Headers

- Authorization: Bearer {{AccessToken}}
- Accept: application/json

Input

- **InvoiceId**: use the field **InvoiceId** returned by Upload invoice draft call

```
{
    "InvoiceNoteNumber": "..."
}
```

- **InvoiceNoteNumber**: the number of the credit note created by the merchant using the POS

Output

If the call is successful the endpoint will return HTTP status code 200 (OK) without any data.

Error codes

If the invoice has not been created on Stamp side when calling Get invoice pdf, Get invoice note pdf, Get customer data or Get invoice status the system will return HTTP status code 400 with the following body:

```
{
    "ErrorCode": 101,
    "ErrorMessage": "Not found"
}
```