



UNIONCAMERE

Unione italiana delle Camere di commercio, industria, artigianato e agricoltura

Ministero della Transizione Ecologica

DIREZIONE GENERALE PER L'ECONOMIA CIRCOLARE (ECi)

“Divisione IV - Pianificazione, tracciabilità e vigilanza sulla gestione dei rifiuti”

“Comitato Nazionale dell'Albo dei gestori ambientali”

Registro Elettronico Nazionale

Tracciabilità dei Rifiuti

Laboratorio Sperimentale Prototipazione Funzionale

DECRETO-LEGGE 14 dicembre 2018, n. 135

Disposizioni urgenti in materia di sostegno e semplificazione per le imprese
e per la pubblica amministrazione.

“Art. 6 - Disposizioni in merito alla tracciabilità dei dati ambientali inerenti rifiuti”
(come modificato in sede di conversione dalla Legge 11 febbraio 2019, n. 12)

Applicazione delle linee di indirizzo AGID

Modello di interoperabilità tecnica

Gestione movimenti nei registri

23 giugno 2021

Versione: 05-00

Data: 04-04-2022

Identificatore: 20210706_ModInterop_Movimenti.RENTRI.v05.BOZZA.docx

Redattori: Michele Rubin, Filippo Babolin

Approvato da: Maurizio Zampieri

Contatti: itec@ecocerved.it



Sommario

1. NOVITÀ INTRODOTTE RISPETTO ALLA PRECEDENTE EMISSIONE.....	3
2. RIFERIMENTI	3
2.1. DOCUMENTI.....	3
2.2. LINK.....	3
2.3. TERMINI E DEFINIZIONI	3
3. OBIETTIVI	4
4. PREREQUISITI.....	4
5. API.....	4
5.1. CATALOGO API.....	6
5.2. STRUTTURA DEL MODELLO RELAZIONALE DEL DATABASE E MEMORIZZAZIONE DEI MOVIMENTI	6
5.3. LOG DELLE REQUEST E DELLE RESPONSE	8
5.4. GESTIONE DELLA CATENA (CHAIN) DELLE RICHIESTE ACCETTATE.....	8
5.5. WORKFLOW TIPICI.....	8
5.5.1. <i>Caricamento di un insieme di movimenti in un registro.....</i>	<i>8</i>
5.5.2. <i>Rettifica di un insieme di movimenti di un registro.....</i>	<i>10</i>
5.5.3. <i>Comunicazione della “quantità verificata a destino” per i movimenti di un registro ...</i>	<i>10</i>
5.6. VALIDAZIONE ED ERRORI GESTITI	11
6. ASSISTENZA.....	11

Avvertenza

- I documenti sono disponibili in copia magnetica originale sul sito di progetto.
- Ogni copia cartacea si ritiene copia non controllata ed è responsabilità di chi utilizza copie non controllate verificarne il livello di aggiornamento.
- Le informazioni contenute in questo documento sono di esclusiva proprietà di UNIONCAMERE
- Questo documento non può essere riprodotto per intero o in parte senza il consenso scritto di UNIONCAMERE



1. NOVITÀ INTRODOTTE RISPETTO ALLA PRECEDENTE EMISSIONE

Versione/Release n°:	05-00	Data Versione/Release:	04-04-2022
Nome documento:	20210706_ModInterop_Movimenti.RENTRI.v05.BOZZA.docx		
Descrizione modifiche:	Aggiornata la tabella degli endpoint a seguito del rilascio delle API v1.3.		
Motivazioni:	Quarta emissione		

2. RIFERIMENTI

2.1. Documenti

[d1]	Modello di interoperabilità – Regole tecniche 00 linea di indirizzo interoperabilita tecnica.pdf (agid.gov.it) 01 pattern interazione 0.pdf (agid.gov.it) 02 pattern sicurezza.pdf (agid.gov.it) 03 profili di interoperabilita.pdf (agid.gov.it) 04 raccomandazioni di implementazione.pdf (agid.gov.it)
[d2]	Applicazione delle linee di indirizzo - AGID Modello di interoperabilità tecnica - Utilizzo, configurazione e test del servizio 20210505_ModInterop.RENTRI_v01.pdf
[d3]	Registri di Carico / Scarico - Comunicazione dati al R.E.N.T.Ri. 20210522_Registri-CS.pdf

2.2. Link

[L0]	Linee Guida Tecnologie e standard sicurezza API Linee Guida Tecnologie e standard per la sicurezza dell'interoperabilità tramite API dei sistemi informatici (italia.it) https://docs.italia.it/italia/piano-triennale-ict/lg-modellointeroperabilita-docs/it/bozza/index.html
[L1]	Linee Guida sull'interoperabilità tecnica Linee Guida sull'interoperabilità tecnica delle Pubbliche Amministrazioni (italia.it) https://docs.italia.it/italia/piano-triennale-ict/lg-modellointeroperabilita-docs/it/bozza/index.html
[L2]	Representational State Transfer - Wikipedia https://it.wikipedia.org/wiki/Representational_State_Transfer
[L3]	RFC 7519 - JSON Web Token (JWT) (ietf.org) https://tools.ietf.org/html/rfc7519
[L4]	RFC 7807 - Problem Details for HTTP APIs (ietf.org) https://tools.ietf.org/html/rfc7807

2.3. Termini e definizioni

Termine	Definizione
AgID	Agenzia per l'Italia digitale



UI	Interfaccia utente
REST	Representational state transfer – architettura per la trasmissione di dati su HTTP senza ulteriori livelli, quali ad esempio SOAP
API	Application programming interface: insieme di procedure atte all’espletamento di un dato compito
Endpoint	Singolo punto di ingresso di una API
Payload	Contenuto della richiesta, tipicamente in formato JSON contenente quindi i dati ricevuti (nelle request) e trasmessi (nelle response)

3. OBIETTIVI

In questo documento verranno indicate le modalità, relativamente ai **pattern di interazione**, per la gestione dei movimenti tramite API nei registri di carico/scarico definiti nel sistema RENTRI.

Non si descrivono i pattern di sicurezza e di integrità e il loro utilizzo, già oggetto dello specifico documento [d2].

Anche gli aspetti base dell’architettura sono descritti in [d2] e quindi sono considerati già definiti come, ad esempio, l’utilizzo dell’architettura “RESTful API”, il fatto che i messaggi sono trasmessi in formato JSON.

4. PREREQUISITI

Il fruitore del servizio, ossia il soggetto che interagisce tramite l’interoperabilità con il sistema, deve essere stato preventivamente accreditato al sistema. L’accreditamento preventivo deve essere effettuato tramite procedura web messa a disposizione attraverso il portale RENTRI, e permette di associare il fruitore del servizio applicativo al soggetto per il quale lo stesso opera.

Nell’ambito dell’interoperabilità, il riconoscimento del fruitore si basa sullo scambio di header nella richiesta HTTP. Tali header sono firmati tramite un certificato X.509 rilasciato da una CA riconosciuta, in modo da garantire l’adeguato livello di sicurezza ed integrità della trasmissione dei dati.

Inoltre, il fruitore deve aver definito, sempre tramite procedura web messa a disposizione attraverso il portale RENTRI uno o più registri nei quali vanno inseriti i movimenti.

5. API

L’API rilasciata in una prima versione è implementata secondo il pattern di interazione bloccante AgID [BLOCK_REST] (vedi [d1]) ritenuto adeguato in relazione agli obiettivi del prototipo.

È comunque possibile prevedere fin da ora l’evoluzione verso un pattern di interazione non bloccante ([NONBLOCK_PUSH_REST] oppure [NONBLOCK_PULL_REST]) più adatto a supportare la scalabilità del sistema nell’ottica del rilascio del sistema in versione non più per fini prototipali / sperimentali.

Per la gestione dei movimenti sono attualmente a disposizione i seguenti endpoint della **API v1.3**:



Metodo HTTP	Endpoint	Descrizione
GET	/api/v1.3/soggetti	Ottiene un elenco dei soggetti gestibili da un fruitore
GET	/api/v1.3/soggetti/associazioni-categoria/soggetti-associati/{identificativoAssociazione}	Ottiene l'elenco di soggetti e siti che hanno delegato l'Associazione di Categoria
POST	/api/v1.3/soggetti/associazioni-categoria/soggetti-associati/{identificativoAssociazione}	Carica una richiesta di deleghe per un'Associazione di Categoria e restituisce l'URL per chiedere lo stato di elaborazione
DELETE	/api/v1.3/soggetti/associazioni-categoria/soggetti-associati/{identificativoAssociazione}	Effettua la dissociazione tra un'Associazione di Categoria ed un sito
GET	/api/v1.3/soggetti/associazioni-categoria/soggetti-associati/{idRichiesta}/status	Restituisce lo stato di elaborazione di una richiesta di deleghe di un'Associazione di Categoria
GET	/api/v1.3/soggetti/associazioni-categoria/soggetti-associati/{idRichiesta}/result	Restituisce l'esito dell'elaborazione di una richiesta di deleghe di un'Associazione di Categoria
GET	/api/v1.3/strutture	Ottiene un elenco delle strutture riferite ad uno specifico soggetto ed accessibili al fruitore
GET	/api/v1.3/registri	Ottiene un elenco dei registri attivi per il fruitore
POST	/api/v1.3/registri/associazioni-categoria/registri-associati/{identificativoStruttura}	Crea un nuovo registro gestito da un'Associazione di Categoria
PUT	/api/v1.3/registri/associazioni-categoria/registri-associati/{identificativoRegistro}	Modifica i dati di un registro gestito da un'Associazione di Categoria
DELETE	/api/v1.3/registri/associazioni-categoria/registri-associati/{identificativoRegistro}	Elimina un registro gestito da un'Associazione di Categoria
PUT	/api/v1.3/registri/associazioni-categoria/registri-associati/{identificativoRegistro}/chiudi	Chiude un registro gestito da un'Associazione di Categoria
GET	/api/v1.3/registri/{identificativoRegistro}/last-idchain	Ottiene l'ultimo "anello della catena" al quale agganciarsi per effettuare la successiva operazione di scrittura relativa ai movimenti in un registro. Si veda il cap. 5.4 per maggiori informazioni
GET	/api/v1.3/registri/{identificativoRegistro}/movimenti	Ottiene l'elenco dei movimenti relativi ad un Registro, filtrati in base ai criteri specificati e specificando un valore per il filtro dataAl, si ottiene l'elenco dei movimenti correnti a quel momento
GET	/api/v1.3/registri/{identificativoRegistro}/movimenti/count	Ottiene il conteggio dei movimenti relativi ad un Registro, filtrati in base ai criteri specificati e specificando un valore per il filtro dataAl, si ottiene il conteggio dei movimenti correnti a quel momento



GET	/api/v1.3/registri/{identificativoRegistro}/movimento-dettaglio/{identificativoMovimento}	Ottiene il dettaglio di un movimento e specificando un valore per il filtro dataAl, si ottiene il dettaglio del movimento corrente a quel momento
GET	/api/v1.3/registri/{identificativoRegistro}/movimento-dettaglio/{anno}/{progressivo}	Ottiene il dettaglio di un movimento e specificando un valore per il filtro dataAl, si ottiene il dettaglio del movimento corrente a quel momento
GET	/api/v1.3/registri/{identificativoRegistro}/movimento-history/{identificativoMovimento}	Ottiene la storia di un movimento (tutte le sue variazioni)
GET	/api/v1.3/registri/{identificativoRegistro}/movimento-history/{anno}/{progressivo}	Ottiene la storia di un movimento (tutte le sue variazioni)
POST	/api/v1.3/registri/{identificativoRegistro}/movimenti	Crea un insieme di movimenti in un registro
PUT	/api/v1.3/registri/{identificativoRegistro}/movimenti	Rettifica un insieme di movimenti in un registro (anche quelli precedentemente annullati)
PUT	/api/v1.3/registri/{identificativoRegistro}/movimenti-quantita-verificate	Aggiorna le informazioni relative alla Quantità verifica a destino per un insieme di movimenti in un registro
PUT	/api/v1.3/registri/{identificativoRegistro}/movimenti-annullamenti	Annulla un insieme di movimenti

Tabella 1 - Principali endpoint disponibili per la gestione dei movimenti delle API v1.3

I dettagli descrittivi di questi endpoint sono resi disponibili in un catalogo come specificato nel capitolo successivo.

Oltre a questi endpoint, sono resi disponibili altri relativi alle tabelle di lookup utilizzate dal sistema, in grado di fornire i valori dei dizionari che saranno accettati nelle request, ovvero che saranno ritornati nelle response dell'API. Si rimanda sempre al catalogo disponibile online per i dettagli.

5.1. Catalogo API

Il catalogo delle API viene esposto tramite la specifica OpenApi 3, ed è navigabile anche attraverso l'interfaccia Swagger apponendo dopo l'indirizzo del servizio la rotta `"/swagger"`.

È attraverso questo catalogo che sono forniti tutti i dettagli degli endpoint resi disponibili.

Di seguito vengono indicati gli indirizzi dei servizi nei vari ambienti disponibili.

Test: https://testprototipo.rentri.it/api/swagger
Demo: https://demoprototipo.rentri.it/api/swagger
Produzione: https://prototipo.rentri.it/api/swagger

5.2. Struttura del modello relazionale del database e memorizzazione dei movimenti

Limitatamente alla parte del database che afferisce alla gestione dei movimenti nei registri di carico/scarico, l'organizzazione delle informazioni si presenta come nel disegno seguente, nel quale si espongono dal punto di vista concettuale solo le entità più rappresentative.

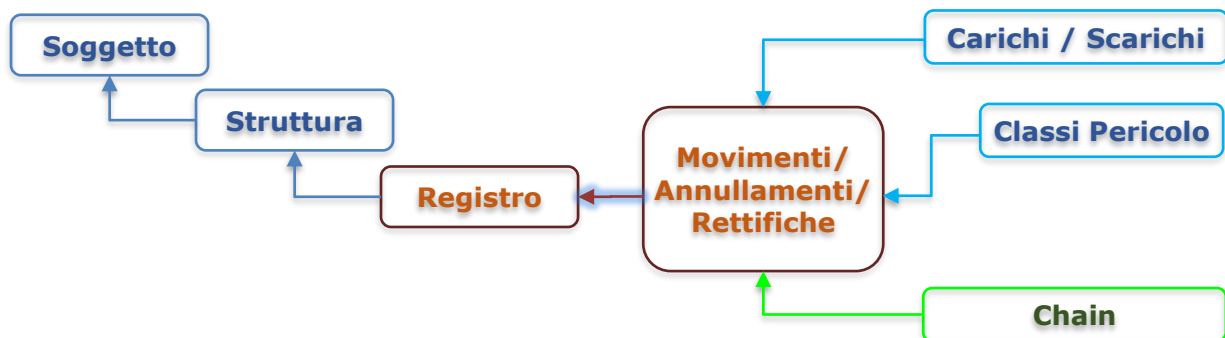


Figura 1: entità database più rappresentative dei movimenti

- **Soggetti:** è la tabella anagrafica dei soggetti censiti in RENTRI dove per soggetto si intende la sede legale di una impresa, un ente pubblico oppure una organizzazione, associazione o professionista. Tipicamente un soggetto ha come identificativo univoco il suo codice fiscale;
- **Strutture:** è la tabella anagrafica delle organizzazioni associate ai soggetti. Ogni struttura ha un identificativo univoco generato da RENTRI;
- **Registri:** è la tabella dei registri. Ogni registro ha un identificativo univoco con formato “REGnnnA”, utilizzabile per le request quando è indicato {identificativoRegistro}. Ogni registro è associato a una struttura;
- **Movimenti:** è la tabella principale dei movimenti alla quale sono relazionate altre tabelle, non rappresentate per semplicità, per la memorizzazione di informazioni strutturate in relazione 1 a molti, quali le classi di pericolo. Ogni movimento ha un identificativo univoco generato dal sistema ed utilizzabile nelle request quando è indicato {identificativoMovimento}. In alternativa, i movimenti possono essere identificati anche tramite la coppia di valori Anno/Progressivo;
- **Classi Pericolo:** consente di definire la correlazione tra un movimento e le sue classi di pericolosità;
- **Carichi / Scarichi:** consente di definire la correlazione tra un movimento di carico e uno di scarico;
- **Chain:** è una particolare tabella di servizio che consente la gestione della “catena” delle richieste di scrittura dei dati dei movimenti rispettando la sequenzialità effettuata dai fruitori. Per maggiori dettagli si veda il cap. 5.4.

Il sistema tiene traccia di tutte le versioni precedenti dei Movimenti, quindi le varie Rettifiche e gli eventuali Annullamenti. Per quanto riguarda le rettifiche, viene memorizzato il timestamp [*DataUltimaRettifica*] in formato UTC (*Coordinated Universal Time*) dell'ultima rettifica applicata. Per quanto riguarda gli annullamenti, viene memorizzato il timestamp [*DataAnnullamento*] in formato UTC (*Coordinated Universal Time*) dell'avvenuto annullamento, il quale rappresenta di fatto una cancellazione logica del movimento.

In ogni momento, il sistema è in grado di ricostruire lo stato attuale di un movimento e delle relative relazioni, così come l'intera cronologia delle modifiche subite nel tempo.

Ogni inserimento porta il timestamp [*DataCreazione*] in formato UTC (*Coordinated Universal Time*) che identifica l'istante esatto della scrittura e sono correlabili in modo certo alle request effettuate attraverso le API, tutte tracciate nel log.



5.3. Log delle request e delle response

È prevista la registrazione di tutte le request e di tutte le rispettive response correlate tra di loro, complete di tutte le informazioni relative all'identificazione certa delle fonti di provenienza, del timestamp di esecuzione e del payload. Tali informazioni possono essere utilizzate per ricostruire l'esatta sequenza che ha prodotto la scrittura di tutti i dati relativi ai movimenti nei registri.

5.4. Gestione della catena (chain) delle richieste accettate

Al fine di garantire la totale tracciabilità tra le request effettuate tramite le API che inseriscono dati relativi ai movimenti, è previsto un sistema di validazione della "catena" delle richieste. Tale sistema ha l'obiettivo di assicurare che tutte le richieste che applicano modifiche ai dati dei movimenti di uno specifico registro siano concatenate, sequenziali, e con la possibilità di dimostrare la loro immutabilità.

Relativamente a questo ultimo aspetto, si fa riferimento a quanto descritto nel documento [d1] e specificatamente al fatto che tutte le request e tutte le response sono firmate digitalmente, comprendendo anche un digest SHA256 del body della request ovvero di quello della response.

Per la concatenazione, il meccanismo prevede il rilascio di un valore univoco di tipo GUID legato allo specifico registro, per ogni response andata a buon fine (HTTP status 20x) relativa alle chiamate che operano sui movimenti. Tale valore è presente nell'header HTTP custom denominato **X-Rentri-IdChain-New**. Questo valore deve essere inserito nella successiva request di una chiamata che opera sui movimenti, nell'header http custom denominato **X-Rentri-IdChain**.

Per la prima richiesta sarà previsto un valore iniziale generato automaticamente con la creazione del registro, recuperabile tramite l'endpoint `/registri/{identificativoRegistro}/last-idchain`.

5.5. Workflow tipici

Sono realizzabili diverse procedure attraverso l'implementazione di chiamate agli endpoint resi disponibili. Di seguito si descrive un primo workflow ma altri tipici saranno aggiunti per rappresentare ulteriori possibili modalità di comunicazione dei movimenti.

5.5.1. Caricamento di un insieme di movimenti in un registro

La creazione dei movimenti è sicuramente l'operazione principale effettuata in un registro. L'esecuzione di una request **POST** con l'endpoint:

`.../api/v1.3/registri/{identificativoRegistro}/movimenti`

prevede i seguenti step principali lato erogatore;

- 1) Validazione dell'autenticazione. Si veda [d2] per i dettagli;
- 2) Validazione del registro {identificativoRegistro}: l'utente deve essere autorizzato all'accesso in scrittura e il registro deve essere aperto;
- 3) Validazione del GUID della chain, ottenuto secondo quanto indicato nel capitolo 5.4;
- 4) Validazione del payload trasmesso (validazione semantica del modello);
- 5) Verifica che il numero di movimenti sia > 0 e minore o uguale di un numero massimo (*da definire*);
- 6) Elaborazione del payload. Tale payload prevede un array di movimenti (si veda il catalogo API, **schema CreateMovimento**, per i dettagli). Quindi, in transazione:
 - a. Per ogni movimento si validano tutte le proprietà per tutti i movimenti ovvero si applicano tutte le validazioni previste;
- 7) Se non ci sono errori:



- a. Si procede alla scrittura dell'intero pacchetto, tramite l'inserimento di record nelle tabelle previste;
- b. Viene generato il GUID della chain associato alla response accettata;
- c. La response ritorna un **HTTP status 201 Ok**, un array con tutti gli identificativi dei movimenti generati e, nell'header **X-Rentri-IdChain-New**, il GUID della chain da utilizzare nella successiva chiamata.

8) Se ci sono errori:

- a. Non viene scritto alcun dato (ad esclusione dei log);
- b. La response ritorna un **HTTP status 400 Bad Request** e gli errori specifici dell'insieme dei movimenti ovvero di specifici movimenti;
- c. NON viene generato un nuovo GUID della chain.

Di seguito un caso d'uso semplificato, rappresentato come diagramma di sequenza, per il caricamento progressivo di più insiemi di movimenti nello stesso registro. Per semplicità sono stati omessi gli step relativi al processo di autenticazione applicativa che quindi si intende andato a buon fine. Si noti la rappresentazione dell'utilizzo dell'header **X-Rentri-IdChain** e il ritorno del successivo "anello della catena" con l'header **X-Rentri-IdChain-New**.

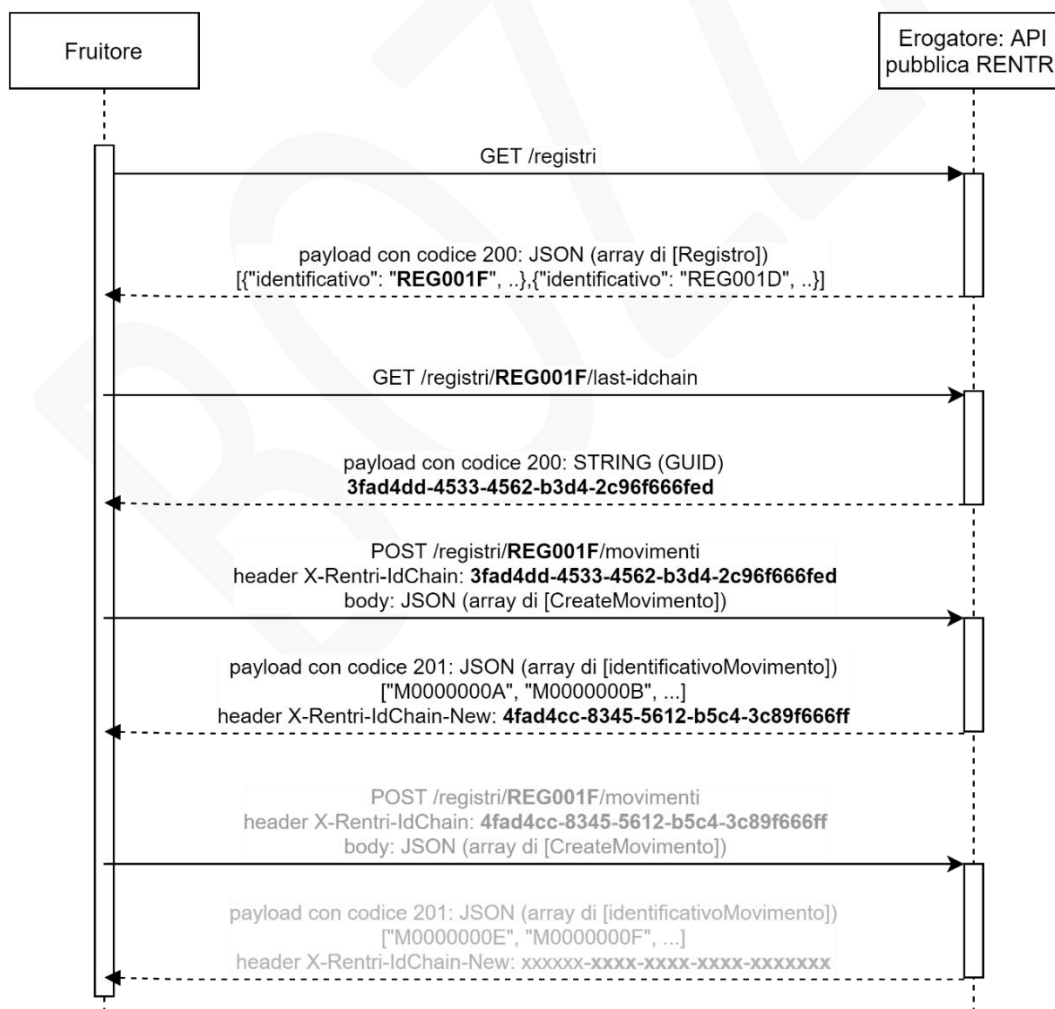


Figura 2: diagramma di sequenza semplificato per la creazione dei movimenti in un registro



5.5.2. Rettifica di un insieme di movimenti di un registro

Il sistema prevede che sia possibile rettificare tutte le informazioni di uno o più movimenti (anche precedentemente annullati), **ad esclusione dei valori di Anno e Progressivo**.

L'esecuzione di una request **PUT** con l'endpoint:

`.../api/v1.3/registri/{identificativoRegistro}/movimenti`

prevede i seguenti step principali lato erogatore;

- 1) Validazione dell'autenticazione. Si veda [d2] per i dettagli;
- 2) Validazione del registro {identificativoRegistro}: l'utente deve essere autorizzato all'accesso in scrittura e il registro deve essere aperto;
- 3) Validazione del GUID della chain, ottenuto secondo quanto indicato nel capitolo 5.4;
- 4) Validazione del payload trasmesso (validazione semantica del modello);
- 5) Verifica che il numero di movimenti sia > 0 e minore o uguale di un numero massimo (*da definire*);
- 6) Elaborazione del payload. Tale payload prevede un array rettifiche (si veda il catalogo API, **schema CreateMovimentoRettifica** per i dettagli). Quindi, in transazione:
 - a. Per ogni movimento si validano tutte le proprietà per tutti i movimenti ovvero si applicano tutte le validazioni previste. In particolare, l'**IdentificativoMovimento** o la coppia di valori **Anno/Progressivo**, devono permettere di identificare un movimento esistente e questo deve essere gestibile dal fruitore;
- 7) Se non ci sono errori:
 - a. Si procede alla scrittura dell'intero pacchetto, tramite l'aggiornamento dei record nelle tabelle previste;
 - b. Viene generato il GUID della chain associato alla response accettata;
 - c. La response ritorna un **HTTP status 204 No Content** e, nell'header **X-Rentri-IdChain-New**, il GUID della chain da utilizzare nella successiva chiamata.
- 8) Se ci sono errori:
 - a. Non viene scritto alcun dato (ad esclusione dei log);
 - b. La response ritorna un **HTTP status 400 Bad Request** e gli errori specifici dell'insieme dei movimenti ovvero di specifici movimenti;
 - c. NON viene generato un nuovo GUID della chain.

5.5.3. Comunicazione della "quantità verificata a destino" per i movimenti di un registro

Il sistema prevede che sia possibile comunicare la quantità verificata a destino relativamente ad uno o più movimenti.

L'esecuzione di una request **PUT** con l'endpoint:

`.../api/v1.3/registri/{identificativoRegistro}/movimenti-quantita-verificate`

prevede i seguenti step principali lato erogatore;

- 1) Validazione dell'autenticazione. Si veda [d2] per i dettagli;



- 2) Validazione del registro {identificativoRegistro}: l'utente deve essere autorizzato all'accesso in scrittura e il registro deve essere aperto;
- 3) Validazione del GUID della chain, ottenuto secondo quanto indicato nel capitolo 5.4;
- 4) Validazione del payload trasmesso (validazione semantica del modello);
- 5) Verifica che il numero di movimenti riferiti sia > 0 e minore o uguale di un numero massimo (da definire);
- 6) Elaborazione del payload. Tale payload prevede un array di quantità verificate (si veda il catalogo API, **schema UpdateQuantitaVerificata** per i dettagli). Quindi, in transazione:
 - a. Per ogni movimento si validano tutte le proprietà per tutti i movimenti ovvero si applicano tutte le validazioni previste. In particolare, l'**IdentificativoMovimento** o la coppia di valori **Anno/Progressivo**, devono permettere di identificare un movimento esistente e questo deve essere gestibile dal fruitore;
- 7) Se non ci sono errori:
 - a. Si procede alla scrittura dell'intero pacchetto, tramite l'aggiornamento dei record nelle tabelle previste;
 - b. Viene generato il GUID della chain associato alla response accettata;
 - c. La response ritorna un **HTTP status 204 No Content** e, nell'header **X-Rentri-IdChain-New**, il GUID della chain da utilizzare nella successiva chiamata.
- 8) Se ci sono errori:
 - a. Non viene scritto alcun dato (ad esclusione dei log);
 - b. La response ritorna un **HTTP status 400 Bad Request** e gli errori specifici dell'insieme dei movimenti ovvero di specifici movimenti;
 - c. NON viene generato un nuovo GUID della chain.

5.6. Validazione ed errori gestiti

Nel processamento eseguito a seguito delle richieste possono essere ritornati degli errori. Le regole generali applicate, previste dal pattern di interazione AgID [BLOCK_REST] identificano i seguenti casi:

- in caso di dati errati, si restituisce **HTTP status 400 Bad Request** fornendo nel body di risposta dettagli circa gli errori;
- In caso in caso di rappresentazione semanticamente non corretta, si ritorna **HTTP status 422 Unprocessable Entity**;
- in caso di errori non dipendenti dalla richiesta, si restituisce **HTTP status 5xx**.

Per l'identificazione e descrizione degli errori previsti dal sistema, è disponibile un endpoint dedicato:

[.../api/v1.0/lookup/codici-errore](#)

6. ASSISTENZA

Per supporto tecnico ed ogni altra richiesta di assistenza, è necessario utilizzare la sezione "Assistenza" accessibile dalla home page del portale.

Le richieste inoltrate saranno gestite dallo stesso staff di supporto, indifferentemente che siano state inserite dall'area dimostrativa oppure pervengano dall'area di lavoro effettiva.



The screenshot shows a web form titled "Assistenza" (Assistance) within a system interface. The form contains the following elements:

- Email***: A text input field with a red error message below it: "Verificare la corretta digitazione dell'indirizzo email".
- Reinscrisci email***: A text input field.
- Priorità**: A dropdown menu with the option "[Selezionare Priorità]".
- Contesto***: A dropdown menu with the option "[Selezionare Contesto]".
- Descrizione***: A large text area for entering details.
- Allegati**: A section with a green "Acciungi" (Add) button.
- Codice Captcha***: A captcha field showing the number "6542".
- INVIA**: A blue button at the bottom of the form.

Figura 3 - Form per la richiesta di assistenza