

**GESTISCO ITALIA**

Via A.Kramer, 19  
20129 Milano (MI)  
Tel. +39 02 320622233  
E-mail: [info@gestisco.it](mailto:info@gestisco.it)



# **Manuale Web Services**

v.1.9 - 15 Aprile 2020

---

Web Services FatturaPRO.click

Documentazione API

---

Nessuna parte di questo documento può essere riprodotta, interamente o in parte, senza l'espressa autorizzazione scritta di Gestisco Italia  
No part of this document may be reproduced, in whole or in part, without the express written permission of Gestisco Italia  
Aucune partie de ce document ne peut être reproduite, en tout ou en partie, sans l'autorisation écrite expresse de Gestisco Italia  
Kein Teil dieses Dokuments darf ohne ausdrückliche schriftliche Genehmigung von Gestisco Italia ganz oder teilweise reproduziert werden

# Sommario

---

<b>Documentazione API</b>	<b>3</b>
userGetKey	6
userLogin	7
loadDocument	8
createDocument	9
getStatus	10
getList	11
getDocument	12
processDocument	13
getKey (Partner)	14
Login (Partner)	15
<b>Errori</b>	<b>16</b>
Elenco errori	16
<b>Domande e Risposte</b>	<b>17</b>
<b>Suggerimenti</b>	<b>18</b>
ASP.NET Base64 string corruption	18

## Documentazione API

---

Le API di FatturaPRO sono in continuo sviluppo, questo manuale sarà sempre aggiornato all'ultima release, pertanto non spediremo mai aggiornamenti.

### URL di accesso al pannello DEMO:

<https://dev.gestisconet.it/test/gestpro/fatturapa>

### Endpoint

<https://dev.gestisconet.it/test/gestpro/fatturapa/api/api.php>

### Note

Nella chiamata API occorre indicare l'attività da svolgere, ogni attività ha permessi di accesso diversi legati al tipo di abbonamento sottoscritto.

Se siete in possesso di una chiave Partner, questa non deve mai essere installata sui vostri client, in quanto ha permessi tali da minacciare la sicurezza. Per le attività di recupero chiave e SSO devono essere utilizzate le funzioni client (userGetKey e userLogin), mentre le funzioni Partner (getKey e Login) possono essere implementate solo sui server del Partner.

Chiamata preliminare per ottenere la chiave utente e memorizzarla nel client, questa chiamata genera ed assegna una chiave al client ed attiva l'abbonamento (se ancora non attivo):

- **userGetKey (utenze PREMIUM e PREMIUM+)**
  - La API si aspetta in input la P.IVA e la password dell'utenza per cui si richiede la ApiKey.
  - Se l'utenza ha attivo o attivabile un abbonamento PREMIUM o PREMIUM PLUS verrà restituita la relativa ApiKey e attivato l'abbonamento, qualora non dovesse essere ancora attivo.

Chiamata di servizio, per aprire il pannello FatturaPRO in SSO senza autenticazione (se utilizzata è necessario dotare il client delle necessarie misure di sicurezza (password operatore))

- **userLogin (utenze PREMIUM e PREMIUM+)**
  - Per le utenze PREMIUM PLUS è possibile specificare il codice fiscale dell'operatore che ha accesso all'utenza.

Chiamate operative:● **loadDocument (utenze PREMIUM+)**

- La API si aspetta in input un file XML in standard SDI;
- Il documento viene verificato, nel caso si riscontrassero anomalie, l'API fornirà in risposta un errore di XML non valido;
- Il documento validato viene integrato con i dati del terzo intermediario, caricato nella sezione "Documenti importati" ed immediatamente messo in coda per l'invio;

● **createDocument (utenze PREMIUM e PREMIUM+)**

- La API si aspetta in input un file XML in standard SDI;
- Dal file vengono prelevati:
  - Numero
  - Data
  - Causale
  - Destinatario
  - Righe del documento
- La sezione mittente viene compilata in base alle impostazioni del profilo utente;
- La causale viene compilata in base alle impostazioni avanzate del profilo (assicurarsi che sia impostata onde evitare un errore per mancata compilazione);
- I totali vengono calcolati in automatico;
- Il documento viene generato nella sezione "Fatture e Note di Credito" e verificato, nel caso si riscontrassero anomalie, l'API fornirà in risposta l'elenco degli errori.
- Il documento validato non viene inviato automaticamente.

● **getStatus (utenze PREMIUM+)**

- La API restituisce lo stato di un documento importato;
- Gli stati in rosso indicano
- ***Dati restituiti:***
  - status
  - status\_desc
  - error\_txt
  - b64\_sdi\_message
  - sdi\_message\_type
- ***Stati previsti:***
  - Attesa invio
  - Inviato SDI
  - Consegnato
  - Non consegnato
  - Consegnato PA
  - Accettato PA
  - Rifiutato PA
  - Accettato DT
  - Accettato AT
  - Errori SDI
  - Rifiutato SDI

- **getList (utenze PREMIUM+)**
  - La API restituisce l'elenco dei documenti ricevuti (acquisti);
  - Vengono restituiti i seguenti dati:
    - token
  - La lista contiene i documenti ricevuti per cui non si è ancora effettuata una `getDocument`, con il limite di 100.
- **getDocument (utenze PREMIUM+)**
  - La API si aspetta in input il token associato al documento che si richiede.
  - Vengono restituiti i seguenti dati:
    - filename
    - b64
- **processDocument (utenze PA PREMIUM+)**
  - La API si aspetta in input il token del documento per cui trasmettere l'accettazione o il rifiuto con motivazione.
  - Viene restituito l'esito

Chiamate Partner, per l'utilizzo sui server del Partner, portali / piattaforme cloud.

- **getKey (utenze Partner)**
  - La API si aspetta in input la P.IVA dell'utenza per cui si richiede la ApiKey.
  - Se l'utenza ha attivo o attivabile un abbonamento PREMIUM o PREMIUM PLUS verrà restituita la relativa ApiKey e attivato l'abbonamento, qualora non attivo.
- **Login (utenze Partner)**
  - La API si aspetta in input la P.IVA dell'utenza per cui si richiede il SSO.
  - Per le utenze PREMIUM PLUS è possibile specificare il codice fiscale dell'operatore che ha accesso all'utenza.

*userGetKey***INPUT**

Nome	Tipo	Descrizione
action	String	nome della api da chiamare: impostare a <b>userGetKey</b>
partita_iva	String	partita iva dell'utenza per cui si richiede la apiKey utente
password	String	MD5 della password dell'utenza per cui si richiede la apiKey utente

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione
- key: ApiKey dell'utenza

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>api key utente</description>
  <key>ac154c025682e08a2997c161b4288007</key>
</result>
```

---

**Errore:**

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

*userLogin***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey <b>Utente</b> fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>userLogin</b>
operatore	String	codice fiscale dell'operatore che effettua il login (facoltativo)

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione
- token: codice identificativo per il SSO
- login\_url: url per effettuare il SSO

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>login token generato</description>
  <token>5a8cad8bb3a5d</token>
  <login_url>https://.....</login_url>
</result>
```

---

**Errore:**

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

*loadDocument***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey utente fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>loadDocument</b>
base64encodedxml	base64 String	stringa xml da cui creare il documento, codificata in base64

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione
- docId: di tipo int. Id del documento creato

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>documento creato</description>
  <token>codice_token</token>
</result>
```

---

**Errore:**

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```



*createDocument***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey utente fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>createDocument</b>
base64encodedxml	base64 String	stringa xml da cui creare il documento, codificata in base64

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione
- docId: di tipo int. Id del documento creato

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>documento creato</description>
  <token>codice_token</token>
</result>
```

**Errore:**

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

**Esempio:**

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

**Lista Errori:**

- 5 - terminale non autorizzato, effettuare registrazione
- 9 - documento creato con errori
  - [token] > token documento
  - [errori] > lista errori
- 11 - base64encodedxml non è un base64 valido
- 11 - errori xml: <lista errori>
- 11 - errori check xml: <lista errori>
- 13 - account non valido
- 15 - documento già presente
- 1004 - target non valido

**Tipologie di Errore comuni:**

- 9 - errori di validazione
- 11 - errori XML

*getStatus***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey utente fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>getStatus</b>
token	String	token del documento per il quale si intende recuperare lo stato

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- **returnCode**: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- **description**: descrizione esito operazione
- **status**: stato del processo: attivo | concluso | rifiutato
- **status\_desc**: Informazioni sullo stato del processo
- **campi aggiuntivi presenti nei casi di rifiuto**:
  - **error\_txt**: tipologia di rifiuto (Rifiutato PA, Rifiutato SdI)
  - **b64\_sdi\_message**: notifica xml di rifiuto, trasmessa dal SdI
  - **sdi\_message\_type**: tipo di notifica SdI

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>Informazioni sullo stato del documento recuperate</description>
  <status>rifiutato</status>
  <status_desc>Processo concluso con errori</status_desc>
  <error_txt>Rifiutato SdI</error_txt>
  <b64_sdi_message>file_xml_sdi_codificato_base64</b64_sdi_message>
  <sdi_message_type>NS</sdi_message_type>
</result>
```

**Errore:**

- **returnCode**: di tipo int. Codice di errore
- **description**: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

*getList***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey utente fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>getList</b>

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente l'elenco dei tokens di tutti i documenti ricevuti, che non risultano essere stati recuperati

**Successo:**

- **returnCode**: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- **description**: descrizione del risultato
- **token** dei documenti ricevuti (uno per ogni documento)

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>lista FEP</description>
  <tokens>
    <token>token_documento</token>
  </tokens>
</result>
```

---

**Errore:**

- **returnCode**: di tipo int. Codice di errore
- **description**: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

*getDocument***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey utente fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>getDocument</b>
token	String	token del documento per cui si richiede il file ricevuto da SDI

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il documento ricevuto, codificato in base64

**Successo:**

- **returnCode**: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- **raw**: documento ricevuto, codificato in base64

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <document>
    <raw>
      <filename>nomefile</filename>
      <b64>base64</b64>
    </raw>
  </document>
</result>
```

**Errore:**

- **returnCode**: di tipo int. Codice di errore
- **description**: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

*processDocument***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey utente fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>processDocument</b>
token	String	token del documento da elaborare
operation	String	valori ammessi: "accetta" oppure "rifiuta"
description	String	motivazione (facoltativa) in caso di rifiuto

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>accettazione inviata</description>
</result>
```

---

**Errore:**

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

*getKey (Partner)***INPUT**

Nome	Tipo	Descrizione
apiKey	String	apiKey <b>Partner</b> fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>getKey</b>
partita_iva	String	partita iva dell'utenza per cui si richiede la apiKey utente

**OUTPUT**

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

**Successo:**

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione
- key: ApiKey dell'utenza

**Esempio di Successo:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>api key utente</description>
  <key>ac154c025682e08a2997c161b4288007</key>
</result>
```

---

**Errore:**

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

**Esempio di Errore:**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

## Login (Partner)

### INPUT

Nome	Tipo	Descrizione
apiKey	String	apiKey <b>Partner</b> fornita per accedere alle api
action	String	nome della api da chiamare: impostare a <b>Login</b>
partita_iva	String	partita iva dell'utenza per cui effettuare il login
operatore	String	codice fiscale dell'operatore che effettua il login (facoltativo)

### OUTPUT

Tipo	Descrizione
xml	stringa xml contenente il risultato della chiamata: successo o errore

#### Successo:

- returnCode: di tipo int. 0 è il codice di successo, un codice diverso da 0 indica un codice di errore
- description: descrizione esito operazione
- token: codice identificativo per il SSO
- login\_url: url per effettuare il SSO

#### Esempio di Successo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>0</returnCode>
  <description>login token generato</description>
  <token>5a8cad8bb3a5d</token>
  <login_url>https://.....</login_url>
</result>
```

---

#### Errore:

- returnCode: di tipo int. Codice di errore
- description: di tipo string. Breve descrizione dell'errore

#### Esempio di Errore:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<result>
  <returnCode>11</returnCode>
  <description>descrizione errore</description>
</result>
```

## Errori

La tabella sottostante contiene l'elenco completo degli errori che possono essere restituiti dalle API nei casi di fallimento.

### Elenco errori

returnCode	Tipo di errore	Descrizione
5	errore utente	utente non trovato o non attivo
11	xml non valido	lista di errori riscontrati separati da \n (newline)
13	errore account	il tipo di account sottoscritto non consente questa azione
14	partita iva non corrispondente	la partita iva nel xml non corrisponde a quella dell'account
15	documento già presente	un documento con la medesima data e numero è già presente in archivio per l'account
16	accesso negato	L'utente non ha permessi sufficienti per accedere alle azioni richieste
17	errore operatore	operatore non trovato o non abilitato per l'utenza indicata
19	errore azione	azione non valida sul documento
99	maintenance	piattaforma in manutenzione
100	errore parametri	parametro obbligatorio mancante



## Domande e Risposte

---

**D: se una getList fallisce perdo i token ?**

*R: NO, i token vengono restituiti alle successive getList se non è stata effettuata per questi una getDocument, in questo modo si ha la certezza matematica di non perdere alcun token nei casi di interruzione della comunicazione, poiché se riesci a fare una getDocument significa che sei in possesso del token, ovviamente devi essere certo di non perderlo dopo aver fatto una getDocument altrimenti non lo avresti più disponibile nei casi interruzione della comunicazione, situazione che ti costringe ad una successiva chiamata per completare l'operazione.*

**D: La getList restituisce al massimo di 100 token, se i documenti sono di più ?**

*R: i token vengono restituiti da una getList con un limite di 100, questo significa che ne ricevo 100 significa che molto probabilmente ce ne sono altri che non ti abbiamo dato, di conseguenza dopo aver effettuato il download tramite getDocument dei primi 100 dovrai effettuare una seconda chiamata e una terza se necessario, fino a quando i token restituiti sono minori di 100.*

**D: Come mi comporto se ricevo un token che ho già elaborato ?**

*R: Se ti viene restituito un token che già possiedi significa che a noi non risulta per quel token alcuna getDocument e viene pertanto ancora restituito da una getList, in questo caso per eliminarlo e liberare uno spazio per un altro token nella getList devi effettuare una getDocument per quel token anche se non ne hai bisogno. Di solito però questo avviene quando è stato chiesto un reset, quindi è il caso che il documento venga comunque scaricato e sostituito all'attuale poiché se non lo stesso, sicuramente più recente.*

**D: Posso cancellare il token una volta scaricato il documento ?**

*R: NO, il token deve rimanere memorizzato ed associato al documento memorizzato nei tuoi sistemi per permetterti, nel caso di problemi di connessione o reset, di capire se si tratta di token già elaborato, prevenendo così l'inserimento di record duplicati nel tuo database.*

*Diversamente perderesti l'identificativo univoco di relazione con i nostri sistemi e non saresti in grado di prevenire i problemi di cui sopra.*

**D: Qual'è il corretto workflow da utilizzare per evitare problemi ?**

*R: Consigliamo di adottare questo flusso per il download dei documenti:*

- 1. Effettuare una prima getList, se i token restituiti sono 100 eseguire i punti 2 e 3 e tornare al punto 1*
- 2. Memorizzare tutti i token del DB e posizionare i record nello stato "Da elaborare", se il token è presente nel DB come "Elaborato" spostare quel record in "Da elaborare" allo scopo di lanciare su quel token nuovamente la getDocument ed aggiornare i dati.*
- 3. Elaborare tutti i record nello stato "Da elaborare" completando i dati con la getDocument e, solo se scaricato correttamente il documento, spostare il record nello stato di "Elaborato"*

*Per i documenti inviati il token viene restituito all'atto dell'elaborazione, se non ricevete risposta a causa di un'interruzione della connessione effettuate nuovamente l'invio, se il documento è stato correttamente acquisito con la precedente chiamata vi verrà restituito un errore di documento già presente con il relativo token non ricevuto alla precedente chiamata.*

*Questa procedura può essere utilizzata anche successivamente per recuperare il token di un documento trasmesso e in stato di inviato o archiviato, non vale per i documenti rifiutati.*

## Suggerimenti

---

### *ASP.NET Base64 string corruption*

Se lavorate in asp.net, la codifica di una stringa Base64 risulta corrotta a causa del carattere "+" che viene sostituito da un line break.

Per creare un Safe URL è necessario effettuare la sostituzione del carattere prima dell'invio. stando a quanto riportato da alcuni nostri Partner, questo è l'unico carattere da sostituire, anche se diversi hanno optato per una sostituzione completa, come da elenco seguente:

```
strBase64 = strBase64.Replace("+", "%2B")
strBase64 = strBase64.Replace("!", "%21")
strBase64 = strBase64.Replace("#", "%23")
strBase64 = strBase64.Replace("$", "%24")
strBase64 = strBase64.Replace("&", "%26")
strBase64 = strBase64.Replace("'", "%27")
strBase64 = strBase64.Replace("(", "%28")
strBase64 = strBase64.Replace(")", "%29")
strBase64 = strBase64.Replace("*", "%2A")
strBase64 = strBase64.Replace(",", "%2C")
strBase64 = strBase64.Replace("/", "%2F")
strBase64 = strBase64.Replace(":", "%3A")
strBase64 = strBase64.Replace(";", "%3B")
strBase64 = strBase64.Replace("=", "%3D")
strBase64 = strBase64.Replace("?", "%3F")
strBase64 = strBase64.Replace("@", "%40")
strBase64 = strBase64.Replace("[", "%5B")
strBase64 = strBase64.Replace("]", "%5D")
```