

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 uploads 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



The screenshot shows a web interface for creating a new invoice. At the top, the heading 'Nuova fattura' is displayed in a bold, dark blue font. Below it, the instruction 'Digita il numero di cellulare del cliente' is shown in a smaller, grey font. A text input field contains the example number 'ad es. +39 3481234567'. Below the input field is a link that says 'Trova prefisso'. A large, dark blue button with the text 'Continua →' is positioned below the link. At the bottom of the form, there is a horizontal line, followed by the text 'oppure usa' and a button labeled 'WeChat'.

- 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 invoice fields will be pre-filled with the data uploaded by the POS)

Fattura

NUMERO FATTURA ST-9999
DATA EMISSIONE 18/06/2020

AARON WELT

Passaporto AB12345
Paese documento Angola
Paese di residenza Andorra

Federico Merchant

07647380968
Piazza d'Aracoeli, 1
00100 - Roma

Codice	Descrizione	Qtà.	Prezzo IVA inclusa	IVA 0%	Netto
9	Mountain fox cushion	1	€599.99	€0.00	€491.80
Subtotale					€491.80
Sconto netto					€0.80
IVA 0%					€0.00
Totale					€491.00

Tax free invoice status

After a tax free invoice has been created, the customer has 3 months to exit the EU and have his invoice approved by customs. The invoice status will be *pending* until the customer goes through one of the EU customs. The EU customs 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 obtain 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 is created, the [Send debit note number](#) endpoint has to be called.

Cancelled

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

Open Stamp Merchant Portal

After the invoice has been uploaded using the [Upload invoice draft](#) endpoint, the endpoint returns a value called **DraftId** which needs to be passed when the merchant wants to open Stamp's Merchant Portal to complete the tax free invoice. The **DraftId** value should be passed in the link. 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 Customs 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 obtain 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":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Ik9EazRRVFpGUxpvVNU1FRkZNa0pDTVRVek5EazVPRFV6T1VSR01VUkVORVksUTBVME5UazNNZyJ9.eyJodHRwczovL2NsYWltcy5zdGFtcGl0LmNvL3JvbGU0iJtZXJjaGFudCIsImh0dHBzOi8vY2xhaW1zLnN0YW1waXQuY28vaWQiOiI0MThkZDNmNC0xMzdjLTQ0ZTMtODU1YS1hYzIyMWE2OGUzMTkiLCJodHRwczovL2NsYWltcy5zdGFtcGl0LmNvL21lcmNoYW50SWQiOiJkNDlmNjNlMC01ZGVmLTRiMDQtOTVjOS1iZmI4ODMxZmE2ZTMiLCJodHRwczovL2NsYWltcy5zdGFtcGl0LmNvL25hbWUiOiJBbW1pbm1zdHJhemlvmU0iLCJodHRwczovL2NsYWltcy5zdGFtcGl0LmNvL3Blcm1pc3Npb25zIjoiaWwRtaW5pc3RyYXRvciIsIm1zcyI6Imh0dHBzOi8vc3RhbnR0dGVtby5ldS5hdXRoMC5jb20vIiwic3ViIjoiaXV0aDB8NTljYTQ3Yjg3YjM0MzUwNzQ5NGU2YzM5IiwiaXVkiJpbImh0dHBzOi8vbWV5Y2hhbnRhcGktZGVtby5zdGFtcGl0LmNvIiwiaHR0cHM6Ly9zdGFtcC1kZW1vLmV1LmF1dGgwLmNvbS91c2Vyaw5mbyJdLCJpYXQiOiJlNDZMTi0ODksImV4cCI6MTU0MDM5ODg4OSwiYXpwIjoiaSh1rdTRMc2tVd0RrSU03V1h6OVNFeWNNZlpl4VHV2S24iLCJzY29wZSI6Im9wZW5pZCBwcm9maWxlIGVtYWlsIGFkZHI1c3MgcGhvbmUiLCJndHkiOiJwYXNzd29yZCJ9.z-NmD_utz2WDFvyCz4uGs_6xZ9H9JNuzSYwXh90dXMeCsTUtH-lbaZWKJL9w5RXPzv6TUtLF_rx0wUEKelrQz4krU1bmUAkcTYEIZ5ehhV1V96umoLgmm-lQ_DmdBp7pxAiJ3oHpHuc3fQD-Kr07rm3gQoUxphkR1Jmv9hHgTIKRrA4jr5xV5CXB1MR3U2FAQrATyig9D4CQnYyWerHATvse08cscQugBdWx0KAIGM_OnqoiIa5qsEkfi818K_aQ1wuJvBAb_0kIIlejeCddVImRm3QXFDpALSt10HUHYZbb0HGK8ucOMf5QZSU0uAP65gzjhxqzBaSRfjAauX-YAA",
  "ExpiresIn": 86400
}
```

Upload invoice draft

Url: <https://{InvoicingApiUrl}/api/i/invoices/draft>

Method: POST

Headers

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

- Accept: application/json

Input

```
{
  "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 characters.
- 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

```
{
  "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-e707-4e8d-b51f-5cc1ec1cef23/Invoice312-OTELLO.pdf?sv=2017-04-17&sr=b&sig=070iKGD5vIP0D6CvM%2B77ZPaNENiYd%2Fy9FM0ApaL4%2BIw%3D&st=2018-08-27T00%3A57%3A55Z&se=2018-08-28T00%3A57%3A55Z&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-e707-4e8d-b51f-5cc1ec1cef23/Invoice312-OTELLO.pdf?sv=2017-04-17&sr=b&sig=070iKGD5vIP0D6CvM%2B77ZPaNENiYd%2Fy9FM0ApaL4%2BIw%3D&st=2018-08-27T00%3A57%3A55Z&se=2018-08-28T00%3A57%3A55Z&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 vat note number

Url: `https://{InvoicingApiUrl}/api/i/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.

Send credit goods note number

Url: `https://{InvoicingApiUrl}/api/i/invoices/{invoiceId}/note/creditgoods/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 credit goods 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.

Send credit vat note number

Url: `https://{InvoicingApiUrl}/api/i/invoices/{invoiceId}/note/creditvat/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 credit vat 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"  
}
```