

[SEND] Documentazione Tecnica API - v1.0.

Versioni

Data	Versi one	Owner	Dettaglio intervento
25/07/2025	1.0	Roberto Vittorio Muzio	Prima stesura documento

Sommario

-  [Versioni](#)
-  [Obiettivi](#)
-  [Servizi](#)
 -  [Autenticazione](#)
 -  [Scenario 1 - Costruzione Layer di Autenticazione](#)
 -  [Parametri di Chiamata](#)
 -  [Modulo Commessa](#)
 -  [Scenario 1 - Interrogazione Modulo Commessa](#)
 -  [Scenario 2 - Caricamento Modulo Commessa](#)
 -  [Parametri di Chiamata](#)
 -  [Notifiche](#)
 -  [Scenario 1 - Interrogazione Notifiche](#)
 -  [Parametri di Chiamata](#)
 -  [Regolare Esecuzione](#)
 -  [Scenario 1 - Interrogazione Regolare Esecuzione](#)
 -  [Scenario 2 - Caricamento REL Firmata](#)
 -  [Parametri di Chiamata](#)
 -  [Documenti Emessi](#)
 -  [Scenario](#)



⌚ Obiettivi

Obietti vi e tragua rdi	Il presente documento fornisce le specifiche tecniche necessarie per l'integrazione con le API fornite, al fine di riprodurre correttamente e in modo sequenziale il flusso delle chiamate, garantendo così la corretta gestione e verifica delle notifiche all'interno del Portale Fatturazione di SEND.
Refere nti Interni	<p>@Roberto Vittorio lo Muzio</p> <p>@ciro.zambrano</p>

🌐 Servizi

All'interno di questa sezione andiamo a riportare i vari scenari possibili nell'utilizzo dei servizi esposti e le varie configurazioni tecniche da mettere in atto per permettere le chiamate.

Autenticazione

- GET - [**/v1/authentication**](#): Verifica autenticazione.

Regolare Esecuzione

- POST - [**/v1/rel/upload**](#): Permette di caricare la REL firmata con 'multipart/form-data'.
- POST - [**/v1/rel/download**](#): Ritorna il documento REL in formato PDF o HTML.
- GET - [**/v1/rel/periodo**](#): Ritorna anni e mesi relativi alla regolare esecuzione.
- POST - [**/v1/rel/tipolografattura**](#): Ritorna tipologia fattura relative alla regolare esecuzione.
- POST - [**/v1/rel/righe**](#): Ritorna righe regolare esecuzione.
- POST - [**/v1/rel/testata**](#): Ritorna la testata della REL per query.

Notifiche

- GET - [**/v1/notifiche/periodo**](#): Ritorna anni e mesi relativi a notifiche.
- POST - [**/v1/notifiche**](#): Ritorna le notifiche in base alla query richiesta.

- GET - [**/v1/notifiche/flag**](#): Ritorna flag contestazioni.
- GET - [**/v1/notifiche/tipologia**](#): Ritorna le tipologie notifiche.

Modulo Commessa

- GET - [**/v1/modulocommessa/configurazione**](#): Ritorna la configurazione per inserire il modulo commessa. In particolare il campo: idTipoSpedizione
- POST - [**/v1/modulocommessa**](#): Permette di inserire o modificare il modulo commessa nel periodo di riferimento [1,19] mese corrente
- GET - [**/v1/modulocommessa/periodo**](#): Permette di tornare i vari periodi storici in cui è stata inserito un modulo commessa.
- POST - [**/v1/modulocommessa/anno**](#): Permette di tornare un sommario di tutti i moduli commessa inseriti per anno.
- POST - [**/v1/modulocommessa/dettaglio**](#): Permette di tornare il dettaglio modulo commessa per anno e mese.

Documenti Emessi

- POST - [**/v1/fatture/ricerca**](#): Permette di tornare le fatture per periodo e tipologia.
- GET - [**/v1/fatture/periodo**](#): Permette di tornare i vari periodi storici con relativa tipologia di fattura.

Contestazioni

- POST - [**/v1/contestazioni/notifiche/check/download**](#): Ritorna OK/KO per download notifiche per il periodo di riferimento. Se true bisogna aggiornare il file notifiche facendo il download sull'endpoint notifiche, altrimenti no.
- GET - [**/v1/contestazioni/azione/periodo**](#): Ritorna anni e mesi azione relativi alle contestazioni.
- GET - [**/v1/contestazioni/archivio/periodo**](#): Ritorna anni e mesi archivio relativi alle contestazioni.
- POST - [**/v1/contestazioni/recap**](#): Ritorna recap relativo a contestazioni.
- POST - [**/v1/contestazioni/reports**](#): Ritorna report contestazioni per tipologia.
- POST - [**/v1/contestazioni/reports/document**](#): Ritorna sas token documento singolo step and report.

- **POST - [/v1/contestazioni/reports/steps](#)**: Ritorna step relativi a un singolo report contestazioni.
- **GET - [/v1/contestazioni/steps](#)**: Ritorna tipologia step contestazioni.
- **GET - [/v1/contestazioni/tiporeport](#)**: Ritorna tipologia report contestazioni.
- **POST - [/v1/contestazioni/upload](#)**: Ritorna SAS token per upload file.

Models

anniMesiTipologiaByEnteDto{

anno	integer(\$int32)
mese	integer(\$int32)
tipologiaFattura	string

}

authenticationResponse{

idEnte	string
timestamp	string(\$date-time)
functionName	string
stage	string
method	string
uri	string
ipAddress	string
id	string

}

checkDownloadNotificheDto{

dataMassimaContestazione	string(\$date-time)
notificheDaScaricare	boolean
ciSonoContestazioni	boolean

}

checkDownloadNotificheRequest{

dataVerifica	string(\$date-time)
anno	integer(\$int32)
mese	integer(\$int32)

}

contestazioneRecap{

tipologiaFattura	string
idFlagContestazione	integer(\$int32)
flagContestazione	string
totale	integer(\$int32)
totaleNotificheAnalogiche	integer(\$int32)
totaleNotificheDigitali	integer(\$int32)

}

contestazioneStep{

step	integer(\$int32)
descrizione	string

}

contestazioniDocumentRequest{

idReport	integer(\$int32)
step	integer(\$int32)

}

contestazioniRecapEnteApiRequest{

anno	integer(\$int32)
mese	integer(\$int32)

}

contestazioniReportEntePagingRequest{

anno	string
mese	string
idTipologiaReports	[...]
page	integer(\$int32)
pageSize	integer(\$int32)

}

datiConfigurazioneModuloCategoriaCommessaResponse{

idCategoriaSpedizione	integer(\$int32)
percentuale	integer(\$int32)
descrizione	string

}

datiConfigurazioneModuloCommessaResponse{

categorie	[...]
tipi	[...]

}

datiConfigurazioneModuloTipoCommessaResponse{

idTipoSpedizione	integer(\$int32)
mediaNotificaNazionale	number(\$double)
mediaNotificaInternazionale	number(\$double)
descrizione	string

}

datiFatturazioneContatto{

idDatiFatturazione	integer(\$int64)
email	string

}

datiGeneraliDocumentoDto{

tipologia	string
riferimentoNumeroLinea	string
idDocumento	string
data	string
numItem	string
codiceCommissaConvenzione	string
cup	string
cig	string

}

datiModuloCommissaByAnnoResponse{

modifica	boolean
annoValidita	integer(\$int32)
meseValidita	integer(\$int32)
idEnte	string
idTipoContratto	integer(\$int64)
stato	string
prodotto	string
totale	string
dataModifica	string

totaleDigitale	string
totaleAnalogico	string

}

datiModuloCommessaCreateRequest{

moduliCommessa	[...]
-----------------------	--------------

}

datiModuloCommessaCreateSimpleRequest{

numeroNotificheNazionali	integer(\$int32)
numeroNotificheInternazionali	integer(\$int32)
idTipoSpedizione	integer(\$int32)

}

datiModuloCommessaResponse{

idTipoContratto	integer(\$int64)
modifica	boolean
anno	integer(\$int32)
mese	integer(\$int32)
dataModifica	string(\$date-time)
moduliCommessa	[...]
totaleModuloCommessaNotifica	totaleDatiModuloCommessaNotifica{...}
totale	[...]

}

datiModuloCommessaSimpleResponse{

numeroNotificheNazionali	integer(\$int32)
numeroNotificheInternazionali	integer(\$int32)

idTipoSpedizione	integer(\$int32)
totaleNotifiche	integer(\$int32)

}

datiModuloCommessaTotaleCostoDto{

totale	number(\$double)
totaleValuta	string
descrizione	string

}

datiModuloCommessaTotaleDto{

totaleNotifiche	integer(\$int32)
numeroNotificheNazionali	integer(\$int32)
numeroNotificheInternazionali	integer(\$int32)
tipo	string
idTipoSpedizione	integer(\$int32)

}

fatturaDto{

fattura	titoloFatturaDto{...}
----------------	------------------------------

}

fatturaRicercaEnteRequest{

anno	integer(\$int32)
mese	integer(\$int32)
tipologiaFattura	[...]

}

flagContestazione{

id	integer(\$int32)
flag	string
descrizione	string

}

moduloComessaAnnoMeseDto{

anno	integer(\$int32)
mese	integer(\$int32)

}

moduloComessaDocumentoDto{

cup	string
codComessa	string
dataDocumento	string(\$date-time)
splitPayment	string
idDocumento	string
map	string
tipoComessa	string
prodotto	string
pec	string
dataModifica	string(\$date-time)
meseAttivita	string
contatti	[...]
descrizione	string
partitaIva	string
indirizzoCompleto	string
datiModuloComessa	[...]

datiModuloCommessaCosti

[...]

}

moduloCommessaGetAnniMesiRequest{

prodotto

string

session

session{...}

}

moduloCommessaGetByAnnoMeseDettaglioRequest{

anno

integer(\$int32)

mese

integer(\$int32)

}

moduloCommessaGetByAnnoRequest{

anno

integer(\$int32)

}

notificheAnniMesiResponse{

anno

integer(\$int32)

mese

integer(\$int32)

descrizione

string

}

notificheResponse{

linkDocumento

string

stato

integer(\$int32)

uniqueId

string

count

integer(\$int32)

}

notificheRicercaRequestDocs{

anno	integer(\$int32)
mese	integer(\$int32)
prodotto	string
cap	string
profilo	string
tipoNotifica	integer(\$int32) <i>default: IEnum:</i> Array [6]
statoContestazione	[...]
iun	string
recipientId	string

}

posizioniDto{

numeroLinea	integer(\$int32)
testo	string
codiceMateriale	string
quantita	integer(\$int32)
prezzoUnitario	number(\$double)
imponibile	number(\$double)
periodoRiferimento	string

}

relAnniMesiResponse{

anno	integer(\$int32)
mese	integer(\$int32)

descrizione	string
--------------------	---------------

}

relDownloadRequest{

idTestata	string
------------------	---------------

tipoDocumentoREL	string
-------------------------	---------------

}

relDownloadResponse{

linkDocumento	string
----------------------	---------------

}

relRigheByIdTestataRequest{

idTestata	string
------------------	---------------

}

relRigheByIdTestataResponse{

linkDocumento	string
----------------------	---------------

}

relTestataDto{

relTestate	[...]
-------------------	--------------

count	integer(\$int32)
--------------	-------------------------

}

relTestataRicercaExternalRequest{

anno	integer(\$int32)
-------------	-------------------------

mese	integer(\$int32)
-------------	-------------------------

tipologiaFattura	string
-------------------------	---------------

page	integer(\$int32)
pageSize	integer(\$int32)

}

relTipologiaFatturaRequest{

anno	integer(\$int32)
mese	integer(\$int32)

}

relUploadRequest{

idTestata	string Id Testata
file	string PDF firmato da caricare

}

reportContestazioneStepsDto{

reportId	integer(\$int32)
step	integer(\$int32)
descrizioneStep	string
totaleNotificheAnalogiche890	integer(\$int32)
totaleNotificheAnalogicheARNazionaliAR	integer(\$int32)
totaleNotificheAnalogicheARIInternazionaliRI R	integer(\$int32)
totaleNotificheAnalogicheRSNazionaliRS	integer(\$int32)
totaleNotificheAnalogicheRSInternazionaliRI S	integer(\$int32)
totaleNotificheDigitali	integer(\$int32)

totaleNotifiche	integer(\$int32)
link	string
nonContestataAnnullata	integer(\$int32)
contestataEnte	integer(\$int32)
rispostaEnte	integer(\$int32)
accettata	integer(\$int32)
rispostaSend	integer(\$int32)
rispostaRecapitista	integer(\$int32)
rispostaConsolidatore	integer(\$int32)
rifiutata	integer(\$int32)
nonFatturabile	integer(\$int32)
fatturabile	integer(\$int32)
storage	string
nomeDocumento	string
dataCompletamento	string(\$date-time)

}

reportContestazioneStepsRequest{

idReport	integer(\$int32)
-----------------	------------------

}

reportContestazioni{

reportId	string
uniqueId	string
json	string
anno	integer(\$int32)
mese	integer(\$int32)

internalOrganizationId	string
contractId	string
actualContractId	string
utenteId	string
prodotto	string
stato	integer(\$int32)
descrizioneStato	string
dataInserimento	string(\$date-time)
dataStepCorrente	string(\$date-time)
nomeDocumento	string
linkDocumento	string
storage	string
hash	string
contentType	string
contentLanguage	string
tipologiaDocumento	string
categoriaDocumento	string
ragioneSociale	string

}

reportContestazioniList{

reports	[...]
count	integer(\$int32)

}

session{

idEnte	string
--------	--------

timestamp	string(\$date-time)
functionName	string
stage	string
method	string
payload	string
uri	string
ipAddress	string
id	string
ragioneSociale	string
idContratto	string
idTipoContratto	integer(\$int32)
prodotto	string
instanceId	string
profilo	string

}

simpleRelTestata{

idTestata	string
nomeTestata	string
idEnte	string
ragioneSociale	string
idContratto	string
tipologiaFattura	string
anno	integer(\$int32)
mese	integer(\$int32)
totaleAnalogico	number(\$double)

totaleDigitale	number(\$double)
totaleNotificheAnalogiche	integer(\$int32)
totaleNotificheDigitali	integer(\$int32)
totaleNotifiche	integer(\$int32)
totale	number(\$double)
iva	number(\$double)
totaleAnalogicoIva	number(\$double)
totaleDigitaleIva	number(\$double)
totaleIva	number(\$double)
firmata	string
asseverazioneTotaleAnalogico	number(\$double)
asseverazioneTotaleDigitale	number(\$double)
asseverazioneTotaleNotificheAnalogiche	integer(\$int32)
asseverazioneTotaleNotificheDigitali	integer(\$int32)
asseverazioneTotaleNotifiche	integer(\$int32)
asseverazioneTotale	number(\$double)
asseverazioneTotaleAnalogicoIva	number(\$double)
asseverazioneTotaleDigitaleIva	number(\$double)
asseverazioneTotaleIva	number(\$double)
flagConguaglio	string

}

tipologiaReport{

idTipologiaReport	integer(\$int32)
categoriaDocumento	string
tipologiaDocumento	string

}

tipoNotificaResponse{

tipoNotifica	integer(\$int32)
descrizione	string

}

titoloFatturaDto{

totale	number(\$double)
numero	integer(\$int64)
idFattura	integer(\$int64)
dataFattura	string
prodotto	string
identificativo	string
tipologiaFattura	string
istitutioID	string
onboardingTokenID	string
ragioneSociale	string
tipoContratto	string
idContratto	string
tipoDocumento	string
divisa	string
metodoPagamento	string
causale	string
split	boolean
inviata	integer(\$int32)
elaborazione	integer(\$int32)

sollecito	string
datiGeneraliDocumento	[...]
posizioni	[...]

}

totaleDatiModuloCommessa{

idCategoriaSpedizione	integer(\$int32)
totaleValoreCategoriaSpedizione	number(\$double)

}

totaleDatiModuloCommessaNotifica{

totaleNumeroNotificheNazionali	integer(\$int32)
totaleNumeroNotificheInternazionali	integer(\$int32)
totaleNumeroNotificheDaProcessare	integer(\$int32)
totale	number(\$double)

}

uploadContestazioniEnteApiRequest{

anno	integer(\$int32)
mese	integer(\$int32)

}

Autenticazione

All'interno di questa sezione, riportiamo i dettagli tecnici e funzionali alle chiamate necessarie per verificare l'accesso all'ambiente PagoPa mediante il servizio di Autenticazione.

Scenario 1 - Costruzione Layer di Autenticazione

Per poter accedere alle API di PagoPa è necessario registrare l'indirizzo IP all'interno della WhiteList presente nel Portale Fatturazione, all'interno del Menù API KEY, e procedere con la chiamata da uno degli indirizzi IP caricati.

The screenshot shows the 'API KEY' section of the PagoPA Fatturazione portal. At the top right are links for 'Manuale operativo', 'Assistenza', and 'Esci'. Below is a button for 'Genera API KEY secondaria'. On the left, a sidebar lists 'Notifiche', 'Regolare esecuzione / Documenti di cortesia', 'Download documenti', and 'API key' (which is selected). The main area is titled 'IP RANGE' and contains five entries for IP addresses and their corresponding CIDR ranges, each with a trash icon for deletion.

Fig.1 - Menù API Key presente all'interno del Portale Fatturazione

Qualora l'indirizzo IP chiamante non fosse presente all'interno della Whitelist, l'API risponderà con un errore relativo all'indirizzo non autorizzato.

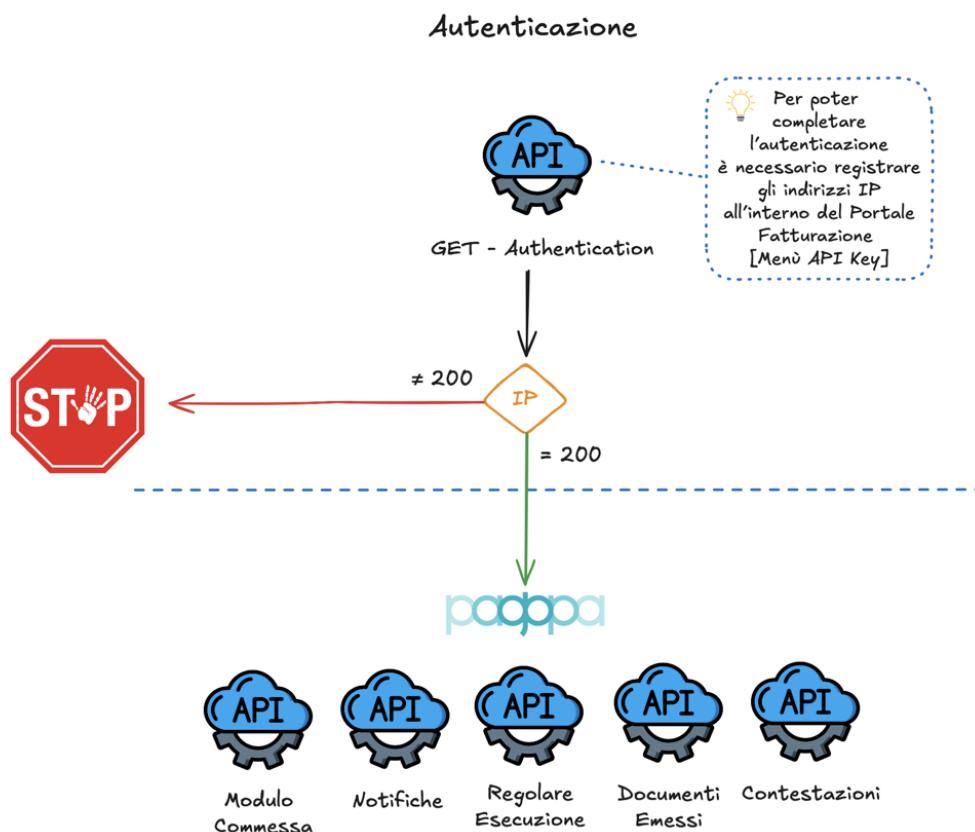


Fig.2 - Flusso logico di chiamata Scenario 1 - Autenticazione

Parametri di Chiamata

L'obiettivo dell'API **Authentication** è quello di costruire un layer di autenticazione per fornire agli Aderenti la possibilità di consumare i servizi del Portale Fatturazione (in breve PF) attraverso un insieme di chiavi

che possono ruotare e un insieme di indirizzi da dove fare le chiamate API.

[GET] Authentication - [/v1/authentication](#)

- Creazione sul db di una table che permetta di memorizzare le informazioni, con associato l'id dell'ente.
- Creazione di un sistema di crittografia per nascondere le informazioni, la chiave crittografica viene memorizzata nel Vault del PF.
- Creazione del codice per scrivere le informazioni di autenticazione sul db e recuperarle. Inoltre la possibilità di “cancellarle.”
- Log delle attività di CRUD dei dati su PF lato utente autorizzato.
- Creazione in PF dell’interfaccia CRUD di queste informazioni de sua gestione. Comparirà sul menù a sinistra la voce “API KEY” o altra label solo nel caso in cui l’aderente/ente è abilitato.
- Creazione di una azure function che esponga le api di autenticazione. Questo layer può essere considerato un Gateway di autenticazione.

In caso di RC 200, avremo il seguente Output:

```
1 {
2   "idEnte": "string",
3   "timestamp": "2025-07-24T15:43:39.949Z",
4   "functionName": "string",
5   "stage": "string",
6   "method": "string",
7   "uri": "string",
8   "ipAddress": "string",
9   "id": "string"
10 }
```

I possibili ReturnCode restituibili dall’API sono i seguenti:

RC	STATO
200	Returns 'Auth Response' on successful authentication.
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

Modulo Commessa

All'interno di questa sezione, riportiamo i dettagli tecnici e funzionali relativi alle chiamate necessarie per caricare e interrogare il Modulo Commessa.

Scenario 1 - Interrogazione Modulo Commessa

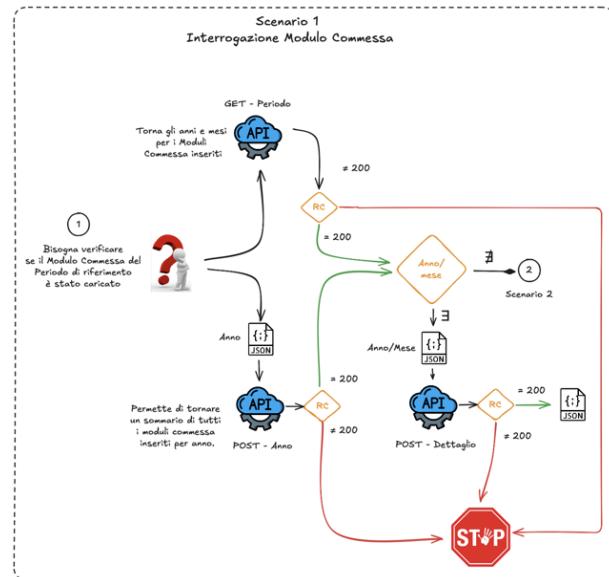


Fig.3 - Flusso logico di chiamata Scenario 1 - Modulo Commessa

Scenario 2 - Caricamento Modulo Commessa

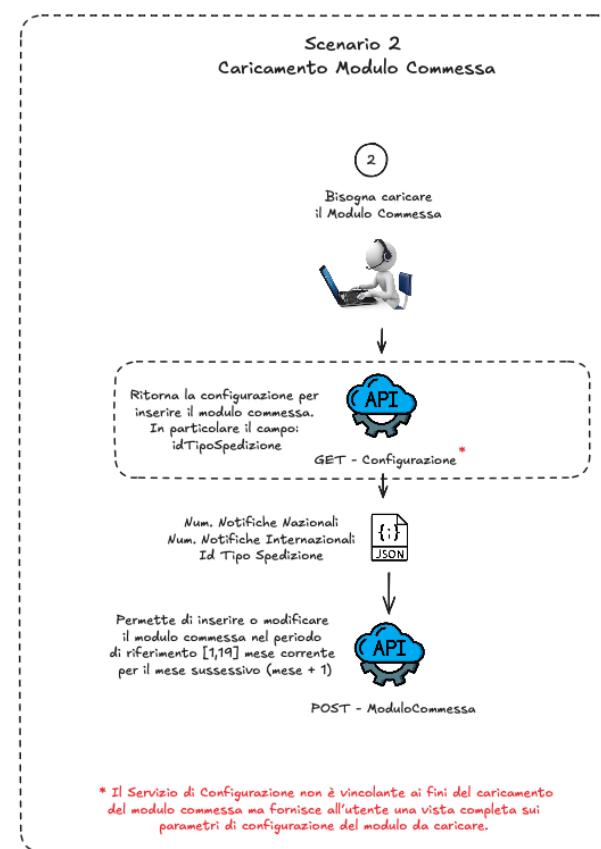


Fig.4 - Flusso logico di chiamata Scenario 2 - Modulo Commessa

Parametri di Chiamata

[GET] Periodo - [/v1/modulocommessa/periodo](#)

L'obiettivo dell'API **Periodo** è quello di ritornare i vari periodi storici in cui è stata inserito un modulo commessa.

In Input non è necessario alcun parametro e in output fornirà un array contenente gli anni e mesi per i Modulo Commessa inseriti.

```

1  [
2    {
3      "anno": 0,
4      "mese": 0,
5      "descrizione": "string"
6    }
7  ]
  
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Anno - [/v1/modulocommessa/anno](#)

L'obiettivo dell'API **Anno** è quello di ritornare un sommario di tutti i moduli commessa inseriti per anno.

In Input viene indicato l'anno per poter estrarre la lista dei moduli commessa inseriti

```
1 {  
2   "anno": 0  
3 }
```

In output fornirà un array contenente un sommario di tutti i moduli commessa inseriti per anno.

```
1 [  
2   {  
3     "modifica": true,  
4     "annoValidita": 0,  
5     "meseValidita": 0,  
6     "idEnte": "string",  
7     "idTipoContratto": 0,  
8     "stato": "string",  
9     "prodotto": "string",  
10    "totale": "string",  
11    "dataModifica": "string",  
12    "totaleDigitale": "string",  
13    "totaleAnalogico": "string"  
14  }  
15 ]
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.

401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Dettaglio - [/v1/modulocommessa/dettaglio](#)

L'obiettivo dell'API **Dettaglio** è quello di ritornare il dettaglio del modulo commessa interrogato con anno e mese.

In Input viene indicato l'anno ed il mese per poter estrarre il modulo commessa interessato.

```

1 {
2   "anno": 0,
3   "mese": 0
4 }
```

In output fornirà un array contenente il dettaglio del modulo commessa interrogato.

```

1 {
2   "cup": "string",
3   "codCommessa": "string",
4   "dataDocumento": "2025-07-25T10:08:51.357Z",
5   "splitPayment": "string",
6   "idDocumento": "string",
7   "map": "string",
8   "tipoCommessa": "string",
9   "prodotto": "string",
10  "pec": "string",
11  "dataModifica": "2025-07-25T10:08:51.357Z",
12  "meseAttivita": "string",
13  "contatti": [
14    {
15      "idDatiFatturazione": 0,
16      "email": "string"
17    }
18  ],
19  "descrizione": "string",
20  "partitaIva": "string",
21  "indirizzoCompleto": "string",
22  "datiModuloCommessa": [
23    {
24      "totaleNotifiche": 0,
25      "numeroNotificheNazionali": 0,
26      "numeroNotificheInternazionali": 0,
27      "tipo": "string",
28      "idTipoSpedizione": 0
29    }
30  ],
31  "datiModuloCommessaCosti": [
32    {
33      "totale": 0,
34      "totaleValuta": "string",
35      "descrizione": "string"
36    }
]
```

```
37     ]
38 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Configurazione - [/v1/modulocommessa/configurazione](#)

L'API **Configurazione** ritorna la configurazione per inserire il modulo commessa, in particolare il campo: idTipoSpedizione.

In Input non è necessario alcun parametro e in output fornirà un array contenente le configurazioni per inserire il modulo commessa.

```
1  {
2    "categorie": [
3      {
4        "idCategoriaSpedizione": 0,
5        "percentuale": 0,
6        "descrizione": "string"
7      }
8    ],
9    "tipi": [
10      {
11        "idTipoSpedizione": 0,
12        "mediaNotificaNazionale": 0,
13        "mediaNotificaInternazionale": 0,
14        "descrizione": "string"
15      }
16    ]
17 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.

401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] ModuloCommessa - [/v1/modulocommessa](#)

L'obiettivo dell'API **ModuloCommessa** è quello di inserire o modificare il modulo commessa nel periodo di riferimento [1,19] del mese corrente.

In Input bisogna indicare i dati del modulo commessa da caricare.

```

1  {
2    "moduliCommessa": [
3      {
4        "numeroNotificheNazionali": 0,
5        "numeroNotificheInternazionali": 0,
6        "idTipoSpedizione": 0
7      }
8    ]
9 }
```

In output fornirà un array contenente il dettaglio del modulo commessa interrogato.

```

1  {
2    "idTipoContratto": 0,
3    "modifica": true,
4    "anno": 0,
5    "mese": 0,
6    "dataModifica": "2025-07-25T10:11:25.292Z",
7    "moduliCommessa": [
8      {
9        "numeroNotificheNazionali": 0,
10       "numeroNotificheInternazionali": 0,
11       "idTipoSpedizione": 0,
12       "totaleNotifiche": 0
13     }
14   ],
15   "totaleModuloCommessaNotifica": {
16     "totaleNumeroNotificheNazionali": 0,
17     "totaleNumeroNotificheInternazionali": 0,
18     "totaleNumeroNotificheDaProcessare": 0,
19     "totale": 0
20   },
21   "totale": [
22     {
23       "idCategoriaSpedizione": 0,
24       "totaleValoreCategoriaSpedizione": 0
25     }
26   ]
27 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

Notifiche

All'interno di questa sezione, riportiamo i dettagli tecnici e funzionali relativi alle chiamate necessarie per visualizzare le Notifiche.

Scenario 1 - Interrogazione Notifiche

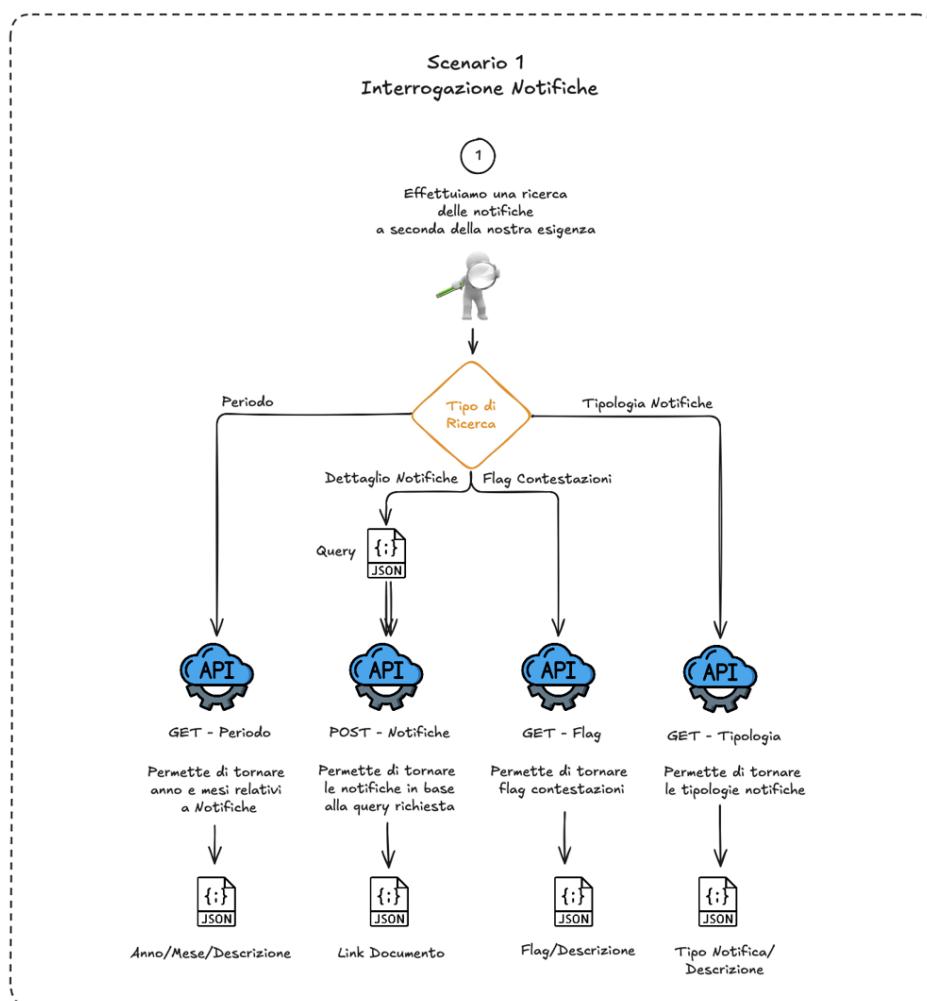


Fig.5 - Flusso logico di chiamata Scenario 1 - Notifiche

Parametri di Chiamata

[POST] Notifiche - [/v1/notifiche](#)

L'obiettivo dell'API **Notifiche** è quello di ritornare il link al documento con il dettaglio delle notifiche da scaricare in base alla query richiesta in input.

In Input bisogna indicare i parametri di ricerca alle notifiche (opzionali).

```
1 {
2   "anno": 0,
3   "mese": 0,
4   "prodotto": "string",
5   "cap": "string",
6   "profilo": "string",
7   "tipoNotifica": 1,
8   "statoContestazione": [
9     0
10    ],
11   "iun": "string",
12   "recipientId": "string"
13 }
```

In output fornirà un array contenente il link al documento con il dettaglio delle notifiche.

```
1 {
2   "linkDocumento": "string",
3   "stato": 0,
4   "uniqueId": "string",
5   "count": 0
6 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Periodo - [/v1/notifiche/periodo](#)

L'obiettivo dell'API **Periodo** è quello di ritornare anni e mesi relativi alle notifiche dell'Ente.

In Input non è necessario alcun parametro e in output fornirà un array contenente gli anni e mesi relativi alle notifiche dell'Ente.

```
1 [
```

```

2   {
3     "anno": 0,
4     "mese": 0,
5     "descrizione": "string"
6   }
7 ]

```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Flag - [/v1/notifiche/flag](#)

L'obiettivo dell'API **Flag** è quello di ritornare lo stato delle contestazioni.

In Input non è necessario alcun parametro e in output fornirà un array contenente lo stati delle contestazioni.

```

1 [
2   {
3     "id": 0,
4     "flag": "string",
5     "descrizione": "string"
6   }
7 ]

```

Di seguito la rappresentazione dei flag contestazioni possibili:

ID	Flag	Descrizione
1	Non Contestata	Non contestata
2	Annnullata	Annnullata
3	Contestata Ente	Contestazione
4	Risposta Send	Contestazione
5	Risposta Recapitista	Contestazione
6	Risposta Consolidatore	Contestazione

7	Risposta Ente	Contestazione
8	Accettata	Risoluzione
9	Rifiutata	Risoluzione

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Tipologia - [/v1/notifiche/tipologia](#)

L'obiettivo dell'API **Tipologia** è quello di ritornare le tipologie delle notifiche

In Input non è necessario alcun parametro e in output fornirà un array contenente le tipologie delle notifiche.

```

1  [
2    {
3      "id": 0,
4      "flag": "string",
5      "descrizione": "string"
6    }
7  ]

```

Di seguito la rappresentazione delle tipologie delle notifiche:

Descrizione
Digitale
Analog. A/R
Analog. L. 890/82
Analog. R/S

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

Regolare Esecuzione

All'interno di questa sezione, riportiamo i dettagli tecnici e funzionali relativi alle chiamate necessarie per caricare e interrogare la Regolare Esecuzione.

Scenario 1 - Interrogazione Regolare Esecuzione

Scenario 2 - Caricamento REL Firmata

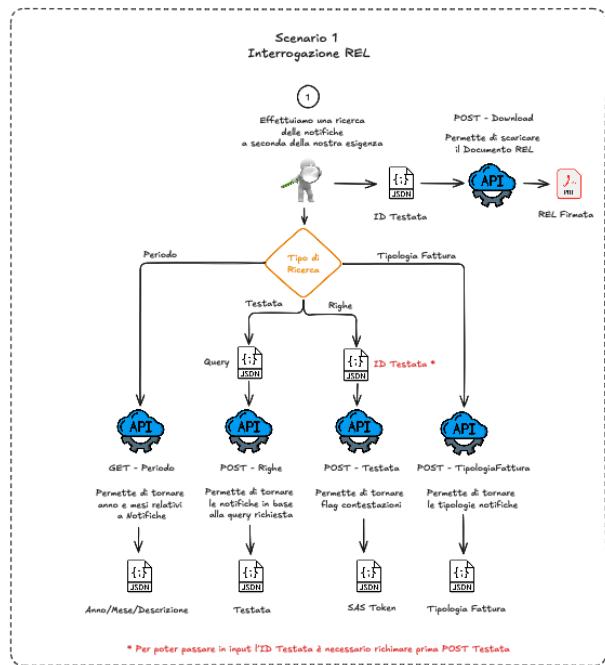


Fig.6 - Flusso logico di chiamata Scenario 1 - Regolare Esecuzione

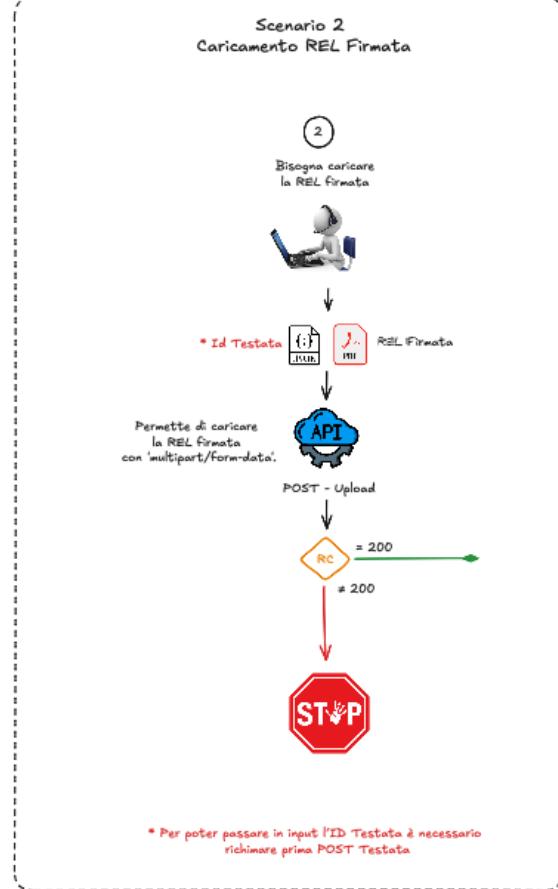


Fig.7 - Flusso logico di chiamata Scenario 2- Regolare Esecuzione

Parametri di Chiamata

[POST] Download - [/v1/rel/download](#)

L'obiettivo dell'API **Download** è quello di permettere di scaricare il documento REL in formato PDF o HTML.

In Input bisogna interrogare l'API inserendo l'IdTestata ed il TipoDocumento.

```
1 {
2   "idTestata": "string",
3   "tipoDocumentoREL": "string"
4 }
```

In output fornirà un SAS token che punterà al documento REL in formato PDF o HTML. Nel primo caso 'application/pdf' altrimenti 'text/html'.

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Periodo - [/v1/rel/periodo](#)

L'obiettivo dell'API **Periodo** è quello di ritornare anni e mesi relativi alla regolare esecuzione.

In Input non è necessario alcun parametro e in output fornirà un array contenente gli anni e mesi relativi alle regolari esecuzioni.

```
1 [
2   {
3     "anno": 0,
4     "mese": 0,
5     "descrizione": "string"
6   }
7 ]
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK

400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Tipologia Fattura - [/v1/rel/tipolografattura](#)

L'obiettivo dell'API **TipologiaFattura** è quello di ritornare la tipologia fattura relative alla regolare esecuzione.

In Input bisogna indicare i parametri di ricerca alle notifiche (opzionali).

```

1 {
2   "anno": 0,
3   "mese": 0
4 }
```

In output fornirà un array contenente la tipologia fattura relative alla regolare esecuzione.

```

1 [
2   "string"
3 ]
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Righe - [/v1/rel/righe](#)

L'obiettivo dell'API **Righe** è quello di ritornare un SAS token per scaricare le righe REL.

In Input bisogna indicare i parametri di ricerca alle notifiche (opzionali).

```

1 {
2   "idTestata": "string"
3 }
```

In output fornirà un SAS token per scaricare le righe REL.

```
1 {
2   "linkDocumento": "string"
3 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Testata - [/v1/rel/testata](#)

L'obiettivo dell'API **Testata** è quello di ritornare la testata della REL in base ai parametri di input forniti in chiamata.

In Input bisogna indicare i parametri di ricerca alle notifiche (opzionali).

```
1 {
2   "anno": 0,
3   "mese": 0,
4   "tipologiaFattura": "string",
5   "page": 0,
6   "pageSize": 0
7 }
```

In output fornirà un array contenente i dati relativi alla REL.

```
1 {
2   "relTestate": [
3     {
4       "idTestata": "string",
5       "nomeTestata": "string",
6       "idEnte": "string",
7       "ragioneSociale": "string",
8       "idContratto": "string",
9       "tipologiaFattura": "string",
10      "anno": 0,
11      "mese": 0,
12      "totaleAnalogico": 0,
13      "totaleDigitale": 0,
14      "totaleNotificheAnalogiche": 0,
15      "totaleNotificheDigitali": 0,
16      "totaleNotifiche": 0,
17      "totale": 0,
18      "iva": 0,
```

```

19     "totaleAnalogicoIva": 0,
20     "totaleDigitaleIva": 0,
21     "totaleIva": 0,
22     "firmata": "string",
23     "asseverazioneTotaleAnalogico": 0,
24     "asseverazioneTotaleDigitale": 0,
25     "asseverazioneTotaleNotificheAnalogiche": 0,
26     "asseverazioneTotaleNotificheDigitali": 0,
27     "asseverazioneTotaleNotifiche": 0,
28     "asseverazioneTotale": 0,
29     "asseverazioneTotaleAnalogicoIva": 0,
30     "asseverazioneTotaleDigitaleIva": 0,
31     "asseverazioneTotaleIva": 0,
32     "flagConguaglio": "string"
33   }
34 ],
35   "count": 0
36 }

```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Upload - [/v1/rel/upload](#)

L'obiettivo dell'API **Upload** è quello di permettere di caricare la REL firmata con 'multipart/form-data'.

In Input bisogna passare il PDF da caricare e l'ID Testata.

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

Documenti Emessi

All'interno di questa sezione, riportiamo i dettagli tecnici e funzionali relativi alle chiamate necessarie per interrogare e scaricare i Documenti Emessi/Fatture.

Scenario

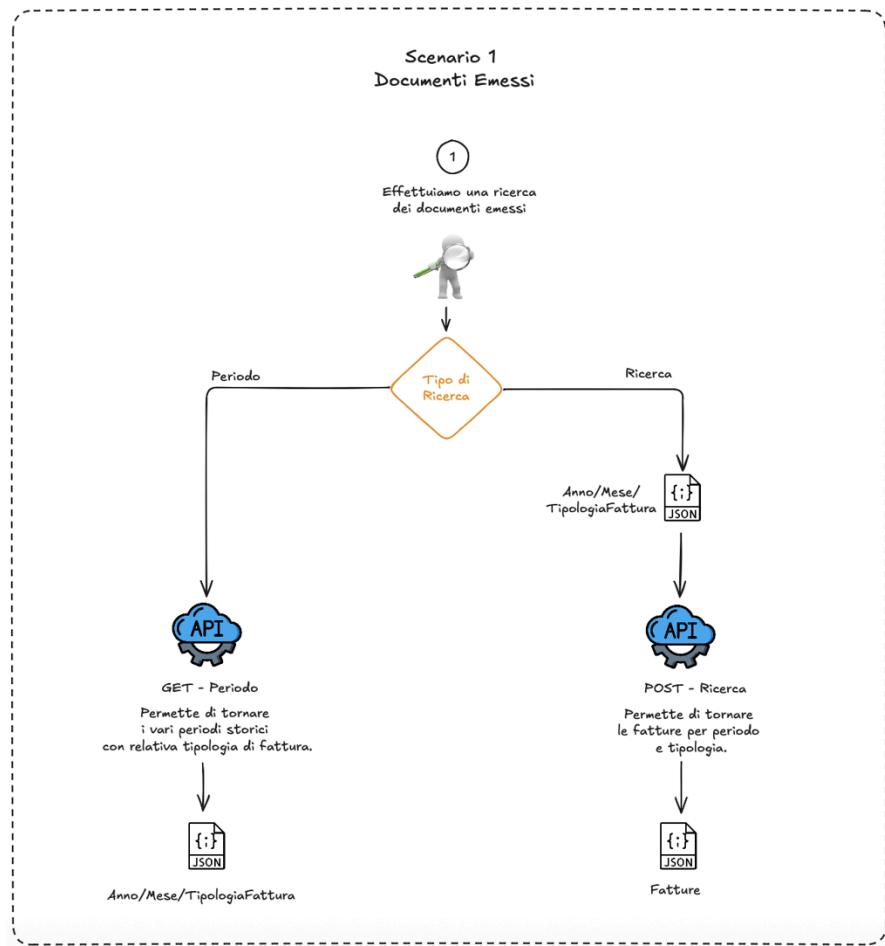


Fig.8 - Flusso logico di chiamata Scenario 1 - Documenti Emessi

Parametri di Chiamata

[POST] Ricerca - [/v1/fatture/ricerca](#)

L'obiettivo dell'API **Ricerca** è quello di ritornare i dettagli della fattura ricercata per Periodo e Tipologia.

In Input bisogna indicare i parametri di ricerca (opzionali).

```
1 {  
2   "anno": 0,  
3   "mese": 0,  
4   "tipologiaFattura": [  
5     "string"  
6   ]  
7 }
```

Di seguito le tipologie fatture disponibili:

TipologiaFattura

ACCONTO

ANTICIPO

PRIMO SALDO

SECONDO SALDO

VAR. SEMESTRALE

In output fornirà un array contenente i dati relativi alla fattura per Tipologia e Periodo.

```
1  [
2    {
3      "fattura": {
4        "totale": 0,
5        "numero": 0,
6        "idFattura": 0,
7        "dataFattura": "string",
8        "prodotto": "string",
9        "identificativo": "string",
10       "tipologiaFattura": "string",
11       "istitutioID": "string",
12       "onboardingTokenID": "string",
13       "ragioneSociale": "string",
14       "tipoContratto": "string",
15       "idContratto": "string",
16       "tipoDocumento": "string",
17       "divisa": "string",
18       "metodoPagamento": "string",
19       "causale": "string",
20       "split": true,
21       "inviata": 0,
22       "elaborazione": 0,
23       "sollecito": "string",
24       "datiGeneraliDocumento": [
25         {
26           "tipologia": "string",
27           "riferimentoNumeroLinea": "string",
28           "idDocumento": "string",
29           "data": "string",
30           "numItem": "string",
31           "codiceComessaConvenzione": "string",
32           "cup": "string",
33           "cig": "string"
34         }
35       ],
36       "posizioni": [
37         {
38           "numeroLinea": 0,
39           "testo": "string",
40           "codiceMateriale": "string",
41           "quantita": 0,
```

```

42     "prezzoUnitario": 0,
43     "imponibile": 0,
44     "periodoRiferimento": "string"
45   }
46 ]
47 }
48 ]
49 ]

```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Periodo - [/v1/fatture/periodo](#)

L'obiettivo dell'API **Periodo** è quello di ritornare i vari periodi storici con relativa tipologia di fattura.

In Input non è necessario alcun parametro e in output fornirà un array contenente i vari periodi storici con relativa tipologia di fattura.

```

1 [
2 {
3   "anno": 0,
4   "mese": 0,
5   "tipologiaFattura": "string"
6 }
7 ]

```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

Contestazione

All'interno di questa sezione, riportiamo i dettagli tecnici e funzionali relativi alle chiamate necessarie per caricare e interrogare le Contestazioni.

Scenario 1 - Caricamento Contestazione

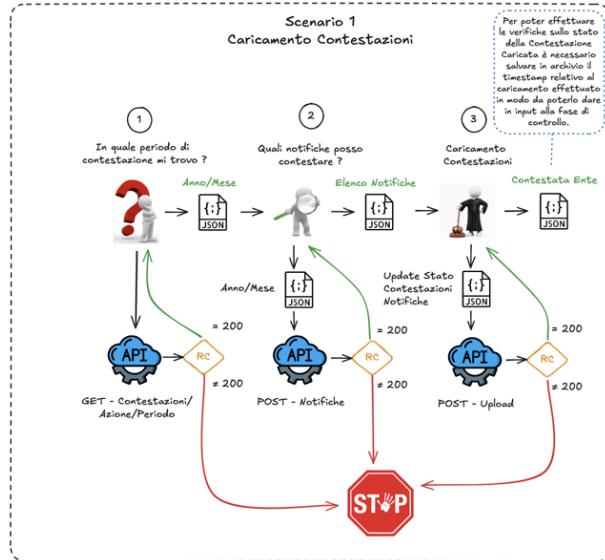


Fig.9 - Flusso logico di chiamata Scenario 1 - Contestazioni

Scenario 2 - Interrogazione Contestazioni

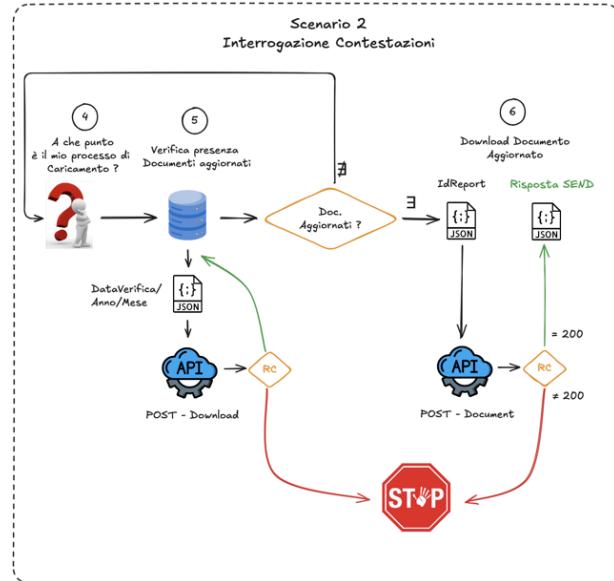


Fig.10 - Flusso logico di chiamata Scenario 2 - Contestazioni

Scenario 3 - Verifica Stato Contestazioni

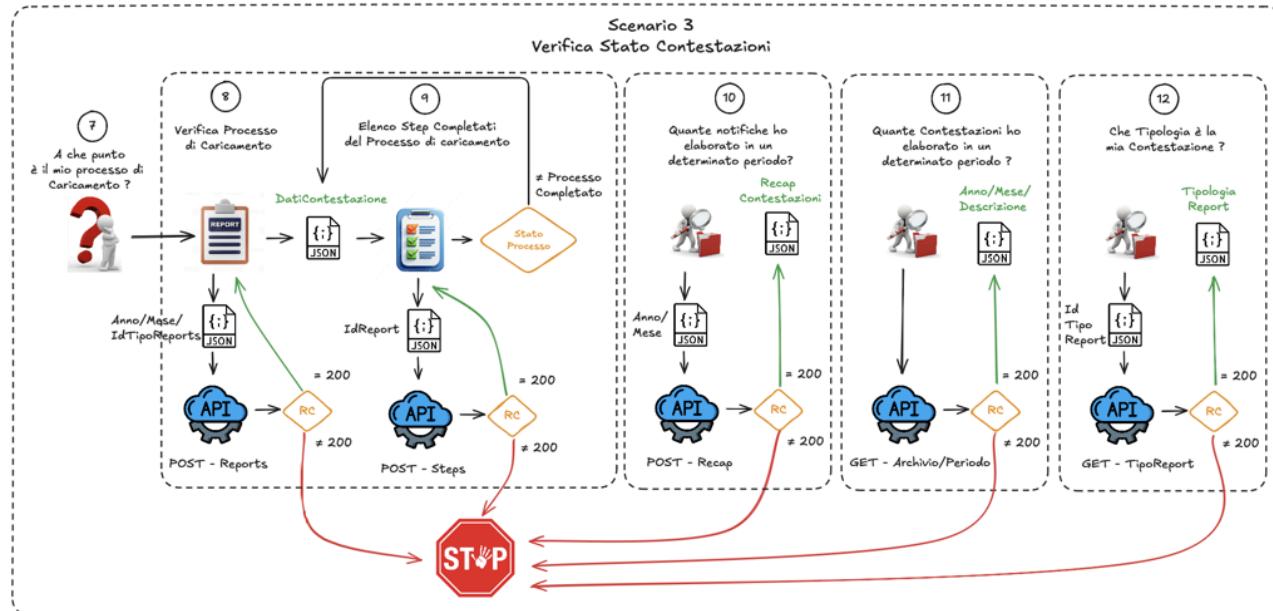


Fig.11 - Flusso logico di chiamata Scenario 3 - Contestazioni

Parametri di Chiamata

[GET] Azione/Periodo - [/v1/contestazioni/azione/periodo](#)

L'obiettivo dell'API **Azione/Periodo** è quello di ritornare anni e mesi relativi alle contestazioni solo se è incluse nel periodo di calendario attivo di contestazione.

In Input non è necessario alcun parametro e in output fornirà un array contenente gli anni e mesi relativi alle Contestazioni.

```
1 [  
2 {  
3   "anno": 0,  
4   "mese": 0,  
5   "descrizione": "string"  
6 }  
7 ]
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Archivio/Periodo - [/v1/contestazioni/archivio/periodo](#)

L'obiettivo dell'API **Archivio/Periodo** è quello di ritornare anni e mesi relativi alle contestazioni incluso lo storico di ciò che è stato contestato in passato.

In Input non è necessario alcun parametro e in output fornirà un array contenente gli anni e mesi relativi alle Contestazioni.

```
1 [  
2 {  
3   "anno": 0,  
4   "mese": 0,  
5   "descrizione": "string"  
6 }  
7 ]
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
----	-------

200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Recap - [/v1/contestazioni/recap](#)

L'obiettivo dell'API **Recap** è quello di ritornare l'elenco dei processi consolidati suddivisi per anno/mese.

In Input bisogna indicare i parametri di ricerca alle notifiche (opzionali).

```

1 {
2   "anno": 0,
3   "mese": 0
4 }
```

In output fornirà un array contenente le contestazioni consolidate dal processo.

```

1 [
2   {
3     "tipologiaFattura": "string",
4     "idFlagContestazione": 0,
5     "flagContestazione": "string",
6     "totale": 0,
7     "totaleNotificheAnalogiche": 0,
8     "totaleNotificheDigitali": 0
9   }
10 ]
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

nb. Si precisa che verrà visualizzato solo i processi consolidati all'interno della fase di caricamento e non quelli in corso.

[POST] Reports - [/v1/contestazioni/reports](#)

L'obiettivo dell'API **Reports** è quello di produrre un report con all'interno tutte le contestazioni per data/mese con la possibilità di impaginazione da parte del chiamante. Verranno restituiti tanti blocchi JSON per ogni report che è stato generato.

In Input bisogna indicare i parametri di ricerca alle notifiche (opzionali).

```
1 {
2   "anno": "string",
3   "mese": "string",
4   "idTipologiaReports": [
5     0
6   ],
7   "page": 0,
8   "pageSize": 0
9 }
```

In output fornirà un array tanti blocchi per ogni report generato.

```
1 {
2   "reports": [
3     {
4       "reportId": "string",
5       "uniqueId": "string",
6       "json": "string",
7       "anno": 0,
8       "mese": 0,
9       "internalOrganizationId": "string",
10      "contractId": "string",
11      "actualContractId": "string",
12      "utenteId": "string",
13      "prodotto": "string",
14      "stato": 0,
15      "descrizioneStato": "string",
16      "dataInserimento": "2025-07-14T13:49:33.988Z",
17      "dataStepCorrente": "2025-07-14T13:49:33.988Z",
18      "nomeDocumento": "string",
19      "linkDocumento": "string",
20      "storage": "string",
21      "hash": "string",
22      "contentType": "string",
23      "contentLanguage": "string",
24      "tipologiaDocumento": "string",
25      "categoriaDocumento": "string",
26      "ragioneSociale": "string"
27    }
28  ],
29  "count": 0
30 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

nb. Si precisa che verrà visualizzato solo i processi consolidati all'interno della fase di caricamento e non quelli in corso.

[POST] Steps - [/v1/contestazioni/reports/steps](#)

L'obiettivo dell'API **Reports/Steps** è quello di ritornare un elenco degli step relativi a un singolo report contestazioni.

In Input bisogna indicare i parametri di ricerca.

```

1  {
2    "idReport": 0
3 }
```

In output fornirà un array tanti blocchi per ogni report generato.

```

1 [
2   {
3     "reportId": 0,
4     "step": 0,
5     "descrizioneStep": "string",
6     "totaleNotificheAnalogiche890": 0,
7     "totaleNotificheAnalogicheARNazionaliAR": 0,
8     "totaleNotificheAnalogicheARInternazionaliRIR": 0,
9     "totaleNotificheAnalogicheRSNazionaliRS": 0,
10    "totaleNotificheAnalogicheRSInternazionaliRIS": 0,
11    "totaleNotificheDigitali": 0,
12    "totaleNotifiche": 0,
13    "link": "string",
14    "nonContestataAnnullata": 0,
15    "contestataEnte": 0,
16    "rispostaEnte": 0,
17    "accettata": 0,
18    "rispostaSend": 0,
19    "rispostaRecapitista": 0,
20    "rispostaConsolidatore": 0,
21    "rifiutata": 0,
22    "nonFatturabile": 0,
23    "fatturabile": 0,
24    "storage": "string",
25    "nomeDocumento": "string",
26    "dataCompletamento": "2025-07-14T13:54:40.068Z"
27  }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

nb. Si precisa che verrà visualizzato solo i processi consolidati all'interno della fase di caricamento e non quelli in corso.

[GET] Steps - [/v1/contestazioni/steps](#)

L'obiettivo dell'API **Steps** è quello di ritornare un elenco delle tipologie di contestazioni.

In Input non è necessario alcun parametro e in output fornirà un array contenente le tipologie di contestazione.

```

1 [
2   {
3     "step": 0,
4     "descrizione": "string"
5   }
6 ]

```

Le tipologie di contestazioni sono:

IdTipoContestazione	TipoContestazione
2	Mancato recapito
3	Danneggiamento o manomissione
4	Mancato espletamento servizio
5	Mancato rispetto normativa
6	Errore di stampa

7	Dematerializzazione incorretta
8	Malfunzionamento - Interruzione del Servizio
9	Mancato rispetto degli SLA (mera segnalazione)

nb. Si precisa che verrà visualizzato solo i processi consolidati all'interno della fase di caricamento e non quelli in corso.

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[GET] Tipo Report - </v1/contestazioni/tiporeport>

L'obiettivo dell'API **TipoReport** è quello di ritornare un elenco delle tipologie di report.

In Input non è necessario alcun parametro e in output fornirà un array contenente le tipologie di report.

```

1 [ 
2   {
3     "idTipologiaReport": 0,
4     "categoriaDocumento": "string",
5     "tipologiaDocumento": "string"
6   }
7 ]

```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Upload - [/v1/contestazioni/upload](#)

L'obiettivo dell'API **Upload** è quello di ritornare un SAS token per effettuare l'upload del file.

In Input è necessario inserire Anno e Mese, come di seguito:

```
1 {
2   "anno": 0,
3   "mese": 0
4 }
```

In output avremo un SAS Token dove è possibile caricare il file di Upload.

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Check/Download - [/v1/contestazioni/upload](#)

L'obiettivo dell'API **Download** è quello di fornire al chiamante un'indicazione in merito ad eventuali aggiornamenti in merito ad ultimi caricamenti effettuati.

In Input è necessario inserire Anno, Mese e data ultimo download notifiche o caricamento contestazioni, come di seguito:

```
1 {
2   "dataVerifica": "2025-07-28T12:18:34.945Z",
3   "anno": 0,
4   "mese": 0
5 }
```

In output avremo l'indicazione con DataMassimaContestazione, NotificheDaScaricare (OK/KO) e se CiSonoContestazioni(SI/NO).

```
1 {
2   "dataMassimaContestazione": "2025-07-28T13:57:52.290Z",
3   "notificheDaScaricare": true,
4   "ciSonoContestazioni": true
5 }
```

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.

[POST] Reports/Document - [/v1/contestazioni/upload](#)

Ritorna SAS token per upload file.

L'obiettivo dell'API **Document** è quello di tornare un SAS token con all'interno un documento singolo, con step e report.

In Input è necessario IdReport e Step:

```

1 {
2   "idReport": 0,
3   "step": 0
4 }
```

In output avremo un SAS token con all'interno un documento singolo, con step e report.

I possibili ReturnCode restituibili dall'API sono i seguenti:

RC	STATO
200	OK
400	Error logging data missing or malformed.
401	Unauthorized: Invalid or missing API Key.
403	Forbidden: Invalid or missing IP address.