

# MANUALE SVILUPPO API



[www.cloudfinance.it](http://www.cloudfinance.it)

**Versione 1.5.1**

**Aggiornato il 20/01/2026**

## Sommario

Storia documento	3
Informazioni generali	4
Attivare un account sandbox	5
Inviare una fattura non firmata	6
Inviare una fattura firmata dal cliente	7
Elencare le fatture inviate	8
Elencare le fatture ricevute	10
Elencare le notifiche ricevute	11
Visualizzare i dettagli di una fattura	13
Stati di una fattura	16
Visualizzare i dettagli di una notifica	17
<b>Ricevere aggiornamenti delle fatture tramite webhook</b>	<b>19</b>
<b>Lotti pacchetto api disponibili</b>	<b>20</b>
<b>Importare un documento</b>	<b>22</b>
<b>Elencare i documenti</b>	<b>23</b>

## Storia documento

Versione	Data	Modifiche
1.5.2	01/02/2026	Aggiunta documentazione per gestione documenti.
1.5.1	20/01/2026	Documentato parametro skipSend per l'invio della fattura.
1.5.0	09/05/2024	Modificato indirizzo ambiente sandbox.
1.4.0	27/09/2022	Aggiunto "invoiceSdild" (codice SDI) ai dettagli della fattura.
1.3.1	21/10/2020	Aggiunti nuovi parametri per la visualizzazione dettagli fattura.
1.3.0	14/10/2020	Aggiunta documentazione webhook e richiesta lotti pacchetto api.
1.2.0	05/02/2019	Aggiunto ora di trasmissione file a SdI, sender, receiver e dati documento.
1.1.1	10/01/2019	Aggiunta sezione "Stati di una fattura".
1.1.0	21/12/2018	Corretti refusi. Aggiunto campo invoiceFileXmlBase64 ai dettagli della fattura.
1.0.0	27/11/2018	Prima release.

## Informazioni generali

Le API sono esposte dal seguente **endpoint**:

SANDBOX	<a href="https://sandbox.cloudfinance.it/api/v1/">https://sandbox.cloudfinance.it/api/v1/</a>
PRODUZIONE Free Invoice	<a href="https://webapp.freeinvoice.it/api/v1/">https://webapp.freeinvoice.it/api/v1/</a>
PRODUZIONE BPilot	<a href="https://webaccounting.cloudfinance.it/api/v1/">https://webaccounting.cloudfinance.it/api/v1/</a>

NB: Tutte le chiamate devono essere effettuate utilizzando il protocollo HTTPS. Non è possibile invocare le API utilizzando HTTP.

Nello specifico, tutte le chiamate avranno la seguente forma:

**METHOD** <https://endpoint/api/v1/comando>

Dove **comando** definisce l'operazione che si vuole effettuare. Tutte le risposte restituiranno codice http 200 e saranno in formato JSON.

In caso di errore nella gestione della richiesta la risposta avrà la seguente forma:

```
{
  "errors": [
    {
      "code": "001",           // Identifica l'errore
      "message": "Api Key non valida", // Descrive l'errore
    }
    // Altri errori
  ]
}
```

Si noti che gli errori potrebbero essere più di uno.

Le API permettono di eseguire le seguenti operazioni:

- Inviare una fattura non firmata (la fattura sarà firmata dal nostro sistema);
- Inviare una fattura già firmata dal cliente;
- Elencare le fatture inviate;
- Elencare le fatture ricevute;
- Elencare le notifiche ricevute;
- Visualizzare i dettagli di una fattura;
- Visualizzare i dettagli di una notifica.

## Attivare un account sandbox

L'ambiente sandbox consente di simulare l'invio di fatture senza effettivamente trasmetterle al Sistema di Interscambio (SDI). Di seguito vengono forniti i passaggi per creare un account sandbox da utilizzare per testare le proprie applicazioni.

Per poter attivare un account sandbox è necessario:

1. Creare un account su <https://cloudfinance.it>
2. Inviare una mail riportando i seguenti dati:

<b>A:</b>	<a href="mailto:sandbox@cloudfinance.it">sandbox@cloudfinance.it</a>
<b>Oggetto:</b>	Attivazione account sandbox
<b>Contatto e-mail:</b>	Utilizzato in fase di registrazione su cloudfinance.it al passo 1.
<b>Recapito telefonico:</b>	Numero di telefono da contattare per eventuali problemi.

3. Entro 24 ore lavorative riceverai una mail di avvenuta attivazione o sarai contattato in caso di problemi. Al ricevimento della mail potrai accedere all'ambiente di sandbox esclusivamente da questo indirizzo:

<https://sandbox.cloudfinance.it/>

Dopo l'accesso, potrai generare le tue apiKey alla pagina:

<https://sandbox.cloudfinance.it/apiKey/list>

## Inviare una fattura non firmata

La chiamata che dovrai fare per inviare una fattura sarà la seguente:

POST <https://endpoint/api/v1/invoices/usend>

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **invoiceFileBase64**: contiene l'XML della fattura convertito in base 64;
- **signer**: (opzionale) identificativo dell'account di firma da utilizzare per firmare la fattura;
- **skipSend**: (opzionale): se *true* la fattura non sarà trasmessa al SDI.

Formato risposta valida:

```
{  
  "invoiceId": "123abc",  
  "timestamp": "2018-10-10T12:29:17.000000Z"  
}
```

Il campo `invoiceId` identifica univocamente la fattura sul sistema ed è utilizzabile, successivamente, per richiedere dettagli sulla fattura inviata.

Il campo `timestamp` identifica quando la fattura è stata aggiunta al sistema. Si noti che questo timestamp può differire dall'ora di invio o ricezione fattura disponibile nei dettagli della fattura.

Possibili errori:

Codice	Messaggio
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.
200	Formato fattura non valido.
201	Progressivo invio non valido.
202	Dati cedente prestatore errati.

 In sandbox

Le fatture trasmesse tramite questa chiamata non saranno inviate al sistema di interscambio. L'invio sarà solo simulato e si concluderà con la ricezione di una *fittizia* notifica di ricezione. Inoltre, le fatture trasmesse saranno inoltrate allo stesso account (partita IVA) d'invio come fattura di costo simulando la ricezione di una fattura.

## Inviare una fattura firmata dal cliente

La chiamata che dovrai fare per inviare una fattura sarà la seguente:

POST <https://endpoint/api/v1/invoices/ssend>

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **invoiceFileBase64**: contiene l'XML firmato (o formato binario) della fattura convertito in base 64;
- **method**: indica il metodo che è stato utilizzato per firmare il documento. I soli valori ammissibili sono "CADES-BES" e "XAdES-BES";
- **skipSend**: (opzionale): se *true* la fattura non sarà trasmessa al SDI.

Formato risposta valida:


```
{  
  "invoiceId": "123abc",  
  "timestamp": "2018-10-10T12:29:17.000000Z"  
}
```

Il campo `invoiceId` identifica univocamente la fattura sul sistema ed è utilizzabile, successivamente, per richiedere dettagli sulla fattura inviata.

Il campo `timestamp` identifica quando la fattura è stata aggiunta al sistema. Si noti che questo timestamp può differire dall'ora di invio o ricezione fattura disponibile nei dettagli della fattura.

Possibili errori:

Codice	Messaggio
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.
200	Formato fattura non valido.
201	Progressivo invio non valido.
202	Dati cedente prestatore errati.

 In sandbox

Le fatture trasmesse tramite questa chiamata non saranno inviate al sistema di interscambio. L'invio sarà solo simulato e si concluderà con la ricezione di una *fittizia* notifica di ricezione. Inoltre, le fatture trasmesse saranno inoltrate allo stesso account (partita IVA) d'invio come fattura di costo simulando la ricezione di una fattura.

## Elencare le fatture inviate

La chiamata che dovrai fare per elencare le fatture inviate:

```
GET https://endpoint/api/v1/invoices/sentlist
```

Questa richiesta permette di caricare la lista degli `invoiceId` delle fatture inviate in un certo intervallo di tempo ordinate cronologicamente.

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **page**: (default = 1) pagina da caricare;
- **withinAfter**: il formato deve essere YYYY-MM-DD HH:MM:SS. Nella lista saranno presenti *solo* le fatture caricate nel sistema ad un orario successivo o uguale a quello specificato;
- **withinBefore**: (opzionale) il formato deve essere YYYY-MM-DD HH:MM:SS. Nella lista saranno presenti *solo* le fatture caricate nel sistema ad un orario successivo o uguale a `withinAfter` e precedente o uguale a `withinBefore`.

Formato risposta valida:

```
{
  "data": [
    {
      "invoiceId": "124abc",
      "timestamp": "2018-10-10T12:29:17.000000Z",
    },
    {
      "invoiceId": "123abc",
      "timestamp": "2018-10-10T12:35:26.000000Z",
    }
    // Altri dati fino ad un massimo di 1000
  ],
  "meta": {
    "current_page": 1,
    "last_page": 3,
    "per_page": 15,
    "from": 1,
    "to": 10,
    "total": 35
  }
}
```

Il campo `invoiceId` identifica univocamente la fattura sul sistema ed è utilizzabile, successivamente, per richiedere dettagli sulla fattura inviata.

Il campo `timestamp` identifica quando la fattura è stata aggiunta al sistema. Si noti che questo timestamp può differire dall'ora di invio o ricezione fattura disponibile nei dettagli della fattura.

Si noti che la risposta conterrà fino ad un massimo di 1000 elementi con timestamp crescente.

Possibili errori:

<b>Codice</b>	<b>Messaggio</b>
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.

## Elencare le fatture ricevute

La chiamata che dovrai fare per elencare le fatture ricevute:

GET `https://endpoint/api/v1/invoices/receivedlist`

Questa richiesta permette di caricare la lista degli `invoiceId` delle fatture ricevute in un certo intervallo di tempo. I parametri della richiesta e della risposta è quello specificato nel paragrafo precedente.

! In sandbox

Per poter simulare la ricezione di una fattura basta [inviare una fattura utilizzando l'apposito endpoint](#).

## Elencare le notifiche ricevute

La chiamata che dovrai fare per elencare le notifiche ricevute:

GET `https://endpoint/api/v1/notifications`

Questa richiesta permette di caricare la lista delle notifiche ricevute in un certo intervallo di tempo.

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **page**: (default = 1) pagina da caricare;
- **withinAfter**: il formato deve essere YYYY-MM-DD HH:MM:SS. Nella lista saranno presenti *solo* le notifiche ricevute ad un orario successivo o uguale a quello specificato;
- **withinBefore**: (opzionale) il formato deve essere YYYY-MM-DD HH:MM:SS. Nella lista saranno presenti *solo* le notifiche ricevute ad un orario successivo o uguale a **withinAfter** e precedente o uguale a **withinBefore**.

Formato risposta valida:

```
{
  "data": [
    {
      "notificationId": "456def",
      "notificationKind": "RicevutaConsegna",
      "invoiceId": "123abc",
      "timestamp": "2018-01-01 13:10:00",
    },
    {
      "notificationId": "456def",
      "notificationKind": "RicevutaConsegna",
      "invoiceId": "123abc",
      "timestamp": "2018-11-26T10:52:17.000000Z",
    }
  ]
  // Altre notifiche fino ad un massimo di 1000
  "meta": {
    "current_page": 1,
    "last_page": 3,
    "per_page": 15,
    "from": 1,
    "to": 10,
    "total": 35
  }
}
```

Il campo `notificationId` identifica univocamente la notifica nel sistema ed è utilizzabile, successivamente, per richiedere dettagli sulla notifica.

Il campo `timestamp` identifica quando la notifica è stata aggiunta al sistema. Si noti che questo timestamp può differire dall'ora di invio o ricezione notifica disponibile nei dettagli della notifica.

Si noti che la risposta conterrà fino ad un massimo di 1000 elementi in ordine cronologico.

Possibili errori:

<b>Codice</b>	<b>Messaggio</b>
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.

## Visualizzare i dettagli di una fattura

La chiamata che dovrai fare per visualizzare i dettagli di una fattura:

```
GET https://endpoint/api/v1/invoices/invoiceId
```

Parametri url:

- **invoiceId**: identifica la fattura di cui si vuole visualizzare i dati.

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **withFile**: (valori ammessi: "true", "false", default: "true") se "true" nella risposta sarà presente il campo "invoiceFileBase64" contenente il file fattura firmato codificato in base64;
- **withFileXml**: (valori ammessi: "true", "false", default: "true") se "true" nella risposta sarà presente il campo "invoiceFileXmlBase64" contenente il file fattura xml non firmato codificato in base64;
- **withFilePdf**: (valori ammessi: "true", "false", default: "false") se "true" nella risposta sarà presente il campo "invoiceFilePdfBase64" contenente il file fattura in formato PDF codificato in base64;

Formato risposta valida:

```
{
  "data": {
    "invoiceId": 14,
    "invoiceKind": "ricavo",
    "invoiceType": "FPR12",
    "invoiceStatus": 8,
    "invoiceStatusName": "Consegnata",
    "invoiceFileName": "IT01234567890_00000.xml.p7m",
    "invoiceSdiId": "123456...000",
    "sender": {
      "description": "MARIO ROSSI",
      "countryCode": "IT",
      "vatCode": "01234567890",
      "fiscalCode": "01234567890"
    },
    "receiver": {
      "description": "LUIGI VERDI",
      "countryCode": "IT",
      "vatCode": "98765432109",
      "fiscalCode": null
    },
    "invoiceBody": {
```

```

        "documentType": "TD04",
        "date": "2019-02-05",
        "number": "1"
    },
    "invoiceSdiTransmissionDateTime": "2019-02-05T12:42:22.000000Z",
    "timestamp": "2019-02-05T13:24:59.000000Z",
    "invoiceFileBase64": "FA684CDAEC3D6E0A0...AA==",
    "invoiceFileXmlBase64": "AA94B8DEE8912AC781...BA4=",
    "invoiceFilePdfBase64": "AC78FBA8912678FF10...AA3=",
    "invoiceMetadataFileName": null,
    "invoiceMetadataFileBase64": null,
    "notifications": [
        {
            "notificationId": "456def",
            "notificationKind": "RicevutaConsegna",
            "invoiceId": "584agd",
            "timestamp": "2018-01-01 13:10:00"
        }
    ]
}

```

Dove:

- **invoiceKind**: assume il valore “costo” per le fatture ricevute, “ricavo” per le fatture inviate;
- **invoiceStatus**: indica lo stato della fattura;
- **invoiceFileBase64**: contenuto del file xml della fattura firmato come trasmesso/ricevuto all’Sdl in base 64;
- **invoiceFileXmlBase64**: contenuto del file xml della fattura privo di firma in base 64;
- **invoiceFileName**: è il nome del file xml come trasmesso/ricevuto all’Sdl;
- **metadatiFileBase64**: (solo per fatture di tipo costo) contenuto del file metadati come ricevuto dall’Sdl in base 64;
- **invoiceSdiId**: è il codice SDI come trasmesso/ricevuto all’Sdl;
- **invoiceSdiTransmissionDateTime**: (solo per fatture di tipo ricavo) se presente, indicata la data di trasmissione del file al Sdl.

Possibili errori:

Codice	Messaggio
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.

005	Parametro richiesta mancante.
006	Parametro richiesta errato.
300	Fattura non trovata.

Si noti che l'invio della fattura al Sistema di Interscambio dopo essere stata trasmessa tramite le API potrebbe non essere immediato. Durante questa fase di "elaborazione" i campi `invoiceFileName` e `invoiceFileBase64` assumeranno valore `null`.

<b>! In sandbox</b>
<p>La fattura non sarà firmata. Il campo <code>invoiceFileName</code> conterrà sempre <code>IT01589730629_BOZZA.xml</code> e il contenuto di <code>invoiceFileBase64</code> sarà sempre il file fattura come trasmesso dall'utente (privo di firma). Allo stesso modo il campo <code>invoiceMetadataFileName</code> conterrà sempre <code>IT01589730629_BOZZA_MT_001.xml</code> e <code>invoiceMetadataFileBase64</code> un file metadati con firma non vera.</p> <p>Al contrario, in produzione <code>invoiceFileName</code> conterrà il nome effettivo della fattura con estensione ".xml" per i file firmati in XAdES-BES o ".xml.p7m" per i file firmati in CAdES-BES. Il contenuto di <code>invoiceFileBase64</code> sarà sempre il file fattura firmato. Allo stesso modo il campo <code>invoiceMetadataFileName</code> conterrà sempre il nome del file metadati come ricevuto dal SdI e <code>invoiceMetadataFileBase64</code> un file metadati con firma firmato in XAdES-BES dal SdI.</p>

## Stati di una fattura

La fattura può assumere diversi stati indicati dai campi `invoiceStatus` e `invoiceStatusName`. I possibili valori sono i seguenti:

<code>invoiceStatus</code>	<code>invoiceStatusName</code>	Descrizione
1	Bozza	-
2	Verificata	-
3	Pronta per l'invio	-
4	Scartata	SdI ha risposto all'invio della fattura con una notifica di scarto. La fattura risulta NON emessa e può essere rinviata dopo le correzioni richieste.
5	Elaborazione	La fattura è in fase di invio al SdI. Stato temporaneo in attesa di notifiche da parte del SdI.
6	Non inviata	-
7	Inviata	La fattura è stata inviata al SdI ma non ha ancora ricevuto notifiche di risposta. Stato temporaneo in attesa di notifiche da parte del SdI.
8	Consegnata	La fattura è stata correttamente inviata al SdI e consegnata al canale del cessionario committente. La fattura risulta emessa.
9	Non consegnata	La fattura è stata correttamente inviata al SdI ma il SdI non è stato in grado di consegnarla al canale del cessionario committente. La fattura risulta emessa.
10	Esito SI	Solo fatture verso PA. La fattura è stata correttamente inviata al SdI e la PA ha accettato la fattura. La fattura risulta emessa.
11	Esito NO	Solo fatture verso PA. La fattura è stata correttamente inviata al SdI ma la PA non ha accettato la fattura. La fattura risulta NON emessa e può essere rinviata dopo le correzioni richieste.
12	Decorrenza termini	Solo fatture verso PA. La fattura è stata correttamente inviata al SdI. E' stata ricevuta una notifica di decorrenza termini dall'SdI che decreta l'impossibilità di ricevere ulteriori notifiche di esito da parte della PA. La fattura risulta emessa.

## Visualizzare i dettagli di una notifica

La chiamata che dovrai fare per visualizzare i dettagli di una fattura:

GET <https://endpoint/api/v1/notifications/notificationId>

Parametri url:

- **notificationId**: identifica la notifica di cui si vuole visualizzare i dati.

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;

Formato risposta valida:

```
{
  "data": {
    "notificationId": "456def",
    "invoiceId": "123abc",
    "fileName": "IT01234567890_00000_RC_001.xml",
    "fileBase64": "5hcm1vQ29tbWl0dGVudGU+PERhdGlB...FncmFmaWNpPj",
    "messageKind": "RicevutaConsegna",
    "timestamp": "2018-01-01 13:00:01"
  }
}
```

Dove:

- **notificationId**: identifica univocamente la notifica (coincide con quello inviato nella richiesta);
- **invoiceId**: identifica univocamente la fattura a cui si riferisce la notifica;
- **fileName**: nome del file notifica come trasmesso dal sistema di interscambio;
- **fileBase64**: contenuto della notifica codificata in base 64;
- **messageKind**: dichiara la tipologia del messaggio;
- **timestamp**: l'ora in cui è stata aggiunta la notifica al sistema Free Invoice.

Possibili errori:

Codice	Messaggio
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.
400	Notifica non trovata.

! In sandbox

Il campo `fileName` conterrà sempre `IT01589730629_BOZZA_RC_001.xml` e il contenuto di `notificationFileBase64` sarà sempre un file notifica con firma non vera.

## Ricevere aggiornamenti delle fatture tramite webhook

Nella sezione di configurazione API nella piattaforma Free Invoice, è possibile configurare degli endpoint a cui saranno inoltrate notifiche di ad ogni aggiornamento di fattura (sia di costo che di ricavo). In altre parole, ad ogni modifica dello stato fattura (ad esempio da “Inviata” a “Consegnata”), Free Invoice invierà una chiamata HTTP all’endpoint specificato con tutti i dettagli della fattura. Questa è la soluzione ideale per chi vuole utilizzare le API di Free Invoice in e-commerce e siti web.

Per poter funzionare, l’e-commerce deve esporre una rotta HTTP raggiungibile pubblicamente in grado di accettare chiamate POST con body in formato JSON.

Se l’utente ha configurato un endpoint “https://mioecommerce.it/freeinvoicewebhooks” ad ogni aggiornamento di una fattura sarà fatta una richiesta POST del tipo:

```
POST /invoiceId HTTP/1.1
Host: mioecommerce.it
Content-Type: application/json
x-freeinvoice-signature: cb5c4c6a2fa924e0848cd21b6935961d6851b9415d96ec8a...87c0a
x-freeinvoice-timestamp: 1602575616

Content-Length: 690

{
  "type": "invoice_status_changed",
  "data": {
    "invoiceId": "12",
    ...
  }
}
```

NB: il campo “data” mostrato nell’esempio è identico al campo ricevuto in risposta alla richiesta dettagli fattura. Per maggiori dettagli vedere la sezione “Visualizza i dettagli di una fattura”.

Per verificare la chiamata ricevuta sono SEMPRE NECESSARIE le seguenti operazioni:

1. assicurarsi che il timestamp UNIX ricevuto nell’header “X-Freeinvoice-Timestamp” non sia distante dal timestamp attuale del sistema di oltre 5 minuti. Al fine di verificare questa condizione, assicurarsi che l’orario del webserver sia aggiornato correttamente.
2. assicurarsi che la firma del messaggio ricevuta nell’header “X-Freeinvoice-Signature” sia uguale a

```
var timestamp = HEADER['X-Freeinvoice-Timestamp']
var payload = BODY
var apiKey = 'mia-api-key'

hash_hmac('sha256', "{timestamp}.{payload}", apiKey);
```

Se una delle due condizioni non è valida RIGETTARE la richiesta con HTTP status code diverso da 200. Per accettare la richiesta rispondere con HTTP status code 200.

NB: tutte le richieste rigettate saranno ritrasmesse ad intervalli regolari per tre giorni.

## Lotti pacchetto api disponibili

Tramite api è possibile richiedere informazioni sul numero di lotti fattura consumati e quelli ancora disponibili.

La chiamata che dovrai fare per visualizzare i dettagli di una fattura:

```
GET https://endpoint/api/v1/stocks
```

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;

Formato risposta valida:

```
{
  "data": {
    "type": "M",
    "stocks": [
      {
        "validityBegin": "01/01/2019",
        "validityExpiration": "31/12/2019",
        "expired": true,
        "consumedOperationsByAccount": 50,
        "operations": 1000,
        "consumedOperationsByAllAccounts": 100,
        "availableOperations": 900,
      },
      {
        "validityBegin": "01/01/2020",
        "validityExpiration": "31/12/2020",
        "expired": false,
        "consumedOperationsByAccount": 0,
        "operations": 1000,
        "consumedOperationsByAllAccounts": 0,
        "availableOperations": 1000,
      }
    ]
  }
}
```

Il valore type può assumere:

- **M**: se la chiamata viene effettuata da una api key "Master";
- **S**: se la chiamata viene effettuata da una api key relativa a un sottoaccount.

Il numero di operazioni totali sarà disponibile solo per api key “Master”.

Dove stocks è una collezione di uno o più elementi (uno per ogni pacchetto api acquistato).

Dove:

- **validityBegin**: indica la data di inizio validità del pacchetto api;
- **validityExpiration**: indica la data di scadenza del pacchetto api;
- **expired**: indica se il pacchetto api è scaduto o meno;
- **operations**: (disponibile solo per api key Master) il numero di operazioni su fatture (invio/ricezione) del pacchetto;
- **consumedOperationsByAccount**: il numero di operazioni su fatture (invio/ricezione) del pacchetto utilizzate dall’api key richiedente;
- **consumedOperationsByAllAccount**: (disponibile solo per api key Master) il numero totale di operazioni su fatture (invio/ricezione) del pacchetto utilizzate;
- **availableOperations**: il numero di operazioni su fatture (invio/ricezione) del pacchetto ancora disponibili;

Ad ogni richiesta di invio/ricezione fattura il valore “consumedOperationsByAccount” e “consumedOperationsByAllAccount” sarà decrementato di una unità. Sarà possibile continuare a inviare/ricevere fatture tramite api finchè esisterà almeno un pacchetto non scaduto e “availableOperations” maggiore di 0.

## Importare un documento

La chiamata che dovrai fare per inviare una fattura sarà la seguente:

POST <https://endpoint/api/v1/document/usend>

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **invoiceFileBase64**: contiene l'XML della fattura convertito in base 64;
- **invoiceType**: possibili valori ["ddt", "ordini", "preventivi", "proforma", "ricevute"];
- **customerEmail**: (opzionale): email identificativa dell'anagrafica
- **receiverCode**: (opzionale): codice identificativo dell'anagrafica

Formato risposta valida:

```
{  
  "invoiceId": "123abc",  
  "timestamp": "2018-10-10T12:29:17.000000Z"  
}
```

Il campo `invoiceId` identifica univocamente il documento sul sistema ed è utilizzabile, successivamente, per richiedere dettagli.

Il campo `timestamp` identifica quando il documento è stato aggiunto al sistema. Si noti che questo timestamp può differire dall'ora di invio o ricezione fattura disponibile nei dettagli della fattura.

Possibili errori:

Codice	Messaggio
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.
200	Formato fattura non valido.
201	Progressivo invio non valido.
202	Dati cedente prestatore errati.

## Elencare i documenti

La chiamata che dovrai fare per elencare le fatture inviate:

```
GET https://endpoint/api/v1/documents/list
```

Questa richiesta permette di caricare la lista degli `invoiceId` delle fatture inviate in un certo intervallo di tempo ordinate cronologicamente.

Parametri della richiesta:

- **apiKey**: API KEY associate alla partita iva del mittente della fattura che si vuole inviare;
- **invoiceType**: possibili valori ["ddt", "ordini", "preventivi", "proforma", "ricevute"];
- **page**: (default = 1) pagina da caricare;
- **withinAfter**: il formato deve essere YYYY-MM-DD HH:MM:SS. Nella lista saranno presenti *solo* le fatture caricate nel sistema ad un orario successivo o uguale a quello specificato;
- **withinBefore**: (opzionale) il formato deve essere YYYY-MM-DD HH:MM:SS. Nella lista saranno presenti *solo* le fatture caricate nel sistema ad un orario successivo o uguale a `withinAfter` e precedente o uguale a `withinBefore`.

Formato risposta valida:

```
{
  "data": [
    {
      "invoiceId": "124abc",
      "timestamp": "2018-10-10T12:29:17.000000Z",
    },
    {
      "invoiceId": "123abc",
      "timestamp": "2018-10-10T12:35:26.000000Z",
    }
  ]
  // Altri dati fino ad un massimo di 1000
},
"meta": {
  "current_page": 1,
  "last_page": 3,
  "per_page": 15,
  "from": 1,
  "to": 10,
  "total": 35
}
}
```

Il campo `invoiceId` identifica univocamente il documento sul sistema ed è utilizzabile, successivamente, per richiedere dettagli.

Il campo `timestamp` identifica quando la fattura è stata aggiunta al sistema. Si noti che questo timestamp può differire dall'ora di invio o ricezione fattura disponibile nei dettagli della fattura.

Si noti che la risposta conterrà fino ad un massimo di 1000 elementi con timestamp crescente.

Possibili errori:

<b>Codice</b>	<b>Messaggio</b>
000	Errore generico.
001	Api key mancante.
002	Api key non valida.
003	Api key non ancora attiva
004	Api key scaduta.
005	Parametro richiesta mancante.
006	Parametro richiesta errato.