MEXAL

PASSCOM

MANUALE WEBAPI PASSEPARTOUT

Versione 3.2

6 ottobre 2026

PASSEPARTOUT®
Software as a Service

Sommario

Introduzione 7
Cosa è 7
Glossario 8
Come si usa 8
Configurazione e Accesso 10
Installazioni locali 10
Installazioni LIVE 12
Concetti Generali 14
Autenticazione 14
Formato Request 15
Formato Response 17
Gestione Errori 20
Paginazione 23
Ricerca e Filtri 26
Controllo sulla Validità dei Dati (Data Ultima Modifica) 30
Guida Rapida 32
Quick Start 32
Riferimento Endpoint 36
Entità: Alias Articoli 36
Entità: Anagrafica Contatti 44
Entità: Anagrafica Unica 54
Entità: Articoli 71
Entità: Articoli Abbinati 86
Entità: Aziende 90
Entità: Banche 95
Entità: Clienti 100
Entità: Fornitori 108
Entità: Conti 116
Entità: Dati Generali 126
Entità: DBA (Distinte Base Articoli) 141
Entità: Distinte Base 144

Entità: Documenti ... 158
Entità: Impegni Bolle di Lavorazione ... 184
Entità: Indirizzi Spedizione ... 196
Entità: Liste Prelievo ... 206
Entità: Lotti ... 216
Entità: Mappa Articoli ... 226
Entità: MyDB ... 231
Entità: Prima Nota ... 240
Entità: Progressivi Articoli ... 254
Entità: Referenti ... 257
Entità: Scadenzario ... 268
Entità: Ubicazioni ... 274
Canale Servizi ... 286
Introduzione ... 286
Differenza tra Endpoint REST e Canale Servizi ... 286
Formato della Request ... 286
Paginazione ... 287
Header obbligatori ... 287
Elenco dei Servizi ... 287
Servizio: Collage Server Remoto ... 288
Servizio: Sviluppo Distinta Base ... 289
Servizio: Avanzamento di Produzione ... 291
Servizio: Spezza Riga Bolle di Lavorazione ... 293
Servizio: Condizioni Documento ... 294
Servizio: Calcolo Esposizione ... 297
Servizio: Progressivi totale Ubicazioni ... 300
Servizio: Totali Documento e Totali Riga ... 302
Servizio: Località Italiane (CAP e Zone) ... 305
Servizio: Conteggio Operazioni Contabili ... 307
Servizio: Pratiche Dichiarativi ... 308
Servizio: Deleghe Dichiarativi ... 310
Servizio: Sblocco deleghe ... 317
Servizio: Registro Cancellati ... 319

Servizio: Dichiarazioni IVA Annuali 322
Servizio: Lettura Comunicazioni LIPE 324
Servizio: Stampa PDF di Documenti 325
Servizio: Inserimento Righe nei Documenti 327
Servizio: Lista Allegati Docuvision 331
Servizio: Lettura Allegati Docuvision 338
Servizio: Associazioni Docuvision 341
Servizio: Lista Strutture MyDB 342
Servizio: Proprietà Numeratori 344
Servizio: Autenticazione MyDB 346
Servizio: Calendari di Produzione 347
Servizio: Multilotto 348
Servizio: Modifica IBAN e Tipo Invio 349
Servizio: Inserimento-Aggiornamento IBAN 352
Servizio: Storia Ordine 355
Servizio: Upload Files 357
Appendici 361
Codici di Errore 361
Glossario 369
Changelog v3.2 372
Changelog v3.1 373
Changelog v3.0 374
Changelog v2.34 374
Changelog v2.33 375
Changelog v2.32 375
Changelog v2.31 376
Changelog v2.30 377
Changelog v2.29 377
Changelog v2.28 378
Changelog v2.27 379
Changelog v2.26 379
Changelog v2.25 380
Changelog v2.24 380

Changelog v2.23 381
Changelog v2.22 382
Changelog v2.21 383
Changelog v2.20 384
Changelog v2.19 384
Changelog v2.18 385
Changelog v2.17 386
Changelog v2.16 387
Changelog v2.15 387
Changelog v2.14 388
Changelog v2.13 389
Changelog v2.12 390
Changelog v2.11 390
Changelog v2.10 391
Changelog v2.9 392
Changelog v2.8 392
Changelog v2.7 393
Changelog v2.6 395
Changelog v2.5 396
Changelog v2.4 397
Changelog v2.3 397
Changelog v2.2 398
Changelog v2.1 398
Changelog v2.0 399
Changelog v1.10 400
Changelog v1.9 401
Changelog v1.8 402
Changelog v1.7 403
Changelog v1.6 404
Changelog v1.5 405
Changelog v1.4 405
Changelog v1.3 406
Changelog v1.2 406

Changelog v1.1 407

Introduzione

Cosa è

WebApi è, come suggerisce il nome, una collezione di API Passepartout basate su tecnologia Web tramite protocollo https. Il servizio è fruibile utilizzando richieste https secondo lo stile architetturale RESTful: **R**epresentational **S**tate **T**ransfer. Sono implementati tutti i metodi CRUD (**C**reate, **R**ead, **U**pdate, **D**elete) per mezzo dei metodi http `POST`, `GET`, `PUT`, `DELETE`.

Secondo il paradigma REST il servizio è fortemente orientato alle entità (gli archivi del gestionale) le quali saranno raggiungibili mediante path specifici.

Documentazione integrata (Self-Discovery)

Le Passepartout WebAPI **non espongono uno schema OpenAPI/Swagger**. La documentazione degli endpoint e dei campi disponibili è invece integrata direttamente nel servizio stesso attraverso due meccanismi:

1. **Elenco endpoint disponibili** — una richiesta GET sull'entità speciale `help` restituisce la lista completa di tutti gli endpoint con i metodi supportati:

```
GET /webapi/risorse/help HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
```

   Aggiungendo il parametro `extended=true` si ottengono informazioni aggiuntive per ciascun endpoint (supporto paginazione, encoding, chiavi, ecc.):

```
GET /webapi/risorse/help?extended=true HTTP/1.1
```

2. **Dettaglio campi di un endpoint** — aggiungendo il parametro `info=true` alla query string di qualsiasi endpoint si ottiene la struttura completa dei campi disponibili con tipo, descrizione e indicazioni di obbligatorietà:

```
GET /webapi/risorse/articoli?info=true HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

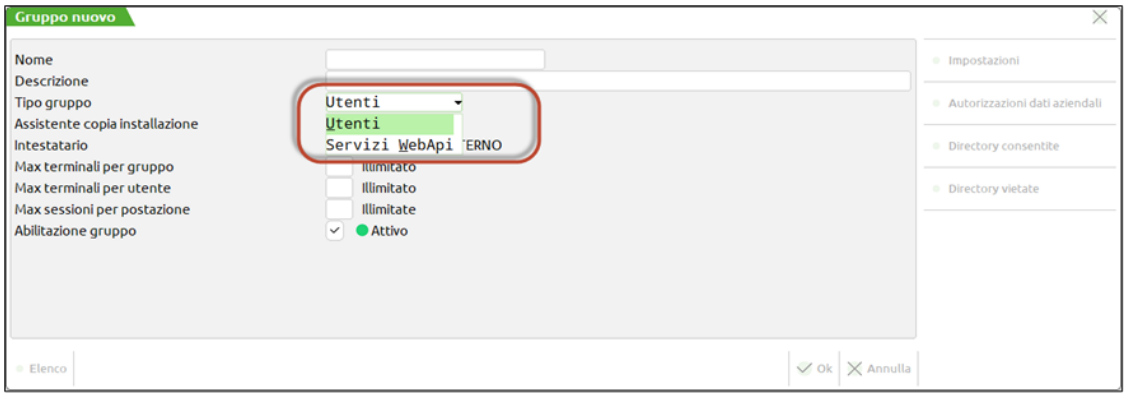
> ℹ **Nota:** Questi due strumenti (`help` e `info=true`) rappresentano il riferimento sempre aggiornato sulla struttura degli endpoint. Si consiglia quindi di interrogarli per scoprire dinamicamente le risorse disponibili e la struttura dei campi, senza dipendere da documentazione statica esterna.

Glossario

Per la raccolta completa dei termini tecnici utilizzati in questo manuale, consultare il Glossario in appendice.

Come si usa

WebApi è disponibile sia in ambiente locale che live dopo aver attivato la suite MDS tramite YouPass. L'attivazione del servizio abiliterà in Mexal/Passcom un gruppo particolare di utenti "Servizi WebApi". Solo gli utenti appartenenti a gruppi di questo tipo potranno utilizzare solo ed esclusivamente i servizi WebApi.



Gestione utenti

WebApi locale: è possibile accedere su installazione locale specificando l'indirizzo del server più la porta su cui è in ascolto il servizio. Il servizio deve essere abilitato lato server Mexal/Passcom tramite il menu *Servizi → Configurazioni → Configurazione moduli → WebApi*. In questa videata sarà possibile verificare la corretta abilitazione del modulo e la porta assegnata al servizio. L'accesso sarà possibile specificando nell'header della richiesta https le credenziali di accesso più altri parametri per identificare l'azienda e la data di esercizio.

WebApi Live: in ambiente live, se attivato da codice contratto, il servizio è già disponibile e raggiungibile tramite l'indirizzo `https://services.passepartout.cloud/`. In questo caso l'autenticazione avviene sempre tramite credenziali di accesso nell'header della richiesta https (come per le installazioni locali) ma usando come *base address* `https://services.passepartout.cloud/webapi/risorse` (vedi Installazioni LIVE).

Il gruppo Servizi WebAPI ha i seguenti limiti operativi:

Parametro	Limite
Utenti per gruppo Servizi WebAPI	max 3
Servizi contemporanei per utente (pool)	max 5
Timeout servizi inattivi	5 minuti

Il primo servizio del pool resta sempre attivo per velocizzare le connessioni successive; gli altri 4 vengono liberati dopo 5 minuti di inattività.

In pratica, ogni utente WebAPI può gestire fino a 5 richieste concorrenti (da client diversi o dallo stesso client). Le richieste eccedenti vengono rifiutate con errore 503 fino a quando un servizio del pool non si libera.

> ℹ **Nota:** Il limite complessivo per installazione è di 3 utenti × 5 servizi = 15 sessioni concorrenti massime.

Gestione uso esclusivo

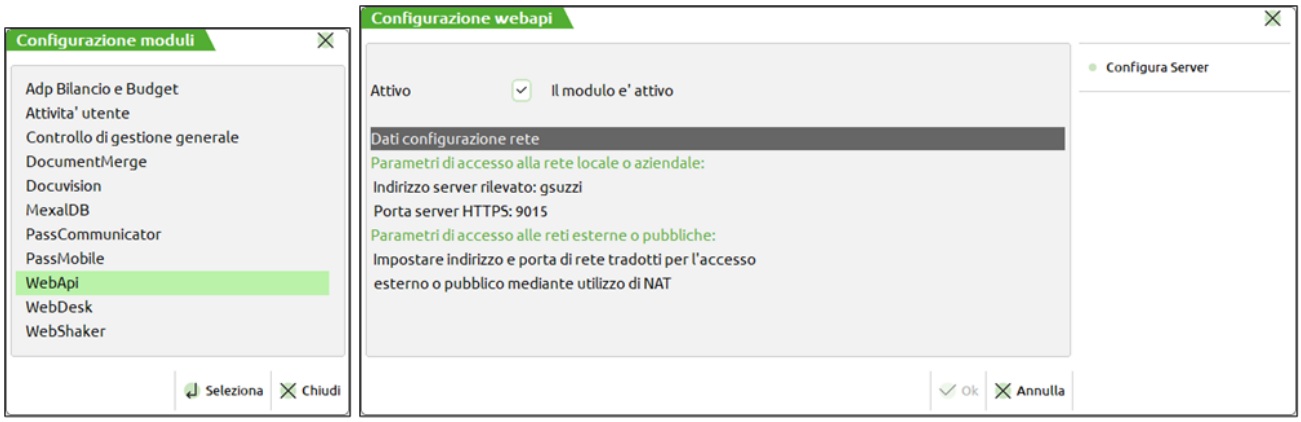
Lato gestionale (Mexal/Passcom) in tutte le parti del programma in cui è richiesto l'uso esclusivo (es. LiveUpdate fase di switch-to, anagrafica azienda, configurazione moduli, etc.), la procedura chiederà la chiusura di eventuali processi WebAPI attivi. Questo si rende necessario in quanto, a differenza di altri terminali con interfaccia o di quelli senza interfaccia ma che si chiudono per time-out, i servizi WebAPI per loro natura non hanno interfaccia e hanno sicuramente un processo che rimane sempre attivo per velocizzare le connessioni. Se il processo WebAPI sta eseguendo una elaborazione si chiuderà non appena l'elaborazione sarà terminata, mentre se il processo è in uno stato *idle* verrà immediatamente chiuso.

In questo modo la procedura ad uso esclusivo può continuare senza interruzioni legate alla presenza di altri terminali.

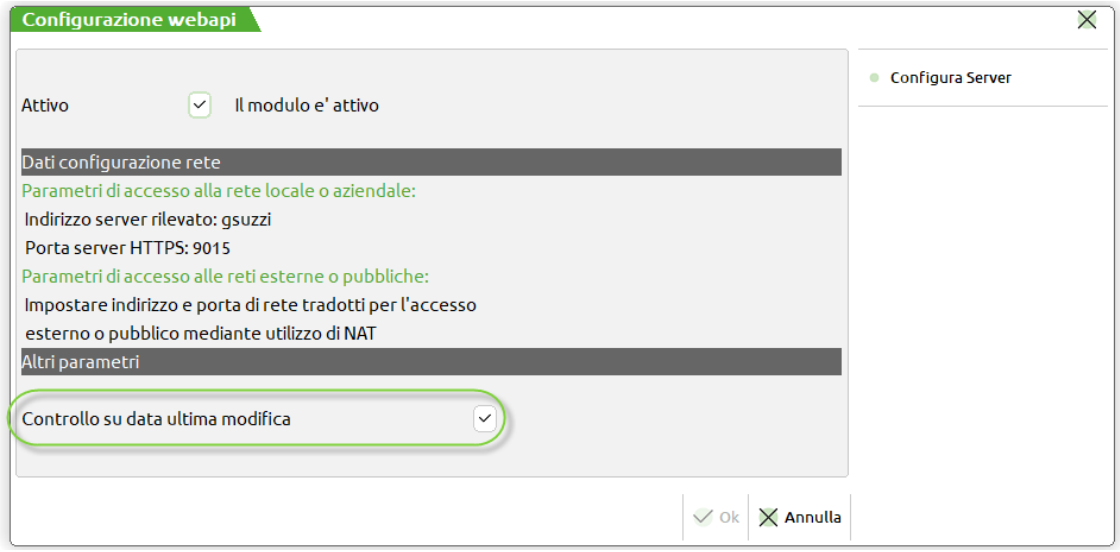
Configurazione e Accesso

Installazioni locali

Per utilizzare WebApi in installazioni locali è necessario attivare il modulo dal menu Servizi → Configurazioni → Configurazione moduli → WebApi.



In questa videata è possibile verificare la corretta abilitazione del modulo, la porta assegnata al servizio e configurare il **controllo sulla data di ultima modifica**.



Configurazione moduli WebAPI

Controllo sulla validità dei dati (data ultima modifica)

Il controllo sulla data di ultima modifica è configurabile da questa stessa videata. Per il funzionamento dettagliato del meccanismo (campi `data_ult_mod` e `dt_ult_mod_orig`), vedere Controllo sulla Validità dei Dati.

Le informazioni necessarie per l'accesso sono:

Parametro	Valore
URL	`<indirizzo_server>`
PORTA	`<porta_server_https>`
USER	`<utente_servizi_webapi>`
PWD	`<password_utente_servizi_webapi>`

Se la richiesta prevede l'utilizzo di un comando che coinvolge una determinata azienda, è necessario fornire anche le seguenti informazioni:

Parametro	Valore	Note
AZIE	`<azienda>`	
SOTTOAZ	`<sottoazienda>`	opzionale
ANNO_AZ	`<anno_di_apertura>`	
MAGAZZINO	`<numero_magazzino_default>`	opzionale

> ℹ **Nota:** se non viene utilizzato il numero di magazzino, tutte le operazioni che coinvolgono il magazzino utilizzeranno come magazzino di default il numero 1.

Se sull'installazione locale è attivo anche il controllo delle credenziali di sistema operativo è necessario utilizzare anche queste credenziali:

Parametro	Valore
USER_SYS	`<utente_di_sistema>`
PWD_SYS	`<password_utente_di_sistema>`

Indirizzo di accesso

L'indirizzo per raggiungere WebApi in locale dovrà essere formato in questo modo:

```
https://URL:PORTA/webapi/risorse/
```

Esempi:

- `PUT` `https://miodominio:9004/webapi/risorse/clienti/codice_cliente` (aggiorna il cliente: dati nel corpo della richiesta)
- `POST` `https://miodominio:9004/webapi/risorse/articoli` (crea un nuovo articolo: dati nel corpo della richiesta)
- `GET` `https://miodominio:9004/webapi/risorse/articoli` (legge tutti gli articoli)
- `DELETE` `https://miodominio:9004/webapi/risorse/articoli/codice_articolo` (cancella un articolo)

Autenticazione

Per il dettaglio completo sul meccanismo di autenticazione (codifica credenziali, formato header, credenziali di sistema operativo), vedere Autenticazione.

Di seguito il formato sintetico degli header per installazioni locali:

```
Authorization: Passepartout <credenziali_b64>
Content-type: application/json
Coordinate-Gestionale: Azienda=<AZIE> SottoAzienda=<SOTTOAZ> Anno=<ANNO_AZ>
Magazzino=<MAGAZZINO>
```

Esempi di header per installazioni locali

Caso di autenticazione con solo le credenziali dell'utente dell'installazione:

```
{
  "Authorization": "Passepartout d2ViYXBpOndlYmFwaQ==",
  "Content-type": "application/json",
  "Coordinate-Gestionale": "Azienda=DEM Anno=2021 Magazzino=2"
}
```

Caso di autenticazione in cui sono necessarie anche le credenziali di sistema operativo (login=1):

```
{
  "Authorization": "Passepartout d2ViYXBpOndlYmFwaQ== dXNlcl9zeXM6cHdk",
  "Content-type": "application/json",
  "Coordinate-Gestionale": "Azienda=DEM Anno=2021"
}
```

Certificati SSL

Nel caso di installazioni locali, l'accesso alle risorse tramite protocollo sicuro https avviene utilizzando un certificato autoprodotto (self-signed). Le librerie utilizzate dall'applicazione client dovranno quindi poter gestire questo caso.

In alternativa, se non fosse possibile utilizzare i certificati self-signed, è necessario dotare l'installazione di certificati validi (non self signed). Questo è possibile farlo inserendo i certificati tramite il menu Servizi → Configurazioni → Configurazione moduli → WebApi nella sezione Configurazione Server (in modo del tutto analogo a quanto avviene per WebDesk).

Installazioni LIVE

Per poter accedere in ambiente Live, i parametri necessari per l'accesso sono i seguenti:

```
https://services.passepartout.cloud/webapi/risorse
```

Parametro	Valore

Parametro	Valore
DOMINIO	`<dominio_della_installazione_live>`
USER	`<utente_servizi_webapi>`
PWD	`<password_utente_servizi_webapi>`

Se la richiesta prevede l'utilizzo di un comando che coinvolge una determinata azienda, è necessario fornire anche le seguenti informazioni:

Parametro	Valore	Note
AZIE	`<azienda>`	
SOTTOAZ	`<sottoazienda>`	opzionale
ANNO_AZ	`<anno_di_apertura>`	
MAGAZZINO	`<numero_magazzino_default>`	opzionale

Autenticazione in ambiente Live

In ambiente Live, nell'header `Authorization` va specificato anche il dominio dell'installazione. Per il meccanismo completo vedere Autenticazione.

Formato sintetico:

```
Authorization: Passepartout <credenziali_b64> Dominio=<DOMINIO>
Content-type: application/json
Coordinate-Gestionale: Azienda=<AZIE> SottoAzienda=<SOTTOAZ> Anno=<ANNO_AZ>
```

Esempio di header per installazioni Live

```
{
  "Authorization": "Passepartout d2ViYXBpOndlYmFwaQ== Dominio=MIODOMINIO",
  "Content-type": "application/json",
  "Coordinate-Gestionale": "Azienda=DEM Anno=2021"
}
```

Concetti Generali

Autenticazione

Passepartout WebAPI utilizza un meccanismo di autenticazione proprietario basato su credenziali codificate in Base64 e trasmesse tramite l'header HTTP `Authorization`.

Formato delle credenziali

Per autenticarsi è necessario:

1. Concatenare il nome utente e la password nel formato `USER:PWD` (es. `webapi:webapi`)
2. Codificare la stringa risultante in Base64

Il valore ottenuto costituisce le **credenziali codificate** (`<credenziali_b64>`) da inserire nell'header di ogni richiesta.

Header di autenticazione

L'header `Authorization` deve essere incluso in ogni richiesta HTTP e utilizza il prefisso `Passepartout` seguito dalle credenziali codificate.

Installazioni locali:

```
Authorization: Passepartout <credenziali_b64>
```

Installazioni Live (cloud):

```
Authorization: Passepartout <credenziali_b64> Dominio=<DOMINIO>
```

Nelle installazioni Live è necessario specificare anche il dominio dell'installazione tramite il parametro `Dominio` aggiunto nella stessa riga dell'header `Authorization`.

Credenziali di sistema operativo

Se sull'installazione locale è attivo il controllo delle credenziali di sistema operativo (`login=1`), è necessario aggiungere un secondo blocco di credenziali codificate in Base64 (formato `USER_SYS:PWD_SYS`):

```
Authorization: Passepartout <credenziali_b64> <credenziali_sistema_b64>
```

Header Coordinate-Gestionale

Quando una richiesta coinvolge una specifica azienda, è necessario includere l'header `Coordinate-Gestionale` per indicare il contesto operativo:

```
Coordinate-Gestionale: Azienda=<AZIE> SottoAzienda=<SOTTOAZ> Anno=<ANNO_AZ>
Magazzino=<MAGAZZINO>
```

Parametro	Descrizione	Obbligatorio
Azienda	Codice dell'azienda di lavoro	Sì
SottoAzienda	Codice della sotto-azienda	No
Anno	Anno di apertura dell'azienda	Sì
Magazzino	Numero del magazzino di default (default: 1)	No

Esempio di request autenticata

Installazione locale — lettura degli articoli per l'azienda DEM, anno 2021, magazzino 2:

```
GET /webapi/risorse/articoli HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=2
```

In questo esempio le credenziali `webapi:webapi` sono state codificate in Base64 ottenendo il valore `d2ViYXBpOndlYmFwaQ==`.

Installazione Live — lettura degli articoli per l'azienda DEM sul dominio MIODOMINIO:

```
GET /webapi/risorse/articoli HTTP/1.1
Host: services.passepartout.cloud
Authorization: Passepartout d2ViYXBpOndlYmFwaQ== Dominio=MIODOMINIO
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021
```

> ℹ **Nota:** Per i dettagli su Content-Type, charset e header obbligatori, vedere Formato Request.

Formato Request

Tutte le richieste verso Passepartout WebAPI devono rispettare il formato e le convenzioni descritte in questa sezione. Il mancato rispetto di questi requisiti produce un errore nella risposta.

Formato dati

Il formato dei dati scambiati con WebAPI è **JSON** (JavaScript Object Notation). Lo standard JSON prevede che la codifica del charset sia **UTF-8**.

Eventuali dati non codificati correttamente in UTF-8 generano un errore. È necessario utilizzare opportune librerie di encoding UTF-8 per garantire che tutti i dati inviati a WebAPI rispettino la codifica corretta.

Header HTTP richiesti

Ogni richiesta HTTP deve includere i seguenti header:

Nome	Valore	Descrizione
Authorization	`Passepartout <credenziali_b64>`	Credenziali di autenticazione codificate in Base64. Può includere anche il dominio per installazioni Live e le credenziali di sistema operativo se richieste.
Content-type	`application/json`	Tipo di contenuto della richiesta. Obbligatorio per i metodi POST e PUT. Opzionale per GET e DELETE.
Coordinate-Gestionale	`Azienda=<AZIE> SottoAzienda=<SOTTOAZ> Anno=<ANNO_AZ> Magazzino=<MAGAZZINO>`	Coordinate dell'azienda e dell'anno di lavoro. Richiesto per i comandi che operano su dati aziendali.

Content-Type e charset

Il Content-Type supportato è esclusivamente `application/json`. Il charset di default è UTF-8 e non è necessario specificarlo esplicitamente.

Se si vuole indicare anche il charset, l'unico valore accettato è:

```
Content-type: application/json;charset=utf-8
```

Qualsiasi altro tipo di charset verrà rifiutato con un errore.

> **Nota:** L'header `Content-type` è obbligatorio solo per le richieste che prevedono un body (POST e PUT). Per i metodi GET e DELETE può essere omesso.

Esempio di request HTTP completa

L'esempio seguente mostra una richiesta POST per creare un nuovo articolo, con tutti gli header richiesti e il body JSON:

```
POST /webapi/risorse/articoli HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=1

{
  "codice": "ART001",
  "descrizione": "Articolo di esempio",
  "cos_ult": 25.50,
  "um_principale": "NR"
}
```

L'esempio seguente mostra una richiesta GET per leggere un articolo, dove il body non è presente e il Content-type è opzionale:

```
GET /webapi/risorse/articoli/ART001 HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=1
```

Formato Response

Tutte le risposte delle WebAPI Passepartout vengono restituite in formato **JSON** con encoding **UTF-8**. L'header di risposta include sempre:

```
Content-Type: application/json; charset=utf-8
```

La struttura della response varia in base al metodo HTTP utilizzato e all'esito dell'operazione.

Struttura standard della response

Le risposte di successo seguono convenzioni differenti a seconda del tipo di operazione eseguita:

Metodo HTTP	Codice successo	Contenuto response
GET	200 OK	Oggetto JSON con i dati della risorsa o della collezione richiesta
POST	201 Created	Header `Location` con l'URL della nuova risorsa creata
PUT	204 No Content	Nessun body (operazione completata con successo)
DELETE	204 No Content	Nessun body (operazione completata con successo)

Campi della response (GET collezione)

La response di una richiesta GET su una collezione restituisce un oggetto JSON con i seguenti campi:

Nome	Tipo	Significato
`dati`	Array	Array di oggetti, ciascuno rappresentante un record della collezione con i campi specifici dell'entità
`data_ric_elab`	String	Timestamp di elaborazione della richiesta, formato `AAAAMMGG HHMMSS`
`next`	String	Token opaco di paginazione. Presente solo quando esistono ulteriori record da recuperare. Va passato "as-is" nella request successiva

Campi della response (GET singola risorsa)

La response di una richiesta GET su una singola risorsa (con chiave nel path) restituisce direttamente l'oggetto JSON con i campi dell'entità, senza wrapper.

Header della response (POST creazione)

Nome	Tipo	Significato
`Location`	String (header)	URL della risorsa appena creata, restituito nell'header HTTP della response a una POST con codice 201

Lettura di una collezione vs singola risorsa

- **Collezione** (GET senza chiave): la response contiene un oggetto JSON con il campo `dati` (array di oggetti), il campo `data_ric_elab` (timestamp di elaborazione) e, se i risultati sono paginati, il campo `next` (token di paginazione). Se specificato il parametro `fields`, ogni oggetto nell'array contiene solo i campi richiesti.
- **Singola risorsa** (GET con chiave): la response contiene un singolo oggetto JSON con tutti i campi dell'entità (o i soli campi specificati tramite `fields`), senza alcun wrapper.

Esempio di response — Lettura di una collezione

La seguente response viene restituita da una richiesta GET su una collezione di articoli con filtro sui campi:

```
GET /webapi/risorse/articoli?fields=codice,descrizione,cos_ult&max=3 HTTP/1.1
Host: mioserver:9014
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Response (HTTP 200 OK):

```
{
  "dati": [
    {
      "codice": "ART001",
      "descrizione": "Bicicletta MTB 29 pollici",
      "cos_ult": 899.00
    },
    {
      "codice": "ART002",
      "descrizione": "Casco ciclismo professionale",
      "cos_ult": 129.50
    },
    {
      "codice": "ART003",
```

```
      "descrizione": "Lucchetto antifurto rinforzato",
      "cos_ult": 45.90
    }
  ],
  "data_ric_elab": "20240601 093012",
  "next": "QVJUMDAz..."
}
```

In questo esempio:

- L'oggetto root contiene il campo `dati` con l'array dei record, ciascuno con i soli campi richiesti (`codice`, `descrizione`, `cos_ult`)
- Il parametro `max=3` limita il numero di record restituiti a 3
- Il campo `next` contiene un token opaco da passare nella request successiva per ottenere la pagina seguente (`?next=QVJUMDAz...`)
- Il campo `data_ric_elab` riporta il timestamp di elaborazione della richiesta

Esempio di response — Lettura di una singola risorsa

```
GET /webapi/risorse/articoli/ART001 HTTP/1.1
Host: mioserver:9014
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Response (HTTP 200 OK):

```
{
  "codice": "ART001",
  "descrizione": "Bicicletta MTB 29 pollici",
  "tipologia": "A",
  "um_principale": "NR",
  "cos_ult": 450.00,
  "cos_std": 0.00,
  "alq_iva": " 22,0",
  "categoria_omogenea": "",
  "note": ""
}
```

Esempio di response — Creazione risorsa (POST)

Quando una risorsa viene creata con successo, il server restituisce un codice **201 Created** con l'header `Location` che contiene l'URL per accedere alla nuova risorsa:

```
HTTP/1.1 201 Created
Location: /webapi/risorse/articoli/ART999
Content-Type: application/json; charset=utf-8
```

Compressione della response

Per ridurre la dimensione del payload, è possibile richiedere la compressione della response aggiungendo nell'header della request:

```
Accept-Encoding: gzip
```

In questo caso il body della response sarà compresso in formato gzip e il server includerà l'header:

```
Content-Encoding: gzip
```

> ℹ **Nota:** la compressione è consigliata per response di grandi dimensioni, come la lettura di collezioni con molti record.

Gestione Errori

Passepartout WebAPI restituisce informazioni strutturate sugli errori attraverso una response JSON con un formato standard. Ogni risposta di errore include un codice HTTP appropriato e un corpo JSON con un oggetto `error` che descrive la natura del problema.

Struttura dell'oggetto errore

Quando una richiesta non può essere elaborata con successo, la response contiene un oggetto JSON `error` con i seguenti campi:

Campo	Tipo	Descrizione
`response-code`	Integer	Codice HTTP della response
`response-detail`	String	Codice errore numerico + descrizione. Formato: `<codice> - <messaggio> [<dettaglio>]`
`response-message`	String	Messaggio HTTP standard (es. "Bad Request", "Unauthorized")
`request-uri`	String	URI della richiesta che ha generato l'errore
`request-method`	String	Metodo HTTP della richiesta (GET, POST, PUT, DELETE)
`request-id`	String	UUID univoco della richiesta per il tracciamento nei log
`warsion`	String	Versione interna del motore WebAPI
`remote`	String	Indirizzo IP:porta del client
`timestamp`	String	Data/ora del server (formato `dd/MM/yyyy HH:mm:ss`)

Il campo `response-detail` è il più importante per la diagnosi: contiene un codice numerico (1001, 3002, 5001, 6001) che identifica la famiglia dell'errore, seguito da una descrizione leggibile e da dettagli tra parentesi quadre.

Codici HTTP utilizzati

La tabella seguente riporta i codici di stato HTTP restituiti dall'API con il relativo significato e contesto d'uso:

Codice	Significato	Uso
200	OK	La richiesta è stata elaborata con successo
400	Bad Request	Parametri non validi, campi mancanti, errore gestionale, coordinate assenti
401	Unauthorized	Autenticazione fallita: credenziali assenti o non valide
404	Not Found	La risorsa richiesta non esiste o il path dell'endpoint non è valido
408	Request Timeout	Il servizio ha superato il tempo massimo di esecuzione
500	Internal Server Error	Errore imprevisto, conflitto uso esclusivo, errore inoltro comando
503	Service Unavailable	Pool servizi esaurito (max 5 servizi simultanei per utente)

Esempi di response di errore

Errore 401 — Autenticazione fallita:

Questo errore si verifica quando le credenziali fornite nell'header `Authorization` non sono valide o sono assenti.

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json

{
  "error": {
    "response-code": 401,
    "response-detail": "3002 - errore in fase di richiesta webapi dal pool
[notifica server errore] [Utente o password non validi - Accesso non possibile]
[WebapiHttpException]",
    "request-uri": "/webapi/risorse/articoli",
    "response-message": "Unauthorized",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "45a192d7-cdf0-4aa8-a9ee-c06eecdab6a8",
    "remote": "192.168.3.180:41934",
    "timestamp": "11/06/2026 09:19:01"
  }
}
```

Errore 400 — Errore gestionale (campo non valido):

Questo errore si verifica quando i dati inviati contengono un campo non riconosciuto dall'endpoint.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error": {
    "response-code": 400,
    "response-detail": "6001 - errore gestionale [Nome campo 'campo_inesistente'
non valido]",
    "request-uri": "/webapi/risorse/articoli/ART001",
    "response-message": "Bad Request",
    "warsion": "00700",
    "request-method": "PUT",
    "request-id": "911f829c-d073-4b5f-b769-60e95a18995e",
    "remote": "192.168.3.180:49858",
    "timestamp": "11/06/2026 09:17:59"
  }
}
```

Errore 404 — Risorsa non trovata:

Questo errore si verifica quando si tenta di accedere a una risorsa inesistente. La response è un semplice HTTP 404 senza corpo JSON strutturato.

```
HTTP/1.1 404 Not Found
```

> ℹ **Nota:** Il campo `request-id` è un UUID univoco utile per correlare le response di errore con i log del server in caso di escalation al supporto tecnico. Per l'elenco completo dei codici numerici e la guida diagnostica, vedere la sezione Codici di Errore in appendice.

Paginazione

Quando una richiesta GET restituisce una collezione di dati, il numero di record nella response può essere molto elevato. Passepartout WebAPI supporta un meccanismo di paginazione basato su cursore che suddivide i risultati in blocchi successivi.

Quando si attiva la paginazione

La paginazione può attivarsi in **due modi**:

1. **Esplicitamente** — il client specifica il parametro `max` in query string per controllare la dimensione di ogni pagina
2. **Automaticamente** — il server pagina autonomamente quando la risposta viola uno dei seguenti limiti interni:
    - **Tempo di elaborazione** superiore a 30 secondi
    - **Dimensione del payload** superiore a 50 MB

In entrambi i casi, se esistono ulteriori record da restituire, la response conterrà il campo `next` con il token per recuperare la pagina successiva.

> ⚠ **Nota Bene:** Anche specificando `max`, il server potrebbe restituire meno record del richiesto se uno dei limiti interni viene raggiunto prima. Ad esempio, con `max=1000` potrebbero essere restituiti solo 700 record se l'elaborazione supera i 30 secondi o i 50 MB.

Parametri di paginazione nella request

La paginazione viene controllata tramite parametri di query string nella URL della richiesta:

Parametro	Tipo	Posizione	Descrizione
`max`	Integer	query	Numero massimo di record da restituire nella response. Se omesso, il server restituisce tutti i record che riesce a processare entro i limiti interni (30s / 50MB).
`next`	String	query	Punto di partenza per la pagina successiva. Il valore da utilizzare è quello restituito nel campo `next` della response precedente.

Campi di paginazione nella response

Quando i risultati di una GET su collezione sono paginati, la response JSON è un oggetto con il campo `dati` (array dei record) e il campo `next` al livello root:

Campo	Tipo	Descrizione
`dati`	Array	Array contenente gli oggetti richiesti
`data_ric_elab`	String	Timestamp di elaborazione della richiesta (formato `AAAAMMGG HHMMSS`)
`next`	String	Token opaco di paginazione. Presente solo quando esistono ulteriori record da recuperare. Il valore va passato “as-is” nella request successiva come parametro query string `?next=<valore>`

Se il campo `next` è presente nell’oggetto di response, significa che esistono altri dati disponibili. Se il campo `next` è assente, significa che non ci sono ulteriori pagine.

Come iterare tra le pagine

Per ottenere tutti i risultati di una collezione in modo paginato:

1. Eseguire la prima richiesta, con o senza il parametro `max`
2. Verificare se nella response è presente il campo `next`
3. Eseguire la richiesta successiva aggiungendo il parametro `next` con il valore ottenuto dalla response precedente (passarlo “as-is”, senza modificarlo)
4. Ripetere i passi 2 e 3 fino a quando la response non contiene più il campo `next`

> ℹ **Nota:** Specificare `max` è consigliato per controllare la dimensione delle pagine, ma non è obbligatorio. Senza `max`, il server restituisce il massimo numero di record possibile entro i propri limiti, paginando automaticamente se necessario.

Gestione interna dei token next

I valori `next` sono gestiti tramite un sistema di semafori per evitare conflitti tra richieste concorrenti. Ogni richiesta è identificata da un **session ID** univoco, concatenato alla chiave del record per determinare il punto di partenza della query. È possibile riutilizzare lo stesso valore di `next` più volte fino a quando l’intero archivio non è stato iterato. Tuttavia, quando viene raggiunta l’ultima chiave `next`, il semaforo viene resettato e, se vengono inviate chiavi `next` precedenti, la richiesta restituirà un errore.

> ⚠ **Nota Bene:** Si consiglia di utilizzare immediatamente i valori di `next` restituiti, senza conservarli per un uso futuro, poiché potrebbero scadere o diventare obsoleti. Utilizzare tempestivamente le chiavi `next` garantisce l’integrità e l’affidabilità del processo di recupero dei dati.

Esempio di request paginata

La seguente richiesta recupera i primi 3 articoli della collezione:

```
GET /webapi/risorse/articoli?fields=codice,descrizione,cos_ult&max=3 HTTP/1.1
Host: mioserver:9014
```

```
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Response (HTTP 200 OK):


```
{
  "dati": [
    {
      "codice": "ART001",
      "descrizione": "Bicicletta MTB 29 pollici",
      "cos_ult": 899.00
    },
    {
      "codice": "ART002",
      "descrizione": "Casco ciclismo professionale",
      "cos_ult": 129.50
    },
    {
      "codice": "ART003",
      "descrizione": "Lucchetto antifurto rinforzato",
      "cos_ult": 45.90
    }
  ],
  "data_ric_elab": "20240601 093012",
  "next": "QVJUMDAz..."
}
```


Il campo next è presente, indicando che esistono altri articoli da recuperare. Per ottenere la pagina successiva, si utilizza il token ricevuto nel parametro next:

```
GET
/webapi/risorse/articoli?fields=codice,descrizione,cos_ult&max=3&next=QVJUMDAz...
HTTP/1.1
Host: mioserver:9014
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Response (HTTP 200 OK):


```
{
  "dati": [
    {
      "codice": "ART004",
      "descrizione": "Pompa ad aria portatile",
      "cos_ult": 34.00
    },
    {
      "codice": "ART005",
      "descrizione": "Kit riparazione camera d'aria",
      "cos_ult": 12.90
```

```
    }
  ],
  "data_ric_elab": "20240601 093015"
}
```

In questo caso la response non contiene il campo `next`. Questo indica che non esistono ulteriori pagine: tutti i risultati della collezione sono stati recuperati.

> **Nota:** Il parametro `max` può essere combinato con altri parametri di query string come `fields` per controllare sia il numero di record restituiti sia i campi inclusi in ciascun record.

Ricerca e Filtri

La risorsa `ricerca` è una risorsa speciale comune a tutti i gruppi di risorse (clienti, articoli, documenti, ubicazioni, ecc.). Consente di effettuare ricerche avanzate specificando filtri complessi nel body della request.

Meccanismo Generale

La ricerca si effettua tramite metodo `POST` sull'endpoint `/ricerca` di qualsiasi entità:

```
POST https://base_address/webapi/risorse/<entità>/ricerca
```

Il significato semantico è: "crea una ricerca su questa collezione di risorse". Il risultato è equivalente a una GET con filtri, ma i criteri vengono specificati nel body JSON anziché in query string.

Come per il comando GET, è possibile combinare i filtri del body con i parametri in query string (`fields`, `max`, `next`).

Struttura dei Filtri

I filtri devono essere specificati nel body secondo il seguente pattern:

```
{
  "filtri": [
    {
      "campo": "<nome_campo>",
      "indice1": "<indice1_campo_array>",
      "indice2": "<indice2_campo_array>",
      "condizione": "<operatore>",
      "case_insensitive": true,
      "valore": "<valore da testare>"
    }
  ]
}
```

Campi del filtro

Campo	Tipo	Obbl.	Descrizione
`campo`	string	Sì	Nome del campo su cui filtrare
`indice1`	integer	No	Indice del primo livello (per campi di tipo array)
`indice2`	integer	No	Indice del secondo livello (per array a 2 livelli, come nelle liste)
`condizione`	string	Sì	Operatore di confronto
`case_insensitive`	boolean	No	Se `true`, il confronto su stringhe ignora maiuscole/minuscole (default: `false`)
`valore`	string/number/array	Sì	Valore da confrontare. Se è un array, i valori sono in logica OR tra loro

Operatori Disponibili

Operatore	Descrizione	Applicabile a
`=`	Uguale	Tutti i tipi
`<>`	Diverso da	Tutti i tipi
`>`	Maggiore di	Numeri, date, stringhe
`<`	Minore di	Numeri, date, stringhe
`>=`	Maggiore o uguale	Numeri, date, stringhe
`<=`	Minore o uguale	Numeri, date, stringhe
`contiene`	Contiene la sottostringa	Stringhe
`inizia_per`	Inizia con la sottostringa	Stringhe

Logica di Combinazione

- Più filtri nello stesso array sono combinati in **logica AND** tra loro.
- Il campo `valore` può essere un **array di valori**: in questo caso i valori sono combinati in **logica OR** tra loro all'interno dello stesso filtro.

Parametri Query String

I seguenti parametri in query string sono utilizzabili insieme alla ricerca (identici alla GET):

Parametro	Descrizione
`fields`	Lista campi da restituire, separati da virgola (es. `?fields=codice,descrizione`)
`max`	Numero massimo di record da restituire

Parametro	Descrizione
next	Token di paginazione per la pagina successiva

Esempi Pratici

Ricerca articoli con filtri multipli

```
POST /webapi/risorse/articoli/ricerca?fields=codice,descrizione HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "tipologia",
      "condizione": "=",
      "valore": "A"
    },
    {
      "campo": "cos_ult",
      "condizione": ">",
      "valore": 100
    },
    {
      "campo": "descrizione",
      "condizione": "contiene",
      "case_insensitive": true,
      "valore": ["BICI", "MTB", "BDC"]
    }
  ]
}
```

Risultato: vengono restituiti solo gli articoli di tipo merce (tipologia=A) **AND** con costo ultimo maggiore di 100 **AND** la cui descrizione contiene “BICI” **OR** “MTB” **OR** “BDC”. Del risultato saranno visualizzati solo i campi `codice` e `descrizione`.

Ricerca clienti per ragione sociale

```
POST /webapi/risorse/clienti/ricerca?fields=codice,ragione_sociale HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "ragione_sociale",
      "condizione": "inizia_per",
```

```
      "case_insensitive": true,
      "valore": "ROSSI"
    }
  ]
}
```

Ricerca documenti per data

```
POST /webapi/risorse/documenti/ordini-clienti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "data_documento",
      "condizione": ">=",
      "valore": "20240101"
    },
    {
      "campo": "data_documento",
      "condizione": "<=",
      "valore": "20240131"
    }
  ]
}
```

Ricerca con campi array (indice)

Per entità che espongono campi come array (es. documenti, liste di prelievo), è possibile specificare un indice per filtrare su un elemento specifico dell'array:

```
{
  "filtri": [
    {
      "campo": "codice_articolo",
      "indice1": 1,
      "condizione": "=",
      "valore": "ART001"
    }
  ]
}
```

Note Importanti

- La ricerca è disponibile su **tutte le entità** che supportano il metodo GET per la lista.
- Il campo `case_insensitive` ha effetto solo su confronti tra stringhe (default: `false`).
- Per le entità con struttura master-detail (documenti, liste-prelievo, prima-nota), la ricerca sulle testate e sulle righe utilizza endpoint separati (es. `/righe/ricerca`).

- Il campo `indice1` è facoltativo e serve solo quando il campo da filtrare è di tipo array. Per le liste, gli array sono di massimo 2 livelli, quindi possono essere specificati al massimo 2 indici (`indice1` e `indice2`).

Esempio (Live)

La seguente ricerca è stata testata su installazione reale (Azienda UB1, Anno 2026):

```
POST /webapi/risorse/articoli/ricerca?fields=codice,descrizione&max=5 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "descrizione",
      "condizione": "contiene",
      "case_insensitive": true,
      "valore": "bici"
    }
  ]
}
```

Response (200 OK):

```
{
  "dati": [
    {"codice": "BICCOMPLETAAUTO", "descrizione": "BICI COMPLETA AUTOMA"},
    {"codice": "BICI", "descrizione": "BICI ASSEMBLATA"},
    {"codice": "BICI1234", "descrizione": "BICI ASSEMBLATA 12 3"},
    {"codice": "BICI23DE", "descrizione": "BICI ASSEMBLATA 23 D"},
    {"codice": "BICI3456", "descrizione": "BICI ASSEMBLATA 34 5"}
  ],
  "data_ric_elab": "20260610 163302",
  "next":
"57415f34393366306664352d363464392d313166312d396633362d396436653165336262303264-
ea00000000000000"
}
```

La ricerca con `case_insensitive: true` e operatore `contiene` restituisce tutti gli articoli la cui descrizione contiene “bici” indipendentemente da maiuscole/minuscole.

Controllo sulla Validità dei Dati (Data Ultima Modifica)

In fase di revisione di una risorsa (`PUT` o `POST` nel caso di trasformazione documenti) può capitare che i dati precedentemente letti da una GET non siano più validi perché modificati nel frattempo da altri

attori. Per evitare disallineamenti, le WebAPI implementano un meccanismo di **optimistic locking** basato sulla data di ultima modifica.

Scenario di esempio

1. Il client legge un documento con 3 righe (GET)
2. Nel frattempo, lato gestionale, viene aggiunta una quarta riga
3. Il client invia la revisione con le 3 righe lette in origine (PUT)
4. Senza il controllo, la quarta riga verrebbe cancellata

Il controllo sulla data di ultima modifica impedisce questo scenario: se la data nel payload non corrisponde a quella attuale della risorsa, la revisione viene rifiutata con un errore.

Campi per il controllo

È necessario specificare nel payload, oltre ai dati da modificare, anche le date di ultima modifica delle risorse che si stanno revisionando:

Campo	Tipo	Quando usarlo
`data_ult_mod`	Scalare	In tutti i casi di revisione (PUT) di una singola risorsa
`dt_ult_mod_orig`	Array	Quando si effettua una trasformazione documento con metodo POST (un elemento per ogni testata di origine)

Il valore di `data_ult_mod` si ottiene dalla risposta GET sulla risorsa: è il campo restituito dal server che rappresenta il timestamp dell'ultima modifica. Va ritrasmesso "as-is" nella PUT successiva.

Configurazione

Il controllo è configurabile lato gestionale alla voce di menu *Servizi → Configurazioni → Configurazione moduli → WebAPI*.

> ⚠ **Nota Bene:** Se il controllo è attivo e la data di ultima modifica inviata non corrisponde a quella attuale della risorsa sul server, la revisione viene rifiutata con un errore. Questo garantisce che nessuna modifica concorrente vada persa.
>
> ℹ **Nota:** Il controllo è opzionale e disattivabile dalla configurazione del modulo. Quando è disattivo, le revisioni vengono accettate senza verificare la data di ultima modifica (last-write-wins).

Guida Rapida

Quick Start

Questa guida accompagna lo sviluppatore dalla preparazione delle credenziali fino al completamento di una lettura (GET) e una scrittura (POST) tramite Passepartout WebAPI. Al termine del flusso sarai in grado di autenticarti, leggere dati e creare nuove risorse.

Prerequisiti

Prima di iniziare, verificare di disporre dei seguenti elementi:

Elemento	Descrizione
URL base (locale)	`https://<server>:<porta>/webapi/risorse/`
URL base (Live)	`https://services.passepartout.cloud/webapi/risorse`
Credenziali	Utente e password di tipo Servizi WebAPI dell'installazione (es. `webapi` / `webapi`)
Versione minima	Mexal/Passcom 2020 o superiore con modulo WebAPI attivo
Azienda di lavoro	Codice azienda, anno di apertura e magazzino di default

> ℹ **Nota:** per le installazioni Live è necessario disporre anche del nome del dominio dell'installazione, da specificare nell'header Authorization.

Step 1 — Preparazione delle credenziali

Concatenare il nome utente e la password nel formato `USER:PWD` e codificare il risultato in Base64.

Esempio:

Credenziali: `webapi:webapi`

Codifica Base64: `d2ViYXBpOndlYmFwaQ==`

Il valore ottenuto (`d2ViYXBpOndlYmFwaQ==`) è quello da inserire nell'header `Authorization` di ogni richiesta, preceduto dal prefisso `Passepartout`:

```
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
```

Step 2 — Prima richiesta GET (lettura articoli)

Una volta preparate le credenziali, eseguire una richiesta GET per verificare la connessione e ottenere la lista degli articoli.

Request:

```
GET /webapi/risorse/articoli HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=1
```

Response (200 OK):

```
{
  "dati": [
    {
      "codice": "ART001",
      "descrizione": "Articolo di prova",
      "tipologia": "A",
      "um_principale": "NR"
    },
    {
      "codice": "ART002",
      "descrizione": "Secondo articolo",
      "tipologia": "A",
      "um_principale": "NR"
    }
  ],
  "data_ric_elab": "20240601 093012"
}
```

Se la response contiene un oggetto con il campo `dati` (array degli articoli presenti in archivio), la connessione e l'autenticazione sono correttamente configurate.

Step 3 — Richiesta POST (creazione articolo)

Dopo aver verificato la lettura, creare un nuovo articolo inviando una richiesta POST con il body JSON contenente i dati del nuovo record.

Request:

```
POST /webapi/risorse/articoli HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=1
```

```
{
  "codice": "ART100",
  "descrizione": "Nuovo articolo da WebAPI",
  "cos_ult": 49.90,
  "um_principale": "NR"
}
```

Response (201 Created):

```
HTTP/1.1 201 Created
Location: /webapi/risorse/articoli/ART100
Content-Type: application/json; charset=utf-8
```

La response non contiene body. L'informazione significativa è l'header `Location`, che riporta l'URL della risorsa appena creata. Per il formato completo delle response, vedere Formato Response.

Step 4 — Verifica (lettura articolo creato)

Per confermare la corretta creazione, eseguire una richiesta GET sul singolo articolo appena inserito, specificando il codice nel path dell'URL.

Request:

```
GET /webapi/risorse/articoli/ART100 HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=1
```

Response (200 OK):

```
{
  "codice": "ART100",
  "descrizione": "Nuovo articolo da WebAPI",
  "tipologia": "",
  "um_principale": "NR",
  "cos_ult": 49.90
}
```

I dati restituiti corrispondono a quanto inviato nello step precedente: l'articolo è stato creato e salvato correttamente.

Gestione errore di autenticazione

Se le credenziali sono errate, assenti o codificate in modo scorretto, la WebAPI restituisce un errore 401:

Request con credenziali errate:

```
GET /webapi/risorse/articoli HTTP/1.1
Host: miodominio:9004
Authorization: Passepartout RVJSQVRP
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2021 Magazzino=1
```

Response (401 Unauthorized):

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json

{
  "error": {
    "response-code": 401,
    "response-detail": "3002 - errore in fase di richiesta webapi dal pool
[notifica server errore] [Utente o password non validi - Accesso non possibile]
[WebapiHttpException]",
    "request-uri": "/webapi/risorse/articoli",
    "response-message": "Unauthorized",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "45a192d7-cdf0-4aa8-a9ee-c06eecdab6a8",
    "remote": "192.168.3.180:41934",
    "timestamp": "11/06/2026 09:19:01"
  }
}
```

Per la struttura completa dell'oggetto errore e i codici numerici, vedere Gestione Errori.

Causa: il valore Base64 nell'header `Authorization` non corrisponde a una coppia utente/password valida nell'installazione del gestionale.

Risoluzione:

1. Verificare che utente e password siano corretti accedendo direttamente al gestionale Mexal/Passcom
2. Verificare che la stringa sia nel formato `USER:PWD` (con i due punti come separatore) prima della codifica Base64
3. Verificare che la codifica Base64 sia stata eseguita correttamente, senza spazi o caratteri aggiuntivi
4. Per installazioni Live, verificare che il parametro `Dominio=` sia presente e corretto nell'header Authorization

⚠ **Nota Bene:** l'header `Authorization` deve contenere il prefisso `Passepartout` seguito da uno spazio e dalle credenziali Base64. L'assenza del prefisso o un formato diverso generano lo stesso errore 401.

Riferimento Endpoint

Entità: Alias Articoli

Gestione degli alias (codici alternativi) associati agli articoli di magazzino. Permette di associare codici cliente, fornitore o personalizzati agli articoli del catalogo.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/alias-articoli`	Inserimento alias articolo
GET	`/alias-articoli`	Lista alias articoli
POST	`/alias-articoli/ricerca`	Ricerca alias articoli
GET	`/alias-articoli/{cod_alias}+{prog_alias}+{cod_articolo}`	Lettura alias articolo
PUT	`/alias-articoli/{cod_alias}+{prog_alias}+{cod_articolo}`	Modifica alias articolo
DELETE	`/alias-articoli/{cod_alias}+{prog_alias}+{cod_articolo}`	Cancellazione alias articolo

Inserimento Alias Articolo

Crea un nuovo alias per un articolo esistente. Consente di associare un codice alternativo a un articolo di magazzino.

`POST /alias-articoli`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
cod_alias	string	Sì	body	Codice alias da assegnare
prog_alias	integer	Sì	body	Progressivo alias (uguali)
cod_articolo	string	Sì	body	Codice articolo associato
descrizione	string	No	body	Descrizione dell'alias
nota1	string	No	body	Nota 1
nota2	string	No	body	Nota 2
nr_colli	integer	No	body	Numero colli
quantita	decimal	No	body	Quantità
um	string	No	body	Unità di misura

Esempio Request

```
POST /webapi/risorse/alias-articoli HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cod_alias": "ALT001",
  "prog_alias": 1,
  "cod_articolo": "ART001",
  "descrizione": "Codice alternativo fornitore",
  "quantita": 1.0,
  "um": "PZ"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/alias-articoli/ALT001+1+ART001
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Alias creato con successo
400	Bad Request	Parametri mancanti o non validi
409	Conflict	Alias già esistente

Note Operative

- La response restituisce l'header `Location` con l'URL della risorsa creata
- Richiede coordinate gestionale (Azienda + Anno)
- Versione minima: 88000

Lista Alias Articoli

Restituisce l'elenco di tutti gli alias articoli definiti nell'azienda corrente. Supporta paginazione.

`GET /alias-articoli`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione per la pagina successiva
fields	string	No	query	Lista campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/alias-articoli?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response


```
{
  "dati": [
    {
      "id": 17,
      "prog_alias": 0,
      "cod_alias": "8056734230635",
      "cod_articolo": "121212",
      "descrizione": "N",
      "nota1": "",
      "nota2": "",
      "nr_colli": 0,
      "quantita": 1.0,
      "um": "P",
      "data_crea": "20240313",
      "data_ult_mod": "20250429"
    },
    {
      "id": 18,
      "prog_alias": 0,
      "cod_alias": "8056734230636",
      "cod_articolo": "121212",
      "descrizione": "N",
      "nota1": "",
      "nota2": "",
      "nr_colli": 0,
      "quantita": 1.0,
```

```
      "um": "P",
      "data_crea": "20250429",
      "data_ult_mod": "20250429"
    }
  ],
  "data_ric_elab": "20260616 162820",
  "next": "token_paginazione_successiva"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti

Note Operative

- Supporta paginazione tramite parametri `max` e `next`
- Supporta selezione campi tramite parametro `fields`
- Versione minima: 82400

Ricerca Alias Articoli

Ricerca alias articoli con filtri avanzati. Accetta criteri di ricerca nel body della richiesta.

`POST /alias-articoli/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Lista campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/alias-articoli/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "cod_articolo",
```

```
      "condizione": "=",
      "valore": "ART001"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "id": 17,
      "prog_alias": 0,
      "cod_alias": "8056734230635",
      "cod_articolo": "121212",
      "descrizione": "N",
      "nota1": "",
      "nota2": "",
      "nr_colli": 0,
      "quantita": 1.0,
      "um": "P",
      "data_crea": "20240313",
      "data_ult_mod": "20250429"
    }
  ],
  "data_ric_elab": "20260616 162820"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata con successo
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- I filtri nel body seguono il formato standard: `campo`, `condizione`, `valore`
- Condizioni supportate: `=`, `<`, `<=`, `>`, `>=`, `<>`, `contiene`, `inizia_per`
- Versione minima: 82400

Lettura Alias Articolo

Recupera i dettagli di uno specifico alias articolo tramite la chiave composta.

`GET /alias-articoli/{cod_alias}+{prog_alias}+{cod_articolo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
cod_alias	string	Sì	path	Codice alias
prog_alias	integer	Sì	path	Progressivo alias
cod_articolo	string	Sì	path	Codice articolo associato

Esempio Request

```
GET /webapi/risorse/alias-articoli/ALT001+1+ART001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "id": 17,
  "prog_alias": 0,
  "cod_alias": "8056734230635",
  "cod_articolo": "121212",
  "descrizione": "N",
  "nota1": "",
  "nota2": "",
  "nr_colli": 0,
  "quantita": 1.0,
  "um": "P",
  "data_crea": "20240313",
  "data_ult_mod": "20250429"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Alias trovato e restituito
400	Bad Request	Formato chiave non valido
404	Not Found	Alias non trovato

Note Operative

- La chiave è composta da tre segmenti separati da +
- I campi stringa con caratteri speciali (es. /, \) possono richiedere encoding base32 o hex
- Versione minima: 88000

Modifica Alias Articolo

Aggiorna i dati di un alias articolo esistente.

PUT `/alias-articoli/{cod_alias}+{prog_alias}+{cod_articolo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
cod_alias	string	Sì	path	Codice alias
prog_alias	integer	Sì	path	Progressivo alias
cod_articolo	string	Sì	path	Codice articolo associato
descrizione	string	No	body	Nuova descrizione dell'alias
nota1	string	No	body	Nota 1
nota2	string	No	body	Nota 2
nr_colli	integer	No	body	Numero colli
quantita	decimal	No	body	Quantità
um	string	No	body	Unità di misura

Esempio Request

```
PUT /webapi/risorse/alias-articoli/ALT001+1+ART001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "descrizione": "Codice aggiornato fornitore principale",
  "quantita": 2.5
}
```

Esempio Response

```
{
  "id": 17,
  "prog_alias": 1,
  "cod_alias": "ALT001",
  "cod_articolo": "ART001",
  "descrizione": "Codice aggiornato fornitore principale",
  "nota1": "",
  "nota2": "",
  "nr_colli": 0,
  "quantita": 2.5,
  "um": "PZ",
  "data_crea": "20240313",
  "data_ult_mod": "20260616"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Alias aggiornato con successo
400	Bad Request	Dati non validi

Codice	Descrizione	Causa
404	Not Found	Alias non trovato

Note Operative

- Inviare nel body solo i campi da modificare
- I campi chiave non sono modificabili
- Versione minima: 88000

Cancellazione Alias Articolo

Elimina un alias articolo esistente.

```
DELETE /alias-articoli/{cod_alias}+{prog_alias}+{cod_articolo}
```

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
cod_alias	string	Sì	path	Codice alias
prog_alias	integer	Sì	path	Progressivo alias
cod_articolo	string	Sì	path	Codice articolo associato

Esempio Request

```
DELETE /webapi/risorse/alias-articoli/ALT001+1+ART001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Alias eliminato con successo
404	Not Found	Alias non trovato

Note Operative

- L'operazione è irreversibile
- Versione minima: 88000

Entità: Anagrafica Contatti

Gestione dell'anagrafica contatti aziendali. Permette operazioni CRUD complete sui contatti associati all'azienda.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/anagrafica-contatti`	Inserimento anagrafica contatti
GET	`/anagrafica-contatti`	Lista anagrafica contatti
POST	`/anagrafica-contatti/ricerca`	Ricerca anagrafica contatti
GET	`/anagrafica-contatti/{codice}`	Lettura anagrafica contatti
PUT	`/anagrafica-contatti/{codice}`	Modifica anagrafica contatti
DELETE	`/anagrafica-contatti/{codice}`	Cancellazione anagrafica contatti

Campi Disponibili

Campo	Tipo	Chiave	Descrizione
codice	Alfanumerico	Sì	Codice contatto
descrizione	Alfanumerico	—	Descrizione (ragione sociale)
indirizzo	Alfanumerico	—	Indirizzo
cap	Alfanumerico	—	C.A.P.
localita	Alfanumerico	—	Località
provincia	Alfanumerico	—	Provincia
nazionalita	Alfanumerico	—	Nazionalità
paese_iso	Alfanumerico	—	Codice ISO paese
telefono	Alfanumerico	—	Numero telefono
fax	Alfanumerico	—	Numero fax
email	Alfanumerico	—	Indirizzo e-mail
url	Alfanumerico	—	Indirizzo internet
note	Alfanumerico	—	Note

Campo	Tipo	Chiave	Descrizione
nome_ricerca	Alfanumerico	—	Nome di ricerca
partita_iva	Alfanumerico	—	Partita IVA
codice_fiscale	Alfanumerico	—	Codice fiscale
gest_per_fisica	Alfanumerico	—	Persona fisica (S/N)
codice_pdc	Alfanumerico	—	Codice piano dei conti
id_pagamento	Numerico	—	Codice pagamento
id_listino	Numerico	—	Numero listino
cat_sconto	Numerico	—	Categoria sconto
cat_provvigione	Numerico	—	Categoria provvigione
cod_iva	Alfanumerico	—	Codice assoggettamento IVA
cod_lingua	Numerico	—	Codice lingua estera
cod_agente	Alfanumerico	—	Codice agente
cod_cond_agente	Numerico	—	Codice condizioni agente
valuta_estera	Numerico	—	Valuta estera
cat_statistica	Numerico	—	Categoria statistica
zona	Numerico	—	Zona
addebito_bollo	Alfanumerico	—	Addebito bollo (S/N)
gest_cig_cup	Alfanumerico	—	Gestione CIG-CUP (S/N)
rg_bolle	Alfanumerico	—	Raggruppamenti bolle
rg_effetti	Alfanumerico	—	Raggruppamento effetti
proposto_e_ord	Alfanumerico	—	Proposto evadibilità ordini
sconto_icz	Alfanumerico	—	Sconto incondizionato
trasporto_mezzo	Alfanumerico	—	Trasporto a mezzo
cod_vett_abituale	Alfanumerico	—	Codice vettore abituale
tipo_porto	Alfanumerico	—	Tipo porto
descr_porto	Alfanumerico	—	Descrizione porto
codice_incoterms	Alfanumerico	—	Codice INCOTERMS
gest_calc_colli	Alfanumerico	—	Modalità calcolo colli
gest_calc_peso	Alfanumerico	—	Modalità calcolo peso
tipo_spese_sped	Alfanumerico	—	Tipo spese spedizione
val_spese_sped	Numerico con virgola	—	Valore spese spedizione
asp_est_beni	Numerico	—	Aspetto esteriore beni
fino_val_spese_sped	Numerico con virgola	—	Fino a valore spese spedizione

Campo	Tipo	Chiave	Descrizione
data_validita	Alfanumerico	—	Data validità
data_creazione	Alfanumerico	—	Data creazione
dt_anagrafica	Alfanumerico	—	Data aggiornamento anagrafica
utente_ult_mod	Alfanumerico	—	Utente ultima modifica (sola lettura)
data_ult_mod	Data e ora	—	Data ora ultima modifica (sola lettura, formato AAAAMMGG HHMMSS)

Inserimento Anagrafica Contatti

Crea un nuovo contatto nell'anagrafica.

`POST /anagrafica-contatti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	body	Codice contatto
descrizione	string	Sì	body	Descrizione (ragione sociale)
indirizzo	string	No	body	Indirizzo
cap	string	No	body	CAP
localita	string	No	body	Località
provincia	string	No	body	Provincia (sigla 2 caratteri)
telefono	string	No	body	Numero di telefono
email	string	No	body	Indirizzo email
(altri campi)		No	body	Vedi tabella Campi Disponibili

Esempio Request

```
POST /webapi/risorse/anagrafica-contatti HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "codice": "15",
  "descrizione": "Mario Rossi Srl",
  "indirizzo": "Via Roma 1",
  "cap": "40100",
  "localita": "BOLOGNA",
  "provincia": "BO",
```

```
  "nazionalita": "I",
  "paese_iso": "IT",
  "telefono": "051123456",
  "email": "info@mariorossi.it"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/anagrafica-contatti/15
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Contatto creato con successo
400	Bad Request	Parametri mancanti o non validi

Note Operative

- Richiede coordinate gestionale (Azienda + Anno)
- L'header `Location` nella response contiene l'URL della risorsa creata
- Versione minima: 87100

Lista Anagrafica Contatti

Restituisce l'elenco dei contatti dell'azienda corrente.

`GET /anagrafica-contatti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Lista campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/anagrafica-contatti?max=25 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response


```
{
  "dati": [
    {
      "codice": 1,
      "url": "",
      "descrizione": "Mario Rossi Srl",
      "note": "",
      "indirizzo": "Via Roma 1",
      "data_validita": "",
      "cap": "35031",
      "data_creazione": "20220318",
      "localita": "ABANO TERME",
      "dt_anagrafica": "20241004",
      "provincia": "PD",
      "partita_iva": "P",
      "nazionalita": "I",
      "codice_fiscale": "ABCDEF12G34H567I",
      "paese_iso": "IT",
      "gest_per_fisica": "N",
      "telefono": "049123456",
      "nome_ricerca": "ROSSI",
      "fax": "",
      "email": "info@rossi.it",
      "codice_pdc": "501.10011",
      "id_pagamento": 2,
      "id_listino": 2,
      "cat_sconto": 1,
      "cat_provvigione": 1,
      "cod_iva": "N08,1",
      "cod_lingua": 0,
      "cod_agente": "",
      "cod_cond_agente": 0,
      "valuta_estera": 1,
      "cat_statistica": 0,
      "zona": 0,
      "addebito_bollo": "N",
      "gest_cig_cup": "N",
      "rg_bolle": "N",
      "rg_effetti": "N",
      "proposto_e_ord": "N",
      "sconto_icz": "",
      "trasporto_mezzo": "",
      "cod_vett_abituale": "",
      "tipo_porto": "",
      "descr_porto": "",
      "codice_incoterms": "",
      "gest_calc_colli": "",
      "gest_calc_peso": "",
      "tipo_spese_sped": "",
```

```
      "val_spese_sped": 0.0,
      "asp_est_beni": 0,
      "fino_val_spese_sped": 0.0,
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20250226 164823"
    }
  ],
  "data_ric_elab": "20260616 163517",
  "next": "token_paginazione_successiva"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti

Note Operative

- Supporta paginazione e selezione campi
- Versione minima: 87100

Ricerca Anagrafica Contatti

Ricerca contatti con filtri avanzati.

`POST /anagrafica-contatti/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/anagrafica-contatti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "descrizione",
```

```
      "condizione": "contiene",
      "valore": "Rossi"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": 1,
      "descrizione": "Mario Rossi Srl",
      "localita": "ABANO TERME",
      "provincia": "PD"
    }
  ],
  "data_ric_elab": "20260616 163517"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Formato filtri standard WebAPI
- Versione minima: 87100

Lettura Anagrafica Contatti

Recupera i dati di uno specifico contatto.

`GET /anagrafica-contatti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice del contatto

Esempio Request

```
GET /webapi/risorse/anagrafica-contatti/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response


```
{
  "codice": 1,
  "url": "",
  "descrizione": "Mario Rossi Srl",
  "note": "",
  "indirizzo": "Via Roma 1",
  "data_validita": "",
  "cap": "35031",
  "data_creazione": "20220318",
  "localita": "ABANO TERME",
  "dt_anagrafica": "20241004",
  "provincia": "PD",
  "partita_iva": "P",
  "nazionalita": "I",
  "codice_fiscale": "ABCDEF12G34H567I",
  "paese_iso": "IT",
  "gest_per_fisica": "N",
  "telefono": "049123456",
  "nome_ricerca": "ROSSI",
  "fax": "",
  "email": "info@rossi.it",
  "codice_pdc": "501.10011",
  "id_pagamento": 2,
  "id_listino": 2,
  "cat_sconto": 1,
  "cat_provvigione": 1,
  "cod_iva": "N08,1",
  "cod_lingua": 0,
  "cod_agente": "",
  "cod_cond_agente": 0,
  "valuta_estera": 1,
  "cat_statistica": 0,
  "zona": 0,
  "addebito_bollo": "N",
  "gest_cig_cup": "N",
  "rg_bolle": "N",
  "rg_effetti": "N",
  "proposto_e_ord": "N",
  "sconto_icz": "",
  "trasporto_mezzo": "",
  "cod_vett_abituale": "",
  "tipo_porto": "",
  "descr_porto": "",
  "codice_incoterms": "",
  "gest_calc_colli": "",
  "gest_calc_peso": "",
  "tipo_spese_sped": "",
  "val_spese_sped": 0.0,
  "asp_est_beni": 0,
```

```
  "fino_val_spese_sped": 0.0,
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20250226 164823"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Contatto trovato
404	Not Found	Contatto non trovato

Note Operative

- Chiave alfanumerica semplice
- Versione minima: 87100

Modifica Anagrafica Contatti

Aggiorna i dati di un contatto esistente.

`PUT /anagrafica-contatti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice del contatto
descrizione	string	No	body	Descrizione (ragione sociale)
indirizzo	string	No	body	Nuovo indirizzo
cap	string	No	body	Nuovo CAP
localita	string	No	body	Nuova località
provincia	string	No	body	Nuova provincia
telefono	string	No	body	Nuovo telefono
email	string	No	body	Nuova email
(altri campi)		No	body	Vedi tabella Campi Disponibili

Esempio Request

```
PUT /webapi/risorse/anagrafica-contatti/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "telefono": "051999888",
```

```
  "email": "nuovaemail@mariorossi.it"
}
```

Esempio Response

```
{
  "codice": 1,
  "descrizione": "Mario Rossi Srl",
  "indirizzo": "Via Roma 1",
  "localita": "ABANO TERME",
  "provincia": "PD",
  "telefono": "051999888",
  "email": "nuovaemail@mariorossi.it",
  "data_ult_mod": "20260616 163600"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Contatto aggiornato
400	Bad Request	Dati non validi
404	Not Found	Contatto non trovato

Note Operative

- Inviare nel body solo i campi da modificare
- Versione minima: 87100

Cancellazione Anagrafica Contatti

Elimina un contatto dall'anagrafica.

`DELETE /anagrafica-contatti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice del contatto da eliminare

Esempio Request

```
DELETE /webapi/risorse/anagrafica-contatti/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Contatto eliminato
404	Not Found	Contatto non trovato

Note Operative

- Operazione irreversibile
- Versione minima: 87100

Entità: Anagrafica Unica

Gestione dell'anagrafica unica centralizzata. Supporta versionamento con data di fine validità e consultazione dello storico. L'archivio è **sovraziendale**: non è necessario specificare le Coordinate-Gestionale nell'header della request.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/anagrafica-unica`	Inserimento anagrafica unica
GET	`/anagrafica-unica`	Lista anagrafiche uniche
POST	`/anagrafica-unica/ricerca`	Ricerca anagrafica unica
GET	`/anagrafica-unica/{id_anagrafica}`	Lettura anagrafica unica (versione attiva)
GET	`/anagrafica-unica/{id_anagrafica}+{data_fine_valid}`	Lettura anagrafica unica alla data
PUT	`/anagrafica-unica/{id_anagrafica}`	Modifica anagrafica unica
DELETE	`/anagrafica-unica/{id_anagrafica}+{data_fine_valid}`	Cancellazione storicizzazione
GET	`/anagrafica-unica/storico`	Lista anagrafiche uniche (storico)
POST	`/anagrafica-unica/storico/ricerca`	Ricerca storico anagrafica unica

Campi Disponibili

L'endpoint espone oltre 90 campi, organizzati nei gruppi seguenti.

Dati Identificativi e Sistema

Campo	Tipo	Descrizione
`id_anagrafica`	Numerico	Codice anagrafico (chiave)
`annullato`	Alfanumerico	Annullata logicamente (`""` o `"N"`)
`data_creaz`	Alfanumerico	Data creazione (YYYYMMDD)
`data_modif`	Alfanumerico	Data modifica (YYYYMMDD)
`utente_ult_mod`	Alfanumerico	Utente ultima modifica
`data_ult_mod`	Data e ora	Data ora ultima modifica (YYYYMMDD HHMMSS)

Dati Anagrafici (anag_*)

Campo	Tipo	Descrizione
`anag_tipo_sogg`	Alfanumerico	Tipo soggetto (`M`/`F` persona fisica, `P` soc. persone, `C` soc. capitali, `E` ente)
`anag_cod_fis`	Alfanumerico	Codice fiscale
`anag_piva`	Alfanumerico	Partita IVA
`anag_cognome`	Alfanumerico	Cognome (persona fisica)
`anag_nome`	Alfanumerico	Nome (persona fisica)
`anag_rag_soc`	Alfanumerico	Ragione sociale
`anag_desc`	Alfanumerico	Descrizione
`anag_denom`	Alfanumerico	Denominazione
`anag_stor_nota`	Alfanumerico	Nota storicizzazione

Nascita (nasc_*)

Campo	Tipo	Descrizione
`nasc_data`	Alfanumerico	Data di nascita (YYYYMMDD)
`nasc_comune`	Alfanumerico	Comune di nascita
`nasc_cap`	Alfanumerico	CAP di nascita
`nasc_prov`	Alfanumerico	Provincia di nascita
`nasc_irpef`	Alfanumerico	Codice IRPEF nascita

Residenza (res_*)

Campo	Tipo	Descrizione

Campo	Tipo	Descrizione
res_indir	Alfanumerico	Indirizzo residenza
res_cap	Alfanumerico	CAP residenza
res_comune	Alfanumerico	Comune residenza
res_prov	Alfanumerico	Provincia residenza
res_nazione	Alfanumerico	Nazione residenza
res_localita	Alfanumerico	Località residenza
res_irpef	Alfanumerico	Codice IRPEF residenza
res_stesso_dom	Alfanumerico	Uguale a sede legale (S/N)
res_stor_nota	Alfanumerico	Nota storicizzazione residenza

Domicilio Fiscale (dom_*)

Campo	Tipo	Descrizione
dom_indir	Alfanumerico	Indirizzo domicilio fiscale
dom_cap	Alfanumerico	CAP domicilio fiscale
dom_comune	Alfanumerico	Comune domicilio fiscale
dom_prov	Alfanumerico	Provincia domicilio fiscale
dom_nazione	Alfanumerico	Nazione domicilio fiscale
dom_localita	Alfanumerico	Località domicilio fiscale
dom_irpef	Alfanumerico	Codice IRPEF domicilio fiscale
dom_stor_nota	Alfanumerico	Nota storicizzazione domicilio

Residenza Estera (est_*)

Campo	Tipo	Descrizione
est_cod_fiscale	Alfanumerico	Codice fiscale estero
est_cod_stato	Alfanumerico	Codice stato estero
est_nazione	Alfanumerico	Stato estero
est_stato_fed	Alfanumerico	Stato federato
est_localita	Alfanumerico	Località estera
est_indir	Alfanumerico	Indirizzo estero
est_stor_nota	Alfanumerico	Nota storicizzazione residenza estera

Contatti

Campo	Tipo	Descrizione
telefono	Alfanumerico	Numero telefono
telefono2	Alfanumerico	Numero telefono 2

Campo	Tipo	Descrizione
telefono3	Alfanumerico	Numero telefono 3
cellulare	Alfanumerico	Numero cellulare
fax	Alfanumerico	Numero fax
email	Alfanumerico	Indirizzo email
email2	Alfanumerico	Indirizzo email 2
pec	Alfanumerico	Indirizzo PEC
pec2	Alfanumerico	Indirizzo PEC 2
internet	Alfanumerico	Sito internet
nota_telefono	Alfanumerico	Nota telefono
nota_telefono2	Alfanumerico	Nota telefono 2
nota_telefono3	Alfanumerico	Nota telefono 3
nota_cellulare	Alfanumerico	Nota cellulare
nota_fax	Alfanumerico	Nota fax
nota_email	Alfanumerico	Nota email
nota_email2	Alfanumerico	Nota email 2

Legale Rappresentante (leg_*)

Campo	Tipo	Descrizione
leg_id_anag	Numerico	Codice ID anagrafica legale rappr.
leg_cod_fiscale	Alfanumerico	Codice fiscale legale rappr.
leg_sesso	Alfanumerico	Sesso legale rappr.
leg_cognome	Alfanumerico	Cognome legale rappr.
leg_nome	Alfanumerico	Nome legale rappr.
leg_cod_carica	Alfanumerico	Codice carica
leg_data_decor	Alfanumerico	Data decorrenza
leg_nasc_data	Alfanumerico	Data nascita legale rappr.
leg_nasc_comune	Alfanumerico	Comune nascita legale rappr.
leg_nasc_prov	Alfanumerico	Provincia nascita legale rappr.
leg_dom_indir	Alfanumerico	Indirizzo domicilio legale rappr.
leg_dom_cap	Alfanumerico	CAP domicilio legale rappr.
leg_dom_comune	Alfanumerico	Comune domicilio legale rappr.
leg_dom_prov	Alfanumerico	Provincia domicilio legale rappr.
leg_telefono	Alfanumerico	Telefono legale rappr.
leg_soc_id_anag	Numerico	Codice ID anagrafica società
leg_soc_cod_fis	Alfanumerico	Codice fiscale società

Campo	Tipo	Descrizione
leg_soc_rag_soc	Alfanumerico	Ragione sociale società
leg_stor_nota	Alfanumerico	Nota storicizzazione legale rappr.

Documenti di Identificazione (docum_*)

Campo	Tipo	Descrizione
docum_tipo	Alfanumerico	Tipo documento
docum_numero	Alfanumerico	Numero documento
docum_aut_ril	Alfanumerico	Autorità di rilascio
docum_data_ril	Alfanumerico	Data rilascio
docum_data_scad	Alfanumerico	Data scadenza
docum_stor_nota	Alfanumerico	Nota storicizzazione documento

Persona Fisica (pf_*)

Campo	Tipo	Descrizione
pf_stato_civ	Alfanumerico	Stato civile
pf_tit_studio	Alfanumerico	Titolo di studio
pf_nazione	Alfanumerico	Nazionalità
pf_qualif	Alfanumerico	Professione/qualifica
pf_titolo	Numerico	Titolo
pf_stor_nota	Alfanumerico	Nota storicizzazione persona fisica

Persona Giuridica (pg_*)

Campo	Tipo	Descrizione
pg_nat_giurid	Numerico	Natura giuridica
pg_stato_soc	Alfanumerico	Stato della società
pg_sit_societa	Alfanumerico	Situazione società
pg_stor_nota	Alfanumerico	Nota storicizzazione persona giuridica

Note

Campo	Tipo	Descrizione
nota	Array (5 elementi)	Note anagrafica (array di coppie [indice, testo])
nota_stor_nota	Alfanumerico	Nota storicizzazione note

Inserimento Anagrafica Unica

Crea una nuova anagrafica unica.

`POST /anagrafica-unica`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
anag_cod_fis	string	Sì	body	Codice fiscale
anag_rag_soc	string	Sì	body	Ragione sociale
anag_piva	string	Sì	body	Partita IVA
anag_tipo_sogg	string	No	body	Tipo soggetto (M/F/P/C/E)
res_indir	string	No	body	Indirizzo residenza
res_cap	string	No	body	CAP residenza
res_comune	string	No	body	Comune residenza
res_prov	string	No	body	Provincia residenza

> ℹ **Nota:** tutti i campi elencati nella sezione Campi Disponibili possono essere specificati nel body di inserimento, oltre a quelli obbligatori sopra indicati.

Esempio Request

```
POST /webapi/risorse/anagrafica-unica HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "anag_rag_soc": "Azienda Test Srl",
  "anag_cod_fis": "12345678901",
  "anag_piva": "12345678901",
  "anag_tipo_sogg": "C",
  "res_indir": "Via Milano 10",
  "res_cap": "20100",
  "res_comune": "Milano",
  "res_prov": "MI",
  "res_nazione": "IT"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/anagrafica-unica/125
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Anagrafica creata
400	Bad Request	Dati non validi o campi obbligatori mancanti

Note Operative

- NON richiede Coordinate-Gestionale (archivio sovraziendale)
- Il tag Location nella response restituisce l'id per referenziare la risorsa creata
- Versione minima: 85000

Lista Anagrafiche Uniche

Restituisce l'elenco delle anagrafiche uniche attive (versione corrente).

`GET /anagrafica-unica`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Lista campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/anagrafica-unica?max=30 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Response


```
{
  "dati": [
    {
      "id_anagrafica": 1,
      "annullato": "",
      "anag_tipo_sogg": "C",
      "anag_cod_fis": "FFFGGG98Y05G234E",
      "anag_piva": "00000000333",
      "anag_cognome": "",
      "anag_nome": "",
      "anag_rag_soc": "UBISOFT",
      "res_indir": "Via Roma 4, 13",
      "res_cap": "47600",
      "res_comune": "RIMINI",
      "res_prov": "RN",
      "res_nazione": "IT",
      "telefono": "0541123456",
      "email": "info@ubisoft.it",
      "data_ult_mod": "20231207 100629"
    }
  ],
```

```
  "data_ric_elab": "20260616 164344",
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita

Note Operative

- NON richiede Coordinate-Gestionale (archivio sovraziendale)
- Restituisce solo le anagrafiche con versione attiva (non scadute)
- Supporta paginazione tramite `max` e `next`
- Versione minima: 85000

Ricerca Anagrafica Unica

Ricerca anagrafiche uniche con filtri.

`POST /anagrafica-unica/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/anagrafica-unica/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "filtri": [
    {
      "campo": "anag_cod_fis",
      "condizione": "=",
      "valore": "FFFGGG98Y05G234E"
    }
  ]
}
```

Esempio Response


```
{
  "dati": [
    {
      "id_anagrafica": 1,
      "annullato": "",
      "anag_tipo_sogg": "C",
      "anag_cod_fis": "FFFGGG98Y05G234E",
      "anag_piva": "00000000333",
      "anag_rag_soc": "UBISOFT",
      "res_indir": "Via Roma 4, 13",
      "res_cap": "47600",
      "res_comune": "RIMINI",
      "res_prov": "RN"
    }
  ],
  "data_ric_elab": "20260616 164352"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Versione minima: 85000

Lettura Anagrafica Unica

Recupera una specifica anagrafica unica. Se la data di validità viene omessa, restituisce la versione attiva alla data di apertura della sessione WebAPI.

`GET /anagrafica-unica/{id_anagrafica}` oppure `/anagrafica-unica/{id_anagrafica}+{data_fine_valid}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id_anagrafica	integer	Sì	path	ID anagrafica
data_fine_valid	string	No	path	Data fine validità (YYYYMMDD). Se omessa, restituisce la versione attiva

Esempio Request (versione attiva)

```
GET /webapi/risorse/anagrafica-unica/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Request (storicizzazione a una data)

```
GET /webapi/risorse/anagrafica-unica/6+20230601 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Response


```
{
  "id_anagrafica": 1,
  "data_iniz_valid": "20231022",
  "data_fine_valid": "",
  "annullato": "N",
  "anag_tipo_sogg": "C",
  "anag_cod_fis": "FFFGGG98Y05G234E",
  "anag_piva": "IT00000000333",
  "anag_rag_soc": "UBISOFT",
  "res_indir": "Via Roma 4, 13",
  "res_cap": "47600",
  "res_comune": "RIMINI",
  "res_prov": "RN",
  "res_nazione": "IT",
  "res_localita": "RIMINI",
  "res_stesso_dom": "S",
  "telefono": "asd",
  "data_creaz": "20231005",
  "data_modif": "20231207",
  "data_ult_mod": "20231207 100629"
}
```


> ℹ **Nota:** nella lettura singola vengono restituiti anche i campi `data_iniz_valid`, `data_fine_valid` e i campi di storicizzazione (`xxx_stor_iniz`, `xxx_stor_data`) non presenti nella lista.

Codici di Stato

Codice	Descrizione	Causa
200	OK	Anagrafica trovata
400	Bad Request	Data validità errata
404	Not Found	Anagrafica non trovata

Note Operative

- Se `data_fine_valid` viene omessa nella chiave, viene utilizzata la data di apertura della sessione WebAPI (data odierna per l'anno corrente). In questo caso le Coordinate-Gestionale devono essere presenti nell'header per determinare l'anno
- Per leggere un dato storicizzato ad una certa data, passare la data effettiva di storicizzazione (visibile nell'endpoint `/storico`)
- Versione minima: 85000

Modifica Anagrafica Unica

Aggiorna un'anagrafica unica esistente.

`PUT /anagrafica-unica/{id_anagrafica}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id_anagrafica	integer	Sì	path	ID anagrafica
storicizza	string	No	query	Modalità storicizzazione (S o R)

Nel body: qualsiasi campo documentato nella sezione Campi Disponibili.

Esempio Request (modifica semplice)

```
PUT /webapi/risorse/anagrafica-unica/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "res_indir": "Via Torino 20",
  "res_comune": "Torino",
  "res_prov": "TO"
}
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Anagrafica aggiornata
400	Bad Request	Dati non validi
404	Not Found	Anagrafica non trovata

Note Operative

- Versione minima: 85000

Cancellazione Anagrafica Unica

Elimina una storicizzazione di un'anagrafica unica. Non è possibile cancellare l'intera anagrafica da WebAPI, ma solo eventuali storicizzazioni.

`DELETE /anagrafica-unica/{id_anagrafica}+{data_fine_valid}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id_anagrafica	integer	Sì	path	ID anagrafica
data_fine_valid	string	Sì	path	Data fine validità della storicizzazione da eliminare (YYYYMMDD)

Esempio Request

```
DELETE /webapi/risorse/anagrafica-unica/6+20230201 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Storicizzazione eliminata
404	Not Found	Anagrafica o storicizzazione non trovata

Note Operative

- La DELETE elimina un'intera storicizzazione alla data: se più blocchi di informazioni hanno la stessa data di storicizzazione verranno eliminati tutti
- Versione minima: 85000

Lista Anagrafiche Uniche (Storico)

Restituisce l'elenco delle anagrafiche uniche includendo le versioni storiche (scadute).

`GET /anagrafica-unica/storico`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/anagrafica-unica/storico?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Response


```
{
  "dati": [
    {
      "id_anagrafica": 1,
      "data_iniz_valid": "",
      "data_fine_valid": "20230810",
      "annullato": "",
      "tp_dati": "R",
      "sottotp_dati": 0,
      "res_indir": "Via Roma, 66",
      "res_cap": "47600",
      "res_comune": "RIMINI",
      "res_prov": "RN",
      "res_stor_nota": "vecchio indirizzo",
      "data_ult_mod": "20231023 102013"
    },
    {
      "id_anagrafica": 1,
      "data_iniz_valid": "",
      "data_fine_valid": "20230930",
      "annullato": "",
      "tp_dati": "R",
      "sottotp_dati": 0,
      "res_indir": "Via Roma, 17",
      "res_cap": "47600",
      "res_comune": "RIMINI",
      "res_prov": "RN",
      "res_stor_nota": "fino a fine settembre",
      "data_ult_mod": "20231023 113550"
    }
  ],
  "data_ric_elab": "20260616 164351",
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Storico restituito

Note Operative

- Include anche le versioni con data_fine_valid nel passato
- Lo storico include campi aggiuntivi: `data_iniz_valid`, `tp_dati`, `sottotp_dati`, `xxx_stor_iniz`, `xxx_stor_data`
- Versione minima: 85100

Ricerca Storico Anagrafica Unica

Ricerca nello storico delle anagrafiche uniche con filtri.

`POST /anagrafica-unica/storico/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/anagrafica-unica/storico/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "filtri": [
    {
      "campo": "id_anagrafica",
      "condizione": "=",
      "valore": 1
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
```

```
      "id_anagrafica": 1,
      "data_iniz_valid": "",
      "data_fine_valid": "20230810",
      "tp_dati": "R",
      "res_indir": "Via Roma, 66",
      "res_stor_nota": "vecchio indirizzo"
    },
    {
      "id_anagrafica": 1,
      "data_iniz_valid": "",
      "data_fine_valid": "20230930",
      "tp_dati": "R",
      "res_indir": "Via Roma, 17",
      "res_stor_nota": "fino a fine settembre"
    }
  ],
  "data_ric_elab": "20260616 164351"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Permette di cercare in tutte le versioni storiche dell'anagrafica
- Versione minima: 85100

Approfondimenti Operativi

Tipo Soggetto e Campi Specifici

Tipo	Valore `anag_tipo_sogg`	Campi utilizzati
Persona fisica	`"M"` o `"F"`	`anag_cognome`, `anag_nome`
Società di persone	`"P"`	`anag_rag_soc` + campi `leg_*` (legale rappresentante)
Società di capitali	`"C"`	`anag_rag_soc` + campi `leg_*`
Ente non commerciale	`"E"`	`anag_rag_soc` + campi `leg_*`

Per il legale rappresentante: se `leg_id_anag` referenzia un'anagrafica esistente, i dati vengono letti da quella; se `leg_id_anag=0`, vengono utilizzati i valori nei campi `leg_*`.

Modifica con Storicizzazione

È possibile storicizzare un'anagrafica durante la modifica specificando in query string il parametro `storicizza`:

Valore	Modalità	Descrizione
`storicizza=S`	Inserimento/modifica storicizzazione	Inserisce o modifica una storicizzazione esistente
`storicizza=R`	Revisione con storicizzazione	Storicizza i dati in linea e contestualmente li modifica

Gruppi Storicizzabili

Gruppo	Prefisso campi	Descrizione
Dati anagrafici testata	`anag_xxx`	Ragione sociale, tipo soggetto, ecc.
Residenza	`res_xxx`	Indirizzo, CAP, città, provincia
Domicilio	`dom_xxx`	Domicilio fiscale
Residenza estera	`est_xxx`	Dati per soggetti esteri
Legale rappresentante	`leg_xxx`	Dati legale rappresentante
Persona fisica	`pf_xxx`	Dati specifici persona fisica
Persona giuridica	`pg_xxx`	Dati specifici persona giuridica
Nota anagrafica	`nota`	Note

Ogni gruppo ha tre campi di storicizzazione: `xxx_stor_data`, `xxx_stor_iniz`, `xxx_stor_nota`.

Esempio: Storicizzazione indirizzo (storicizza=S)

```
PUT /webapi/risorse/anagrafica-unica/6?storicizza=S HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "res_indir": "Via Roma, 77",
  "res_stor_data": "20230810",
  "res_stor_nota": "vecchio indirizzo valido fino ad agosto"
}
```

Dove `res_stor_data` è la data di **fine validità**: l'indirizzo "Via Roma, 77" era valido fino al 10 agosto 2023.

Esempio: Revisione con storicizzazione (storicizza=R)

```
PUT /webapi/risorse/anagrafica-unica/6?storicizza=R HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "res_indir": "Via Roma, 78",
  "res_stor_iniz": "20230811",
  "res_stor_nota": "vecchio indirizzo valido fino ad agosto"
}
```

In questo caso l'indirizzo viene attualizzato con "Via Roma, 78" a partire dall'11/08/2023 e contestualmente il vecchio indirizzo viene storicizzato al 10/08/2023.

> ⚠ **Nota Bene:** `xxx_stor_iniz` e `xxx_stor_data` sono mutuamente esclusivi. Se specificati entrambi viene considerata solo la data inerente la modalità di storicizzazione scelta.

Storico — Tipi di Dati

L'endpoint `anagrafica-unica/storico` contiene tutte le storicizzazioni. Ogni record ha un campo `tp_dati`:

Valore	Tipo Dati
A	Dati anagrafici (anag_xxx)
R	Residenza (res_xxx)
I	Domicilio fiscale (dom_xxx)
E	Residenza estera (est_xxx)
L	Legale rappresentante (leg_xxx)
T	Documenti di identificazione (docum_xxx)
F	Ulteriori dati persona fisica (pf_xxx)
G	Ulteriori dati persona giuridica (pg_xxx)
N	Nota anagrafica (nota_xxx)

Esempio: Ricerca Storicizzazioni Modificate

```
POST /webapi/risorse/anagrafica-unica/storico/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json

{
  "filtri": [
    {
      "campo": "data_ult_mod",
      "condizione": ">",
      "valore": "20230101 000000"
```

```
    },
    {
      "campo": "id_anagrafica",
      "condizione": "=",
      "valore": 25
    }
  ]
}
```

Per avere la situazione aggiornata ad una delle date di storicizzazione ritornate, fare una GET con la data:

```
GET /webapi/risorse/anagrafica-unica/25+20230506 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Entità: Articoli

Gestione completa dell'anagrafica articoli di magazzino. Include operazioni CRUD, gestione allegati (immagini), progressivi e articoli abbinati.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/articoli`	Inserimento articolo
GET	`/articoli`	Lista articoli
POST	`/articoli/ricerca`	Ricerca articoli
GET	`/articoli/{codice}`	Lettura articolo
PUT	`/articoli/{codice}`	Modifica articolo
DELETE	`/articoli/{codice}`	Cancellazione articolo
GET	`/articoli/{codice}/allegati/{tipo}`	Lettura allegato articolo
GET	`/articoli/{codice}/progressivi`	Lista progressivi articolo
POST	`/articoli/{codice}/abbinati`	Inserimento abbinato
GET	`/articoli/{codice}/abbinati/{codice_abb}+{prog_abbinati}`	Lettura abbinato articolo

Metodo	URL	Descrizione
PUT	`/articoli/{codice}/abbinati/{codice_abb}+{prog_abbinati}`	Modifica abbinato articolo
DELETE	`/articoli/{codice}/abbinati/{codice_abb}+{prog_abbinati}`	Cancellazione abbinato articolo

Inserimento Articolo

Crea un nuovo articolo nell'anagrafica di magazzino.

`POST /articoli`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	body	Codice articolo
descrizione	string	Sì	body	Descrizione articolo
um_principale	string	No	body	Unità di misura primaria
alq_iva	string	No	body	Aliquota IVA
cod_natura	string	No	body	Codice natura articolo
cod_grp_merc	string	No	body	Codice gruppo merceologico
prz_listino	array	No	body	Prezzi listino (array indicizzato per listino)

Esempio Request

```
POST /webapi/risorse/articoli HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "codice": "ART001",
  "descrizione": "Prodotto esempio",
  "um_principale": "PZ",
  "alq_iva": "22",
  "cod_grp_merc": "ELETTR"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/articoli/ART001
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Articolo creato con successo
400	Bad Request	Parametri non validi
409	Conflict	Codice articolo già esistente

Note Operative

- Richiede coordinate gestionale
- Versione minima: 81500

Lista Articoli

Restituisce l'elenco degli articoli di magazzino.

`GET /articoli`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/articoli?max=100&fields=codice,descrizione,um_principale
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "ART001",
      "descrizione": "Prodotto esempio",
      "um_principale": "PZ"
    },
    {
      "codice": "ART002",
```

```
      "descrizione": "Secondo prodotto",
      "um_principale": "KG"
    }
  ],
  "data_ric_elab": "20260617 091346",
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Note Operative

- Supporta paginazione e selezione campi
- Il campo `data_ric_elab` indica la data/ora di elaborazione della richiesta
- Versione minima: 81500

Ricerca Articoli

Ricerca articoli con filtri avanzati.

`POST /articoli/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/articoli/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "descrizione",
      "condizione": "contiene",
```

```
      "valore": "prodotto"
    },
    {
      "campo": "cos_ult",
      "condizione": ">",
      "valore": 10.00
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "ART001",
      "descrizione": "Prodotto esempio",
      "um_principale": "PZ"
    }
  ],
  "data_ric_elab": "20260617 091359"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Più filtri sono combinati in AND
- Versione minima: 81500

Lettura Articolo

Recupera tutti i dati di uno specifico articolo.

`GET /articoli/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo

Esempio Request

```
GET /webapi/risorse/articoli/ART001 HTTP/1.1
Host: server:porta
```

```
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "codice": "ART001",
  "descrizione": "Prodotto esempio",
  "um_principale": "PZ",
  "alq_iva": "22",
  "cod_natura": "9",
  "cod_grp_merc": "ELETTR",
  "cos_ult": 25.50
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Articolo trovato
404	Not Found	Articolo non trovato

Note Operative

- Supporta encoding base32/hex per codici con caratteri speciali (chiave encodable)
- La response contiene direttamente l'oggetto articolo (senza wrapper `dati`)
- Versione minima: 81500

Modifica Articolo

Aggiorna i dati di un articolo esistente.

`PUT /articoli/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo
descrizione	string	No	body	Nuova descrizione
um_principale	string	No	body	Nuova unità di misura primaria
alq_iva	string	No	body	Nuova aliquota IVA

Esempio Request

```
PUT /webapi/risorse/articoli/ART001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

```
{
  "descrizione": "Prodotto esempio aggiornato",
  "alq_iva": "22"
}
```

Esempio Response

```
{
  "codice": "ART001",
  "descrizione": "Prodotto esempio aggiornato",
  "alq_iva": "22"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Articolo aggiornato
400	Bad Request	Dati non validi
404	Not Found	Articolo non trovato

Note Operative

- Inviare solo i campi da aggiornare
- Versione minima: 81500

Cancellazione Articolo

Elimina un articolo dall'anagrafica.

`DELETE /articoli/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo da eliminare

Esempio Request

```
DELETE /webapi/risorse/articoli/ART001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa

Codice	Descrizione	Causa
200	OK	Articolo eliminato
404	Not Found	Articolo non trovato
409	Conflict	Articolo utilizzato in documenti

Note Operative

- Potrebbe fallire se l'articolo è referenziato in documenti esistenti
- Versione minima: 81500

Lettura Allegato Articolo

Recupera un allegato (immagine, icona, immagine-catalogo) associato all'articolo.

`GET /articoli/{codice}/allegati/{tipo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo
tipo	string	Sì	path	Tipo allegato: `immagine`, `icona`, `immagine-catalogo`

Esempio Request

```
GET /webapi/risorse/articoli/ART001/allegati/immagine HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
Content-Type: image/jpeg

<contenuto binario immagine>
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Allegato trovato
404	Not Found	Allegato non presente

Note Operative

- Valori ammessi per `tipo`: `immagine`, `icona`, `immagine-catalogo`
- La response contiene il binario dell'immagine con Content-Type appropriato

- Versione minima: 81600

Lista Progressivi Articolo

Restituisce i progressivi (quantità, giacenze per magazzino) di uno specifico articolo.

`GET /articoli/{codice}/progressivi`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo (campo `cod_articolo`)

Esempio Request

```
GET /webapi/risorse/articoli/ART001/progressivi HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "dati": [
    {
      "cod_magazzino": 1,
      "giacenza": 150.00,
      "ordinato_clienti": 20.00,
      "ordinato_fornitori": 50.00
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Progressivi restituiti
404	Not Found	Articolo non trovato

Note Operative

- Restituisce dati per tutti i magazzini dove l'articolo ha movimenti
- Versione minima: 82300

Inserimento Abbinato

Associa un articolo abbinato a un articolo principale.

`POST /articoli/{codice}/abbinati`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo principale (campo `codice_articolo`)
codice_abb	string	Sì	body	Codice articolo abbinato
prog_abbinati	integer	Sì	body	Progressivo abbinamento
quantita	number	No	body	Quantità abbinata

Esempio Request

```
POST /webapi/risorse/articoli/ART001/abbinati HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "codice_abb": "ACC001",
  "prog_abbinati": 1,
  "quantita": 2
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/articoli/ART001/abbinati/ACC001+1
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Abbinato creato
400	Bad Request	Dati non validi
404	Not Found	Articolo principale non trovato

Note Operative

- Versione minima: 81500

Lettura Abbinato Articolo

Recupera i dati di un abbinamento specifico.

`GET /articoli/{codice}/abbinati/{codice_abb}+{prog_abbinati}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo principale
codice_abb	string	Sì	path	Codice articolo abbinato
prog_abbinati	integer	Sì	path	Progressivo abbinamento

Esempio Request

```
GET /webapi/risorse/articoli/ART001/abbinati/ACC001+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "codice_articolo": "ART001",
  "codice": "ACC001",
  "prog_abbinati": 1,
  "quantita": 2
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Abbinato trovato
404	Not Found	Abbinato non trovato

Note Operative

- Versione minima: 88000

Modifica Abbinato Articolo

Aggiorna i dati di un abbinamento esistente.

PUT `/articoli/{codice}/abbinati/{codice_abb}+{prog_abbinati}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo principale
codice_abb	string	Sì	path	Codice articolo abbinato
prog_abbinati	integer	Sì	path	Progressivo abbinamento
quantita	number	No	body	Nuova quantità

Esempio Request

```
PUT /webapi/risorse/articoli/ART001/abbinati/ACC001+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "quantita": 3
}
```

Esempio Response

```
{
  "codice_articolo": "ART001",
  "codice": "ACC001",
  "prog_abbinati": 1,
  "quantita": 3
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Abbinato aggiornato
404	Not Found	Abbinato non trovato

Note Operative

- Versione minima: 88000

Cancellazione Abbinato Articolo

Rimuove un abbinamento da un articolo.

DELETE /articoli/{codice}/abbinati/{codice_abb}+{prog_abbinati}

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice articolo principale
codice_abb	string	Sì	path	Codice articolo abbinato
prog_abbinati	integer	Sì	path	Progressivo abbinamento

Esempio Request

```
DELETE /webapi/risorse/articoli/ART001/abbinati/ACC001+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Abbinato eliminato
404	Not Found	Abbinato non trovato

Note Operative

- Versione minima: 88000

Esempi

Allegati - Accesso alle Immagini Articolo

Solo per gli archivi che hanno allegati (come le immagini articoli) è disponibile un'entità dedicata per l'accesso ai documenti. Nell'header della risposta sarà specificato il `Content-Type` corretto per il file restituito (ad esempio `image/jpeg` per le immagini).

I tre tipi di allegato disponibili per gli articoli sono:

Tipo Allegato	Descrizione
`icona`	Icona piccola dell'articolo
`immagine-catalogo`	Immagine per il catalogo
`immagine`	Immagine principale dell'articolo

Lettura dell'icona di un articolo:

```
GET /webapi/risorse/articoli/FUELEX97/allegati/icona HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Lettura dell'immagine di catalogo:

```
GET /webapi/risorse/articoli/FUELEX97/allegati/immagine-catalogo HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Lettura dell'immagine articolo:

```
GET /webapi/risorse/articoli/FUELEX97/allegati/immagine HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Abbinati - Dettaglio Operativo

L'end-point `articoli` include un sotto end-point `abbinati` che segue la stessa logica dei progressivi degli articoli e restituisce gli abbinati riferiti all'articolo scelto.

È disponibile anche un end-point specifico `articoli-abbinati` su cui si possono ricercare e listare tutti gli abbinati di tutti gli articoli.

Gli abbinati devono essere referenziati obbligatoriamente dal **codice abbinato** e da un **progressivo**.

Creazione di un abbinato con campi dettagliati:

```
POST /webapi/risorse/articoli/121212/abbinati HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "prog_abbinati": 0,
  "codice": "121213",
  "cat_abbinamento": "ACCESSORI",
  "descr_cat_abbina": "Accessori compatibili",
  "nota_01": "Nota aggiuntiva 1",
  "nota_02": "Nota aggiuntiva 2"
}
```

Lettura di un singolo abbinato (articolo 121212, abbinato 121213 progressivo 0):

```
GET /webapi/risorse/articoli/121212/abbinati/121213+0 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Modifica abbinato:

```
PUT /webapi/risorse/articoli/121212/abbinati/121213+0 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "nota_01": "Nota aggiornata"
}
```

Cancellazione abbinato:

```
DELETE /webapi/risorse/articoli/121212/abbinati/121213+0 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempi (live)

I seguenti esempi sono stati testati su un'installazione reale Passepartout (Azienda UB1, Anno 2026).

GET — Lista articoli con selezione campi

```
GET /webapi/risorse/articoli?fields=codice,descrizione&max=5 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Response (200 OK):


```
{
  "dati": [
    {
      "codice": "&''UN TEST CATTIVO",
      "descrizione": "Test cattivo"
    },
    {
      "codice": "/ \\",
      "descrizione": "TEST DATA ULT MOD"
    },
    {
      "codice": "/ASD",
      "descrizione": "asd sdjkfh sdkjfh sd"
    },
    {
      "codice": "/\\",
      "descrizione": "Articolo Test WebAPI"
    },
    {
      "codice": "121212",
      "descrizione": "Descrizione molto lu"
    }
  ],
  "data_ric_elab": "20260610 163226",
  "next":
"57415f3333626366396362d363464392d313166312d396138392d396436653165336262303264-
3801000000000000"
}
```


> ℹ **Nota:** Il campo `data_ric_elab` indica la data e ora di elaborazione della richiesta. Il campo `next` è il token di paginazione per la pagina successiva.

POST /ricerca — Ricerca articoli con filtro "contiene"

```
POST /webapi/risorse/articoli/ricerca?fields=codice,descrizione&max=5 HTTP/1.1
Host: gsuzzi:9014
```

```
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "descrizione",
      "condizione": "contiene",
      "case_insensitive": true,
      "valore": "bici"
    }
  ]
}
```

Response (200 OK):

```
{
  "dati": [
    {
      "codice": "BICCOMPLETAAUTO",
      "descrizione": "BICI COMPLETA AUTOMA"
    },
    {
      "codice": "BICI",
      "descrizione": "BICI ASSEMBLATA"
    },
    {
      "codice": "BICI1234",
      "descrizione": "BICI ASSEMBLATA 12 3"
    },
    {
      "codice": "BICI23DE",
      "descrizione": "BICI ASSEMBLATA 23 D"
    },
    {
      "codice": "BICI3456",
      "descrizione": "BICI ASSEMBLATA 34 5"
    }
  ],
  "data_ric_elab": "20260610 163302",
  "next":
"57415f34393366306664352d363464392d313166312d396633362d396436653165336262303264-
ea00000000000000"
}
```

Entità: Articoli Abbinati

Consultazione trasversale degli abbinamenti articoli. Permette di visualizzare e ricercare tutti gli abbinamenti definiti nell'azienda indipendentemente dall'articolo principale.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	/articoli-abbinati	Lista abbinati articolo
POST	/articoli-abbinati/ricerca	Ricerca abbinati articolo

Struttura Campi

Campo	Tipo	Chiave	Descrizione
codice_articolo	Alfanumerico	Sì	Codice articolo principale
codice	Alfanumerico	Sì	Codice articolo abbinato
prog_abbinati	Numerico	Sì	Progressivo articoli abbinati
cat_abbinamento	Alfanumerico	No	Categoria abbinamento
descr_cat_abbina	Alfanumerico	No	Descrizione categoria abbinamento
nota_01	Alfanumerico	No	Nota 1
nota_02	Alfanumerico	No	Nota 2
utente_ult_mod	Alfanumerico	No	Utente ultima modifica
data_ult_mod	Data e ora	No	Data/ora ultima modifica (formato AAAAMMGG HHMMSS)

Lista Abbinati Articolo

Restituisce l'elenco completo di tutti gli abbinamenti articoli dell'azienda.

GET /articoli-abbinati

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/articoli-abbinati?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=XXX Anno=YYYY
```

Esempio Response


```
{
  "dati": [
    {
      "prog_abbinati": 0,
      "codice": "ATT001",
      "cat_abbinamento": "",
      "descr_cat_abbina": "",
      "codice_articolo": "ART001",
      "nota_01": "Nota di esempio",
      "nota_02": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20251120 150019"
    },
    {
      "prog_abbinati": 0,
      "codice": "ART002",
      "cat_abbinamento": "CAT1",
      "descr_cat_abbina": "Categoria principale",
      "codice_articolo": "ART003",
      "nota_01": "",
      "nota_02": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20251119 094045"
    }
  ],
  "data_ric_elab": "20260617 092044",
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate gestionale mancanti o non valide

Note Operative

- Vista trasversale: elenca tutti gli abbinamenti di tutti gli articoli dell'azienda
- Supporta paginazione tramite parametri `max` e `next` (vedi Paginazione)
- Supporta selezione campi tramite parametro `fields`
- Versione minima: 88000

Ricerca Abbinati Articolo

Ricerca abbinamenti con filtri.

POST `/articoli-abbinati/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/articoli-abbinati/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=YYYY

{
  "filtri": [
    {
      "campo": "codice_articolo",
      "condizione": "=",
      "valore": "ART001"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "prog_abbinati": 0,
      "codice": "ATT001",
      "cat_abbinamento": "",
      "descr_cat_abbina": "",
      "codice_articolo": "ART001",
      "nota_01": "Nota di esempio",
      "nota_02": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20251120 150019"
    }
  ],
  "data_ric_elab": "20260617 092044"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata

Codice	Descrizione	Causa
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Supporta tutte le condizioni di filtro disponibili (vedi Ricerca e Filtri)
- La chiave composita è formata da: `codice_articolo` + `codice` + `prog_abbinati`
- Versione minima: 88000

Entità: Aziende

Gestione delle aziende configurate nell'installazione Passepartout. Permette di elencare le aziende disponibili e crearne di nuove.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	`/aziende`	Lista aziende
POST	`/aziende`	Creazione azienda

> ℹ **Nota:** L'endpoint `aziende` NON supporta ricerca con filtri (`POST /aziende/ricerca` non è disponibile). Non supporta inoltre `?info=true` per la discovery automatica dei campi.

Lista Aziende

Restituisce l'elenco delle aziende configurate nell'installazione. Non richiede coordinate gestionale.

`GET /aziende`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione per la pagina successiva
fields	string	No	query	Lista campi da restituire (separati da virgola)

Campi Restituiti

Campo	Tipo	Descrizione
id	integer	Identificativo numerico progressivo

Campo	Tipo	Descrizione
sigla	string	Sigla azienda (3 caratteri)
ragione_sociale	string	Ragione sociale
descrizione	string	Descrizione aggiuntiva
cod_fis_piva	string	Codice fiscale o partita IVA
stato	string	Stato dell'azienda (`Ok`, `Da aggiornare`, `Anno non gestito`)
tipo	string	Tipologia societaria (`Soc. Capitali`, `Soc. Persone`, `Pers. Fisica`, `Ente non commerciale`)
gest_impresa_prof	string	Tipo gestione (`Impresa`, `Professionista`)
gestione_fiscale	string	Regime fiscale (`Ordinaria`, `Semplificata`)
primo_anno_gest	integer	Primo anno gestito
ult_anno_gest	integer	Ultimo anno gestito
raggruppamento	string	Codice raggruppamento (vuoto se non impostato)
livello_azienda	string	Livello di abilitazione dell'azienda

Esempio Request

```
GET /webapi/risorse/aziende HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
```

Esempio Response


```
{
  "Aziende": [
    {
      "id": 21,
      "sigla": "BIK",
      "ragione_sociale": "VELO BIKE LTD",
      "descrizione": "PRODUZIONE BICI DA CORSA",
      "cod_fis_piva": "04139120374",
      "stato": "Ok",
      "tipo": "Soc. Capitali",
      "gest_impresa_prof": "Impresa",
      "gestione_fiscale": "Ordinaria",
      "primo_anno_gest": 2017,
      "ult_anno_gest": 2026,
      "raggruppamento": "",
      "livello_azienda": "3"
    },
    {
      "id": 74,
      "sigla": "UB1",
      "ragione_sociale": "UBISOFT",
```

```
      "descrizione": "COPIA DI UBI IN DATA 16/02/2024",
      "cod_fis_piva": "FFFGGG98Y05G234E",
      "stato": "Ok",
      "tipo": "Soc. Capitali",
      "gest_impresa_prof": "Impresa",
      "gestione_fiscale": "Ordinaria",
      "primo_anno_gest": 2017,
      "ult_anno_gest": 2026,
      "raggruppamento": "",
      "livello_azienda": "3"
    }
  ]
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
401	Unauthorized	Credenziali non valide

Note Operative

- NON richiede coordinate gestionale (Azienda/Anno)
- Supporta paginazione tramite `max` e `next`
- Supporta selezione campi tramite `fields`
- Il wrapper della response è `Aziende` (non `dati`)
- Il campo `stato` indica se l'azienda è utilizzabile: solo le aziende con stato `Ok` e con `ult_anno_gest` che copra l'anno richiesto sono operabili nelle altre API
- NON supporta ricerca con filtri
- Versione minima: 85000

Creazione Azienda

Crea una nuova azienda nell'installazione Passepartout. La creazione richiede il collegamento ad un soggetto **già presente in anagrafica unica** (vedi Entità: Anagrafica Unica). Prima della creazione dell'azienda è necessario usare `POST /anagrafica-unica` per creare il soggetto.

`POST /aziende`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
in_sigla_azienda	string	Sì	body	Sigla azienda (3 caratteri alfanumerici)
in_data_inizio_gestione	string	Sì	body	Data di creazione (formato AAAAMMGG)

Campo	Tipo	Obbl.	Posizione	Descrizione
in_gest_impresa_professionista	string	Sì	body	P = Professionista, I = Impresa
in_gestione_fiscale	string	Sì	body	Tipo gestione fiscale (vedi tabella sotto)
in_contabilita_cassa	string	Condizionale	body	Solo se impresa + semplificata (vedi tabella sotto)
in_id_anagrafica	integer	Sì	body	ID anagrafica unica collegata
in_mese_inizio_anno_contabile	integer	Sì	body	Mese inizio anno contabile (1-12)

Valori Gestione Fiscale (in_gestione_fiscale)

Valore	Descrizione
0	Ordinaria
S	Semplificata
U	Super minimo
I	Solo attività istituzionale
Y	Forfetaria 2015

Valori Contabilità per Cassa (in_contabilita_cassa)

Valore	Descrizione
S	Reale
V	Virtuale

> ⚠ **Nota Bene:** Il campo `in_contabilita_cassa` deve essere specificato **esclusivamente** se `in_gest_impresa_professionista=I` e `in_gestione_fiscale=S`. In tutti gli altri casi non viene accettato e la procedura restituisce un errore.

Esempio Request

```
POST /webapi/risorse/aziende HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json

{
  "in_sigla_azienda": "MI2",
  "in_data_inizio_gestione": "20230301",
  "in_gest_impresa_professionista": "P",
```

```
  "in_gestione_fiscale": "Y",
  "in_contabilita_cassa": "V",
  "in_id_anagrafica": 12,
  "in_mese_inizio_anno_contabile": 1
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/aziende/MI2
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Azienda creata con successo
400	Bad Request	Parametri non validi o mancanti
409	Conflict	Sigla azienda già esistente

Note Operative

- NON richiede coordinate gestionale
- La response restituisce l'header `Location` con l'URL della risorsa creata
- Richiede un soggetto pre-esistente in anagrafica unica (`in_id_anagrafica`)
- Versione minima: 86000

Esempi Verificati (Live)

GET — Lista aziende

Installazione: `gsuzzi:9014` — Credenziali: `webapi:w`

```
GET /webapi/risorse/aziende?max=2 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
```

Response (200 OK):

```
{
  "Aziende": [
    {
      "id": 1,
      "sigla": "AA2",
      "ragione_sociale": "Velo Bike",
      "descrizione": "",
      "cod_fis_piva": "DUCAAAAAAZ5",
      "stato": "Anno non gestito",
      "tipo": "Soc. Capitali",
      "gest_impresa_prof": "Impresa",
```

```
      "gestione_fiscale": "Semplificata",
      "primo_anno_gest": 2023,
      "ult_anno_gest": 2023,
      "raggruppamento": "80",
      "livello_azienda": "3"
    },
    {
      "id": 2,
      "sigla": "AA3",
      "ragione_sociale": "MTB SPETTACOLO",
      "descrizione": "",
      "cod_fis_piva": "DUCAAAAAA",
      "stato": "Anno non gestito",
      "tipo": "Soc. Capitali",
      "gest_impresa_prof": "Impresa",
      "gestione_fiscale": "Semplificata",
      "primo_anno_gest": 2023,
      "ult_anno_gest": 2023,
      "raggruppamento": "80",
      "livello_azienda": "3"
    }
  ]
}
```


Entità: Banche

Consultazione dell'archivio banche configurate nell'azienda. Endpoint di sola lettura con supporto paginazione, selezione campi e ricerca con filtri.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	/banche	Lista banche
POST	/banche/ricerca	Ricerca banche con filtri

Campi Disponibili

Campo	Tipo	Descrizione
codice	Numerico	Codice identificativo della banca
banca	Alfanumerico	Denominazione della banca
filiale	Alfanumerico	Denominazione della filiale
abi	Alfanumerico	Coordinate bancarie - ABI
cab	Alfanumerico	Coordinate bancarie - CAB

Campo	Tipo	Descrizione
bic	Alfanumerico	Coordinate bancarie - BIC (SWIFT)
indirizzo	Alfanumerico	Indirizzo della filiale
cap	Alfanumerico	CAP
comune	Alfanumerico	Comune
provincia	Alfanumerico	Provincia (sigla)
telefono	Alfanumerico	Numero di telefono
fax	Alfanumerico	Numero di fax
email	Alfanumerico	Indirizzo e-mail
url	Alfanumerico	Indirizzo internet
sigla_azienda	Alfanumerico	Sigla azienda
intendenza	Alfanumerico	Intendenza di finanza che ha rilasciato l'autorizzazione al bollo virtuale
nr_autorizza	Numerico	Numero autorizzazione dell'intendenza di finanza
data_autorizza	Alfanumerico	Data autorizzazione dell'intendenza di finanza
tipo_invio_arch	Alfanumerico	Tipo invio archivio
gest_cr_lf	Alfanumerico	Flag aggiungi CR/LF
cf_piva_cli_for	Alfanumerico	Riporta codice fiscale o partita IVA cliente/fornitore
cf_piva_azienda	Alfanumerico	Riporta codice fiscale o partita IVA azienda
gest_esito_pg	Alfanumerico	Gestione esito pagamento
gest_bonif_cbi	Alfanumerico	Gestione bonifici CBI - Coordinate ordinante
gest_mav_cbi	Alfanumerico	Gestione MAV CBI - Coordinate ordinante
gest_rid_cbi	Alfanumerico	Gestione RID CBI - Coordinate ordinante
tipo_codice_rid	Alfanumerico	RID - Tipo codice azienda creditrice
tipo_inc_rid	Alfanumerico	RID - Tipo incasso

Nota: L'endpoint non espone operazioni di scrittura (POST per creazione, PUT, DELETE). La gestione delle banche avviene esclusivamente dal gestionale.

Lista Banche

Restituisce l'elenco delle banche configurate nell'azienda corrente.

`GET` /banche

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione per la pagina successiva
fields	string	No	query	Lista campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/banche?max=3 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "codice": 1,
      "banca": "BANCA D'ITALIA",
      "filiale": "BONIFICI DA FILIALE",
      "abi": "01000",
      "cab": "01001",
      "bic": "",
      "indirizzo": "VIA ARSENALE, 8",
      "cap": "10121",
      "comune": "TORINO",
      "provincia": "TO",
      "telefono": "",
      "fax": "",
      "email": "",
      "url": "",
      "sigla_azienda": "BIK",
      "intendenza": "",
      "nr_autorizza": 0,
      "data_autorizza": "",
      "tipo_invio_arch": "C",
      "gest_cr_lf": "N",
      "cf_piva_cli_for": "",
      "cf_piva_azienda": "",
      "gest_esito_pg": "N",
      "gest_bonif_cbi": "S",
      "gest_mav_cbi": "N",
      "gest_rid_cbi": "N",
      "tipo_codice_rid": "4",
      "tipo_inc_rid": ""
    },
    {
```

```
      "codice": 2,
      "banca": "BANCA D'ITALIA",
      "filiale": "ROMA VIA DEI MILLE,52 - ROMA SUCCURS",
      "abi": "01000",
      "cab": "03201",
      "bic": "",
      "indirizzo": "VIA DEI MILLE,52 - ROMA SUCCUR",
      "cap": "00185",
      "comune": "ROMA",
      "provincia": "RM",
      "telefono": "",
      "fax": "",
      "email": "",
      "url": "",
      "sigla_azienda": "",
      "intendenza": "",
      "nr_autorizza": 0,
      "data_autorizza": "",
      "tipo_invio_arch": "C",
      "gest_cr_lf": "N",
      "cf_piva_cli_for": "",
      "cf_piva_azienda": "",
      "gest_esito_pg": "N",
      "gest_bonif_cbi": "S",
      "gest_mav_cbi": "N",
      "gest_rid_cbi": "N",
      "tipo_codice_rid": "4",
      "tipo_inc_rid": ""
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti o non valide

Note Operative

- Richiede coordinate gestionale (Azienda + Anno)
- Supporta paginazione tramite parametri `max` e `next` (vedi Paginazione)
- Supporta selezione campi tramite parametro `fields`
- Non è disponibile la lettura di una singola banca per chiave
- Versione minima: 84200

Ricerca Banche

Ricerca banche con filtri avanzati.

`POST /banche/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Lista campi da restituire (separati da virgola)
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/banche/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "banca",
      "condizione": "contiene",
      "valore": "INTESA"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": 3,
      "banca": "INTESA SANPAOLO SPA",
      "filiale": "PARMA TARGET",
      "abi": "03069",
      "cab": "12712",
      "bic": "",
      "indirizzo": "VIA LANGHIRANO 1",
      "cap": "43100",
      "comune": "PARMA",
      "provincia": "PR",
      "telefono": "",
```

```
      "fax": "",
      "email": "",
      "url": "",
      "sigla_azienda": "",
      "intendenza": "",
      "nr_autorizza": 0,
      "data_autorizza": "",
      "tipo_invio_arch": "C",
      "gest_cr_lf": "N",
      "cf_piva_cli_for": "",
      "cf_piva_azienda": "",
      "gest_esito_pg": "N",
      "gest_bonif_cbi": "S",
      "gest_mav_cbi": "N",
      "gest_rid_cbi": "N",
      "tipo_codice_rid": "4",
      "tipo_inc_rid": ""
    }
  ]
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata con successo
400	Bad Request	Filtri non validi o coordinate gestionale mancanti

Note Operative

- I filtri seguono il formato standard WebAPI (vedi Ricerca e Filtri)
- Condizioni supportate: `=`, `<`, `<=`, `>`, `>=`, `<>`, `contiene`, `inizia_per`
- I campi filtrabili corrispondono ai campi restituiti nella response
- Versione minima: 84200

Entità: Clienti

Gestione completa dell'anagrafica clienti. Supporta operazioni CRUD con paginazione e ricerca avanzata.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/clienti`	Inserimento cliente

Metodo	URL	Descrizione
GET	`/clienti`	Lista clienti
POST	`/clienti/ricerca`	Ricerca clienti
GET	`/clienti/{codice}`	Lettura cliente
PUT	`/clienti/{codice}`	Modifica cliente
DELETE	`/clienti/{codice}`	Cancellazione cliente

Inserimento Cliente

Crea un nuovo cliente nell'anagrafica.

POST `/clienti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	body	Codice cliente (formato `NNN.NNNNN`, es. `501.00001`)
ragione_sociale	string	Sì	body	Ragione sociale
indirizzo	string	No	body	Indirizzo
cap	string	No	body	CAP
localita	string	No	body	Località
provincia	string	No	body	Provincia (sigla 2 caratteri)
partita_iva	string	No	body	Partita IVA (con prefisso nazione, es. `IT01234567890`)
codice_fiscale	string	No	body	Codice fiscale
cod_pagamento	integer	No	body	Codice condizione di pagamento (numerico)
telefono	string	No	body	Numero telefono
email	string	No	body	Indirizzo e-mail
pec	string	No	body	Indirizzo PEC
codice_sdi	string	No	body	Codice destinatario SDI
cod_listino	integer	No	body	Codice listino
cod_agente	string	No	body	Codice conto agente (formato `NNN.NNNNN`)

Esempio Request

```
POST /webapi/risorse/clienti HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "codice": "501.00100",
  "ragione_sociale": "Cliente Esempio Srl",
  "indirizzo": "Via Verdi 5",
  "cap": "00100",
  "localita": "ROMA",
  "provincia": "RM",
  "partita_iva": "IT01234567890",
  "cod_pagamento": 9
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/clienti/501.00100
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Cliente creato
400	Bad Request	Dati non validi
409	Conflict	Codice già esistente

Note Operative

- Richiede coordinate gestionale
- Versione minima: 81500

Lista Clienti

Restituisce l'elenco dei clienti dell'azienda.

`GET /clienti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/clienti?max=50&fields=codice,ragione_sociale,localita HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "501.00001",
      "ragione_sociale": "Cliente Esempio Srl",
      "localita": "ROMA"
    }
  ],
  "data_ric_elab": "20260616 171456",
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Note Operative

- Supporta paginazione e selezione campi
- Versione minima: 81500

Ricerca Clienti

Ricerca clienti con filtri avanzati.

`POST /clienti/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/clienti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "ragione_sociale",
      "condizione": "contiene",
      "valore": "Esempio"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "501.00001",
      "ragione_sociale": "Cliente Esempio Srl",
      "localita": "ROMA"
    }
  ],
  "data_ric_elab": "20260616 171503"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Versione minima: 81500

Lettura Cliente

Recupera i dati di uno specifico cliente.

`GET /clienti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice cliente (formato NNN.NNNNN)

Esempio Request

```
GET /webapi/risorse/clienti/501.00001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "codice": "501.00001",
  "ragione_sociale": "Cliente Esempio Srl",
  "indirizzo": "Via Verdi 5",
  "cap": "00100",
  "localita": "ROMA",
  "provincia": "RM",
  "partita_iva": "IT01234567890",
  "cod_pagamento": 9
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Cliente trovato
404	Not Found	Cliente non trovato

Note Operative

- Supporta encoding per codici con caratteri speciali (la chiave ha `encodable: true`)
- Versione minima: 81500

Modifica Cliente

Aggiorna i dati di un cliente esistente.

`PUT /clienti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice cliente (formato NNN.NNNNN)
ragione_sociale	string	No	body	Nuova ragione sociale
indirizzo	string	No	body	Nuovo indirizzo
localita	string	No	body	Nuova località

Campo	Tipo	Obbl.	Posizione	Descrizione
telefono	string	No	body	Nuovo telefono

Esempio Request

```
PUT /webapi/risorse/clienti/501.00001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "indirizzo": "Via Garibaldi 10",
  "localita": "MILANO",
  "provincia": "MI"
}
```

Esempio Response

```
{
  "codice": "501.00001",
  "ragione_sociale": "Cliente Esempio Srl",
  "indirizzo": "Via Garibaldi 10",
  "localita": "MILANO",
  "provincia": "MI"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Cliente aggiornato
400	Bad Request	Dati non validi
404	Not Found	Cliente non trovato

Note Operative

- Inviare solo i campi da modificare
- Versione minima: 81500

Cancellazione Cliente

Elimina un cliente dall'anagrafica.

`DELETE /clienti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice cliente (formato NNN.NNNNN)

Esempio Request

```
DELETE /webapi/risorse/clienti/501.00001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Cliente eliminato
404	Not Found	Cliente non trovato
409	Conflict	Cliente con documenti associati

Note Operative

- Potrebbe fallire se il cliente ha documenti associati
- Versione minima: 81500

Esempi

GET — Lista clienti con selezione campi

```
GET
/webapi/risorse/clienti?fields=codice,ragione_sociale,partita_iva,localita,provinc
ia&max=5 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Response (200 OK):

```
{
  "dati": [
    {
      "codice": "501.00001",
      "ragione_sociale": "Test è",
      "partita_iva": "IT00009876543",
      "localita": "CREMONA",
      "provincia": "CR"
    },
```

```
    {
      "codice": "501.00002",
      "ragione_sociale": "ORIZZONTE PEDALE SRL",
      "partita_iva": "IT01195030596",
      "localita": "RICCIONE",
      "provincia": "RN"
    },
    {
      "codice": "501.00003",
      "ragione_sociale": "HOTEL ADLON DI MONTANARI LUIGI E C. SAS",
      "partita_iva": "IT02275110407",
      "localita": "RICCIONE",
      "provincia": "RN"
    },
    {
      "codice": "501.00004",
      "ragione_sociale": "ARMANTI BICICLETTE SNC",
      "partita_iva": "IT00342630373",
      "localita": "BOLOGNA",
      "provincia": "BO"
    },
    {
      "codice": "501.00005",
      "ragione_sociale": "SELECT SRL",
      "partita_iva": "IT04139120374",
      "localita": "BOLOGNA",
      "provincia": "BO"
    }
  ],
  "data_ric_elab": "20260610 163244",
  "next":
"57415f33653636303563392d363464392d313166312d613966632d396436653165336262303264-5500000000000000"
}
```


Nota: Il campo `codice` per clienti e fornitori è nel formato `NNN.NNNNN` (es. `501.00001` per i clienti, `601.00001` per i fornitori). Il prefisso 501 identifica il mastro clienti, 601 il mastro fornitori.

Entità: Fornitori

Gestione completa dell'anagrafica fornitori. Supporta operazioni CRUD con paginazione e ricerca avanzata.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/fornitori`	Inserimento fornitore

Metodo	URL	Descrizione
GET	`/fornitori`	Lista fornitori
POST	`/fornitori/ricerca`	Ricerca fornitori
GET	`/fornitori/{codice}`	Lettura fornitore
PUT	`/fornitori/{codice}`	Modifica fornitore
DELETE	`/fornitori/{codice}`	Cancellazione fornitore

Inserimento Fornitore

Crea un nuovo fornitore nell'anagrafica.

POST `/fornitori`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	body	Codice fornitore (formato `NNN.NNNNN`, es. `601.00001`)
ragione_sociale	string	Sì	body	Ragione sociale
indirizzo	string	No	body	Indirizzo
cap	string	No	body	CAP
localita	string	No	body	Località
provincia	string	No	body	Provincia (sigla 2 caratteri)
partita_iva	string	No	body	Partita IVA (con prefisso nazione, es. `IT01234567890`)
codice_fiscale	string	No	body	Codice fiscale
cod_pagamento	integer	No	body	Codice condizione di pagamento (numerico)
telefono	string	No	body	Numero telefono
email	string	No	body	Indirizzo e-mail
pec	string	No	body	Indirizzo PEC
codice_sdi	string	No	body	Codice destinatario SDI
cod_agente	string	No	body	Codice conto agente (formato `NNN.NNNNN`)

Esempio Request

```
POST /webapi/risorse/fornitori HTTP/1.1
Host: server:porta
```

```
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "codice": "601.00100",
  "ragione_sociale": "Fornitore Materie Prime Srl",
  "indirizzo": "Via Industriale 22",
  "cap": "20100",
  "localita": "MILANO",
  "provincia": "MI",
  "partita_iva": "IT09876543210",
  "cod_pagamento": 9
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/fornitori/601.00100
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Fornitore creato
400	Bad Request	Dati non validi
409	Conflict	Codice già esistente

Note Operative

- Richiede coordinate gestionale
- Versione minima: 81500

Lista Fornitori

Restituisce l'elenco dei fornitori dell'azienda.

`GET /fornitori`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/fornitori?max=50&fields=codice,ragione_sociale,localita
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "601.00001",
      "ragione_sociale": "Fornitore Materie Prime Srl",
      "localita": "MILANO"
    }
  ],
  "data_ric_elab": "20260617 095049",
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Note Operative

- Supporta paginazione e selezione campi
- Versione minima: 81500

Ricerca Fornitori

Ricerca fornitori con filtri avanzati.

`POST /fornitori/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/fornitori/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "localita",
      "condizione": "=",
      "valore": "MILANO"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "601.00001",
      "ragione_sociale": "Fornitore Materie Prime Srl",
      "localita": "MILANO"
    }
  ],
  "data_ric_elab": "20260617 095056"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Versione minima: 81500

Lettura Fornitore

Recupera i dati di uno specifico fornitore.

`GET /fornitori/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice fornitore (formato NNN.NNNNN)

Esempio Request

```
GET /webapi/risorse/fornitori/601.00001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
{
  "codice": "601.00001",
  "ragione_sociale": "Fornitore Materie Prime Srl",
  "indirizzo": "Via Industriale 22",
  "cap": "20100",
  "localita": "MILANO",
  "provincia": "MI",
  "partita_iva": "IT09876543210",
  "cod_pagamento": 9
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Fornitore trovato
404	Not Found	Fornitore non trovato

Note Operative

- Supporta encoding per codici con caratteri speciali (la chiave ha `encodable: true`)
- Versione minima: 81500

Modifica Fornitore

Aggiorna i dati di un fornitore esistente.

`PUT /fornitori/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice fornitore (formato NNN.NNNNN)
ragione_sociale	string	No	body	Nuova ragione sociale
indirizzo	string	No	body	Nuovo indirizzo
localita	string	No	body	Nuova località

Campo	Tipo	Obbl.	Posizione	Descrizione
telefono	string	No	body	Nuovo telefono

Esempio Request

```
PUT /webapi/risorse/fornitori/601.00001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "indirizzo": "Via Nuova 30",
  "localita": "ROMA",
  "provincia": "RM"
}
```

Esempio Response

```
{
  "codice": "601.00001",
  "ragione_sociale": "Fornitore Materie Prime Srl",
  "indirizzo": "Via Nuova 30",
  "localita": "ROMA",
  "provincia": "RM"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Fornitore aggiornato
400	Bad Request	Dati non validi
404	Not Found	Fornitore non trovato

Note Operative

- Inviare solo i campi da modificare
- Versione minima: 81500

Cancellazione Fornitore

Elimina un fornitore dall'anagrafica.

`DELETE /fornitori/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice fornitore (formato NNN.NNNNN)

Esempio Request

```
DELETE /webapi/risorse/fornitori/601.00001 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Fornitore eliminato
404	Not Found	Fornitore non trovato
409	Conflict	Fornitore con documenti associati

Note Operative

- Potrebbe fallire se il fornitore ha documenti associati
- Versione minima: 81500

Esempi

GET — Lista fornitori con selezione campi

```
GET
/webapi/risorse/fornitori?fields=codice,ragione_sociale,partita_iva,localita,provi
ncia&max=5 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Response (200 OK):

```
{
  "dati": [
    {
      "codice": "601.00001",
      "ragione_sociale": "SHIMANO SRL",
      "partita_iva": "",
      "localita": "MILANO",
      "provincia": "MI"
    },
```

```
    {
      "codice": "601.00002",
      "ragione_sociale": "ALFIO MARTINI CONSULENZE",
      "partita_iva": "IT02279980409",
      "localita": "SANTARCANGELO",
      "provincia": "RN"
    },
    {
      "codice": "601.00003",
      "ragione_sociale": "SPECIALIZED BICYCLE COMPONENTS ITALIA SRL",
      "partita_iva": "IT10552220153",
      "localita": "MILANO",
      "provincia": "MI"
    },
    {
      "codice": "601.00004",
      "ragione_sociale": "XENTIS AUSTRIA LTD",
      "partita_iva": "ATU99999999",
      "localita": "Bärnbach",
      "provincia": ""
    },
    {
      "codice": "601.00005",
      "ragione_sociale": "SHOCKBIKE SRL",
      "partita_iva": "IT01314760933",
      "localita": "SPILIMBERGO",
      "provincia": "PN"
    }
  ],
  "data_ric_elab": "20260617 095049",
  "next":
"57415f34316432666430632d366132312d313166312d623439652d306637356532316661303739-
4d00000000000000"
}
```


Nota: Il campo `codice` per clienti e fornitori è nel formato `NNN.NNNNN` (es. `501.00001` per i clienti, `601.00001` per i fornitori). Il prefisso 501 identifica il mastro clienti, 601 il mastro fornitori.

Entità: Conti

Gestione del piano dei conti contabile. Supporta operazioni CRUD complete sui conti di contabilità generale.

Il codice conto utilizza il formato `GGG.NNNNN` dove `GGG` è il gruppo-mastro e `NNNNN` è il sottoconto (es. `101.00001`, `201.00001`).

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	/conti	Inserimento conto
GET	/conti	Lista conti
POST	/conti/ricerca	Ricerca conti
GET	/conti/{codice}	Lettura conto
PUT	/conti/{codice}	Modifica conto
DELETE	/conti/{codice}	Cancellazione conto

Inserimento Conto

Crea un nuovo conto nel piano dei conti.

`POST /conti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	body	Codice conto (formato: GGG.NNNNN)
ragione_sociale	string	Sì	body	Descrizione/ragione sociale del conto
gest_gior_cassa	string	No	body	Giornale di cassa (S/N)
cod_assogg_iva	string	No	body	Assoggettamento IVA
gest_clas_conto	string	No	body	Classe conto (banca/cassa)
gest_rateo_risc	string	No	body	Gestione rateo/risconto (S/N)
bil_ivdir_cee_d	string	No	body	Codice IV direttiva CEE dare
bil_ivdir_cee_a	string	No	body	Codice IV direttiva CEE avere

> ℹ **Nota:** La tabella riporta i campi principali. L'endpoint accetta tutti i campi restituiti dalla lettura (67 campi totali). Per l'elenco completo consultare la risposta di una GET su un conto esistente.

Esempio Request

```
POST /webapi/risorse/conti HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "codice": "201.00099",
  "ragione_sociale": "CASSA CONTANTI SEDE"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/conti/201.00099
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Conto creato
400	Bad Request	Dati non validi o coordinate mancanti
409	Conflict	Codice conto già esistente

Note Operative

- Richiede coordinate gestionale (Azienda e Anno)
- Il campo `codice` deve rispettare il formato `GGG.NNNNN` coerente con il piano dei conti dell'azienda
- Versione minima: 81800

Lista Conti

Restituisce l'elenco dei conti dell'azienda. Supporta paginazione e selezione campi.

`GET /conti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/conti?max=3&fields=codice,ragione_sociale,gest_clas_conto
HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "101.00001",
      "ragione_sociale": "SOCI C/SOTTOSCRIZIONE",
      "gest_clas_conto": ""
    },
    {
      "codice": "101.00002",
      "ragione_sociale": "SOCI C/SOTTOSCRIZIONI RICHIAMATE",
      "gest_clas_conto": ""
    },
    {
      "codice": "102.00001",
      "ragione_sociale": "COSTI DI IMPIANTO",
      "gest_clas_conto": ""
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Note Operative

- Senza il parametro `fields`, vengono restituiti tutti i 67 campi di ciascun conto
- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Versione minima: 81800

Ricerca Conti

Ricerca conti con filtri avanzati.

POST `/conti/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST
/webapi/risorse/conti/ricerca?max=5&fields=codice,ragione_sociale,gest_clas_conto
HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "ragione_sociale",
      "condizione": "contiene",
      "valore": "CASSA"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "201.00001",
      "ragione_sociale": "CASSA",
      "gest_clas_conto": ""
    },
    {
      "codice": "455.00001",
      "ragione_sociale": "DEB.vs CASSA NAZIONALE PREVIDENZA",
      "gest_clas_conto": ""
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Per la sintassi dei filtri vedi Ricerca e Filtri
- I campi filtrabili coincidono con i campi dell'entità (es. `ragione_sociale`, `codice`, `gest_clas_conto`)
- Versione minima: 81800

Lettura Conto

Recupera i dati di uno specifico conto.

`GET /conti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice conto (formato `GGG.NNNNN`)
fields	string	No	query	Campi da restituire

Esempio Request

```
GET
/webapi/risorse/conti/101.00001?fields=codice,ragione_sociale,gest_clas_conto,cod_
assogg_iva,bil_ivdir_cee_d HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "codice": "101.00001",
  "ragione_sociale": "SOCI C/SOTTOSCRIZIONE",
  "gest_clas_conto": "",
  "cod_assogg_iva": "",
  "bil_ivdir_cee_d": "AA"
}
```

> ℹ **Nota:** Senza il parametro `fields`, la response contiene tutti i 67 campi del conto, inclusi dati bancari, conti automatici per immobilizzazioni, classificazioni di bilancio e stato anagrafica.

Codici di Stato

Codice	Descrizione	Causa
200	OK	Conto trovato
404	Not Found	Conto non trovato

Note Operative

- La chiave `codice` è di tipo string encodable — se contiene caratteri speciali usare l'encoding (vedi Formato Request)
- Versione minima: 81800

Modifica Conto

Aggiorna i dati di un conto esistente.

`PUT /conti/{codice}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice conto
codice	string	Sì	body	Codice conto (deve coincidere con il path)
ragione_sociale	string	No	body	Nuova ragione sociale/descrizione
(altri campi)	—	No	body	Qualsiasi campo restituito dalla GET

> ⚠ **Nota Bene:** Il campo `codice` è obbligatorio anche nel body della PUT (`obbligatorio_put: true` nella struttura campi).

Esempio Request

```
PUT /webapi/risorse/conti/101.00003 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "codice": "101.00003",
  "ragione_sociale": "Conto modificato webapi REST"
}
```

Esempio Response

```
{
  "codice": "101.00003",
  "ragione_sociale": "Conto modificato webapi REST"
}
```

Codici di Stato

Codice	Descrizione	Causa

Codice	Descrizione	Causa
200	OK	Conto aggiornato
400	Bad Request	Dati non validi
404	Not Found	Conto non trovato

Note Operative

- Versione minima: 81800

Cancellazione Conto

Elimina un conto dal piano dei conti.

```
DELETE /conti/{codice}
```

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	string	Sì	path	Codice conto

Esempio Request

```
DELETE /webapi/risorse/conti/201.00099 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Conto eliminato
404	Not Found	Conto non trovato
409	Conflict	Conto utilizzato in registrazioni

Note Operative

- La cancellazione può fallire se il conto ha movimenti in prima nota
- Versione minima: 81800

Struttura Campi Completa

Di seguito l'elenco completo dei campi restituiti dall'endpoint, raggruppati per area funzionale.

Campi Principali

Campo	Tipo	Descrizione
codice	Alfanumerico	Codice conto (chiave, formato GGG.NNNNN)
cod_alternativo	Alfanumerico	Codice alternativo
ragione_sociale	Alfanumerico	Ragione sociale / descrizione del conto
nome_ricerca	Alfanumerico	Nome di ricerca / descrizione aggiuntiva

Gestione Contabile

Campo	Tipo	Descrizione
gest_gior_cassa	Alfanumerico	Giornale di cassa (S/N)
cod_assogg_iva	Alfanumerico	Assoggettamento IVA
perc_indetra	Numerico	Percentuale indetraibilità
gest_auto_trasp	Alfanumerico	Automezzo trasportatore (S/N)
tp_manutenzione	Alfanumerico	Tipo manutenzione
perc_manut	Numerico	Percentuale manutenzione
gest_castel_iva	Alfanumerico	Escludi da apertura automatica castelletto IVA (S/N)
gest_clas_conto	Alfanumerico	Classe conto (banca/cassa)
alq_comp	Numerico	Aliquota di compensazione
gest_rateo_risc	Alfanumerico	Gestione rateo/risconto (S/N)

Conti Collegati

Campo	Tipo	Descrizione
cod_costo_k	Alfanumerico	Conto di costo (per tipo K)
cod_effet_z	Alfanumerico	Conto effettivo (per tipo Z)
cod_immob_v	Alfanumerico	Conto immobilizzazione (per tipo V)

Classificazioni di Bilancio

Campo	Tipo	Descrizione
bil_34dir_ue_d	Alfanumerico	Codice 34 direttiva UE dare
bil_34dir_ue_a	Alfanumerico	Codice 34 direttiva UE avere
bil_ivdir_cee_d	Alfanumerico	Codice IV direttiva CEE dare
bil_ivdir_cee_a	Alfanumerico	Codice IV direttiva CEE avere
bil_xbrl_rsm_d	Alfanumerico	Codice XBRL RSM dare
bil_xbrl_rsm_a	Alfanumerico	Codice XBRL RSM avere

Conti Automatici (Immobilizzazioni — tipo I)

Campo	Tipo	Descrizione
cod_amm_odn	Alfanumerico	Conto ammortamento ordinario
cod_amm_ant	Alfanumerico	Conto ammortamento anticipato
cod_amm_no_ded	Alfanumerico	Conto ammortamento non deducibile
cod_fnd_odn	Alfanumerico	Conto fondo ordinario
cod_fnd_ant	Alfanumerico	Conto fondo anticipato
cod_fnd_no_ded	Alfanumerico	Conto fondo non deducibile
cod_plv	Alfanumerico	Conto plusvalenze
cod_plv_no_ded	Alfanumerico	Conto plusvalenze non tassabili
cod_mnv	Alfanumerico	Conto minusvalenze
cod_mnv_no_ded	Alfanumerico	Conto minusvalenze non deducibili

Dati Bancari

Campo	Tipo	Descrizione
id_banca	Numerico	ID banca
cod_abi	Numerico	Codice ABI
cod_cab	Numerico	Codice CAB
codice_cuc	Alfanumerico	Codice univoco CBI
codice_sia	Alfanumerico	Codice SIA
codice_soc_rid	Alfanumerico	Codice identificatore impresa RID
firma_societa	Alfanumerico	Firma azienda
conto_corren_b	Alfanumerico	Numero conto corrente
tp_conto_corren	Alfanumerico	Tipo conto corrente
cin_bban	Alfanumerico	Codice CIN su BBAN
intesta_iban_ban	Alfanumerico	Intestazione IBAN
dt_inizio_rapp	Alfanumerico	Data inizio rapporto
dt_fine_rapp	Alfanumerico	Data fine rapporto
fido_scoperto	Numerico	Fido scoperto c/c
fido_castel_eff	Numerico	Fido castelletto effetti
dt_tasso_attivo	Array[6]	Date tasso attivo
alq_tasso_att	Array[6]	Aliquote tasso attivo
dt_tasso_pas	Array[6]	Date tasso passivo
alq_tasso_pas	Array[6]	Aliquote tasso passivo
rif_signor	Alfanumerico	Riferimento Signor
rif_settore	Alfanumerico	Riferimento settore

Campo	Tipo	Descrizione
rif_telefono	Alfanumerico	Riferimento telefono
rif_note	Alfanumerico	Riferimento note

Spese Sanitarie

Campo	Tipo	Descrizione
cod_sp_sanit_rg	Alfanumerico	Tipo spesa raggruppamento
cod_sp_sanit	Alfanumerico	Tipo spesa aziendale
cod_part_sanit	Alfanumerico	Particolarità spesa aziendale

Stato Anagrafica e Metadati

Campo	Tipo	Descrizione
anag_dt_crea	Alfanumerico	Data creazione conto (formato AAAAMMGG)
anag_dt_mod	Alfanumerico	Data modifica conto (formato AAAAMMGG)
anag_mod	Alfanumerico	Conto modificato (S/N)
stato_anag_att	Array[12]	Stato anagrafica attiva (per mirroring)
stato_anag_var	Array[12]	Stato anagrafica variata (per mirroring)
conto_precanc	Alfanumerico	Conto precancellato (S/N)
utente_ult_mod	Alfanumerico	Utente ultima modifica
data_ult_mod	Data e ora	Data/ora ultima modifica (formato AAAAMMGG HHMMSS)
tp_merce_rsm	Numerico	Tipo merce aziende RSM

Entità: Dati Generali

Raccolta di tabelle e configurazioni generali dell'azienda. Include parametri installazione, utenti, valute, pagamenti, magazzini, listini, gruppi merceologici, e numerose altre tabelle di configurazione. Molte sotto-risorse sono di sola consultazione (GET + ricerca), mentre alcune supportano CRUD completo.

Riepilogo Endpoint

Configurazione Sistema (sola lettura, senza coordinate)

Metodo	URL	Descrizione
GET	`/dati-generali/installazione`	Lettura dati installazione
GET	`/dati-generali/utenti`	Lista utenti

Tabelle di Sola Lettura (con paginazione e/o ricerca)

Metodo	URL	Descrizione
GET	/dati-generali/parametri-aziendali	Lista parametri aziendali
GET	/dati-generali/valute	Lista valute
POST	/dati-generali/valute/ricerca	Ricerca valute
GET	/dati-generali/gruppi-mastri	Lista gruppi mastri
POST	/dati-generali/gruppi-mastri/ricerca	Ricerca gruppi mastri
GET	/dati-generali/serie-documenti	Lista serie documenti
POST	/dati-generali/serie-documenti/ricerca	Ricerca serie documenti
GET	/dati-generali/magazzini	Lista magazzini
POST	/dati-generali/magazzini/ricerca	Ricerca magazzini
GET	/dati-generali/sconti-listini	Lista sconti listini
POST	/dati-generali/sconti-listini/ricerca	Ricerca sconti listini
GET	/dati-generali/abbinamenti-colori	Lista abbinamenti colori
POST	/dati-generali/abbinamenti-	Ricerca abbinamenti colori

Metodo	URL	Descrizione
	colori/ricerca	
GET	/dati-generali/taglie	Lista taglie
POST	/dati-generali/taglie/ricerca	Ricerca taglie
GET	/dati-generali/pagamenti	Lista pagamenti
POST	/dati-generali/pagamenti/ricerca	Ricerca pagamenti
GET	/dati-generali/gruppi-merceologici	Lista gruppi merceologici
POST	/dati-generali/gruppi-merceologici/ricerca	Ricerca gruppi merceologici
GET	/dati-generali/esenzioni-iva	Lista esenzioni IVA
POST	/dati-generali/esenzioni-iva/ricerca	Ricerca esenzioni IVA
GET	/dati-generali/nature-articoli	Lista nature articoli
POST	/dati-generali/nature-articoli/ricerca	Ricerca nature articoli
GET	/dati-generali/strutture-articoli	Lista strutture articoli
POST	/dati-generali/strutture-articoli/ricerca	Ricerca strutture articoli
GET	/dati-generali/categ	Lista categorie statistiche articoli

Metodo	URL	Descrizione
	orie-statistiche-articoli	
POST	/dati-generali/categorie-statistiche-articoli/ricerca	Ricerca categorie statistiche articoli
GET	/dati-generali/causali-movimenti	Lista causali movimenti
POST	/dati-generali/causali-movimenti/ricerca	Ricerca causali movimenti
GET	/dati-generali/omaggi-abbuoni	Lista omaggi e abbuoni
POST	/dati-generali/omaggi-abbuoni/ricerca	Ricerca omaggi e abbuoni
GET	/dati-generali/listini	Lista listini
POST	/dati-generali/listini/ricerca	Ricerca listini
GET	/dati-generali/provvigioni-listini	Lista provvigioni listini
POST	/dati-generali/provvigioni-listini/ricerca	Ricerca provvigioni listini
GET	/dati-generali/sconti-quantita	Lista sconti quantità
POST	/dati-generali/sconti-quantita/ricer	Ricerca sconti quantità

Metodo	URL	Descrizione
	ca	
GET	/dati-generali/particolarita	Lista particolarità
POST	/dati-generali/particolarita/ricerca	Ricerca particolarità
GET	/dati-generali/imballi	Lista imballi
POST	/dati-generali/imballi/ricerca	Ricerca imballi
GET	/dati-generali/posizione-referenti	Lista posizioni referenti
GET	/dati-generali/cartella-abbinamenti	Lista cartelle abbinamenti
POST	/dati-generali/cartella-abbinamenti/ricerca	Ricerca cartelle abbinamenti
GET	/dati-generali/centri-costo-ricavo	Lista centri costo ricavo
POST	/dati-generali/centri-costo-ricavo/ricerca	Ricerca centri costo ricavo
GET	/dati-generali/classi-docuvision	Lista classi Docuvision
GET	/dati-generali/correlazione-unita-misura	Lista correlazioni unità misura
GET	/dati-generali/condizioni-generali-pagamenti	Lista condizioni pagamenti

Tabelle CRUD Complete

Metodo	URL	Descrizione
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-prezzi[/{codice}]	Categorie prezzi articoli
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-provvigioni[/{codice}]	Categorie provvigioni
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-sconti[/{codice}]	Categorie sconti cli/for
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-provvigioni-articoli[/{codice}]	Categorie provvigioni articoli
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-sconti-articoli[/{codice}]	Categorie sconti articoli
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-statistiche-cli-for[/{codice}]	Categorie statistiche cli/for
POST/GET/GET{id}/PUT/DELETE	/dati-generali/aspetto-esteriore-beni[/{codice}]	Aspetto esteriore beni
POST/GET/POST ricerca/GET{id}/PUT/DELETE	/dati-generali/fasi-lavorazione[/{codice}]	Fasi di lavorazione
POST/GET/POST ricerca/GET{tipo}/PUT/DELETE	/dati-generali/tipi-lotti-matricole[/{ti	Tipi lotti matricole

Metodo	URL	Descrizione
	po_lotto}]	
POST/GET/GET{id}/PUT/DELETE	/dati-generali/nature-articoli[/{codice}]	Nature articoli (CRUD)
POST/GET/GET{id}/PUT/DELETE	/dati-generali/categorie-statistiche-articoli[/{codice}]	Cat. statistiche articoli (CRUD)
POST/GET/GET{id}/PUT/DELETE	/dati-generali/zone-clienti-fornitori[/{codice}]	Zone clienti fornitori
POST/GET/GET{id}/PUT/DELETE	/dati-generali/lingue-straniere[/{codice}]	Lingue straniere
POST/GET/GET{id}/PUT/DELETE	/dati-generali/taglie-articoli[/{codice}]	Taglie articoli
POST/GET/GET{id}/PUT/DELETE	/dati-generali/sconti-quantita-articoli[/{codice}]	Sconti quantità articoli

Configurazione Sistema

Lettura Dati Installazione

Restituisce informazioni sull'installazione Passepartout (versione, tipo, configurazione).

`GET /dati-generali/installazione`

Parametri

Nessun parametro richiesto.

Esempio Request

```
GET /webapi/risorse/dati-generali/installazione HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Response

```
{
  "codice_prodotto": "2.9.17",
  "versione_prodotto": 89000,
  "scadenza_sap": "20260902",
  "id_installazione": 1924148000,
  "utenti_connessi": 1
}
```

Campi Response

Campo	Tipo	Descrizione
codice_prodotto	string	Versione del prodotto WebAPI (es. "2.9.17")
versione_prodotto	integer	Versione numerica del gestionale (es. 89000)
scadenza_sap	string	Data scadenza contratto SAP in formato YYYYMMDD
id_installazione	integer	Identificativo univoco dell'installazione
utenti_connessi	integer	Numero di utenti attualmente connessi

Note Operative

- NON richiede coordinate gestionale
- Versione minima: 81500

Lista Utenti

Restituisce l'elenco degli utenti configurati nell'installazione.

`GET /dati-generali/utenti`

Esempio Request

```
GET /webapi/risorse/dati-generali/utenti HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
```

Esempio Response

```
{
  "Utenti": [
    {
      "id": 626892910,
      "nome": "ADMIN",
      "nome_completo": "",
      "tipo": "Amministratore 1",
      "gruppo": "Amministratore 1"
```

```
    },
    {
      "id": 198187508,
      "nome": "webapi",
      "nome_completo": "",
      "tipo": "Utente",
      "gruppo": "WEBAPI"
    }
  ]
}
```

Campi Response

Campo	Tipo	Descrizione
`Utenti`	array	Array di oggetti utente (chiave root con iniziale maiuscola)
`Utenti[].id`	integer	Identificativo univoco dell'utente
`Utenti[].nome`	string	Nome utente (login)
`Utenti[].nome_completo`	string	Nome completo dell'utente (può essere vuoto)
`Utenti[].tipo`	string	Tipologia utente (es. `"Amministratore 1"`, `"Utente"`)
`Utenti[].gruppo`	string	Gruppo di appartenenza

> ℹ **Nota:** La chiave root della response è `"Utenti"` con iniziale maiuscola, a differenza della maggior parte degli altri endpoint che usano `"dati"`.

Note Operative

- NON richiede coordinate gestionale
- Versione minima: 81500

Tabelle di Sola Lettura

Le seguenti sotto-risorse espongono un pattern di consultazione: - `GET /dati-generali/{risorsa}` → lista (la maggior parte supporta paginazione: `max`, `next`, `fields`) - `POST /dati-generali/{risorsa}/ricerca` → ricerca con filtri nel body (dove disponibile)

Tutte richiedono coordinate gestionale.

> ℹ **Nota:** Non tutte le sotto-risorse supportano le stesse funzionalità. Consultare la colonna "Ricerca" e "Fields" nella tabella dettaglio per conoscere le capacità esatte di ciascuna.

Esempio Generico (applicabile a tutte le sotto-risorse)

```
GET /webapi/risorse/dati-generali/pagamenti?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

```
POST /webapi/risorse/dati-generali/pagamenti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "descrizione",
      "condizione": "contiene",
      "valore": "30 giorni"
    }
  ]
}
```

Dettaglio Sotto-Risorse

Sotto-risorsa	Versione Min.	Paginazione	Fields	Ricerca	Descrizione
parametri-aziendali	86000	Sì	Sì	No	Parametri configurazione azienda
valute	81500	Sì	Sì	Sì	Tabella valute
gruppi-mastri	81500	Sì	Sì	Sì	Gruppi e mastri del piano dei conti
serie-documenti	81500	Sì	Sì	Sì	Numerazioni documenti
magazzini	81500	Sì	Sì	Sì	Anagrafica magazzini
sconti-listini	81500	Sì	Sì	Sì	Sconti su listini
abbinamenti-colori	81500	Sì	Sì	Sì	Abbinamenti colori articoli
taglie	81600	Sì	Sì	Sì	Tabella taglie
pagamenti	81600	Sì	Sì	Sì	Condizioni di pagamento
gruppi-merceologici	81600	Sì	Sì	Sì	Gruppi merceologici articoli
esenzioni-iva	81600	Sì	Sì	Sì	Codici esenzione IVA
nature-articoli	81600	Sì	Sì	Sì	Nature articoli (lista e ricerca)
strutture-articoli	81600	Sì	Sì	Sì	Strutture

Sotto-risorsa	Versione Min.	Paginazione	Fields	Ricerca	Descrizione
					articoli
categorie-statistiche-articoli	81600	Sì	Sì	Sì	Categorie per statistiche
causali-movimenti	81600	Sì	Sì	Sì	Causali movimenti magazzino
omaggi-abbuoni	81600	Sì	Sì	Sì	Gestione omaggi e abbuoni
listini	81600	Sì	Sì	Sì	Listini prezzi
provvigioni-listini	81600	Sì	Sì	Sì	Provvigioni su listini
sconti-quantita	81600	Sì	Sì	Sì	Sconti per quantità
particolarita	81600	Sì	Sì	Sì	Particolarità prezzo
imballi	82400	Sì	Sì	Sì	Tipi imballo
posizione-referenti	84300	Sì	No	No	Posizioni/ruoli referenti
cartella-abbinamenti	84500	Sì	Sì	Sì	Cartelle abbinamento colori
centri-costo-ricavo	86500	Sì	Sì	Sì	Centri di costo e ricavo
classi-docuvision	87700	Sì	Sì	No	Classi documentali Docuvision
correlazione-unita-misura	87400	Sì	Sì	No	Correlazioni tra unità misura
condizioni-generali-pagamenti	84402	No	No	No	Condizioni generali pagamento (lista semplice)

Tabelle CRUD Complete

Le seguenti sotto-risorse supportano il ciclo completo CRUD con pattern uniforme: - `POST` `/dati-generali/{risorsa}` → inserimento - `GET` `/dati-generali/{risorsa}` → lista - `GET` `/dati-`

generali/{risorsa}/{codice} → lettura singola - PUT /dati-generali/{risorsa}/{codice} → modifica - DELETE /dati-generali/{risorsa}/{codice} → cancellazione

Alcune includono anche ricerca: POST /dati-generali/{risorsa}/ricerca

Esempio CRUD Generico

Inserimento

```
POST /webapi/risorse/dati-generali/categorie-prezzi HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "descrizione": "Categoria prezzi speciale"
}
```

Response: 201 Created con header Location

Lettura Singola

```
GET /webapi/risorse/dati-generali/categorie-prezzi/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Modifica

```
PUT /webapi/risorse/dati-generali/categorie-prezzi/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "descrizione": "Nuova descrizione"
}
```

Cancellazione

```
DELETE /webapi/risorse/dati-generali/categorie-prezzi/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Dettaglio Sotto-Risorse CRUD

Sotto-risorsa	Chiave	Tipo Chiave	Versione Min.	Ricerca
categorie-prezzi	codice	integer	84200 (GET), 88300 (CRUD)	No
categorie-provvigioni	codice	integer	84200 (GET), 88300 (CRUD)	No

Sotto-risorsa	Chiave	Tipo Chiave	Versione Min.	Ricerca
categorie-sconti	codice	integer	84200 (GET), 88300 (CRUD)	No
categorie-provvigioni-articoli	codice	integer	84402 (GET), 88300 (CRUD)	No
categorie-sconti-articoli	codice	integer	84402 (GET), 88300 (CRUD)	No
categorie-statistiche-cli-for	codice	integer	84402 (GET), 87300 (CRUD)	No
aspetto-esteriore-beni	codice	integer	81500 (GET), 88300 (CRUD)	No
fasi-lavorazione	codice	integer	83000 (GET), 88300 (CRUD)	Sì
tipi-lotti-matricole	tipo_lotto	string	83000 (GET), 87500 (CRUD)	Sì
nature-articoli	codice	string	81600 (GET/ricerca), 88300 (CRUD)	Sì (in lista)
categorie-statistiche-articoli	codice	string	81600 (GET/ricerca), 88300 (CRUD)	Sì (in lista)
zone-clienti-fornitori	codice	integer	82400 (GET), 88300 (CRUD)	No
lingue-straniere	codice	integer	87300	No
taglie-articoli	codice	integer	88300	No
sconti-quantita-articoli	codice	integer	88300	No

Codici di Stato (comuni a tutte le operazioni CRUD)

Codice	Descrizione	Causa
200	OK	Lettura/modifica/cancellazione riuscita
201	Created	Inserimento riuscito
400	Bad Request	Dati non validi o coordinate mancanti
404	Not Found	Risorsa non trovata
409	Conflict	Codice già esistente (inserimento) o risorsa in uso (cancellazione)

Note Operative Generali

- Le sotto-risorse di sola lettura (senza CRUD) non supportano POST, PUT, DELETE
- Le sotto-risorse senza ricerca (senza `/ricerca`) accettano solo GET per la lista
- Il tipo di chiave (`integer` vs `string`) determina il formato dell'URL
- Le chiavi di tipo `string` supportano encoding (base32/hex) per caratteri speciali
- L'endpoint `dati-generali/installazione` e `dati-generali/utenti` sono gli unici che NON richiedono coordinate gestionale
- Tutte le altre sotto-risorse richiedono coordinate gestionale nell'header

---

Esempi

Codifica Automatica

L'endpoint `dati-generali/tipi-lotti-matricole` ha una particolarità: la creazione può avvenire solo con **codifica automatica** specificando nel campo `tipo_lotto` la chiave `"**"`.

Esempio: Creazione Tipo Lotto con Campi Personalizzati

```
POST /webapi/risorse/dati-generali/tipi-lotti-matricole HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "tipo_lotto": "**",
  "stato": "C",
  "descrizione": "Test creazione",
  "chiave_univoca": "S",
  "auto_prel_tipo": "L",
  "auto_prel_qta": "S",
  "auto_prel_peso": "S",
  "auto_prel_spez": "S",
  "auto_prel_qres": "N",
  "tipo_ctrl_caric": "A",
  "auto_prel_gest": "E",
  "qta_insuf_scar": "D",
  "tipo_lott_matr": "M",
  "cod_tipo_lott": [
    [1, "**aa"],
    [2, "**bb"]
  ],
  "etichetta": [
    [1, "ANNO"],
    [2, "CONTATORE"]
  ],
  "tipo": [
```

```
    [1, "N"],
    [2, "I"]
  ],
  "dimensione": [
    [1, 4],
    [2, 7]
  ],
  "gest_obbl": [
    [1, "S"],
    [2, "S"]
  ],
  "riga_ord_visual": [
    [1, 1],
    [2, 2]
  ],
  "valore_iniziale": [
    [2, "1"]
  ],
  "nr_elem_cod_ute": [
    [2, 128]
  ],
  "data_obbl": "N",
  "giorni_autocod": 0,
  "giorni_manuale": 0,
  "giorni_soglia": 0
}
```


Descrizione Campi Principali

Campo	Tipo	Descrizione
`tipo_lotto`	string	Chiave (usare `"**"` per codifica automatica)
`tipo_lott_matr`	string	`"M"` = Matricola, `"L"` = Lotto
`chiave_univoca`	string	`"S"` = chiave univoca
`auto_prel_tipo`	string	`"L"` = LIFO/FIFO per prelievo automatico
`auto_prel_gest`	string	`"E"` = gestione prelievo
`qta_insuf_scar`	string	`"D"` = comportamento per quantità insufficiente
`tipo_ctrl_caric`	string	`"A"` = tipo controllo caricamento
`cod_tipo_lott`	array	Codici dei componenti del tipo lotto
`etichetta`	array	Etichette descrittive dei componenti
`tipo`	array	`"N"` = numerico, `"I"` = incrementale
`dimensione`	array	Dimensione in caratteri di ogni componente
`gest_obbl`	array	Gestione obbligatoria (`"S"`/`"N"`)

Nota: Gli altri metodi (PUT, GET, DELETE) seguono la classica logica REST con la chiave `tipo_lotto` nel path.

Entità: DBA (Distinte Base Articoli)

Consultazione delle distinte base automatiche degli articoli. Endpoint di sola lettura che restituisce le singole righe (componenti) delle distinte base definite sugli articoli, con dettaglio su condizioni, tipo scelta e gestione costi.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	`/dba`	Lista righe DBA articoli
POST	`/dba/ricerca`	Ricerca righe DBA articoli con filtri

Struttura Campi

Campo	Tipo	Descrizione
`codice`	Alfanumerico	Codice articolo della distinta automatica (chiave)
`nr_riga`	Numerico	Numero riga all'interno della distinta
`titolo`	Alfanumerico	Titolo della riga
`cod_articolo_cmp`	Alfanumerico	Codice articolo componente
`um`	Alfanumerico	Unità di misura del componente
`quantita`	Alfanumerico	Quantità del componente
`tipo_condizione`	Alfanumerico	Tipo condizione applicata
`cond_articolo`	Alfanumerico	Condizione articolo
`cond_cliente`	Alfanumerico	Condizione cliente
`tipo_scelta`	Alfanumerico	Tipo scelta (es. V = variante)
`part_prz_cmp`	Alfanumerico	Particolarità prezzi su componenti (es. S = sì)
`gest_cos_ini`	Alfanumerico	Costo di partenza (es. U = ultimo costo)

> ℹ **Nota:** Ogni record restituito rappresenta una singola riga/componente della distinta base. Un articolo con distinta base multi-componente produce più record con lo stesso `codice` e valori diversi di `nr_riga`.

Lista Righe DBA Articoli

Restituisce l'elenco delle righe delle distinte base automatiche degli articoli.

`GET /dba`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione (vedi Paginazione)
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/dba?max=3 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=XXX Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "codice": "BICCOMPLETAAUTO",
      "nr_riga": 1,
      "titolo": "",
      "cod_articolo_cmp": "TECAUD800000001",
      "um": "PZ",
      "quantita": "1",
      "tipo_condizione": "",
      "cond_articolo": "",
      "cond_cliente": "",
      "tipo_scelta": "V",
      "part_prz_cmp": "S",
      "gest_cos_ini": "U"
    },
    {
      "codice": "BICCOMPLETAAUTO",
      "nr_riga": 2,
      "titolo": "",
      "cod_articolo_cmp": "SELLSMPCARBON001",
      "um": "PZ",
      "quantita": "1",
      "tipo_condizione": "",
      "cond_articolo": "",
      "cond_cliente": "",
      "tipo_scelta": "V",
      "part_prz_cmp": "S",
      "gest_cos_ini": "U"
    }
  ],
```

```
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti o non valide

Note Operative

- Richiede coordinate gestionale (Azienda, Anno)
- Supporta paginazione tramite `max` e `next`
- Supporta selezione campi tramite `fields`
- Versione minima: 84300

Ricerca Righe DBA Articoli

Ricerca le righe delle distinte base automatiche con filtri.

`POST /dba/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)
filtri	array	Sì	body	Array di oggetti filtro (vedi Ricerca e Filtri)

Esempio Request

```
POST /webapi/risorse/dba/ricerca?max=3 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=2026

{
  "filtri": [
    {
      "campo": "codice",
      "condizione": "inizia_per",
      "valore": "BIC"
```

```
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "BICCOMPLETAAUTO",
      "nr_riga": 1,
      "titolo": "",
      "cod_articolo_cmp": "TECAUD800000001",
      "um": "PZ",
      "quantita": "1",
      "tipo_condizione": "",
      "cond_articolo": "",
      "cond_cliente": "",
      "tipo_scelta": "V",
      "part_prz_cmp": "S",
      "gest_cos_ini": "U"
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata con successo
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Supporta tutte le condizioni di filtro standard: `=`, `<`, `<=`, `>`, `>=`, `<>`, `contiene`, `inizia_per`
- Il campo `codice` è il più utile per filtrare per articolo specifico
- Versione minima: 84300

Entità: Distinte Base

Gestione delle distinte base di produzione. L'endpoint `/distinte-base` si suddivide in due sotto-risorse:

- **`/distinte-base/fasi`** — Gestione completa (CRUD) delle fasi di lavorazione associate a un articolo con distinta base
- **`/distinte-base/componenti`** — Consultazione (sola lettura) dei componenti associati alle fasi

Ogni fase contiene sia i dati di testata della distinta base del prodotto (DBP) sia i dati specifici della fase. I componenti sono restituiti con i campi di testata (articolo e fase) più tutti i campi del singolo componente.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	/distinte-base/fasi	Inserimento fase distinta base
GET	/distinte-base/fasi	Lista fasi distinte base
POST	/distinte-base/fasi/ricerca	Ricerca fasi con filtri
GET	/distinte-base/fasi/{codice}+{nr_fase}	Lettura singola fase
PUT	/distinte-base/fasi/{codice}+{nr_fase}	Modifica fase
DELETE	/distinte-base/fasi/{codice}+{nr_fase}	Cancellazione fase
GET	/distinte-base/componenti	Lista componenti distinte base
POST	/distinte-base/componenti/ricerca	Ricerca componenti con filtri

Struttura Campi — Fasi

I campi restituiti per ogni fase comprendono sia i dati di testata della distinta base (prefisso “DBP”) sia i dati specifici della fase.

Campi Testata Distinta Base (DBP)

Campo	Tipo	Descrizione
codice	Alfanumerico	Codice articolo con distinta base (**chiave**)
cod_art_costo	Alfanumerico	Articolo per calcolo costo
gest_ctrl_disp	Alfanumerico	Controllo esistenza/disponibilità (S/N)
codice_pf_padre	Alfanumerico	Prodotto finito padre di default
cod_tab_abbina	Alfanumerico	Tabella colori di aggancio
categoria_sml	Alfanumerico	Categoria semilavorato

Campo	Tipo	Descrizione
gest_dbvc_dbvo	Alfanumerico	Usa DBVC/DBVO (S/N)
gest_colleg_sml	Alfanumerico	Gestione collegamento PF→SML (S/N)
tp_um_usata	Alfanumerico	Tipo unità di misura utilizzato (P=prima, S=seconda)
cod_mag_mp	Numerico	Magazzino materie prime di default
nr_decimali_qta	Numerico	Numero di decimali quantità
cod_cliente	Alfanumerico	Codice cliente di default
nr_ore_appronta	Numerico con virgola	Ore di approntamento
nr_ore_lotto	Numerico con virgola	Ore per lotto
qta_lotto_prod	Numerico con virgola	Lotto di produzione
coeff_conv	Numerico con virgola	Coefficiente di conversione
consumo_taglia	Numerico con virgola	Consumi per taglia personali (array, 32 elementi)
gest_ril_tempi	Numerico con virgola	Gestione rilevamento tempi (array, 32 elementi)
gest_tipo_costo	Alfanumerico	Tipo costo industriale
cos_indutriale	Numerico con virgola	Valore costo industriale

Campi Fase

Campo	Tipo	Descrizione
nr_fase	Numerico	Numero fase (**chiave**)
descr_fase	Alfanumerico	Descrizione della fase
tp_cond_fase	Alfanumerico	Tipo condizione della fase
descr_agg_fase	Alfanumerico	Ulteriore descrizione della fase
qta_rif_fase	Alfanumerico	Quantità per riferimento
doc_stampa_fase	Alfanumerico	Documenti in stampa
note_fase	Alfanumerico	Gruppo di note della fase (array, 25 elementi)
wip_fase	Numerico	Minuti di ritardo/anticipo (positivi/negativi)
lag_fase	Alfanumerico	Tipo relazione (valori da ‘0’ a ‘3’)
gest_clav_fase	Alfanumerico	Conto lavoro (S/N)
cod_fornit_ctl	Alfanumerico	Conto lavoro — fornitore
cod_mag_mp_ctl	Numerico	Conto lavoro — magazzino
cod_ciclo	Numerico	Codice ciclo
utente_ult_mod	Alfanumerico	Utente ultima modifica
data_ult_mod	Data e ora	Data/ora ultima modifica (formato AAAAMMGG HHMMSS)

i Nota: Nella lettura singola di una fase (`GET` con chiave) vengono restituiti anche i campi `nr_tot_fasi` (numero totale fasi della distinta) e `nr_cmp_fase` (numero componenti nella fase), oltre ai dati dei componenti della fase stessa come array.

Struttura Campi — Componenti

I campi restituiti per ogni componente comprendono il riferimento alla testata (codice articolo, fase) e tutti i campi specifici del componente.

Campo	Tipo	Descrizione
codice	Alfanumerico	Codice articolo con distinta base (**chiave**)
nr_fase	Numerico	Numero fase (**chiave**)
descr_fase	Alfanumerico	Descrizione della fase
indice_cmp	Alfanumerico	Indice componente
codice_mp	Alfanumerico	Codice articolo componente (materia prima)
qta_utilizzo	Alfanumerico	Quantità di utilizzo
coeff_conv_mp	Numerico con virgola	Coefficiente di conversione componente
tp_evidenza	Alfanumerico	Evidenza o fase di disimpegno rapido
nr_unita_misura	Alfanumerico	Unità di misura utilizzata (‘1’=prima, ‘2’=seconda)
cod_mag_impegni	Numerico	Magazzino componente
tp_cond_art	Alfanumerico	Tipo condizione (‘A’ o ‘S’)
cod_art_cond	Alfanumerico	Condizione — codice articolo
cod_cli_cond	Alfanumerico	Codice cliente condizione
dt_inizio_cond	Alfanumerico	Data inizio condizione
dt_fine_cond	Alfanumerico	Data fine condizione
nega_cond_cli	Alfanumerico	Negazione condizione cliente (S/N)
nega_cond_data	Alfanumerico	Negazione condizione data (S/N)
nega_cond_tg	Alfanumerico	Negazione condizione taglie (S/N)
nega_cond_art	Alfanumerico	Negazione condizioni articolo (S/N)
condizione_tg	Alfanumerico	Condizione per taglia (array, 32 elementi)
cod_art_sost	Alfanumerico	Articoli sostitutivi (array, 4 elementi)
tp_sost_cmp	Alfanumerico	Tipo sostituzione componente
tp_dir_ricerca	Alfanumerico	Direzione ricerca (‘P’=precedente, ‘S’=successiva)
gest_escludi_dl	Alfanumerico	Escludi da deposito (S/N)
gest_escludi_sl	Alfanumerico	Escludi da scarico (S/N)
gest_esterna	Alfanumerico	Lavorazione esterna (S/N)
tp_costo_mp	Alfanumerico	Tipo di costo (‘U’, ‘S’, ‘M’, ‘V’ o ‘P’)
cos_lavorazione	Numerico con virgola	Costo lavorazione

Campo	Tipo	Descrizione
cod_fornitore	Alfanumerico	Codice fornitore

Inserimento Fase Distinta Base

Crea una nuova fase nella distinta base di un articolo.

`POST /distinte-base/fasi`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	Alfanumerico	Sì	body	Codice articolo con distinta base
nr_fase	Numerico	Sì	body	Numero fase
descr_fase	Alfanumerico	No	body	Descrizione della fase
cod_ciclo	Numerico	No	body	Codice ciclo
gest_clav_fase	Alfanumerico	No	body	Conto lavoro (S/N)
cod_fornit_ctl	Alfanumerico	No	body	Fornitore conto lavoro
descr_agg_fase	Alfanumerico	No	body	Descrizione aggiuntiva fase

> ℹ **Nota:** Tutti i campi della struttura fasi sono accettati nel body di inserimento. La tabella sopra riporta solo i più comuni.

Esempio Request

```
POST /webapi/risorse/distinte-base/fasi HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "codice": "PF",
  "nr_fase": 5,
  "descr_fase": "ASSEMBLAGGIO FINALE",
  "cod_ciclo": 1,
  "gest_clav_fase": "N"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/distinte-base/fasi/PF+5
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Fase creata con successo
400	Bad Request	Dati non validi o coordinate mancanti
409	Conflict	Fase già esistente per codice+nr_fase

Note Operative

- Richiede header `Coordinate-Gestionale`
- Versione minima: 81800
- La chiave della risorsa creata è `{codice}+{nr_fase}`

Lista Fasi Distinte Base

Restituisce l'elenco paginato di tutte le fasi delle distinte base.

`GET /distinte-base/fasi`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione (da response precedente)
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/distinte-
base/fasi?max=2&fields=codice,nr_fase,descr_fase,cod_ciclo HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "ATT001",
```

```
      "nr_fase": 1,
      "descr_fase": "BOTTIGLIA",
      "cod_ciclo": 1
    },
    {
      "codice": "BOTTIGLIA",
      "nr_fase": 1,
      "descr_fase": "FASE1",
      "cod_ciclo": 5
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate mancanti

Note Operative

- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Il parametro `fields` permette di selezionare solo i campi desiderati
- Senza `fields`, vengono restituiti tutti i campi (testata DBP + fase)
- Versione minima: 81800

Ricerca Fasi Distinte Base

Ricerca fasi con filtri avanzati.

`POST /distinte-base/fasi/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/distinte-base/fasi/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
```

```
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice",
      "condizione": "=",
      "valore": "PF"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "PF",
      "nr_fase": 1,
      "descr_fase": "FASE1",
      "descr_agg_fase": "FASE ESEGUITA SU MACCHINA",
      "cod_ciclo": 5,
      "gest_clav_fase": "N"
    },
    {
      "codice": "PF",
      "nr_fase": 2,
      "descr_fase": "FASE2",
      "descr_agg_fase": "",
      "cod_ciclo": 5,
      "gest_clav_fase": "N"
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Per la sintassi completa dei filtri, vedi Ricerca e Filtri
- Condizioni disponibili: `=`, `<`, `<=`, `>`, `>=`, `<>`, `contiene`, `inizia_per`
- Versione minima: 81800

Lettura Fase Distinta Base

Recupera i dati completi di una specifica fase, inclusi i componenti associati.

`GET /distinte-base/fasi/{codice}+{nr_fase}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	Alfanumerico	Sì	path	Codice articolo con distinta base
nr_fase	Numerico	Sì	path	Numero fase
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/distinte-base/fasi/PF+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "codice": "PF",
  "cod_art_costo": "PF",
  "gest_ctrl_disp": "S",
  "tp_um_usata": "P",
  "nr_decimali_qta": 3,
  "coeff_conv": 1.0,
  "descr_fase": "FASE1",
  "descr_agg_fase": "FASE ESEGUITA SU MACCHINA",
  "nr_fase": 1,
  "nr_tot_fasi": 4,
  "gest_clav_fase": "N",
  "nr_cmp_fase": 3,
  "cod_ciclo": 5,
  "codice_mp": [
    [1, "121212"],
    [2, "121212"],
    [3, "BOTTIGLIA"]
  ],
  "qta_utilizzo": [
    [1, "1"],
    [2, "2"],
    [3, "1"]
  ],
  "nr_unita_misura": [
    [1, "1"],
    [2, "1"],
    [3, "1"]
```

```
  ]
}
```

ℹ Nota: La response della lettura singola include i componenti della fase come array indicizzati (indice, valore). Il campo `nr_cmp_fase` indica il numero totale di componenti nella fase e `nr_tot_fasi` il numero totale di fasi della distinta.

Codici di Stato

Codice	Descrizione	Causa
200	OK	Fase trovata
400	Bad Request	Coordinate mancanti
404	Not Found	Fase non trovata

Note Operative

- Chiave composta: `codice_articolo` + `numero_fase` separati da +
- Se il codice articolo contiene caratteri speciali, usare encoding base32 o hex
- Versione minima: 81800

Modifica Fase Distinta Base

Aggiorna i dati di una fase esistente.

`PUT /distinte-base/fasi/{codice}+{nr_fase}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	Alfanumerico	Sì	path	Codice articolo con distinta base
nr_fase	Numerico	Sì	path	Numero fase
descr_fase	Alfanumerico	No	body	Nuova descrizione fase
descr_agg_fase	Alfanumerico	No	body	Nuova descrizione aggiuntiva
cod_ciclo	Numerico	No	body	Nuovo codice ciclo
gest_clav_fase	Alfanumerico	No	body	Conto lavoro (S/N)

ℹ Nota: Tutti i campi della struttura fasi sono modificabili nel body. La tabella sopra riporta i più comuni.

Esempio Request

```
PUT /webapi/risorse/distinte-base/fasi/PF+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "descr_fase": "ASSEMBLAGGIO E SALDATURA",
  "descr_agg_fase": "FASE ESEGUITA SU MACCHINA CNC"
}
```

Esempio Response

```
{
  "codice": "PF",
  "nr_fase": 1,
  "descr_fase": "ASSEMBLAGGIO E SALDATURA",
  "descr_agg_fase": "FASE ESEGUITA SU MACCHINA CNC",
  "cod_ciclo": 5,
  "gest_clav_fase": "N"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Fase aggiornata
400	Bad Request	Dati non validi
404	Not Found	Fase non trovata

Note Operative

- I campi `codice` e `nr_fase` sono obbligatori anche nel body (marcati `obbligatorio_put`)
- Versione minima: 81800

Cancellazione Fase Distinta Base

Elimina una fase dalla distinta base.

DELETE `/distinte-base/fasi/{codice}+{nr_fase}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	Alfanumerico	Sì	path	Codice articolo con distinta base
nr_fase	Numerico	Sì	path	Numero fase

Esempio Request

```
DELETE /webapi/risorse/distinte-base/fasi/PF+5 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Fase eliminata
404	Not Found	Fase non trovata

Note Operative

- La cancellazione elimina anche tutti i componenti associati alla fase
- Versione minima: 81800

Lista Componenti Distinte Base

Restituisce l'elenco paginato dei componenti delle distinte base. Ogni record corrisponde a un singolo componente di una fase.

`GET /distinte-base/componenti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/distinte-
base/componenti?max=2&fields=codice,nr_fase,codice_mp,qta_utilizzo,nr_unita_misura
HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "dati": [
```

```
    {
      "codice": "ATT001",
      "nr_fase": 1,
      "codice_mp": "PROVA",
      "qta_utilizzo": "1",
      "nr_unita_misura": "pz"
    },
    {
      "codice": "BOTTIGLIA",
      "nr_fase": 1,
      "codice_mp": "PLABS",
      "qta_utilizzo": "1",
      "nr_unita_misura": "kg"
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Note Operative

- Endpoint di sola lettura: non supporta POST/PUT/DELETE per singolo componente
- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Il campo `qta_utilizzo` è alfanumerico (può contenere valori con formule o espressioni)
- Versione minima: 81800

Ricerca Componenti Distinte Base

Ricerca componenti con filtri avanzati.

`POST /distinte-base/componenti/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/distinte-base/componenti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice",
      "condizione": "=",
      "valore": "BOTTIGLIA"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "BOTTIGLIA",
      "descr_fase": "FASE1",
      "nr_fase": 1,
      "codice_mp": "PLABS",
      "qta_utilizzo": "1",
      "nr_unita_misura": "kg",
      "gest_esterna": "N",
      "cos_lavorazione": 0.0
    },
    {
      "codice": "BOTTIGLIA",
      "descr_fase": "FASE1",
      "nr_fase": 1,
      "codice_mp": "PLPLA",
      "qta_utilizzo": "2",
      "nr_unita_misura": "kg",
      "gest_esterna": "N",
      "cos_lavorazione": 0.0
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Per la sintassi completa dei filtri, vedi Ricerca e Filtri
- Permette di trovare tutti i componenti di un articolo, di una fase specifica, o di un certo codice materia prima
- Versione minima: 81800

Entità: Documenti

Gestione dei documenti commerciali e di magazzino. Include ordini clienti, ordini fornitori, ordini matrici, preventivi, movimenti di magazzino e documenti di lavorazione (prodotti, bolle, impegni).

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	`/documenti/moduli-stampa/{sigla}`	Lettura modulo stampa
POST	`/documenti/ordini-clienti`	Inserimento ordine cliente
GET	`/documenti/ordini-clienti`	Lista ordini clienti
POST	`/documenti/ordini-clienti/ricerca`	Ricerca ordini clienti
GET	`/documenti/ordini-clienti/{sigla}+{serie}+{numero}`	Lettura ordine cliente
PUT	`/documenti/ordini-clienti/{sigla}+{serie}+{numero}`	Modifica ordine cliente
DELETE	`/documenti/ordini-clienti/{sigla}+{serie}+{numero}`	Cancellazione ordine cliente
GET	`/documenti/ordini-clienti/righe`	Lettura righe ordini clienti
POST	`/documenti/ordini-`	Ricerca righe ordini clienti

Metodo	URL	Descrizione
	clienti/righe/ricerca	
POST	/documenti/ordini-fornitori	Inserimento ordine fornitore
GET	/documenti/ordini-fornitori	Lista ordini fornitori
POST	/documenti/ordini-fornitori/ricerca	Ricerca ordini fornitori
GET	/documenti/ordini-fornitori/{sigla}+{serie}+{numero}	Lettura ordine fornitore
PUT	/documenti/ordini-fornitori/{sigla}+{serie}+{numero}	Modifica ordine fornitore
DELETE	/documenti/ordini-fornitori/{sigla}+{serie}+{numero}	Cancellazione ordine fornitore
GET	/documenti/ordini-fornitori/righe	Lettura righe ordini fornitori
POST	/documenti/ordini-fornitori/righe/ricerca	Ricerca righe ordini fornitori
POST	/documenti/ordini-matrici	Inserimento ordine matrice
GET	/documenti/ordini-matrici	Lista ordini matrici
POST	/documenti/ordini-matrici/ricerca	Ricerca ordini matrici
GET	/documenti/ordini-matrici/{sigla}+{serie}+{numero}	Lettura ordine matrice

Metodo	URL	Descrizione
PUT	/documenti/ordini-matrici/{sigla}+{serie}+{numero}	Modifica ordine matrice
DELETE	/documenti/ordini-matrici/{sigla}+{serie}+{numero}	Cancellazione ordine matrice
GET	/documenti/ordini-matrici/righe	Lettura righe ordini matrici
POST	/documenti/ordini-matrici/righe/ricerca	Ricerca righe ordini matrici
POST	/documenti/preventivi	Inserimento preventivo
GET	/documenti/preventivi	Lista preventivi
POST	/documenti/preventivi/ricerca	Ricerca preventivi
GET	/documenti/preventivi/{sigla}+{serie}+{numero}	Lettura preventivo
PUT	/documenti/preventivi/{sigla}+{serie}+{numero}	Modifica preventivo
DELETE	/documenti/preventivi/{sigla}+{serie}+{numero}	Cancellazione preventivo
GET	/documenti/preventivi/righe	Lettura righe preventivi
POST	/documenti/preventivi/righe/ricerca	Ricerca righe preventivi
POST	/documenti/movimenti-magazzino	Inserimento movimento di magazzino
GET	/documenti/movimenti-	Lista movimenti di magazzino

Metodo	URL	Descrizione
	magazzino	
POST	/documenti/movimenti-magazzino/ricerca	Ricerca movimenti di magazzino
GET	/documenti/movimenti-magazzino/{sigla}+{serie}+{numero}+{cod_conto}	Lettura movimento
PUT	/documenti/movimenti-magazzino/{sigla}+{serie}+{numero}+{cod_conto}	Modifica movimento
DELETE	/documenti/movimenti-magazzino/{sigla}+{serie}+{numero}+{cod_conto}	Cancellazione movimento
GET	/documenti/movimenti-magazzino/righe	Lettura righe movimenti
POST	/documenti/movimenti-magazzino/righe/ricerca	Ricerca righe movimenti
GET	/documenti/lavorazione/prodotti	Lista prodotti in lavorazione
POST	/documenti/lavorazione/prodotti/ricerca	Ricerca prodotti in lavorazione
POST	/documenti/lavorazione/bolle	Inserimento bolla di lavorazione
GET	/documenti/lavorazione/bolle	Lista bolle di lavorazione
POST	/documenti/lavorazione/bolle/ricerca	Ricerca bolle di lavorazione
GET	/documenti/lavorazione/bolle	Lettura bolla

Metodo	URL	Descrizione
	`/{codice}+{cod_sottobolla}`	
PUT	`/documenti/lavorazione/bolle/{codice}+{cod_sottobolla}`	Modifica bolla
DELETE	`/documenti/lavorazione/bolle/{codice}+{cod_sottobolla}`	Cancellazione bolla
	`/documenti/lavorazione/impegni`	Vedi Entità: Impegni Bolle di Lavorazione

Moduli Stampa

Lettura Modulo Stampa

Restituisce la struttura del modulo di stampa per una sigla documento.

`GET /documenti/moduli-stampa/{sigla}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
sigla	string	Sì	path	Sigla documento (es. OC, OF, FT)

Esempio Request

```
GET /webapi/risorse/documenti/moduli-stampa/OC HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Modulo restituito
404	Not Found	Sigla non valida

Note Operative

- Richiede coordinate gestionale
- Versione minima: 86500

Ordini Clienti

Pattern CRUD standard per documenti con chiave `{sigla}+{serie}+{numero}`. Sigle ammesse: OC, OX.

Inserimento Ordine Cliente

`POST /documenti/ordini-clienti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
sigla	string	Sì	body	Sigla documento (OC, OX)
serie	integer	Sì	body	Serie numerazione
numero	integer	Sì	body	Numero documento (0 = numerazione automatica)
cod_conto	string	Sì	body	Codice cliente (formato mastro.conto)
data_documento	string	No	body	Data documento (YYYYMMDD)
id_riga	array	Sì	body	Identificativo riga (0 = nuova riga)
tp_riga	array	Sì	body	Tipo riga (“R” = riga articolo)
codice_articolo	array	Sì	body	Codice articolo
quantita	array	Sì	body	Quantità
cod_iva	array	Sì	body	Codice aliquota IVA

> ℹ **Nota:** Le righe del documento sono rappresentate da variabili di tipo array con formato `[[indice, valore], ...]`. Per approfondimenti sulla struttura, vedi Struttura Testate e Righe.

Esempio Request

```
POST /webapi/risorse/documenti/ordini-clienti HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "sigla": "OC",
  "serie": 1,
  "numero": 0,
  "cod_conto": "501.00001",
  "data_documento": "20260616",
  "id_riga": [[1, 0]],
  "tp_riga": [[1, "R"]],
```

```
  "codice_articolo": [[1, "ATT001"]],
  "quantita": [[1, 2]],
  "cod_iva": [[1, "22"]]
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/documenti/ordini-clienti/OC+1+541
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Ordine creato
400	Bad Request	Dati non validi

Note Operative

- Il numero viene assegnato automaticamente dal gestionale
- Richiede coordinate gestionale
- Versione minima: 81500

Lista Ordini Clienti

`GET` `/documenti/ordini-clienti`

Restituisce l'elenco delle testate ordini clienti. Supporta `max`, `next`, `fields`.

Esempio Request

```
GET /webapi/risorse/documenti/ordini-clienti?max=20 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Ricerca Ordini Clienti

`POST` `/documenti/ordini-clienti/ricerca`

Ricerca con filtri nel body. Supporta `max`, `next`, `fields`.

Esempio Request

```
POST /webapi/risorse/documenti/ordini-clienti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
```

```
      "campo": "cod_conto",
      "condizione": "=",
      "valore": "C00001"
    }
  ]
}
```

Lettura Ordine Cliente

`GET /documenti/ordini-clienti/{sigla}+{serie}+{numero}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
sigla	string	Sì	path	Sigla (OC, OX)
serie	integer	Sì	path	Serie
numero	integer	Sì	path	Numero documento

Esempio Request

```
GET /webapi/risorse/documenti/ordini-clienti/OC+1+15 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Modifica Ordine Cliente

`PUT /documenti/ordini-clienti/{sigla}+{serie}+{numero}`

Aggiorna testata e/o righe dell'ordine.

Cancellazione Ordine Cliente

`DELETE /documenti/ordini-clienti/{sigla}+{serie}+{numero}`

Elimina l'ordine.

Lettura Righe Ordini Clienti

`GET /documenti/ordini-clienti/righe`

Vista trasversale di tutte le righe degli ordini clienti. Supporta `max`, `next`, `fields`.

Ricerca Righe Ordini Clienti

`POST /documenti/ordini-clienti/righe/ricerca`

Ricerca trasversale nelle righe con filtri.

Ordini Fornitori

Stessa struttura degli ordini clienti. Sigla ammessa: `OF`. Chiave: `{sigla}+{serie}+{numero}`.

Metodo	URL	Descrizione
POST	`/documenti/ordini-fornitori`	Inserimento
GET	`/documenti/ordini-fornitori`	Lista
POST	`/documenti/ordini-fornitori/ricerca`	Ricerca
GET	`/documenti/ordini-fornitori/{sigla}+{serie}+{numero}`	Lettura
PUT	`/documenti/ordini-fornitori/{sigla}+{serie}+{numero}`	Modifica
DELETE	`/documenti/ordini-fornitori/{sigla}+{serie}+{numero}`	Cancellazione
GET	`/documenti/ordini-fornitori/righe`	Lista righe
POST	`/documenti/ordini-fornitori/righe/ricerca`	Ricerca righe

Le operazioni seguono lo stesso pattern documentato per gli ordini clienti.

Ordini Matrici

Gestione ordini matrici (ordini multi-conferma). Sigle ammesse: `MA`, `MF`, `MX`. Chiave: `{sigla}+{serie}+{numero}`.

Metodo	URL	Descrizione
POST	`/documenti/ordini-matrici`	Inserimento
GET	`/documenti/ordini-matrici`	Lista
POST	`/documenti/ordini-matrici/ricerca`	Ricerca
GET	`/documenti/ordini-matrici/{sigla}+{serie}+{numero}`	Lettura
PUT	`/documenti/ordini-matrici/{sigla}+{serie}+{numero}`	Modifica
DELETE	`/documenti/ordini-matrici/{sigla}+{serie}+{numero}`	Cancellazione
GET	`/documenti/ordini-matrici/righe`	Lista righe
POST	`/documenti/ordini-matrici/righe/ricerca`	Ricerca righe

Le operazioni seguono lo stesso pattern degli ordini clienti.

Preventivi

Gestione preventivi. Sigle ammesse: `PC`, `PF`, `PR`, `PX`. Chiave: `{sigla}+{serie}+{numero}`.

Metodo	URL	Descrizione
POST	`/documenti/preventivi`	Inserimento

Metodo	URL	Descrizione
GET	`/documenti/preventivi`	Lista
POST	`/documenti/preventivi/ricerca`	Ricerca
GET	`/documenti/preventivi/{sigla}+{serie}+{numero}`	Lettura
PUT	`/documenti/preventivi/{sigla}+{serie}+{numero}`	Modifica
DELETE	`/documenti/preventivi/{sigla}+{serie}+{numero}`	Cancellazione
GET	`/documenti/preventivi/righe`	Lista righe
POST	`/documenti/preventivi/righe/ricerca`	Ricerca righe

Le operazioni seguono lo stesso pattern degli ordini clienti.

Movimenti di Magazzino

Gestione movimenti di magazzino (DDT, fatture, carichi, scarichi). Chiave: `{sigla}+{serie}+{numero}+{cod_conto}` (il 4° segmento è opzionale).

Inserimento Movimento di Magazzino

POST `/documenti/movimenti-magazzino`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
sigla	string	Sì	body	Sigla documento (2 caratteri maiuscoli)
serie	integer	Sì	body	Serie numerazione
numero	integer	Sì	body	Numero documento (0 = numerazione automatica)
cod_conto	string	No	body	Codice conto cliente/fornitore (formato

Campo	Tipo	Obbl.	Posizione	Descrizione
				mastro.conto)
data_documento	string	No	body	Data documento (YYYYMMDD)
id_riga	array	Sì	body	Identificativo riga (0 = nuova riga)
tp_riga	array	Sì	body	Tipo riga (“R” = riga articolo)
codice_articolo	array	Sì	body	Codice articolo
quantita	array	Sì	body	Quantità
cod_iva	array	Sì	body	Codice aliquota IVA

> ℹ **Nota:** Le righe del documento sono rappresentate da variabili di tipo array con formato `[[indice, valore], ...]`. Per approfondimenti sulla struttura, vedi Struttura Testate e Righe.

Esempio Request

```
POST /webapi/risorse/documenti/movimenti-magazzino HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "sigla": "BC",
  "serie": 1,
  "numero": 0,
  "cod_conto": "501.00001",
  "data_documento": "20260616",
  "id_riga": [[1, 0]],
  "tp_riga": [[1, "R"]],
  "codice_articolo": [[1, "ATT001"]],
  "quantita": [[1, 5]],
  "cod_iva": [[1, "22"]]
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/documenti/movimenti-magazzino/BC+1+12+501.00001
```

Lista/Ricerca/Lettura/Modifica/Cancellazione

Seguono lo stesso pattern documentato sopra.

`GET /documenti/movimenti-magazzino/{sigla}+{serie}+{numero}+{cod_conto}`

Il 4° segmento (cod_conto) è opzionale. Se omesso, cerca solo per sigla+serie+numero.

GET/POST `/documenti/movimenti-magazzino/righe[/ricerca]`

Vista trasversale delle righe dei movimenti.

Lavorazione - Prodotti

Consultazione dei prodotti in lavorazione (sola lettura).

Metodo	URL	Descrizione
GET	`/documenti/lavorazione/prodotti`	Lista prodotti in lavorazione
POST	`/documenti/lavorazione/prodotti/ricerca`	Ricerca prodotti in lavorazione

Esempio Request

```
GET /webapi/risorse/documenti/lavorazione/prodotti?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Lavorazione - Bolle

Gestione bolle di lavorazione. Chiave: {codice}+{cod_sottobolla}.

Inserimento Bolla di Lavorazione

POST `/documenti/lavorazione/bolle`

Esempio Request

```
POST /webapi/risorse/documenti/lavorazione/bolle HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cod_articolo_pf": "PF001",
  "quantita": 100,
  "data_inizio": "20240115"
}
```

Lettura Bolla

`GET /documenti/lavorazione/bolle/{codice}+{cod_sottobolla}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	integer	Sì	path	Codice bolla
cod_sottobolla	integer	Sì	path	Codice sottobolla

Modifica/Cancellazione Bolla

PUT/DELETE `/documenti/lavorazione/bolle/{codice}+{cod_sottobolla}`

Ricerca Bolle

`POST /documenti/lavorazione/bolle/ricerca`

Lavorazione - Impegni

Per la documentazione completa degli impegni bolle di lavorazione (CRUD, struttura campi, chiave composta a 8 segmenti, esempi), vedi Entità: Impegni Bolle di Lavorazione.

Note Operative Generali Documenti

- Tutti gli endpoint documenti richiedono coordinate gestionale
- La chiave per ordini e preventivi è sempre `{sigla}+{serie}+{numero}`
- I movimenti magazzino aggiungono un 4° segmento opzionale `{cod_conto}`
- Le bolle di lavorazione usano `{codice}+{cod_sottobolla}` (integer)
- Gli impegni lavorazione hanno la chiave più lunga (8 segmenti)
- Le sigle documento sono codificate nel gestionale e non modificabili via API
- Tutti gli endpoint di tipo `/righe` e `/righe/ricerca` sono viste trasversali
- Versione minima per documenti base: 81500
- Versione minima per lavorazione impegni: 87000

Uso Avanzato - Struttura Documenti e Operazioni Complesse

Struttura Testate e Righe

L'end-point dei documenti contiene entità complesse, composte generalmente da una o più testate ed *n* righe:

- La **lettura della lista** (GET su collection) ritorna solo le testate
- La **lettura delle righe** (GET su `/righe`) ritorna tutte le righe di quella famiglia, filtrabili poi via POST su `/righe/ricerca`
- La **lettura del singolo documento** (GET con chiave) ritorna il dettaglio completo di testate e righe

Solo referenziando il **singolo documento** è possibile fare modifiche tramite PUT e POST.

Rappresentazione delle Righe come Array

Per il singolo endpoint (es. `ordini-clienti/OC+1+1`) le righe del documento sono rappresentate da variabili di tipo array. Per ogni variabile, l'indice dell'array oltre a referenziare la riga i-esima ne rappresenta la sua **posizione a video** in Mexal/Passcom.

```
{
  "id_riga": [
    [1, 4],
    [2, 1]
  ]
}
```

In questo esempio: la prima posizione a video ha `id_riga=4`, la seconda posizione a video ha `id_riga=1`.

L'ordine degli elementi può anche essere scambiato, l'importante è che gli indici rimangano coerenti. Utilizzando lo stesso esempio potremmo referenziare le stesse righe cambiando l'ordine dei due elementi degli array ma mantenendo coerenza tra indice di riga e `id_riga`:

```
{
  "id_riga": [
    [2, 1],
    [1, 4]
  ]
}
```

In caso di creazione o modifica di un ordine è importante che gli indici di riga mantengano l'ordinamento corretto altrimenti si modifica l'ordine in cui compaiono le righe sul documento in Mexal/Passcom.

> ⚠ **Nota Bene:** Non utilizzare `id_riga` come ordinamento.

Revisione di un Documento (PUT)

In caso di revisione (PUT) **bisogna sempre indicare tutti gli identificativi di riga** (`id_riga`) ed il tipo di riga (`tp_riga`) anche se la riga non è da revisionare. Eventuali id omessi elimineranno la riga corrispondente dal documento.

Modifica solo testata (preservando le righe):

```
PUT /webapi/risorse/documenti/ordini-clienti/OC+1+1?solo_testata=true HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cod_conto": "501.00002"
}
```

Con `solo_testata=true` le righe vengono preservate anche se omesse nel body.

Gestione delle Taglie

Parametro Magazzino	Tipo variabile `qta_taglia`	Comportamento
“Gestione a taglie” = No	Scalare	Riporta sempre il valore `0`
“Gestione a taglie” = Sì, articolo senza taglie	Array vuoto	`"qta_taglia": []`
“Gestione a taglie” = Sì, articolo con taglie	Array valorizzato	`"qta_taglia": [[1, 5], [2, 3]]`

Copia Righe Documenti

È possibile duplicare una riga specificando solo il suo riferimento. La copia è possibile solo in: - **PUT** (revisione di un documento) - **POST con trasformazione** (`sigla_trasformazione=xx`)

Parametri per la copia: - `id_riga`: impostato a **0** (nuova riga) - `tp_riga`: tipo della riga da copiare - `idx_riga_copia`: indice della riga di origine da cui copiare

Esempio - Aggiunta riga copiata in revisione:

```
{
  "sigla": "OC",
  "serie": 1,
  "numero": 358,
  "cod_conto": "501.00001",
  "data_documento": "20221231",
  "id_riga": [
    [1, 1],
    [2, 2],
    [3, 0]
  ],
  "tp_riga": [
    [1, "R"],
    [2, "R"],
    [3, "R"]
  ],
  "idx_riga_copia": [
    [3, 1]
```

```
  ],
  "data_ult_mod": "20230227 145327"
}
```

In questo esempio la riga 3 (`id_riga=0`, nuova) viene copiata dalla riga con indice 1. Non è possibile fare una copia della copia, cioè impostare come `idx_riga_copia` un indice di una riga che è essa stessa copia di un'altra. La copia deve essere sempre riferita ad una riga pre-esistente nel documento.

> ⚠ **Nota Bene:** Nel caso di copia durante la trasformazione di un documento composto da più testate, la riga da copiare deve essere sempre inserita nel punto di rottura tra una testata e la successiva (o comunque tra le righe della stessa testata). Questo perché, per un vincolo tecnico, i riferimenti alle testate (`id_rif_testata`) devono essere progressivi. Ad esempio, con 2 testate ognuna con 2 righe, se si vuole inserire una riga copiandola da una riga della prima testata, non è possibile inserirla in ultima posizione ma deve essere inserita in prima, seconda o terza posizione: in questo modo devono essere aggiornati tutti gli indici degli array per i riferimenti alle righe.

Trasformazione Documenti

Per tutte le risorse sotto l'entità documenti è possibile specificare in query string in fase di creazione (POST) l'eventuale sigla documento in cui si vuole trasformare il documento senza doversi preoccupare di gestire la trasformazione "a mano" eliminando il documento di origine o gestendo i residui delle righe. Eventuali moduli di stampa dovranno essere specificati nel campo dedicato `cod_modulo`.

```
POST /webapi/risorse/documenti/movimenti-magazzino?sigla_trasformazione=FT
HTTP/1.1
```

> Se nel body è specificata una sigla differente, viene ignorata: la query string `sigla_trasformazione` è prioritaria.

Campi Obbligatori per la Trasformazione

Per i movimenti di magazzino bisogna indicare sia i riferimenti al documento di origine che agli ordini:

Campo	Descrizione
`sigla_doc_orig`	Sigla del documento di origine
`serie_doc_orig`	Serie del documento di origine
`numero_doc_orig`	Numero del documento di origine
`data_doc_orig`	Data del documento di origine
`sigla_ordine`	Sigla dell'ordine (solo per movimenti-magazzino)
`serie_ordine`	Serie dell'ordine
`numero_ordine`	Numero dell'ordine
`data_ordine`	Data dell'ordine
`dt_ult_mod_orig`	Data ultima modifica dei documenti di origine (metadati da GET)

Per trasformazioni su endpoint di ordini, preventivi o matrici i riferimenti agli ordini possono essere omessi.

Gestione dei Residui

Per le trasformazioni di documenti, se si vogliono gestire anche i residui di riga, è necessario valorizzare le seguenti variabili di riga (array) a seconda che la riga sia gestita a quantità, taglia, lotto, colli:

Campo	Descrizione
`quantita_trs`	Quantità / Peso Lordo trasformato
`nr_colli_trs`	Numero colli trasformato
`qta_taglia_trs`	Quantità per taglia trasformato
`qta_lotto_trs`	Lotti - Quantità trasformato
`nr_colli_lo_trs`	Lotti - Colli trasformato
`qta_tg_lo_trs`	Lotti - Quantità per taglia trasformato
`ubi_su_res`	S/N - se l'ubicazione deve essere impostata anche sul residuo (il default dipende dal documento di origine: "S" se era già presente una ubicazione, "N" se non era presente)

Le quantità a residuo devono essere intese (come in emissione/revisione documenti) come le quantità che verranno **evase** e la differenza andrà a residuo. Ad esempio se la prima riga ha quantità 10 e viene valorizzata `"quantita_trs": [[1, 3]]`, si intende che della prima riga vengono evase 3 quantità e 7 rimangono a residuo nel documento originale.

Dovranno sempre essere indicati tutti gli indici delle righe che si vogliono trasformare e, nel caso di gestione dei residui, le righe a residuo devono essere lasciate con lo stato originale (tipicamente `"S"`) mentre quelle che devono essere totalmente evase devono essere impostate in evadibili `"E"`:

- `tipo_stato_riga = "E"` → riga totalmente evasa (non rimane nel documento di origine)
- `tipo_stato_riga = "S"` → riga con residuo (rimane aperta nel documento di origine)

Esempio: Trasformazione da 2 bolle (BC) a fattura (FT)

```
POST /webapi/risorse/documenti/movimenti-magazzino?sigla_trasformazione=FT
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "sigla": "FT",
  "serie": 1,
  "numero": 0,
  "cod_conto": "501.00002",
  "data_documento": "20220523",
  "id_riga": [
    [1, 1],
```

```
    [2, 1]
  ],
  "tp_riga": [
    [1, "R"],
    [2, "R"]
  ],
  "codice_articolo": [
    [1, "ATT001"],
    [2, "ATT003"]
  ],
  "quantita": [
    [1, 3],
    [2, 4]
  ],
  "cod_iva": [
    [1, "23"],
    [2, "23"]
  ],
  "sigla_doc_orig": [
    [1, "BC"],
    [2, "BC"]
  ],
  "serie_doc_orig": [
    [1, 1],
    [2, 1]
  ],
  "numero_doc_orig": [
    [1, 2],
    [2, 12]
  ],
  "data_doc_orig": [
    [1, "20220520"],
    [2, "20220523"]
  ],
  "id_rif_testata": [
    [1, 1],
    [2, 2]
  ],
  "dt_ult_mod_orig": [
    [1, "20230101 070555"],
    [2, "20230102 080512"]
  ]
}
```

Esempio: Trasformazione OF → BF con Residuo

```
POST /webapi/risorse/documenti/movimenti-magazzino?sigla_trasformazione=BF
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

```
{
  "sigla": "BF",
  "serie": 1,
  "numero": 14,
  "data_documento": "20220809",
  "cod_conto": "601.00001",
  "id_riga": [
    [1, 1],
    [2, 1]
  ],
  "tp_riga": [
    [1, "R"],
    [2, "R"]
  ],
  "tipo_stato_riga": [
    [1, "S"],
    [2, "S"]
  ],
  "quantita": [
    [1, 20],
    [2, 30]
  ],
  "quantita_trs": [
    [1, 5],
    [2, 6]
  ],
  "cod_iva": [
    [1, "22"],
    [2, "22"]
  ],
  "sigla_doc_orig": [
    [1, "OF"],
    [2, "OF"]
  ],
  "serie_doc_orig": [
    [1, 1],
    [2, 1]
  ],
  "numero_doc_orig": [
    [1, 337],
    [2, 401]
  ],
  "data_doc_orig": [
    [1, "20220809"],
    [2, "20220809"]
  ],
  "id_rif_testata": [
    [1, 1],
    [2, 2]
  ],
  "sigla_ordine": [
```

```
    [1, "OF"],
    [2, "OF"]
  ],
  "serie_ordine": [
    [1, 1],
    [2, 1]
  ],
  "numero_ordine": [
    [1, 337],
    [2, 401]
  ],
  "data_ordine": [
    [1, "20220523"],
    [2, "20220825"]
  ],
  "dt_ult_mod_orig": [
    [1, "20230101 070555"],
    [2, "20230102 080512"]
  ]
}
```

In questo esempio: OF con quantità 20 e 30 → BF con 5 e 6, residuo di 15 e 24 negli OF di origine.

⚠ **Nota Bene:** Il numero per le BF è obbligatorio, non può essere automatico (0).

Esempio: Trasformazione OF → BF con Residuo e Gestione dei Lotti

Può esistere la circostanza per cui un documento in origine non abbia lotti (si pensi ad un ordine fatto al fornitore) e poi solo in fase di trasformazione in bolla di carico vengano assegnati uno o più lotti.

Nell'esempio che segue si trasforma in BF partendo da due ordini al fornitore:

- **OF337**: ordine con articolo senza lotti
- **OF401**: ordine con articolo gestito a lotti

Nella BF risultante per la riga riferita all'ordine senza lotti verrà gestito un residuo assegnando anche i lotti, mentre per la riga riferita all'ordine OF401 (con lotti) viene gestito un residuo riferito ai lotti già presenti.

```
POST /webapi/risorse/documenti/movimenti-magazzino?sigla_trasformazione=BF
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "sigla": "BF",
  "serie": 1,
  "numero": 13,
  "data_documento": "20220809",
  "cod_conto": "601.00001",
```

```
  "id_riga": [
    [1, 1],
    [2, 1]
  ],
  "tp_riga": [
    [1, "R"],
    [2, "R"]
  ],
  "tipo_stato_riga": [
    [1, "S"],
    [2, "S"]
  ],
  "quantita": [
    [1, 20],
    [2, 30]
  ],
  "quantita_trs": [
    [1, 5],
    [2, 6]
  ],
  "qta_lotto_trs": [
    [4, 2],
    [5, 2],
    [6, 2]
  ],
  "pos_righe_lotto": [
    [1, 1],
    [2, 4]
  ],
  "nr_righe_lotto": [
    [1, 3],
    [2, 3]
  ],
  "sigla_doc_orig": [
    [1, "OF"],
    [2, "OF"]
  ],
  "serie_doc_orig": [
    [1, 1],
    [2, 1]
  ],
  "numero_doc_orig": [
    [1, 337],
    [2, 401]
  ],
  "data_doc_orig": [
    [1, "20220809"],
    [2, "20220809"]
  ],
  "id_rif_testata": [
    [1, 1],
```

```
    [2, 2]
  ],
  "sigla_ordine": [
    [1, "OF"],
    [2, "OF"]
  ],
  "serie_ordine": [
    [1, 1],
    [2, 1]
  ],
  "numero_ordine": [
    [1, 337],
    [2, 401]
  ],
  "data_ordine": [
    [1, "20220523"],
    [2, "20220825"]
  ],
  "id_lotto": [
    [1, 175],
    [2, 174],
    [3, 173],
    [4, 173],
    [5, 174],
    [6, 175]
  ],
  "qta_lotto": [
    [1, 1],
    [2, 2],
    [3, 2],
    [4, 15],
    [5, 5],
    [6, 10]
  ],
  "dt_ult_mod_orig": [
    [1, "20230101 070555"],
    [2, "20230102 080512"]
  ]
}
```


Spiegazione dei campi lotto:

- `pos_righe_lotto`: per ogni riga indica a quale posizione dell'array dei lotti (`id_lotto`) iniziano i propri lotti. Per la riga 1 i lotti partono dall'indice 1, per la riga 2 dall'indice 4.
- `nr_righe_lotto`: per ogni riga indica quanti lotti gestisce (in questo caso 3 lotti per riga).
- `qta_lotto_trs`: per ogni lotto già esistente (riga 2, indici 4-5-6) indica le quantità per la gestione del residuo.
- `id_lotto`: array di tutti i lotti presenti nel documento.

- `qta_lotto`: quantità dei lotti. Per la riga di origine che **non aveva lotti** (riga 1, indici 1-2-3) le quantità sono quelle totali che andranno a finire in bolla (la somma deve essere uguale a `quantita_trs`, cioè 1+2+2=5). Per la riga che **già aveva i lotti** (riga 2, indici 4-5-6) si deve indicare il totale dei lotti (la somma deve essere uguale al campo `quantita`, cioè 15+5+10=30).

  ℹ **Nota:** Se la riga di origine non aveva lotti non si deve specificare nessun residuo in `qta_lotto_trs` ma si deve specificare direttamente la quantità totale del lotto nel campo `qta_lotto`.

Trasformazione con Ubicazione

Nel caso di trasformazione con residuo si voglia specificare anche una ubicazione (es. ordine al fornitore di nuovi articoli che poi verranno ubicati solo in fase di ricezione merce), questa di default non viene impostata sul residuo.

Nel caso si voglia impostare l'id ubicazione anche nel documento di origine, oltre a valorizzare l'id nel campo `id_ubicazione`, è possibile specificare nel campo `ubi_su_res` se l'ubicazione impostata deve essere riportata anche sul residuo oppure deve essere impostata esclusivamente sul documento trasformato:

- `"ubi_su_res": "S"` → l'ubicazione viene riportata anche sul residuo
- `"ubi_su_res": "N"` → l'ubicazione resta solo sul documento trasformato (default)

Il campo `ubi_su_res` non è un dato presente in archivio, ma è un parametro di servizio specifico nel caso di trasformazione documenti.

Esempio: Trasformazione con Prelievo Merce da Ubicazioni Diverse

Il caso è quello di un ordine cliente (OC) in cui non è specificata nessuna ubicazione per l'articolo e si vuole prelevare parte della merce da una ubicazione e parte da un'altra.

Nell'esempio che segue si fa riferimento ad un OC con una riga gestita su 3 lotti differenti: 30 pezzi divisi in 15, 5 e 10 pezzi per ogni lotto. Non è specificata nessuna ubicazione. Si vuole trasformare il documento in una BC in cui si evadono parzialmente solo le quantità del primo lotto prelevando 3pz dall'ubicazione 2704 e 5pz dall'ubicazione 2708.

Creazione ordine OC 1/401 con lotti:

```
{
  "sigla": "OC",
  "serie": 1,
  "numero": 401,
  "data_documento": "20220825",
  "cod_conto": "501.00001",
  "id_riga": [[1, 1]],
  "tp_riga": [[1, "R"]],
  "codice_articolo": [[1, "RUOCARB005"]],
  "quantita": [[1, 30]],
  "pos_righe_lotto": [[1, 1]],
  "nr_righe_lotto": [[1, 3]],
  "id_lotto": [[1, 173], [2, 174], [3, 175]],
```

```
  "qta_lotto": [[1, 15], [2, 5], [3, 10]]
}
```

In questo caso, dato che l'ubicazione è una proprietà in rapporto 1:1 con la riga, se si vogliono assegnare più ubicazioni alla stessa merce è necessario che nel documento di destinazione si aggiunga una riga in più. Questa riga non può essere gestita dall'algoritmo di trasformazione che si basa esclusivamente sulla riga originale del documento OC. Deve quindi essere una riga **copiata** dalla riga 1, completamente evasa (in modo che non risulti nell'OC originale), e dato che la sua quantità viene presa dal lotto 173, deve essere aggiustata la quantità totale in modo che la trasformazione utilizzi dati coerenti.

Trasformazione OC → BC con ubicazioni diverse:

```
POST /webapi/risorse/documenti/movimenti-magazzino?sigla_trasformazione=BC
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "sigla": "BC",
  "serie": 1,
  "numero": 0,
  "data_documento": "20220809",
  "cod_conto": "501.00001",
  "id_riga": [
    [1, 1],
    [2, 0]
  ],
  "tp_riga": [
    [1, "R"],
    [2, "R"]
  ],
  "idx_riga_copia": [
    [2, 1]
  ],
  "tipo_stato_riga": [
    [1, "S"],
    [2, "E"]
  ],
  "quantita": [
    [1, 25],
    [2, 5]
  ],
  "quantita_trs": [
    [1, 3]
  ],
  "qta_lotto_trs": [
    [1, 3]
  ],
```

```
  "pos_righe_lotto": [
    [1, 1],
    [2, 4]
  ],
  "nr_righe_lotto": [
    [1, 3],
    [2, 1]
  ],
  "sigla_doc_orig": [
    [1, "OC"]
  ],
  "serie_doc_orig": [
    [1, 1]
  ],
  "numero_doc_orig": [
    [1, 401]
  ],
  "data_doc_orig": [
    [1, "20220809"]
  ],
  "id_riga_orig": [
    [1, 1],
    [2, 1]
  ],
  "id_rif_testata": [
    [1, 1]
  ],
  "sigla_ordine": [
    [1, "OC"]
  ],
  "serie_ordine": [
    [1, 1]
  ],
  "numero_ordine": [
    [1, 401]
  ],
  "data_ordine": [
    [1, "20220523"]
  ],
  "id_ubicazione": [
    [1, 2704],
    [2, 2708]
  ],
  "ubi_su_res": [
    [1, "N"],
    [2, "N"]
  ],
  "id_lotto": [
    [1, 173],
    [2, 174],
    [3, 175],
```

```
    [4, 173]
  ],
  "qta_lotto": [
    [1, 10],
    [2, 5],
    [3, 10],
    [4, 5]
  ],
  "dt_ult_mod_orig": [
    [1, "20230101 070555"]
  ]
}
```

Spiegazione:

- **Riga 1** (originale): stato `"S"` perché deve essere gestito un residuo. La `quantita` è abbassata a 25 (30-5) perché 5pz vengono "spostati" sulla riga copiata. La `quantita_trs` indica che vengono evase solo 3 quantità dall'ubicazione 2704.
- **Riga 2** (copiata con `idx_riga_copia`): stato `"E"` perché deve essere completamente evasa e non deve rimanere traccia sul documento di origine. La `quantita` è 5 (quella "spostata" dalla riga 1).
- `id_riga_orig`: l'indice di riga di origine è il medesimo per entrambe le righe (entrambe derivano dalla riga 1 dell'OC).
- `id_ubicazione`: per ogni riga si specifica l'ubicazione di prelievo merce (3pz da 2704, 5pz da 2708).
- `ubi_su_res`: impostato a `"N"` perché non si vuole che l'ubicazione sia cambiata sul documento di origine.
- `qta_lotto`: la quantità totale dei lotti deve essere coerente con la quantità totale. Dato che 5pz vengono "spostati" sul lotto 4 (173) a partire dal lotto 1 (173), la quantità del lotto 1 viene abbassata (15-5=10). Il lotto 4 è riferito alla riga aggiunta e preleva dal lotto 173.
- `id_rif_testata`: entrambe le righe sono riferite alla stessa testata. Il riferimento alla testata della riga copiata si può anche omettere perché la copia prenderà lo stesso riferimento di testata della riga origine.

⚠ **Nota Bene:** Le righe copiate devono essere ordinate per testata di appartenenza altrimenti verrà dato un errore in fase di esecuzione della procedura.

Entità: Impegni Bolle di Lavorazione

Gestione degli impegni materiali associati alle bolle di lavorazione. Ogni impegno rappresenta una materia prima (articolo componente) richiesta per una specifica fase di lavorazione di un prodotto finito. Supporta CRUD completo con chiave composta a 8 segmenti.

⚠ **Nota Bene:** L'endpoint corretto è `/documenti/lavorazione/impegni`. L'endpoint legacy `/impegni` (versione 81800) è deprecato e non va più utilizzato per nuove integrazioni.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/documenti/lavorazione/impegni`	Inserimento impegno
GET	`/documenti/lavorazione/impegni`	Lista impegni
POST	`/documenti/lavorazione/impegni/ricerca`	Ricerca impegni
GET	`/documenti/lavorazione/impegni/{chiave}`	Lettura impegno
PUT	`/documenti/lavorazione/impegni/{chiave}`	Modifica impegno
DELETE	`/documenti/lavorazione/impegni/{chiave}`	Cancellazione impegno

Struttura Campi

La tabella seguente elenca tutti i campi restituiti dall'endpoint.

Campo	Tipo	Chiave	Descrizione
codice	Numerico	Sì	Numero documento (bolla di lavorazione)
cod_sottobolla	Numerico	Sì	Numero sottodocumento
nr_rif_pf	Numerico	Sì	Numero riferimento prodotto finito: >0 = numero, -1 = residui per depositi, -2 = residui per scarichi
nr_fase	Numerico	Sì	Numero fase prodotto finito
cod_magazzino	Numerico	Sì	Magazzino materie prime
cod_articolo	Alfanumerico	Sì	Codice articolo materia prima
id_ubicazione	Numerico	Sì	Ubicazione
progressivo	Numerico	Sì	Progressivo inserimento

Campo	Tipo	Chiave	Descrizione
dt_sca_riga_bl	Alfanumerico	No	Data scadenza riga bolla
tab_taglie	Numerico	No	Misura: 1-99 taglie; 101 tipo articolo unità misura (um1); 102 tipo articolo unità misura (um2)
nr_colli	Numerico	No	Numero colli o quantità variabile
quantita	Numerico con virgola	No	Quantità o quantità fissa
quantita_taglia	Array[32] numerico	No	Quantità per taglia (32 posizioni)
prezzo	Numerico con virgola	No	Prezzo
tp_evidenza	Alfanumerico	No	Evidenza: `' '`, `'E'`, `'L'`, `'T'`
indice	Alfanumerico	No	Indice
gest_depositato	Alfanumerico	No	Depositato (S/N)
gest_escludi_dl	Alfanumerico	No	Escludi dai depositi (S/N)
gest_escludi_sl	Alfanumerico	No	Escludi dagli scarichi (S/N)
gest_costo_dbp	Alfanumerico	No	Gestione costo DBP (S/N)
gest_prz_bloc	Alfanumerico	No	Gestione prezzo bloccato (S/N)
tp_articolo	Alfanumerico	No	Tipologia articolo
cod_ciclo_fase	Numerico	No	Codice ciclo fase
gest_ril_lavora	Alfanumerico	No	Lavorazione da rilevare (S/N)
gest_lav_rileva	Alfanumerico	No	Lavorazione rilevata (S/N)
pos_in_dbp	Numerico	No	Posizione in distinta base di produzione
utente_ult_mod	Alfanumerico	No	Utente ultima modifica
data_ult_mod	Data e ora	No	Data ora ultima modifica (formato `AAAAMMGG HHMMSS`)

Campi aggiuntivi restituiti nella lettura singola (GET con chiave):

Campo	Tipo	Descrizione
pos_righe_lotto	Numerico	Posizione righe lotto
nr_righe_lotto	Numerico	Numero righe lotto
id_lotto	Array numerico	ID lotti associati
nr_decimali_qta	Array numerico	Numero decimali quantità per lotto
qta_lotto	Array numerico	Quantità per lotto
nr_colli_lotto	Array numerico	Numero colli per lotto

Campo	Tipo	Descrizione
qta_tg_lotto	Array numerico	Quantità taglie per lotto

Inserimento Impegno

Crea un nuovo impegno associato a una bolla di lavorazione.

`POST /documenti/lavorazione/impegni`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	integer	Sì	body	Numero documento (bolla)
cod_sottobolla	integer	Sì	body	Numero sottodocumento
nr_rif_pf	integer	Sì	body	Numero riferimento prodotto finito
nr_fase	integer	Sì	body	Numero fase
cod_magazzino	integer	Sì	body	Magazzino materie prime
cod_articolo	string	Sì	body	Codice articolo materia prima
id_ubicazione	integer	Sì	body	ID ubicazione
quantita	number	No	body	Quantità impegnata
prezzo	number	No	body	Prezzo
tab_taglie	integer	No	body	Codice misura/taglia
nr_colli	integer	No	body	Numero colli

Esempio Request

```
POST /webapi/risorse/documenti/lavorazione/impegni HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=YYYY

{
  "codice": 50,
  "cod_sottobolla": 0,
  "nr_rif_pf": 1,
  "nr_fase": 1,
  "cod_magazzino": 1,
  "cod_articolo": "121212",
  "id_ubicazione": 0,
  "quantita": 3.0
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/documenti/lavorazione/impegni/50+0+1+1+1+121212+0+0
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Impegno creato con successo
400	Bad Request	Dati non validi o coordinate mancanti
409	Conflict	Impegno con la stessa chiave già esistente

Note Operative

- Chiave composta da 8 segmenti separati da +: `codice+cod_sottobolla+nr_rif_pf+nr_fase+cod_magazzino+cod_articolo+id_ubicazione+progressivo`
- Il `progressivo` viene assegnato automaticamente dal sistema e restituito nel header `Location`
- Richiede header `Coordinate-Gestionale` con Azienda e Anno
- Il campo `cod_articolo` supporta encoding (base32/hex) se contiene caratteri speciali
- Versione minima: 87000

Lista Impegni

Restituisce l'elenco paginato degli impegni di tutte le bolle di lavorazione.

`GET /documenti/lavorazione/impegni`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione per la pagina successiva
fields	string	No	query	Elenco campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/documenti/lavorazione/impegni?max=3 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=XXX Anno=YYYY
```

Esempio Response


```
{
  "dati": [
    {
      "codice": 50,
      "cod_sottobolla": 0,
      "nr_rif_pf": 1,
      "nr_fase": 1,
      "cod_magazzino": 1,
      "cod_articolo": "121212",
      "id_ubicazione": 0,
      "progressivo": 0,
      "dt_sca_riga_bl": "",
      "tab_taglie": 101,
      "nr_colli": 0,
      "quantita": 3.0,
      "quantita_taglia": [],
      "prezzo": 0.0,
      "tp_evidenza": "",
      "indice": "",
      "gest_depositato": "N",
      "gest_escludi_dl": "N",
      "gest_escludi_sl": "N",
      "gest_costo_dbp": "N",
      "gest_prz_bloc": "N",
      "tp_articolo": "A",
      "cod_ciclo_fase": 0,
      "gest_ril_lavora": "N",
      "gest_lav_rileva": "N",
      "pos_in_dbp": 1,
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20250514 092311"
    }
  ],
  "data_ric_elab": "20260617 104054",
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti

Note Operative

- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Il campo `data_ric_elab` indica la data/ora di elaborazione della richiesta
- Supporta selezione campi con parametro `fields`

- Versione minima: 87000

Ricerca Impegni

Ricerca impegni con filtri strutturati.

`POST /documenti/lavorazione/impegni/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Ogni oggetto filtro ha la struttura descritta in Ricerca e Filtri.

Esempio Request

```
POST /webapi/risorse/documenti/lavorazione/impegni/ricerca?max=3 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=YYYY

{
  "filtri": [
    {
      "campo": "codice",
      "condizione": "=",
      "valore": 51
    }
  ]
}
```

> ⚠ **Nota Bene:** I campi numerici (come `codice`, `cod_sottobolla`, `nr_fase`, ecc.) richiedono valori di tipo numerico intero nel filtro, non stringhe. Passare `"51"` invece di `51` genera un errore di tipo.

Esempio Response

```
{
  "dati": [
    {
      "codice": 51,
      "cod_sottobolla": 0,
      "nr_rif_pf": 1,
      "nr_fase": 1,
```

```
      "cod_magazzino": 1,
      "cod_articolo": "121212",
      "id_ubicazione": 0,
      "progressivo": 1,
      "dt_sca_riga_bl": "",
      "tab_taglie": 102,
      "nr_colli": 0,
      "quantita": 99.0,
      "quantita_taglia": [],
      "prezzo": 7.0,
      "tp_evidenza": "",
      "indice": "",
      "gest_depositato": "N",
      "gest_escludi_dl": "N",
      "gest_escludi_sl": "N",
      "gest_costo_dbp": "N",
      "gest_prz_bloc": "N",
      "tp_articolo": "A",
      "cod_ciclo_fase": 0,
      "gest_ril_lavora": "N",
      "gest_lav_rileva": "N",
      "pos_in_dbp": 2,
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20241112 104855"
    },
    {
      "codice": 51,
      "cod_sottobolla": 0,
      "nr_rif_pf": 1,
      "nr_fase": 2,
      "cod_magazzino": 1,
      "cod_articolo": "2.53.03.905",
      "id_ubicazione": 1,
      "progressivo": 0,
      "dt_sca_riga_bl": "",
      "tab_taglie": 101,
      "nr_colli": 0,
      "quantita": 1.0,
      "quantita_taglia": [],
      "prezzo": 1250.0,
      "tp_evidenza": "",
      "indice": "",
      "gest_depositato": "N",
      "gest_escludi_dl": "N",
      "gest_escludi_sl": "N",
      "gest_costo_dbp": "N",
      "gest_prz_bloc": "N",
      "tp_articolo": "A",
      "cod_ciclo_fase": 3,
      "gest_ril_lavora": "N",
      "gest_lav_rileva": "N",
```

```
      "pos_in_dbp": 1,
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20241003 154406"
    }
  ],
  "data_ric_elab": "20260617 104103"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o tipo valore errato

Note Operative

- Supporta tutte le condizioni standard: =, <, <=, >, >=, <>, `contiene`, `inizia_per`
- Supporta paginazione e selezione campi
- Versione minima: 87000

Lettura Impegno

Recupera un singolo impegno tramite chiave composta a 8 segmenti.

GET
`/documenti/lavorazione/impegni/{codice}+{cod_sottobolla}+{nr_rif_pf}+{nr_fase}+{cod_magazzino}+{cod_articolo}+{id_ubicazione}+{progressivo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	integer	Sì	path	Numero documento (bolla)
cod_sottobolla	integer	Sì	path	Numero sottodocumento
nr_rif_pf	integer	Sì	path	Numero riferimento prodotto finito
nr_fase	integer	Sì	path	Numero fase
cod_magazzino	integer	Sì	path	Magazzino materie prime
cod_articolo	string	Sì	path	Codice articolo materia prima
id_ubicazione	integer	Sì	path	ID ubicazione
progressivo	integer	Sì	path	Progressivo inserimento
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/documenti/lavorazione/impegni/50+0+1+1+1+121212+0+0 HTTP/1.1
Host: server:porta
```

```
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=XXX Anno=YYYY
```

Esempio Response


```
{
  "codice": 50,
  "cod_sottobolla": 0,
  "nr_rif_pf": 1,
  "nr_fase": 1,
  "cod_magazzino": 1,
  "cod_articolo": "121212",
  "id_ubicazione": 0,
  "progressivo": 0,
  "dt_sca_riga_bl": "",
  "tab_taglie": 101,
  "nr_colli": 0,
  "quantita": 3.0,
  "quantita_taglia": [],
  "prezzo": 0.0,
  "tp_evidenza": "",
  "indice": "",
  "gest_depositato": "N",
  "gest_escludi_dl": "N",
  "gest_escludi_sl": "N",
  "gest_costo_dbp": "N",
  "gest_prz_bloc": "N",
  "tp_articolo": "A",
  "cod_ciclo_fase": 0,
  "gest_ril_lavora": "N",
  "gest_lav_rileva": "N",
  "pos_in_dbp": 1,
  "pos_righe_lotto": 0,
  "nr_righe_lotto": 0,
  "id_lotto": [],
  "nr_decimali_qta": [],
  "qta_lotto": [],
  "nr_colli_lotto": [],
  "qta_tg_lotto": [],
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20250514 092311"
}
```


> ℹ **Nota:** La lettura singola restituisce campi aggiuntivi relativi ai lotti (`pos_righe_lotto`, `nr_righe_lotto`, `id_lotto`, `nr_decimali_qta`, `qta_lotto`, `nr_colli_lotto`, `qta_tg_lotto`) non presenti nella lista.

Codici di Stato

Codice	Descrizione	Causa
200	OK	Impegno trovato

Codice	Descrizione	Causa
404	Not Found	Impegno non esistente con la chiave specificata

Note Operative

- Chiave composta da 8 segmenti separati da +
- Il campo `cod_articolo` supporta encoding (base32/hex) se contiene caratteri come / o \
- Supporta selezione campi con parametro `fields`
- Versione minima: 87000

Modifica Impegno

Aggiorna un impegno esistente identificato dalla chiave composta.

PUT
`/documenti/lavorazione/impegni/{codice}+{cod_sottobolla}+{nr_rif_pf}+{nr_fase}+{cod_magazzino}+{cod_articolo}+{id_ubicazione}+{progressivo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	integer	Sì	path + body	Numero documento (bolla)
cod_sottobolla	integer	Sì	path + body	Numero sottodocumento
nr_rif_pf	integer	Sì	path + body	Numero riferimento prodotto finito
nr_fase	integer	Sì	path + body	Numero fase
cod_magazzino	integer	Sì	path + body	Magazzino materie prime
cod_articolo	string	Sì	path + body	Codice articolo materia prima
id_ubicazione	integer	Sì	path + body	ID ubicazione
progressivo	integer	Sì	path	Progressivo inserimento
quantita	number	No	body	Nuova quantità
prezzo	number	No	body	Nuovo prezzo
nr_colli	integer	No	body	Nuovo numero colli

> ⚠ **Nota Bene:** Tutti i campi chiave (tranne `progressivo`) sono marcati come `obbligatorio_put` e devono essere ripetuti nel body della richiesta.

Esempio Request

```
PUT /webapi/risorse/documenti/lavorazione/impegni/50+0+1+1+1+121212+0+0 HTTP/1.1
Host: server:porta
```

```
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=YYYY

{
  "codice": 50,
  "cod_sottobolla": 0,
  "nr_rif_pf": 1,
  "nr_fase": 1,
  "cod_magazzino": 1,
  "cod_articolo": "121212",
  "id_ubicazione": 0,
  "quantita": 8.0
}
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Impegno aggiornato con successo
400	Bad Request	Dati non validi o campi chiave mancanti nel body
404	Not Found	Impegno non trovato

Note Operative

- Versione minima: 87000

Cancellazione Impegno

Elimina un impegno identificato dalla chiave composta.

DELETE
`/documenti/lavorazione/impegni/{codice}+{cod_sottobolla}+{nr_rif_pf}+{nr_fase}+{cod_magazzino}+{cod_articolo}+{id_ubicazione}+{progressivo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice	integer	Sì	path	Numero documento (bolla)
cod_sottobolla	integer	Sì	path	Numero sottodocumento
nr_rif_pf	integer	Sì	path	Numero riferimento prodotto finito
nr_fase	integer	Sì	path	Numero fase
cod_magazzino	integer	Sì	path	Magazzino materie prime
cod_articolo	string	Sì	path	Codice articolo materia prima

Campo	Tipo	Obbl.	Posizione	Descrizione
id_ubicazione	integer	Sì	path	ID ubicazione
progressivo	integer	Sì	path	Progressivo inserimento

Esempio Request

```
DELETE /webapi/risorse/documenti/lavorazione/impegni/50+0+1+1+1+121212+0+0
HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=XXX Anno=YYYY
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Impegno eliminato con successo
404	Not Found	Impegno non trovato

Note Operative

- Versione minima: 87000

Entità: Indirizzi Spedizione

Gestione degli indirizzi di spedizione alternativi per clienti e fornitori. Supporta inserimento, consultazione, ricerca e modifica. La cancellazione (DELETE) non è supportata.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/indirizzi-spedizione`	Inserimento indirizzo di spedizione
GET	`/indirizzi-spedizione`	Lista indirizzi di spedizione
POST	`/indirizzi-spedizione/ricerca`	Ricerca indirizzi di spedizione
GET	`/indirizzi-spedizione/{id}`	Lettura indirizzo di spedizione
PUT	`/indirizzi-spedizione/{id`	Modifica indirizzo di spedizione

Metodo	URL	Descrizione
	}	

Struttura Campi

Campo	Tipo	Descrizione
id	Alfanumerico	Codice indirizzo di spedizione (chiave, obbligatorio in PUT)
cod_conto	Alfanumerico	Codice Cliente / Fornitore associato (formato mastro.sottoconto)
descrizione	Alfanumerico	Descrizione / ragione sociale della destinazione
indirizzo	Alfanumerico	Indirizzo di spedizione
cap	Alfanumerico	C.A.P.
localita	Alfanumerico	Località (città)
provincia	Alfanumerico	Provincia (sigla)
tp_nazionalita	Alfanumerico	Nazionalità (es. `I` = Italia)
cod_paese	Alfanumerico	Codice paese (es. `IT`)
telefono	Alfanumerico	Numero di telefono
fax	Alfanumerico	Numero di Fax
email	Alfanumerico	Indirizzo e-Mail
url	Alfanumerico	Indirizzo Internet
chiusura_sett	Alfanumerico	Chiusura settimanale o stringa libera
riferimento	Alfanumerico	Riferimento (persona di contatto)
cell_rif	Alfanumerico	Numero di cellulare riferimento
nota	Alfanumerico	Note
data_validita	Alfanumerico	Data validità
data_creazione	Alfanumerico	Data creazione (formato AAAAMMGG)
data_modifica	Alfanumerico	Data aggiornamento (formato AAAAMMGG)
cod_conto_sost	Alfanumerico	Codice Cliente / Fornitore sostitutivo
id_ind_sost	Alfanumerico	Codice indirizzo di spedizione sostitutivo
annullato	Alfanumerico	Annullato (`S`/`N`)
gest_raggruppa	Alfanumerico	Raggruppa documenti (`S`/`N`)
codice_cig	Alfanumerico	Codice CIG
codice_cup	Alfanumerico	Codice CUP
rif_esterno_tp	Alfanumerico	Riferimenti esterni — tipo
rif_esterno_sgl	Alfanumerico	Riferimenti esterni — sigla

Campo	Tipo	Descrizione
rif_esterno_nr	Alfanumerico	Riferimenti esterni — numero
rif_esterno_cod	Alfanumerico	Riferimenti esterni — codice
rif_esterno_dt	Alfanumerico	Riferimenti esterni — data
tp_indirizzo	Alfanumerico	Tipo indirizzo
utente_ult_mod	Alfanumerico	Utente ultima modifica
data_ult_mod	Data e ora	Data/ora ultima modifica (formato AAAAMMGG HHMMSS)

Inserimento Indirizzo di Spedizione

Crea un nuovo indirizzo di spedizione.

`POST /indirizzi-spedizione`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
cod_conto	Alfanumerico	No	body	Codice cliente/fornitore (formato mastro.sottoconto)
descrizione	Alfanumerico	No	body	Descrizione / ragione sociale destinazione
indirizzo	Alfanumerico	No	body	Indirizzo di spedizione
cap	Alfanumerico	No	body	C.A.P.
localita	Alfanumerico	No	body	Località (città)
provincia	Alfanumerico	No	body	Provincia (sigla)
tp_nazionalita	Alfanumerico	No	body	Nazionalità (es. `I`)
cod_paese	Alfanumerico	No	body	Codice paese (es. `IT`)
telefono	Alfanumerico	No	body	Numero di telefono
email	Alfanumerico	No	body	Indirizzo e-Mail
riferimento	Alfanumeri	No	body	Persona di contatto

Campo	Tipo	Obbl.	Posizione	Descrizione
	co			
cell_rif	Alfanumerico	No	body	Cellulare riferimento
nota	Alfanumerico	No	body	Note

Nota: La tabella riporta i campi principali. Tutti i campi della struttura (vedi Struttura Campi) sono accettati in fase di inserimento.

Esempio Request

```
POST /webapi/risorse/indirizzi-spedizione HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "cod_conto": "501.00001",
  "descrizione": "Sede operativa Milano",
  "indirizzo": "Via Roma 15",
  "cap": "20100",
  "localita": "MILANO",
  "provincia": "MI",
  "telefono": "02-12345678",
  "email": "magazzino@cliente.it",
  "riferimento": "Mario Rossi",
  "cell_rif": "333-1234567",
  "nota": "Consegna piano terra"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/indirizzi-spedizione/1
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Indirizzo creato con successo
400	Bad Request	Dati non validi o coordinate mancanti

Note Operative

- Richiede coordinate gestionale (Azienda e Anno)
- L'`id` viene assegnato automaticamente dal sistema
- Il campo `cod_conto` identifica il cliente/fornitore in formato mastro.sottoconto (es. `501.00001`)

- Versione minima: 81600

Lista Indirizzi di Spedizione

Restituisce tutti gli indirizzi di spedizione. Supporta paginazione e selezione campi.

`GET /indirizzi-spedizione`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione (da response precedente)
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/indirizzi-spedizione?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "id": "1",
      "cod_conto": "501.00001",
      "descrizione": "Sede operativa Milano",
      "indirizzo": "Via Roma 15",
      "cap": "20100",
      "localita": "MILANO",
      "provincia": "MI",
      "tp_nazionalita": "I",
      "cod_paese": "",
      "telefono": "02-12345678",
      "fax": "02-9876543",
      "email": "magazzino@cliente.it",
      "url": "www.prova.net",
      "chiusura_sett": "",
      "riferimento": "Mario Rossi",
      "cell_rif": "333-1234567",
      "nota": "Consegna piano terra",
      "data_validita": "",
      "data_creazione": "",
```

```
      "data_modifica": "20260430",
      "cod_conto_sost": "",
      "id_ind_sost": "",
      "annullato": "N",
      "gest_raggruppa": "N",
      "codice_cig": "",
      "codice_cup": "",
      "rif_esterno_tp": "",
      "rif_esterno_sgl": "",
      "rif_esterno_nr": "",
      "rif_esterno_cod": "",
      "rif_esterno_dt": "",
      "tp_indirizzo": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20260430 122142"
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate gestionale mancanti

Note Operative

- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Il parametro `fields` permette di restituire solo i campi desiderati (es. `?fields=id,cod_conto,descrizione`)
- Versione minima: 81600

Ricerca Indirizzi di Spedizione

Ricerca indirizzi di spedizione con filtri. Supporta paginazione e selezione campi.

POST `/indirizzi-spedizione/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)

Campo	Tipo	Obbl.	Posizione	Descrizione
filtri	array	Sì	body	Array di oggetti filtro

Per la sintassi dei filtri, vedi Ricerca e Filtri.

Esempio Request

```
POST /webapi/risorse/indirizzi-spedizione/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "cod_conto",
      "condizione": "=",
      "valore": "501.00001"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "id": "1",
      "cod_conto": "501.00001",
      "descrizione": "Sede operativa Milano",
      "indirizzo": "Via Roma 15",
      "cap": "20100",
      "localita": "MILANO",
      "provincia": "MI",
      "telefono": "02-12345678",
      "email": "magazzino@cliente.it",
      "riferimento": "Mario Rossi",
      "cell_rif": "333-1234567",
      "nota": "Consegna piano terra",
      "annullato": "N",
      "gest_raggruppa": "N"
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Il body deve contenere un array `filtri` con almeno un oggetto filtro
- Condizioni supportate: `=`, `<>`, `<`, `<=`, `>`, `>=`, `contiene`, `inizia_per`
- Versione minima: 81600

Lettura Indirizzo di Spedizione

Recupera uno specifico indirizzo di spedizione tramite il suo identificativo.

`GET /indirizzi-spedizione/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	Alfanumerico	Sì	path	Identificativo numerico dell'indirizzo
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/indirizzi-spedizione/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "id": "1",
  "cod_conto": "501.00001",
  "descrizione": "Sede operativa Milano",
  "indirizzo": "Via Roma 15",
  "cap": "20100",
  "localita": "MILANO",
  "provincia": "MI",
  "tp_nazionalita": "I",
  "cod_paese": "",
  "telefono": "02-12345678",
  "fax": "02-9876543",
  "email": "magazzino@cliente.it",
  "url": "www.prova.net",
  "chiusura_sett": "",
  "riferimento": "Mario Rossi",
  "cell_rif": "333-1234567",
  "nota": "Consegna piano terra",
  "data_validita": "",
  "data_creazione": "",
```

```
  "data_modifica": "20260430",
  "cod_conto_sost": "",
  "id_ind_sost": "",
  "annullato": "N",
  "gest_raggruppa": "N",
  "codice_cig": "",
  "codice_cup": "",
  "rif_esterno_tp": "",
  "rif_esterno_sgl": "",
  "rif_esterno_nr": "",
  "rif_esterno_cod": "",
  "rif_esterno_dt": "",
  "tp_indirizzo": "",
  "utente_ult_mod": "198187508",
  "data_ult_mod": "20260430 122142"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Indirizzo trovato
404	Not Found	Indirizzo non trovato

Note Operative

- La chiave è di tipo stringa ed è encodable (supporta encoding per caratteri speciali)
- Supporta il parametro `fields` per selezionare i campi da restituire
- Versione minima: 81600

Modifica Indirizzo di Spedizione

Aggiorna un indirizzo di spedizione esistente.

`PUT /indirizzi-spedizione/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	Alfanumerico	Sì	path	Identificativo numerico dell'indirizzo
descrizione	Alfanumerico	No	body	Nuova descrizione
indirizzo	Alfanumerico	No	body	Nuovo indirizzo
localita	Alfanumerico	No	body	Nuova località

Campo	Tipo	Obbl.	Posizione	Descrizione
cap	Alfanumerico	No	body	Nuovo C.A.P.
provincia	Alfanumerico	No	body	Nuova provincia
telefono	Alfanumerico	No	body	Nuovo telefono
email	Alfanumerico	No	body	Nuovo indirizzo email
riferimento	Alfanumerico	No	body	Nuovo riferimento
nota	Alfanumerico	No	body	Nuove note

Nota: La tabella riporta i campi principali. Tutti i campi della struttura (vedi Struttura Campi) sono accettati in fase di modifica.

Esempio Request

```
PUT /webapi/risorse/indirizzi-spedizione/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "indirizzo": "Corso Francia 100",
  "cap": "10138",
  "localita": "TORINO",
  "provincia": "TO"
}
```

Esempio Response

```
{
  "id": "1",
  "cod_conto": "501.00001",
  "descrizione": "Sede operativa Milano",
  "indirizzo": "Corso Francia 100",
  "cap": "10138",
  "localita": "TORINO",
  "provincia": "TO",
  "tp_nazionalita": "I",
  "telefono": "02-12345678",
  "email": "magazzino@cliente.it",
  "riferimento": "Mario Rossi",
  "annullato": "N",
  "gest_raggruppa": "N"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Indirizzo aggiornato
400	Bad Request	Dati non validi
404	Not Found	Indirizzo non trovato

Note Operative

- La cancellazione (`DELETE`) non è supportata per questa entità
- Nel body inviare solo i campi da modificare
- Versione minima: 81600

Entità: Liste Prelievo

L'entità liste di prelievo è una struttura **master-detail**: è presente una testata ed *n* righe ad essa associata (come per i documenti). Sono quindi presenti endpoint per accedere alla lista nel suo complesso ed altri per accedere solo alle righe:

- **Liste:** `https://base_address/webapi/risorse/liste-prelievo`
- **Righe:** `https://base_address/webapi/risorse/liste-prelievo/righe`

Come per i documenti, solo la singola lista può essere modificata (PUT) specificando tutte le righe a cui si vogliono apportare modifiche. Ogni riga d'ordine è identificata dalla chiave composta da `sigla`, `serie`, `numero` e `id_riga`.

Sempre come per i documenti, è possibile modificare solo i campi di testata specificando in query string il parametro `solo_testata=true`.

Riepilogo Endpoint

Metodo	URL	Descrizione
`POST`	`/liste-prelievo`	Inserimento prelievo
`GET`	`/liste-prelievo`	Lista prelievi - testate
`POST`	`/liste-prelievo/ricerca`	Ricerca prelievi - testate
`GET`	`/liste-prelievo/{id}`	Lettura prelievo (testata + righe)
`PUT`	`/liste-prelievo/{id}`	Modifica prelievo
`DELETE`	`/liste-prelievo/{id}`	Cancellazione prelievo

Metodo	URL	Descrizione
GET	/liste-prelievo/righe	Lista prelievi - righe
POST	/liste-prelievo/righe/ricerca	Ricerca prelievi - righe

Struttura Campi

Campi di Testata

Campo	Tipo	Descrizione
id	Numerico	Identificatore lista prelievo (chiave, obbligatorio in PUT)
data_creazione	Alfanumerico	Data creazione (formato AAAAMMGG)
stato	Alfanumerico	Stato della lista
descrizione	Alfanumerico	Descrizione
tipo	Alfanumerico	Tipo
utente_ult_mod	Alfanumerico	Utente ultima modifica
data_ult_mod	Data e ora	Data ora ultima modifica (formato AAAAMMGG HHMMSS)

Campi di Riga (Array Dinamici)

Campo	Tipo	Descrizione
sigla	Alfanumerico	Sigla documento
sottoazienda	Numerico	Codice sottoazienda documento
serie	Numerico	Serie documento
numero	Numerico	Numero documento
id_riga	Numerico	Riga univoca documento
prog_riga	Numerico	Progressivo riga (picking)
codice_articolo	Alfanumerico	Codice articolo
descr_articolo	Alfanumerico	Descrizione articolo
data_documento	Alfanumerico	Data documento (formato AAAAMMGG)
nr_riga_art	Numerico	Numero record articolo
id_magazzino	Numerico	Numero magazzino
id_ubicazione	Numerico	Identificatore ubicazione
dt_sca_doc	Alfanumerico	Data scadenza documento
tp_um_articolo	Alfanumerico	Tipo unità di misura

Campo	Tipo	Descrizione
um	Alfanumerico	Unità di misura
nr_colli	Numerico	Numero colli
quantita	Numerico con virgola	Quantità / Peso lordo
decimali_qta	Numerico	Numero decimali quantità articolo
coeff_conv	Numerico con virgola	Rapporto di conversione misura o tara
nota	Alfanumerico	Nota
id_lotto	Numerico	Identificatore lotto
pos_righe_lotto	Numerico	Posizione inizio righe lotto nell'array dei lotti
nr_righe_lotto	Numerico	Numero righe lotto della riga
id_lotto_ord	Numerico	Identificatore lotto su ordine
dec_qta_lo_ord	Numerico	Lotti su ordine - Numero decimali quantità
qta_lotto_ord	Numerico con virgola	Lotti su ordine - Quantità
nr_colli_lo_ord	Numerico	Lotti su ordine - Colli
qta_tg_lo_ord	Numerico	Lotti su ordine - Quantità per taglia (dim. 32)
cod_taglia	Numerico	Tabella taglie
numero_taglia	Numerico	Numero taglia
qta_taglia	Numerico	Quantità per taglia (dim. 32)
id_alias	Numerico	Identificatore alias
cod_conto	Alfanumerico	Codice cliente
cod_agente	Alfanumerico	Codice agente
cod_vettore	Alfanumerico	Codice vettore
ute_ult_mod	Alfanumerico	Id utente ultima modifica riga (sola lettura)
dt_ult_mod	Data e ora	Data ora ultima modifica riga (sola lettura, formato AAAAMMGG HHMMSS)
conto_ind_sped	Alfanumerico	Conto indirizzo di spedizione
ind_spedizione	Alfanumerico	Indirizzo spedizione
nr_righe_lista	Numerico	Numero di righe della lista (sola lettura)
priorita	Numerico	Priorità
gest_residuo	Alfanumerico	Stralcio / residuo
gest_tipo_lav	Alfanumerico	Riga in prelievo magazzino
note_documento	Alfanumerico	Annotazioni documento (dim. 10)
note_articolo	Alfanumerico	Annotazioni articolo (dim. 10)

Inserimento Prelievo

Crea una nuova lista di prelievo.

`POST /liste-prelievo`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	body	Identificatore lista prelievo (valorizzare a **0**)
descrizione	string	No	body	Descrizione della lista

> ⚠ **Nota Bene:** Il campo `id` deve essere presente nel body e valorizzato a **0** per l'inserimento di un nuovo record.

Codici di Stato

Codice	Descrizione	Causa
201	Created	Prelievo creato
400	Bad Request	Dati non validi

Note Operative

- Richiede coordinate gestionale
- Versione minima: 82300

Lista Prelievi - Testate

Restituisce l'elenco delle testate prelievo.

`GET /liste-prelievo`

Supporta `max`, `next`, `fields`.

Esempio Request

```
GET /webapi/risorse/liste-prelievo?max=20 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Ricerca Prelievi - Testate

Ricerca prelievi con filtri.

`POST /liste-prelievo/ricerca`

Supporta `max`, `next`, `fields`. Filtri nel body JSON (vedi Ricerca e Filtri).

Esempio Request

```
POST /webapi/risorse/liste-prelievo/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "stato",
      "condizione": "=",
      "valore": "C"
    }
  ]
}
```

Lettura Prelievo

Restituisce il dettaglio completo di una lista di prelievo (testata + righe).

`GET /liste-prelievo/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	ID lista prelievo

Supporta `fields`.

Esempio Request

```
GET /webapi/risorse/liste-prelievo/4 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "id": 4,
  "data_creazione": "20220610",
  "stato": "C",
  "descrizione": "Lista Prelievo del 10/06/2022",
  "tipo": "V",
  "sigla": [[1, "OC"]],
  "sottoazienda": [],
```

```
  "serie": [[1, 1]],
  "numero": [[1, 330]],
  "id_riga": [[1, 1]],
  "prog_riga": [],
  "codice_articolo": [[1, "ATT001"]],
  "data_documento": [[1, "20220302"]],
  "nr_riga_art": [[1, 49]],
  "id_magazzino": [[1, 1]],
  "id_ubicazione": [],
  "dt_sca_doc": [],
  "tp_um_articolo": [[1, "1"]],
  "um": [[1, "PZ"]],
  "nr_colli": [],
  "quantita": [[1, 1.0]],
  "decimali_qta": [[1, 2]],
  "coeff_conv": [],
  "id_lotto": [],
  "cod_taglia": [],
  "numero_taglia": [],
  "qta_taglia": [],
  "id_alias": [],
  "cod_conto": [[1, "501.00022"]],
  "cod_agente": [[1, "601.00006"]],
  "cod_vettore": [],
  "conto_ind_sped": [],
  "ind_spedizione": [],
  "nr_righe_lista": 0
}
```

Modifica Prelievo

Aggiorna una lista di prelievo esistente (testata e/o righe).

`PUT /liste-prelievo/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	ID lista prelievo

> ⚠ **Nota Bene:** In caso di aggiornamento con PUT è necessario specificare sempre tutti i campi chiave delle righe: `sigla`, `serie`, `numero`, `id_riga`, `prog_riga`.

Modifica Solo Testata

È possibile modificare solo i campi di testata specificando in query string il parametro `solo_testata=true`:

```
PUT /webapi/risorse/liste-prelievo/3?solo_testata=true HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "descrizione": "Lista aggiornata"
}
```

Cancellazione Prelievo

Elimina una lista di prelievo o singole righe.

DELETE `/liste-prelievo/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	ID lista prelievo
sigla	string	No	query	Sigla documento (per cancellazione selettiva)
serie	integer	No	query	Serie documento
numero	integer	No	query	Numero documento
id_riga	integer	No	query	ID riga
prog_riga	integer	No	query	Progressivo riga picking

La cancellazione senza parametri in query string elimina l'intera lista; con parametri elimina solo la riga specificata.

Esempio Request — Cancellazione Intera Lista

```
DELETE /webapi/risorse/liste-prelievo/5 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Request — Cancellazione Selettiva Riga

```
DELETE /webapi/risorse/liste-
prelievo/5?sigla=OC&serie=1&numero=339&id_riga=4&prog_riga=2 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Lista Prelievi - Righe

Restituisce le righe di tutti i prelievi (vista trasversale).

`GET /liste-prelievo/righe`

Supporta `max`, `next`, `fields`.

Esempio Request

```
GET /webapi/risorse/liste-prelievo/righe?max=100 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Ricerca Prelievi - Righe

Ricerca nelle righe prelievo con filtri.

`POST /liste-prelievo/righe/ricerca`

Supporta `max`, `next`, `fields`. Filtri nel body JSON (vedi Ricerca e Filtri).

Esempio Request

```
POST /webapi/risorse/liste-prelievo/righe/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice_articolo",
      "condizione": "=",
      "valore": "ATT001"
    }
  ]
}
```

Righe d'Ordine vs Righe di Picking

La peculiarità di questa entità risiede nel fatto che è possibile inserire un ulteriore livello di dettaglio: le **righe di picking**. Per ogni riga della lista è infatti possibile "creare" ulteriori righe di prelievo (picking) ad essa associata.

Per ogni record (riga) della lista il campo `id_riga` identifica a quale riga d'ordine è associato. Se la riga è una riga di picking l'elemento `prog_riga` identifica il progressivo di picking relativo alla riga d'ordine, altrimenti se per la riga non è presente l'elemento `prog_riga` si intende che si tratta della riga d'ordine.

In sintesi:

- `prog_riga` **non presente** → si tratta della **riga d'ordine**
- `prog_riga` **presente** → si tratta di una **riga di picking** relativa alla riga d'ordine identificata da `id_riga`

Inserimento Riga di Picking

Nel caso si voglia inserire una nuova riga di picking, è sufficiente fare una PUT referenziando la riga d'ordine su cui si vuole agire ed impostando il progressivo di riga a **-1**. In questo caso è necessario specificare, oltre ai dati di chiave, solo le quantità.

> ℹ **Nota:** Se l'articolo è gestito a colli o taglie è necessario utilizzare i campi dedicati (`nr_colli`, `qta_taglia`) anziché `quantita`.

Esempio: Inserimento Riga di Picking

Consideriamo la seguente GET sulla lista di prelievo 4 avente una sola riga d'ordine (OC serie 1 numero 330, articolo ATT001, quantità 1):

```
GET https://base_address/webapi/risorse/liste-prelievo/4
```

La response restituisce la lista con una sola riga d'ordine (vedi Lettura Prelievo).

Per aggiungere una riga di picking relativa alla riga d'ordine è necessario fare una PUT indicando, oltre alla chiave per identificare la riga (`sigla` + `serie` + `numero` + `id_riga` + `prog_riga`), la quantità di picking e impostando il progressivo di riga a **-1** (`"prog_riga": [[1, -1]]`):

```
PUT /webapi/risorse/liste-prelievo/4 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "sigla": [[1, "OC"]],
  "serie": [[1, 1]],
  "numero": [[1, 330]],
  "id_riga": [[1, 1]],
  "prog_riga": [[1, -1]],
  "quantita": [[1, 2]]
}
```

In questo esempio: - `prog_riga` impostato a **-1** indica l'inserimento di una nuova riga di picking - Viene specificata la chiave per identificare la riga d'ordine (`sigla` + `serie` + `numero` + `id_riga`) - La quantità di picking è 2

Gestione dei Lotti

Le variabili per l'identificazione dei lotti sulle righe sono le seguenti:

Campo	Applicabile a	Descrizione
`id_lotto`	Righe di picking (`prog_riga` ≥ 1)	Identificatore lotto sulla riga di picking
`id_lotto_ord`	Righe d'ordine (senza `prog_riga`)	Identificatore lotto sulla riga d'ordine
`pos_righe_lotto`	Righe d'ordine	Posizione in cui iniziano i lotti della riga d'ordine i-esima nell'array dei lotti (`id_lotto_ord`)
`nr_righe_lotto`	Righe d'ordine	Numero di lotti gestiti dalla riga
`dec_qta_lo_ord`	Righe d'ordine	Decimali quantità lotto riga ordine
`qta_lotto_ord`	Righe d'ordine	Quantità lotto riga ordine
`nr_colli_lo_ord`	Righe d'ordine	Numero colli lotto della riga ordine
`qta_tg_lo_ord`	Righe d'ordine	Quantità taglia del lotto della riga ordine

Questi campi servono perché, sebbene tutte le righe siano omogenee, le righe d'ordine possono avere la gestione a multi-lotto: è quindi necessario poter leggere tutti i lotti della riga (il meccanismo è identico a quello dei documenti: per ogni riga c'è l'indice per l'accesso all'array dei lotti + offset). Questo per quanto riguarda l'endpoint `liste-prelievo`.

Per l'endpoint `liste-prelievo/righe` invece il lotto è singolo e compare nell'array `id_lotto` con lo stesso indice della riga di picking. Per le righe d'ordine `id_lotto=0` e sono valorizzati gli array `id_lotto_ord`.

Note Operative Generali

- Tutti gli endpoint richiedono coordinate gestionale
- La chiave per le righe è composta da: `sigla` + `serie` + `numero` + `id_riga`
- Le righe di picking aggiungono `prog_riga` alla chiave
- Per la modifica (PUT) è obbligatorio specificare sempre tutti i campi chiave
- Per modificare solo la testata: `?solo_testata=true`

- La cancellazione senza parametri in query string elimina l'intera lista; con parametri elimina solo la riga specificata
- Versione minima: 82300

Entità: Lotti

Gestione dei lotti e matricole di magazzino. Supporta CRUD completo per la tracciabilità degli articoli tramite lotti, matricole e campi personalizzati definiti sul tipo lotto.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/lotti`	Inserimento lotto
GET	`/lotti`	Lista lotti
POST	`/lotti/ricerca`	Ricerca lotti con filtri
GET	`/lotti/{id}`	Lettura lotto singolo
PUT	`/lotti/{id}`	Modifica lotto
DELETE	`/lotti/{id}`	Cancellazione lotto

Struttura Campi

Campo	Tipo	Descrizione	Note
id	Numerico	Identificatore lotto	Chiave, obbligatorio in PUT
stato	Alfanumerico	Stato del lotto	Valori: C = corrente, B = bloccato, A = annullato
msg_bloc_ann	Alfanumerico	Messaggio stato bloccato/annullato	
caricato	Alfanumerico	Carico effettuato da movimento magazzino	Valori: S/N
tp_ctrl_carico	Alfanumerico	Tipo controllo carico lotto	
cod_articolo	Alfanumerico	Codice articolo associato al lotto	
cod_tipo_lotto	Alfanumerico	Codice tipo lotto/matricola	Riferimento a tipi lotti (vedi Dati Generali)
cod_fornitore	Alfanumerico	Codice fornitore	
data_carico	Alfanumerico	Data di carico	Formato

Campo	Tipo	Descrizione	Note
			AAAAMMGG
data_validita	Alfanumerico	Data di validità/scadenza	Formato AAAAMMGG
data_creazione	Alfanumerico	Data creazione	Solo lettura
data_modifica	Alfanumerico	Data ultima modifica	Solo lettura
cod_ute_lotto	Alfanumerico	Codice utente lotto (numero lotto assegnato)	
num_campi_pers	Numerico	Numero campi personalizzati	Visibile in lettura singola
cod_campi_pers	Array	Codici dei campi personalizzati	Array di coppie [indice, codice]
tp_campi_pers	Array	Tipi dei campi personalizzati	Array di coppie [indice, tipo]
val_campi_pers	Array	Valori dei campi personalizzati	Array di coppie [indice, valore]
gest_stor_lotto	Alfanumerico	Gestione storicizzazione lotto	Valori: S/N
utente_ult_mod	Alfanumerico	Utente ultima modifica	Solo lettura
data_ult_mod	Data e ora	Data/ora ultima modifica	Formato AAAAMMGG HHMMSS, solo lettura

Inserimento Lotto

Crea un nuovo lotto/matricola.

`POST /lotti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
cod_articolo	string	Sì	body	Codice articolo
cod_tipo_lotto	string	Sì	body	Codice tipo lotto
cod_ute_lotto	string	No	body	Codice utente lotto (numero lotto assegnato)
cod_fornitore	string	No	body	Codice fornitore
data_carico	string	No	body	Data di carico (AAAAMMGG)

Campo	Tipo	Obbl.	Posizione	Descrizione
data_validita	string	No	body	Data validità/scadenza (AAAAMMGG)
stato	string	No	body	Stato lotto (C, B, A)
tp_ctrl_carico	string	No	body	Tipo controllo carico
gest_stor_lotto	string	No	body	Gestione storicizzazione (S/N)
val_campi_pers	array	No	body	Valori campi personalizzati

Esempio Request

```
POST /webapi/risorse/lotti HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=AAAA

{
  "cod_articolo": "TELDDDDAA01",
  "cod_tipo_lotto": "ab",
  "cod_ute_lotto": "LOT20260101",
  "data_carico": "20260601",
  "data_validita": "20271231"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/lotti/157
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Lotto creato con successo
400	Bad Request	Dati non validi o coordinate mancanti

Note Operative

- Richiede header `Coordinate-Gestionale` con Azienda e Anno
- Versione minima: 82001
- Il tipo lotto deve essere precedentemente definito nei dati generali (vedi Dati Generali)

Lista Lotti

Restituisce l'elenco dei lotti registrati nell'azienda.

`GET /lotti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Elenco campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/lotti?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=XXX Anno=AAAA
```

Esempio Response


```
{
  "dati": [
    {
      "id": 4,
      "stato": "C",
      "msg_bloc_ann": "",
      "caricato": "S",
      "tp_ctrl_carico": "M",
      "cod_articolo": "TELDDDDAA01",
      "cod_tipo_lotto": "ab",
      "cod_fornitore": "",
      "data_carico": "20160601",
      "data_validita": "",
      "data_creazione": "20160517",
      "data_modifica": "20160517",
      "cod_ute_lotto": "000000000",
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20260108 095824"
    }
  ],
  "data_ric_elab": "20260617 105752",
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti

Note Operative

- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Supporta selezione campi tramite `fields` (vedi Formato Request)
- Il campo `data_ric_elab` nella response indica la data/ora di elaborazione della richiesta
- Versione minima: 82001

Ricerca Lotti

Ricerca lotti applicando filtri sui campi disponibili.

`POST /lotti/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Per la sintassi dei filtri, vedi Ricerca e Filtri.

Esempio Request

```
POST /webapi/risorse/lotti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=AAAA

{
  "filtri": [
    {
      "campo": "cod_articolo",
      "condizione": "=",
      "valore": "TELDDDDAA01"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "id": 4,
      "stato": "C",
      "msg_bloc_ann": "",
```

```
      "caricato": "S",
      "tp_ctrl_carico": "M",
      "cod_articolo": "TELDDDDAA01",
      "cod_tipo_lotto": "ab",
      "cod_fornitore": "",
      "data_carico": "20160601",
      "data_validita": "",
      "data_creazione": "20160517",
      "data_modifica": "20160517",
      "cod_ute_lotto": "000000000",
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20260108 095824"
    }
  ],
  "data_ric_elab": "20260617 105809",
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Versione minima: 82001
- Campi filtrabili: tutti i campi della risorsa (es. `cod_articolo`, `cod_tipo_lotto`, `stato`, `cod_fornitore`)

Lettura Lotto

Recupera i dati completi di uno specifico lotto, inclusi i campi personalizzati.

`GET /lotti/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	Identificatore numerico del lotto

Esempio Request

```
GET /webapi/risorse/lotti/4 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=XXX Anno=AAAA
```

Esempio Response


```
{
  "id": 4,
  "stato": "C",
  "msg_bloc_ann": "",
  "caricato": "S",
  "tp_ctrl_carico": "M",
  "ctrl_carico_mem": "",
  "cod_articolo": "TELDDDDAA01",
  "cod_tipo_lotto": "ab",
  "cod_fornitore": "",
  "data_carico": "20160601",
  "data_validita": "",
  "data_creazione": "20160517",
  "data_modifica": "20160517",
  "cod_ute_lotto": "000000000",
  "num_campi_pers": 2,
  "cod_campi_pers": [
    [1, "pn"],
    [2, "sn"]
  ],
  "tp_campi_pers": [
    [1, "I"],
    [2, "A"]
  ],
  "val_campi_pers": [
    [2, "SN"]
  ],
  "gest_stor_lotto": "N",
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20260108 095824"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lotto trovato
404	Not Found	Lotto con l'ID specificato non esistente

Note Operative

- Chiave numerica semplice (ID intero)
- La lettura singola restituisce anche i campi personalizzati (`num_campi_pers`, `cod_campi_pers`, `tp_campi_pers`, `val_campi_pers`) non presenti nella lista
- I tipi campi personalizzati sono: `I` = intero, `A` = alfanumerico
- Versione minima: 82001

Modifica Lotto

Aggiorna i dati di un lotto esistente.

`PUT /lotti/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	Identificatore numerico del lotto
id	integer	Sì	body	Identificatore lotto (deve corrispondere al path)
stato	string	No	body	Nuovo stato (C, B, A)
msg_bloc_ann	string	No	body	Messaggio blocco/annullamento
data_validita	string	No	body	Nuova data validità (AAAAMMGG)
cod_fornitore	string	No	body	Codice fornitore
cod_ute_lotto	string	No	body	Codice utente lotto
tp_ctrl_carico	string	No	body	Tipo controllo carico
gest_stor_lotto	string	No	body	Gestione storicizzazione (S/N)
val_campi_pers	array	No	body	Valori campi personalizzati

Esempio Request

```
PUT /webapi/risorse/lotti/4 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=XXX Anno=AAAA

{
  "id": 4,
  "data_validita": "20271231",
  "stato": "C"
}
```

Esempio Response

```
{
  "id": 4,
  "stato": "C",
  "msg_bloc_ann": "",
  "caricato": "S",
```

```
  "tp_ctrl_carico": "M",
  "cod_articolo": "TELDDDDAA01",
  "cod_tipo_lotto": "ab",
  "cod_fornitore": "",
  "data_carico": "20160601",
  "data_validita": "20271231",
  "data_creazione": "20160517",
  "data_modifica": "20160517",
  "cod_ute_lotto": "000000000",
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20260617 110000"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lotto aggiornato
400	Bad Request	Dati non validi
404	Not Found	Lotto non trovato

Note Operative

- Il campo `id` è obbligatorio nel body (deve corrispondere all'ID nel path)
- I campi di sola lettura (`data_creazione`, `data_modifica`, `utente_ult_mod`, `data_ult_mod`) vengono aggiornati automaticamente dal sistema
- Versione minima: 82001

Cancellazione Lotto

Elimina un lotto dall'archivio.

DELETE `/lotti/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	Identificatore numerico del lotto

Esempio Request

```
DELETE /webapi/risorse/lotti/4 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=XXX Anno=AAAA
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lotto eliminato
404	Not Found	Lotto non trovato
409	Conflict	Lotto utilizzato in movimenti di magazzino

Note Operative

- Versione minima: 82001
- Non è possibile eliminare un lotto referenziato da movimenti di magazzino esistenti

---

Esempi Verificati (Live)

Test eseguiti su installazione `gsuzzi:9014`, azienda `UB1`, anno `2026`.

GET lista (max=3):


```
{
  "dati": [
    {
      "id": 4,
      "stato": "C",
      "msg_bloc_ann": "",
      "caricato": "S",
      "tp_ctrl_carico": "M",
      "cod_articolo": "TELDDDDAA01",
      "cod_tipo_lotto": "ab",
      "cod_fornitore": "",
      "data_carico": "20160601",
      "data_validita": "",
      "data_creazione": "20160517",
      "data_modifica": "20160517",
      "cod_ute_lotto": "000000000",
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20260108 095824"
    }
  ],
  "data_ric_elab": "20260617 105752",
  "next":
"57415f39666531383834342d366132612d313166312d396238342d3066373565323166661303739-
0600000000000000"
}
```


GET singolo (id=4):


```
{
  "id": 4,
  "stato": "C",
```

```
  "msg_bloc_ann": "",
  "caricato": "S",
  "tp_ctrl_carico": "M",
  "ctrl_carico_mem": "",
  "cod_articolo": "TELDDDDAA01",
  "cod_tipo_lotto": "ab",
  "cod_fornitore": "",
  "data_carico": "20160601",
  "data_validita": "",
  "data_creazione": "20160517",
  "data_modifica": "20160517",
  "cod_ute_lotto": "000000000",
  "num_campi_pers": 2,
  "cod_campi_pers": [[1, "pn"], [2, "sn"]],
  "tp_campi_pers": [[1, "I"], [2, "A"]],
  "val_campi_pers": [[2, "SN"]],
  "gest_stor_lotto": "N",
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20260108 095824"
}
```


POST ricerca (filtro su cod_articolo):


```
{
  "dati": [
    {
      "id": 4,
      "stato": "C",
      "cod_articolo": "TELDDDDAA01",
      "cod_tipo_lotto": "ab",
      "data_carico": "20160601"
    }
  ],
  "data_ric_elab": "20260617 105809",
  "next":
"57415f61613131633562372d366132612d313166312d393062622d3066373565323166613031303739-
0500000000000000"
}
```


Entità: Mappa Articoli

Consultazione della mappa ubicazioni degli articoli di magazzino. Endpoint di sola lettura che fornisce informazioni sulla collocazione fisica degli articoli nei magazzini, comprensive di date delle ultime movimentazioni e parametri di scorta.

Riepilogo Endpoint

Metodo	URL	Descrizione

Metodo	URL	Descrizione
GET	/mappa-articoli	Lista mappa ubicazione articoli
POST	/mappa-articoli/ricerca	Ricerca mappa ubicazione articoli con filtri

Campi Restituiti

Campo	Tipo	Descrizione
codice	Alfanumerico	Codice articolo (chiave)
nr_magazzino	Numerico	Numero magazzino
cod_ubicazione	Alfanumerico	Codice ubicazione
cod_sotto_ubic	Alfanumerico	Codice sotto-ubicazione
id_ubicazione	Numerico	ID ubicazione
dt_ult_carico	Alfanumerico	Ultima data di carico (formato AAAAMMGG)
dt_ult_scarico	Alfanumerico	Ultima data di scarico (formato AAAAMMGG)
dt_ult_ordine	Alfanumerico	Ultima data ordini/impegni (formato AAAAMMGG)
dt_ult_mov	Alfanumerico	Ultima data movimenti (formato AAAAMMGG)
scorta_massima	Numerico	Scorta massima
scorta_minima	Numerico	Scorta minima
dt_ult_car_prec	Alfanumerico	Ultima data di carico precedente (formato AAAAMMGG)
utente_ult_mod	Alfanumerico	Utente ultima modifica
data_ult_mod	Data e ora	Data/ora ultima modifica (formato AAAAMMGG HHMMSS)

Lista Mappa Ubicazione Articoli

Restituisce la mappa delle ubicazioni degli articoli nei magazzini, con date delle ultime operazioni e parametri di scorta.

`GET /mappa-articoli`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire

Campo	Tipo	Obbl.	Posizione	Descrizione
next	string	No	query	Token di paginazione (vedi Paginazione)
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/mappa-articoli?max=5 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "codice": "121212",
      "nr_magazzino": 1,
      "cod_ubicazione": "",
      "cod_sotto_ubic": "",
      "id_ubicazione": 0,
      "dt_ult_carico": "",
      "dt_ult_scarico": "20260515",
      "dt_ult_ordine": "20261231",
      "dt_ult_mov": "20261231",
      "scorta_massima": 1,
      "scorta_minima": 999,
      "dt_ult_car_prec": "20240730",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20260609 141921"
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Mappa restituita con successo
400	Bad Request	Coordinate gestionale mancanti o non valide

Note Operative

- Richiede header `Coordinate-Gestionale` con almeno `Azienda` e `Anno`
- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Supporta selezione campi tramite parametro `fields`
- Le date sono restituite in formato stringa `AAAAMMGG` (vuoto se non valorizzate)

- Un articolo può comparire più volte se presente in più magazzini/ubicazioni
- Versione minima: 83000

Ricerca Mappa Ubicazione Articoli

Ricerca nella mappa articoli con filtri (vedi Ricerca e Filtri).

`POST /mappa-articoli/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/mappa-articoli/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice",
      "condizione": "=",
      "valore": "121212"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "121212",
      "nr_magazzino": 1,
      "cod_ubicazione": "",
      "cod_sotto_ubic": "",
      "id_ubicazione": 0,
      "dt_ult_carico": "",
```

```
      "dt_ult_scarico": "20260515",
      "dt_ult_ordine": "20261231",
      "dt_ult_mov": "20261231",
      "scorta_massima": 1,
      "scorta_minima": 999,
      "dt_ult_car_prec": "20240730",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20260609 141921"
    },
    {
      "codice": "121212",
      "nr_magazzino": 2,
      "cod_ubicazione": "C01S",
      "cod_sotto_ubic": "01R1",
      "id_ubicazione": 1770,
      "dt_ult_carico": "",
      "dt_ult_scarico": "20250603",
      "dt_ult_ordine": "",
      "dt_ult_mov": "20250603",
      "scorta_massima": 0,
      "scorta_minima": 0,
      "dt_ult_car_prec": "20231128",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20260609 141921"
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata con successo
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Il campo `filtri` nel body è obbligatorio e deve contenere almeno un oggetto filtro
- È possibile filtrare per qualsiasi campo della risorsa (es. `codice`, `nr_magazzino`, `cod_ubicazione`)
- Supporta le condizioni standard: `=`, `<`, `<=`, `>`, `>=`, `<>`, `contiene`, `inizia_per`
- Per la sintassi completa dei filtri, vedi Ricerca e Filtri
- Versione minima: 83000

Entità: MyDB

Gestione degli archivi personalizzati MyDB (database custom definiti tramite Passbuilder). La risorsa `mydb` è un contenitore (come la risorsa `documenti`) e quindi **non è direttamente accessibile**: è necessario specificare sempre nel path anche l'archivio MyDB su cui si intende operare.

Il formato del path è:

```
https://base_address/webapi/risorse/mydb/<APP>@<nome_archivio>
```

Dove: - `<APP>` è l'app Passbuilder in cui è presente l'archivio MyDB - `<nome_archivio>` è il nome dell'archivio MyDB

Sono disponibili tutti i metodi REST con la stessa logica delle altre entità.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	`/mydb/{APP}@{archivio}`	Lista records
GET	`/mydb/{APP}@{archivio}/{id}`	Lettura singolo record
POST	`/mydb/{APP}@{archivio}`	Inserimento record
POST	`/mydb/{APP}@{archivio}/ricerca`	Ricerca records
PUT	`/mydb/{APP}@{archivio}/{id}`	Modifica record
DELETE	`/mydb/{APP}@{archivio}/{id}`	Cancellazione record
GET	`/mydb/{APP}@{archivio}/{sigla_doc}`	Lista records (con sigla documento)
GET	`/mydb/{APP}@{archivio}/{sigla_doc}/{id}`	Lettura record (con sigla documento)
POST	`/mydb/{APP}@{archivio}/{sigla_doc}`	Inserimento record (con sigla documento)
POST	`/mydb/{APP}@{archivio}/{sigla_doc}/ricerca`	Ricerca records (con sigla documento)
PUT	`/mydb/{APP}@{archivio}/{sigla_doc}/{id}`	Modifica record (con sigla documento)

Metodo	URL	Descrizione
DELETE	/mydb/{APP}@{archivio}/{sigla_doc}/{id}	Cancellazione record (con sigla documento)

Struttura dei Campi

Campo	Tipo	Chiave	Descrizione
nome_archivio	Alfanumerico	Sì	Nome dell'archivio (formato APP@nome_archivio)
sigla_doc	Alfanumerico	Sì	Sigla documento per archivi MyDB collegati a documenti o parcelle
id	Numerico	Sì	Numero ID del record
numero_campi	Numerico	—	Numero campi archivio (sola lettura)
dati_campi	Array dinamico	—	Dati campi archivio (array di [indice, valore])
tipi_campi	Array dinamico	—	Tipi campi archivio (sola lettura)
etichette_campi	Array dinamico	—	Etichette campi archivio (sola lettura)
riportabil_doc	Array dinamico	—	Riportabilità sui documenti per anagrafiche dinamiche (S/N)
annullato	Alfanumerico	—	Record anagrafica annullato (S/N)
bloccato	Alfanumerico	—	Anagrafica bloccata da Passepartout (S/N) (sola lettura)
utente_ult_mod	Alfanumerico	—	Utente ultima modifica
data_ult_mod	Data e ora	—	Data ora ultima modifica (formato AAAAMMGG HHMMSS)

Lista Records Archivio MyDB

Restituisce la lista di tutti i record dell'archivio MyDB indicato.

GET /mydb/{APP}@{archivio}

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
APP@archivio	string	Sì	path	Identificativo archivio MyDB
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/mydb/MIA_APP@mydb?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "nome_archivio": "MIA_APP@mydb",
      "sigla_doc": "",
      "id": 1,
      "numero_campi": 3,
      "dati_campi": [
        [1, "1"],
        [2, "TREK"],
        [3, "FUEL-EX"]
      ],
      "tipi_campi": [
        [1, "N"],
        [2, "A"],
        [3, "A"]
      ],
      "etichette_campi": [
        [1, "Codice"],
        [2, "Marca"],
        [3, "Modello"]
      ],
      "riportabil_doc": [],
      "annullato": "",
      "bloccato": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20220420 151837"
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti
404	Not Found	Archivio non trovato

Lettura Singolo Record

Recupera un singolo record per ID.

`GET /mydb/{APP}@{archivio}/{id}`

Esempio Request

```
GET /webapi/risorse/mydb/MIA_APP@mydb/42 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "nome_archivio": "MIA_APP@mydb",
  "sigla_doc": "",
  "id": 42,
  "numero_campi": 5,
  "dati_campi": [],
  "tipi_campi": [],
  "etichette_campi": [],
  "riportabil_doc": [],
  "annullato": "",
  "bloccato": "",
  "utente_ult_mod": "198187508",
  "data_ult_mod": "20211217 113408"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Record trovato
404	Not Found	Record o archivio non trovato

Inserimento Record

Crea un nuovo record nell'archivio MyDB specificato.

`POST /mydb/{APP}@{archivio}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
APP@archivio	string	Sì	path	Identificativo archivio MyDB
id	integer	Sì	body	Deve essere presente e valorizzato a `0`
numero_cam	integer	Sì	body	Numero dei campi da

Campo	Tipo	Obbl.	Posizione	Descrizione
pi				valorizzare
dati_campi	array	Sì	body	Array di `[indice, valore]` per ogni campo

Esempio Request

```
POST /webapi/risorse/mydb/MIA_APP@mydb/OC HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "id": 0,
  "numero_campi": 2,
  "dati_campi": [
    [1, "C000OC001.000008"],
    [2, "valore_campo"]
  ]
}
```

> ⚠ **Nota Bene:** L'`id` del record deve essere presente e valorizzato a `0` dentro il body della request. Se omesso o valorizzato diversamente, viene restituito un errore.

Codici di Stato

Codice	Descrizione	Causa
200	OK	Record creato
400	Bad Request	Dati o archivio non validi
404	Not Found	Archivio MyDB non trovato

Modifica Record

Aggiorna un record esistente.

`PUT /mydb/{APP}@{archivio}/{id}`

Esempio Request

```
PUT /webapi/risorse/mydb/MIA_APP@mydb/42 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "numero_campi": 1,
```

```
  "dati_campi": [
    [2, "nuovo_valore"]
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Record aggiornato
404	Not Found	Record non trovato

Cancellazione Record

Elimina un record dall'archivio.

`DELETE /mydb/{APP}@{archivio}/{id}`

Esempio Request

```
DELETE /webapi/risorse/mydb/MIA_APP@mydb/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Record eliminato
404	Not Found	Record non trovato

Ricerca Records

Ricerca record nell'archivio MyDB con filtri.

`POST /mydb/{APP}@{archivio}/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
APP@archivio	string	Sì	path	Identificativo archivio
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/mydb/MIA_APP@mydb/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "dati_campi",
      "condizione": "=",
      "valore": "valore_cercato",
      "indice1": 2
    }
  ]
}
```

Note Operative

- Per filtrare sui campi dell'archivio (che sono array), utilizzare il campo `dati_campi` e specificare `indice1` con il numero del campo da filtrare
- Supporta le stesse condizioni degli altri endpoint REST (vedi Ricerca e Filtri)

Variante con Sigla Documento

Se si vuole interrogare un archivio MyDB collegato a un tipo di documento, va specificata anche la sigla documento nel path.

Lista di tutti i record collegati a un tipo documento

```
GET /webapi/risorse/mydb/MIA_APP@mydb/OC HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Lettura di un singolo record collegato a un documento

```
GET /webapi/risorse/mydb/MIA_APP@mydb/OC/42 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

I dati restituiti avranno valorizzato anche l'elemento `sigla_doc`:

```
{
  "nome_archivio": "MIA_APP@mydb",
  "sigla_doc": "OC",
  "id": 42,
  "numero_campi": 2,
```

```
  "dati_campi": [],
  "tipi_campi": [],
  "etichette_campi": [],
  "riportabil_doc": [],
  "annullato": "",
  "bloccato": "",
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20211217 122426"
}
```

Inserimento con sigla documento

```
POST /webapi/risorse/mydb/MIA_APP@mydb/OC HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "id": 0,
  "numero_campi": 2,
  "dati_campi": [
    [1, "C000OC001.000008"],
    [2, "valore_campo"]
  ]
}
```

> **Nota:** La sigla documento è un codice di 2 lettere maiuscole (es. OC, OF, FT, BC) che identifica il tipo di documento a cui i record MyDB sono collegati. Il campo `dati_campi[1]` per l'estensione contiene il codice del documento nel formato specificato dal manuale MyDB (vedi documentazione Passbuilder).

Note Operative

- I restanti metodi (PUT, DELETE) seguono la stessa logica, aggiungendo la sigla documento nel path
- Versione minima: 81800

Note Operative Generali

- Il formato dell'archivio è `APP@nome_archivio` dove `APP` è il codice dell'app Passbuilder (es. `924148CQWEBAPI@webapi05`)
- I nomi e i tipi dei campi dipendono dalla struttura dell'archivio MyDB definita in Passbuilder — ogni archivio avrà campi diversi
- L'array `dati_campi` contiene coppie `[indice, valore]` dove l'indice è il numero progressivo del campo
- Gli array `tipi_campi` e `etichette_campi` sono in sola lettura e descrivono la struttura dell'archivio

- Per conoscere la lista di archivi MyDB disponibili si può usare il servizio `lista_strutture_mydb` (vedi Servizio: Lista Strutture MyDB)
- Per l'autenticazione su campi etichettati come "Aut" si può usare il servizio `verifica_autenticazione_mydb` (vedi Servizio: Verifica Autenticazione MyDB)

Esempi Verificati (Live)

Esempi ottenuti da test live su installazione `gsuzzi:9014`, azienda UB1, anno 2026.

GET archivio anagrafico senza sigla documento

```
GET /webapi/risorse/mydb/924148CQWEBAPI@webapi05?max=2 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```


```
{
  "dati": [
    {
      "nome_archivio": "924148CQWEBAPI@webapi05",
      "sigla_doc": "",
      "id": 1,
      "numero_campi": 3,
      "dati_campi": [[1, "1"], [2, "TREK"], [3, "FUEL-EX"]],
      "tipi_campi": [[1, "N"], [2, "A"], [3, "A"]],
      "etichette_campi": [[1, "Codice"], [2, "Marca"], [3, "Modello"]],
      "riportabil_doc": [],
      "annullato": "",
      "bloccato": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20220420 151837"
    },
    {
      "nome_archivio": "924148CQWEBAPI@webapi05",
      "sigla_doc": "",
      "id": 3,
      "numero_campi": 3,
      "dati_campi": [[1, "2"], [2, "TREK"], [3, "FUEL-EX 9.7"]],
      "tipi_campi": [[1, "N"], [2, "A"], [3, "A"]],
      "etichette_campi": [[1, "Codice"], [2, "Marca"], [3, "Modello"]],
      "riportabil_doc": [],
      "annullato": "",
      "bloccato": "",
      "utente_ult_mod": "198187508",
      "data_ult_mod": "20220506 160637"
    }
  ],
  "next": "0-0-4-00000004-00000000"
}
```

GET archivio con sigla documento OC

```
GET /webapi/risorse/mydb/924148CQWEBAPI@webapi02/OC?max=1 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "dati": [
    {
      "nome_archivio": "924148CQWEBAPI@webapi02",
      "sigla_doc": "OC",
      "id": 1,
      "numero_campi": 2,
      "dati_campi": [[1, "000OC666.00000200000000"], [2, "PIPPO"]],
      "tipi_campi": [[1, "*"], [2, "A"]],
      "etichette_campi": [[1, "Testate tutti i documenti"], [2, "Test WebAPI"]],
      "riportabil_doc": [],
      "annullato": "",
      "bloccato": "",
      "utente_ult_mod": "626892910",
      "data_ult_mod": "20221124 102744"
    }
  ],
  "next": "0-0-5-00000005-00000000"
}
```

Entità: Prima Nota

Gestione delle registrazioni di prima nota contabile. Supporta due formati di chiave: completa (8 segmenti) e abbreviata (data+progressivo). Include anche consultazione righe.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/prima-nota`	Inserimento primanota
GET	`/prima-nota`	Lista prima nota
POST	`/prima-nota/ricerca`	Ricerca primanota
GET	`/prima-nota/{chiave_completa}`	Lettura prima nota (chiave completa)
PUT	`/prima-nota/{chiave_completa}`	Modifica prima nota (chiave completa)
DELETE	`/prima-nota/{chiave_c`	Cancellazione prima nota (chiave completa)

Metodo	URL	Descrizione
	ompleta}	
GET	/prima-nota/{data}+{progressivo}	Lettura prima nota (chiave abbreviata)
PUT	/prima-nota/{data}+{progressivo}	Modifica prima nota (chiave abbreviata)
DELETE	/prima-nota/{data}+{progressivo}	Cancellazione prima nota (chiave abbreviata)
GET	/prima-nota/righe	Lista righe primanota
POST	/prima-nota/righe/ricerca	Ricerca righe prima nota

Inserimento Primanota

Crea una nuova registrazione di prima nota.

`POST /prima-nota`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
data_registr	string	Sì	body	Data registrazione (YYYYMMDD)
cau_contabile	string	Sì	body	Causale contabile (2 caratteri)
rstro_prot_iva	string	Sì	body	Registro protocollo IVA
serie_prot	integer	No	body	Serie protocollo
nr_protocollo	integer	No	body	Numero protocollo (0 = AUTO)
nr_documento	integer	No	body	Numero documento (0 = AUTO)
data_documento	string	No	body	Data documento (YYYYMMDD)
codice_conto	array	Sì	body	Codici conto per ogni riga
importo_riga	array	Sì	body	Importo per ogni riga (positivo = dare, negativo = avere)

Nota: Le righe contabili sono rappresentate da variabili di tipo array con formato `[[indice, valore], ...]`. Gli importi positivi vanno in dare, quelli negativi in avere. La registrazione deve essere bilanciata (somma importi = 0).

Esempio Request

```
POST /webapi/risorse/prima-nota HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "data_registr": "20260616",
  "cau_contabile": "FE",
  "rstro_prot_iva": "V",
  "serie_prot": 1,
  "nr_protocollo": 0,
  "nr_documento": 0,
  "data_documento": "20260616",
  "codice_conto": [
    [1, "501.00001"],
    [2, "501.00001"]
  ],
  "importo_riga": [
    [1, 1000],
    [2, -1000]
  ]
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/prima-nota/20260616+6
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Registrazione creata
400	Bad Request	Dati non validi o registrazione non bilanciata

Note Operative

- Richiede coordinate gestionale
- Le righe devono essere bilanciate (dare = avere)
- Versione minima: 85000

Lista Prima Nota

Restituisce l'elenco delle registrazioni di prima nota.

`GET /prima-nota`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/prima-nota?max=30 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "dati": [
    {
      "nr_operatore": 1,
      "sottoazienda": 1,
      "data_registr": "20260616",
      "progressivo": 6,
      "cau_contabile": "FE",
      "descrizione": "",
      "rstro_prot_iva": "V",
      "serie_prot": 1,
      "nr_protocollo": "V01.000001",
      "nr_documento": 1,
      "data_documento": "20260616",
      "data_ult_mod": "20260616 153438"
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Note Operative

- Versione minima: 85000

Ricerca Primanota

Ricerca registrazioni con filtri.

`POST /prima-nota/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtro	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/prima-nota/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "cau_contabile",
      "condizione": "=",
      "valore": "FE"
    },
    {
      "campo": "data_registr",
      "condizione": ">=",
      "valore": "20260101"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "data_registr": "20260616",
      "progressivo": 6,
      "cau_contabile": "FE",
      "rstro_prot_iva": "V",
      "serie_prot": 1,
      "nr_protocollo": "V01.000001",
```

```
      "nr_documento": 1,
      "data_documento": "20260616"
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Versione minima: 85000

Lettura Prima Nota (Chiave Completa)

Recupera una registrazione con chiave completa a 8 segmenti.

`GET /prima-nota/{data}+{progressivo}+{causale}+{registro}+{serie_prot}+{nr_protocollo}+{nr_documento}+{data_documento}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
in_data_registr	string	Sì	path	Data registrazione (YYYYMMDD)
in_progressivo	integer	Sì	path	Progressivo giornaliero
in_cau_contabile	string	Sì	path	Causale contabile (2 caratteri)
in_rstro_prot_iva	string	Sì	path	Registro protocollo IVA
in_serie_prot	integer	Sì	path	Serie protocollo
in_nr_protocollo	integer	Sì	path	Numero protocollo
in_nr_documento	integer	Sì	path	Numero documento
in_data_documento	string	Sì	path	Data documento (YYYYMMDD)

Esempio Request

```
GET /webapi/risorse/prima-nota/20240115+1+FV+FA+1+1+1+20240115 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "data_registr": "20260616",
  "progressivo": 6,
  "cau_contabile": "FE",
  "descrizione": "",
  "rstro_prot_iva": "V",
  "serie_prot": 1,
  "nr_protocollo": 1,
  "nr_documento": 1,
  "data_documento": "20260616",
  "codice_conto": [
    [1, "501.00001"],
    [2, "501.00001"]
  ],
  "importo_riga": [
    [1, 1000.0],
    [2, -1000.0]
  ]
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Registrazione trovata
404	Not Found	Registrazione non trovata

Note Operative

- Chiave a 8 segmenti separati da +
- Versione minima: 85000

Lettura Prima Nota (Chiave Abbreviata)

Recupera una registrazione con chiave abbreviata (data + progressivo).

`GET /prima-nota/{data}+{progressivo}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione

Campo	Tipo	Obbl.	Posizione	Descrizione
in_data_registr	string	Sì	path	Data registrazione (YYYYMMDD)
in_progressivo	integer	Sì	path	Progressivo giornaliero

Esempio Request

```
GET /webapi/risorse/prima-nota/20260616+6 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "data_registr": "20260616",
  "progressivo": 6,
  "cau_contabile": "FE",
  "descrizione": "",
  "rstro_prot_iva": "V",
  "serie_prot": 1,
  "nr_protocollo": 1,
  "nr_documento": 1,
  "data_documento": "20260616",
  "codice_conto": [
    [1, "501.00001"],
    [2, "501.00001"]
  ],
  "importo_riga": [
    [1, 1000.0],
    [2, -1000.0]
  ]
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Registrazione trovata
404	Not Found	Registrazione non trovata

Note Operative

- Chiave semplificata a 2 soli segmenti
- Versione minima: 85000

Modifica Prima Nota

Aggiorna una registrazione esistente. Disponibile sia con chiave completa che abbreviata.

`PUT /prima-nota/{chiave}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
(chiave)	vari	Sì	path	Chiave completa (8 segmenti) o abbreviata (2 segmenti)
righe	array	No	body	Nuove righe contabili
nr_documento	integer	No	body	Nuovo numero documento

Esempio Request

```
PUT /webapi/risorse/prima-nota/20260616+6 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "nr_documento": 2
}
```

Esempio Response

```
{
  "data_registr": "20260616",
  "progressivo": 6,
  "cau_contabile": "FE",
  "nr_documento": 2
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Registrazione aggiornata
400	Bad Request	Dati non validi
404	Not Found	Registrazione non trovata

Note Operative

- Versione minima: 85000

Cancellazione Prima Nota

Elimina una registrazione. Disponibile con entrambi i formati di chiave.

`DELETE /prima-nota/{chiave}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
(chiave)	vari	Sì	path	Chiave completa o abbreviata

Esempio Request

```
DELETE /webapi/risorse/prima-nota/20260616+6 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Registrazione eliminata
404	Not Found	Registrazione non trovata

Note Operative

- Versione minima: 85000

Lista Righe Primanota

Restituisce le righe contabili di tutte le registrazioni (vista trasversale).

`GET /prima-nota/righe`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/prima-nota/righe?max=100 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "data_registr": "20260520",
      "progressivo": 1,
      "cau_contabile": "PG",
      "rstro_prot_iva": " ",
      "serie_prot": 0,
      "nr_documento": 0,
      "data_documento": "20260520",
      "codice_conto": "501.00001",
      "importo_riga": 100.0,
      "descr_riga": "",
      "data_ult_mod": "20260520 165536"
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Righe restituite
400	Bad Request	Coordinate mancanti

Note Operative

- Versione minima: 85000

Ricerca Righe Prima Nota

Ricerca nelle righe contabili con filtri.

`POST /prima-nota/righe/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtro	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/prima-nota/righe/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice_conto",
      "condizione": "=",
      "valore": "501.00001"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "data_registr": "20260520",
      "progressivo": 1,
      "cau_contabile": "PG",
      "codice_conto": "501.00001",
      "importo_riga": 100.0
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi

Note Operative

- Utile per estrarre tutti i movimenti di un conto
- Importi positivi indicano dare, importi negativi indicano avere
- Versione minima: 85000

Approfondimenti Operativi

Struttura della Chiave Composta

L'endpoint è strutturato come quello dei documenti ed è suddiviso in testate e righe. I documenti di prima nota sono referenziabili usando una chiave composta:

Campo	Tipo	Descrizione
`data_registr`	string	Data di registrazione (YYYYMMDD)
`progressivo`	numerico	Progressivo di registrazione (1 – 16.777.214)
`cau_contabile`	string	Causale del documento
`rstro_prot_iva`	string	Registro di protocollo
`serie_prot`	numerico	Serie di protocollo
`nr_protocollo`	numerico	Numero di protocollazione
`nr_documento`	numerico	Numero del documento
`data_documento`	string	Data del documento (YYYYMMDD)

Se il **numero progressivo** è omesso (0), i restanti campi diventano obbligatori.

Esempio: Inserimento con Campi Reali

```
POST /webapi/risorse/prima-nota HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "data_registr": "20230302",
  "descrizione": "Registrazione fattura vendita",
  "cau_contabile": "FE",
  "rstro_prot_iva": "V",
  "serie_prot": 1,
  "nr_protocollo": 5,
  "nr_documento": 0,
  "data_documento": "20201231",
  "codice_conto": [
    [1, "501.00001"],
    [2, "501.00001"]
  ],
  "importo_riga": [
    [1, 6666],
    [2, -6666]
  ]
}
```

Formati di Chiave per GET/PUT/DELETE

Con solo il numero progressivo (chiave abbreviata):

```
GET /webapi/risorse/prima-nota/20210103+5022 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Senza progressivo (chiave completa, tutti i campi obbligatori):

```
GET /webapi/risorse/prima-nota/20210103+0+FR+A+1+5+74344+20201234 HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

Ricerca e Righe

```
POST /webapi/risorse/prima-nota/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout <credenziali_b64>
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "filtri": [
    {
      "campo": "cau_contabile",
      "condizione": "=",
      "valore": "FE"
    }
  ]
}
```

L'endpoint `/prima-nota/righe` supporta solo lettura (GET) e ricerca (POST):

```
GET /webapi/risorse/prima-nota/righe HTTP/1.1

POST /webapi/risorse/prima-nota/righe/ricerca HTTP/1.1
```

Nota sulla Revisione (PUT)

Nel caso di revisione di un documento (PUT), la procedura prima esegue una lettura, poi **cancella** il documento e lo riscrive con le modifiche. È quindi possibile che vengano restituiti errori sulla cancellazione se il documento non può essere cancellato:

> **Errore 6001**: errore gestionale — documento senza record di scadenzario, documento non cancellabile.

Entità: Progressivi Articoli

Consultazione dei progressivi di magazzino (giacenze, ordinato, impegnato, scaricato) di tutti gli articoli dell'azienda. Endpoint di sola lettura che restituisce i dati in formato **matrice sparsa colonnare**.

> ⚠ **Nota Bene:** Questo endpoint utilizza un formato di risposta diverso dagli altri endpoint REST. I dati sono organizzati per campo (colonnare) anziché per record. Vedi la sezione Formato Response per i dettagli.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	`/progressivi-articoli`	Lista progressivi di tutti gli articoli

Lista Progressivi di Tutti gli Articoli

Restituisce i progressivi di magazzino per tutti gli articoli dell'azienda. Ogni record rappresenta una combinazione univoca di articolo + magazzino + ubicazione + lotto.

`GET /progressivi-articoli`

Campi Disponibili

Campo	Tipo	Descrizione
cod_articolo	string	Codice articolo
descr_articolo	string	Descrizione articolo
id_magazzino	integer	ID del magazzino
id_ubicazione	integer	ID dell'ubicazione (presente solo se l'articolo ha ubicazione)
id_lotto	integer	ID del lotto/matricola (presente solo se l'articolo gestisce lotti)
um_principale	string	Unità di misura principale
um_secondaria	string	Unità di misura secondaria (presente solo se definita)
qta_inventario	number	Quantità inventario (giacenza risultante da inventario e movimenti)
qta_inventa_tg	array	Quantità inventario per taglia — formato `[indice, taglia, valore]`
tara_inventa	number	Tara inventario
colli_inventa	integer	Colli inventario

Campo	Tipo	Descrizione
qta_carico	number	Quantità caricata
qta_carico_tg	array	Quantità caricata per taglia
tara_carico	number	Tara carico
colli_carico	integer	Colli caricati
qta_scarico	number	Quantità scaricata
qta_scarico_tg	array	Quantità scaricata per taglia
tara_scarico	number	Tara scarico
colli_scarico	integer	Colli scaricati
qta_ord_for	number	Quantità ordinata a fornitori
qta_ord_for_tg	array	Quantità ordinata a fornitori per taglia
qta_ord_imp	number	Quantità impegnata (ordini clienti)
qta_ord_imp_tg	array	Quantità impegnata per taglia
qta_ord_dimp	number	Quantità da impegnare
qta_ord_dimp_tg	array	Quantità da impegnare per taglia
qta_ord_auto	number	Quantità ordini automatici
qta_ord_auto_tg	array	Quantità ordini automatici per taglia
dt_mod_ult_doc	string	Data/ora ultimo documento movimentato — formato `"YYYYMMDD HHMMSS"`
num_prog_art	integer	Numero totale di progressivi (scalare, non array)

Formato Response

La risposta utilizza un formato **matrice sparsa colonnare**: ogni campo è un array indipendente di tuple `[indice, valore]`. Solo i record con valore diverso da zero/vuoto sono inclusi nell'array del rispettivo campo.

Struttura generale:

```
{
  "nome_campo": [
    [indice_progressivo, valore],
    [indice_progressivo, valore]
  ],
  "campo_taglie": [
    [indice_progressivo, indice_taglia, valore]
  ],
  "num_prog_art": numero_totale
}
```

- **`indice_progressivo`**: identificativo numerico del record progressivo (la stessa combinazione articolo+magazzino+ubicazione+lotto ha lo stesso indice in tutti i campi)
- **`indice_taglia`**: presente solo nei campi con suffisso `_tg`, identifica la posizione della taglia

- **num_prog_art**: scalare (non array), indica il numero totale di record progressivi

ℹ Nota: I campi con valori tutti a zero o vuoti non vengono restituiti nella risposta (array vuoto `[]`). Per ricostruire un record completo, è necessario correlare i dati tramite l'`indice_progressivo` attraverso i vari campi.

Esempio Request

```
GET /webapi/risorse/progressivi-articoli HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "cod_articolo": [
    [1, "2.53.03.905"],
    [4, "2.53.03.905"],
    [5, "2.53.03.905"],
    [10, "8007873000313"],
    [60, "ATT001"]
  ],
  "id_magazzino": [
    [1, 1],
    [4, 1],
    [5, 1],
    [10, 1],
    [60, 1]
  ],
  "qta_inventario": [
    [4, -20.0],
    [5, -80.0]
  ],
  "qta_ord_for": [
    [60, 6.0]
  ],
  "qta_ord_imp": [
    [1, 202.0],
    [10, 10.0]
  ],
  "num_prog_art": 317
}
```


ℹ Nota: Nell'esempio, i record con indice 1, 10 e 60 non compaiono in `qta_inventario` perché il loro valore è zero. Analogamente, solo l'indice 60 ha un valore in `qta_ord_for`.

Codici di Stato

Codice	Descrizione	Causa
200	OK	Progressivi restituiti

Codice	Descrizione	Causa
400	Bad Request	Coordinate gestionale mancanti o non valide

Note Operative

- Richiede coordinate gestionale (Azienda e Anno obbligatori)
- **Non supporta ricerca con filtri** (non è disponibile l'endpoint `/progressivi-articoli/ricerca`)
- Differenza con `/articoli/{codice}/progressivi`: questo endpoint restituisce i progressivi di **tutti** gli articoli in un'unica lista; l'endpoint per singolo articolo restituisce solo i progressivi di un articolo specifico
- Il formato colonnare è ottimizzato per dataset grandi: evita la ripetizione di campi con valore zero
- Per gli articoli a taglie, i campi con suffisso `_tg` contengono il dettaglio per singola taglia
- Il campo `dt_mod_ult_doc` con valore `"00000000 000000"` indica che non è stato mai movimentato con documenti
- Versione minima: 84002

Entità: Referenti

Gestione dei referenti (contatti) associati a clienti, fornitori. L'endpoint è diviso in due sotto-risorse: referenti clienti e referenti fornitori.

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/referenti/clienti`	Inserimento referente cliente
GET	`/referenti/clienti`	Lista referenti clienti
POST	`/referenti/clienti/ricerca`	Ricerca referenti clienti
GET	`/referenti/clienti/{codice_cli_for}+{cod_referente}`	Lettura referente cliente
PUT	`/referenti/clienti/{codice_cli_for}+{cod_referente}`	Modifica referente cliente
DELETE	`/referenti/clienti/{codice_cli_for}+{cod_referente}`	Cancellazione referente cliente

Metodo	URL	Descrizione
POST	`/referenti/fornitori`	Inserimento referente fornitore
GET	`/referenti/fornitori`	Lista referenti fornitori
POST	`/referenti/fornitori/ricerca`	Ricerca referenti fornitori
GET	`/referenti/fornitori/{codice_cli_for}+{cod_referente}`	Lettura referente fornitore
PUT	`/referenti/fornitori/{codice_cli_for}+{cod_referente}`	Modifica referente fornitore
DELETE	`/referenti/fornitori/{codice_cli_for}+{cod_referente}`	Cancellazione referente fornitore

Struttura Campi

I campi sono identici per referenti clienti e referenti fornitori.

Campo	Tipo	Chiave	Descrizione
codice_cli_for	Alfanumerico	Sì	Codice del cliente/fornitore; accetta anche un codice contatto
cod_referente	Numerico	Sì	Codice referente — identificativo progressivo all'interno del conto
cod_posizione	Numerico	No	Codice posizione referente (riferimento alla tabella posizioni)
nome	Alfanumerico	No	Nome referente
nota	Alfanumerico	No	Nota referente
telefono	Alfanumerico	No	Telefono referente
cellulare	Alfanumerico	No	Cellulare referente
fax	Alfanumerico	No	Fax referente
email	Alfanumerico	No	Indirizzo e-mail referente
url	Alfanumerico	No	Indirizzo internet referente
tipo	Alfanumerico	No	Tipo referente

> ⚠ **Nota Bene:** Per azzerare il campo `cellulare` occorre impostarlo con uno spazio (" "). Inviare una stringa vuota non ne cancella il contenuto.

Referenti Clienti

Inserimento Referente Cliente

Crea un nuovo referente associato a un cliente.

`POST /referenti/clienti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice_cli_for	Alfanumerico	Sì	body	Codice cliente (o contatto)
cod_posizione	Numerico	No	body	Codice posizione referente
nome	Alfanumerico	No	body	Nome referente
nota	Alfanumerico	No	body	Nota referente
telefono	Alfanumerico	No	body	Telefono
cellulare	Alfanumerico	No	body	Cellulare
fax	Alfanumerico	No	body	Fax
email	Alfanumerico	No	body	Email
url	Alfanumerico	No	body	Indirizzo internet
tipo	Alfanumerico	No	body	Tipo referente

Esempio Request

```
POST /webapi/risorse/referenti/clienti HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "codice_cli_for": "501.00001",
  "cod_posizione": 1,
  "nome": "Marco Bianchi",
  "nota": "Responsabile acquisti",
  "telefono": "051999888",
```

```
  "cellulare": "+39.348.1234567",
  "email": "m.bianchi@cliente.it"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/referenti/clienti/501.00001+1
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Referente creato con successo
400	Bad Request	Dati non validi o coordinate mancanti
404	Not Found	Cliente non trovato

Lista Referenti Clienti

Restituisce l'elenco di tutti i referenti clienti.

`GET /referenti/clienti`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/referenti/clienti?max=50 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "dati": [
    {
      "codice_cli_for": "501.00001",
      "cod_referente": 1,
      "cod_posizione": 1,
      "nome": "Marco Bianchi",
      "nota": "",
      "telefono": "051999888",
      "cellulare": "",
      "fax": "",
      "email": "m.bianchi@cliente.it",
      "url": "",
```

```
      "tipo": "R"
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita
400	Bad Request	Coordinate mancanti

Ricerca Referenti Clienti

Ricerca referenti clienti con filtri.

`POST /referenti/clienti/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Per la struttura dei filtri vedi Ricerca e Filtri.

Esempio Request

```
POST /webapi/risorse/referenti/clienti/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice_cli_for",
      "condizione": "=",
      "valore": "501.00022"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
```

```
      "codice_cli_for": "501.00022",
      "cod_referente": 5,
      "cod_posizione": 4,
      "nome": "Laura Verdi",
      "nota": "Ufficio tecnico",
      "telefono": "051.123456",
      "cellulare": "+39.348.9876543",
      "fax": "051.654321",
      "email": "l.verdi@cliente.it",
      "url": "www.cliente.it",
      "tipo": "R"
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Lettura Referente Cliente

Recupera un singolo referente cliente.

`GET /referenti/clienti/{codice_cli_for}+{cod_referente}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice_cli_for	Alfanumerico	Sì	path	Codice cliente
cod_referente	Numerico	Sì	path	Codice referente
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/referenti/clienti/501.00001+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "codice_cli_for": "501.00001",
  "cod_referente": 1,
  "cod_posizione": 1,
  "nome": "Marco Bianchi",
  "nota": "",
  "telefono": "051999888",
  "cellulare": "+39.348.1234567",
```

```
  "fax": "",
  "email": "m.bianchi@cliente.it",
  "url": "",
  "tipo": "R"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Referente trovato
404	Not Found	Referente non trovato

Modifica Referente Cliente

Aggiorna un referente cliente.

`PUT /referenti/clienti/{codice_cli_for}+{cod_referente}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice_cli_for	Alfanumerico	Sì	path	Codice cliente
cod_referente	Numerico	Sì	path	Codice referente
cod_posizione	Numerico	No	body	Nuovo codice posizione
nome	Alfanumerico	No	body	Nuovo nome
nota	Alfanumerico	No	body	Nuova nota
telefono	Alfanumerico	No	body	Nuovo telefono
cellulare	Alfanumerico	No	body	Nuovo cellulare
fax	Alfanumerico	No	body	Nuovo fax
email	Alfanumerico	No	body	Nuova email
url	Alfanumerico	No	body	Nuovo URL
tipo	Alfanumerico	No	body	Nuovo tipo

> ℹ **Nota:** I campi `codice_cli_for` e `cod_referente` sono obbligatori nel body della PUT in quanto identificano la risorsa.

Esempio Request

```
PUT /webapi/risorse/referenti/clienti/501.00001+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "codice_cli_for": "501.00001",
  "cod_referente": 1,
  "email": "marco.bianchi@nuovaemail.it",
```

```
  "cellulare": "+39.333.9998877"
}
```

Esempio Response

```
{
  "codice_cli_for": "501.00001",
  "cod_referente": 1,
  "cod_posizione": 1,
  "nome": "Marco Bianchi",
  "nota": "",
  "telefono": "051999888",
  "cellulare": "+39.333.9998877",
  "fax": "",
  "email": "marco.bianchi@nuovaemail.it",
  "url": "",
  "tipo": "R"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Referente aggiornato
400	Bad Request	Dati non validi
404	Not Found	Referente non trovato

Cancellazione Referente Cliente

Elimina un referente cliente.

`DELETE /referenti/clienti/{codice_cli_for}+{cod_referente}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
codice_cli_for	Alfanumerico	Sì	path	Codice cliente
cod_referente	Numerico	Sì	path	Codice referente

Esempio Request

```
DELETE /webapi/risorse/referenti/clienti/501.00001+1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa

Codice	Descrizione	Causa
200	OK	Referente eliminato
404	Not Found	Referente non trovato

Referenti Fornitori

La sotto-risorsa referenti fornitori espone gli stessi endpoint con la medesima struttura campi dei referenti clienti, accessibili sotto il path `/referenti/fornitori`.

Inserimento Referente Fornitore

POST `/referenti/fornitori`

Stessa struttura dell'inserimento referente cliente. Il campo `codice_cli_for` indica il codice fornitore (es. `601.00001`).

Lista Referenti Fornitori

GET `/referenti/fornitori`

Stessa struttura della lista referenti clienti.

Ricerca Referenti Fornitori

POST `/referenti/fornitori/ricerca`

Stessa struttura della ricerca referenti clienti.

Lettura Referente Fornitore

GET `/referenti/fornitori/{codice_cli_for}+{cod_referente}`

Stessa struttura della lettura referente cliente.

Modifica Referente Fornitore

PUT `/referenti/fornitori/{codice_cli_for}+{cod_referente}`

Stessa struttura della modifica referente cliente.

Cancellazione Referente Fornitore

DELETE `/referenti/fornitori/{codice_cli_for}+{cod_referente}`

Stessa struttura della cancellazione referente cliente.

Note Operative

- Chiave composta: `codice_cli_for` + `cod_referente` separati da `+`
- Il campo `codice_cli_for` è di tipo stringa e supporta encoding per caratteri speciali (vedi Paginazione)
- Il campo `cod_posizione` fa riferimento alla tabella posizioni referenti, consultabile tramite l'endpoint `dati-generali/posizione-referenti`
- Richiede coordinate gestionale (Azienda e Anno)
- Versione minima: 81500

Esempi Verificati (Live)

Test eseguiti su installazione **gsuzzi:9014**, azienda **UB1**, anno **2026**.

GET lista referenti clienti:

```
GET /webapi/risorse/referenti/clienti?max=2 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Response:


```
{
  "dati": [
    {
      "codice_cli_for": "501.00001",
      "cod_referente": 1,
      "cod_posizione": 1,
      "nome": "TEST",
      "nota": "",
      "telefono": "",
      "cellulare": "",
      "fax": "",
      "email": "",
      "url": "",
      "tipo": "R"
    },
    {
      "codice_cli_for": "501.00022",
      "cod_referente": 5,
      "cod_posizione": 4,
      "nome": "1",
      "nota": "aaa",
      "telefono": "2",
      "cellulare": "aaa",
      "fax": "3",
      "email": "aaa",
```

```
      "url": "4",
      "tipo": "R"
    }
  ]
}
```

GET singolo referente fornitore:

```
GET /webapi/risorse/referenti/fornitori/601.00001+1 HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Response:

```
{
  "codice_cli_for": "601.00001",
  "cod_referente": 1,
  "cod_posizione": 1,
  "nome": "Shimano",
  "nota": "asd",
  "telefono": "55",
  "cellulare": "",
  "fax": "",
  "email": "",
  "url": "shimano.it",
  "tipo": "R"
}
```

Ricerca referenti clienti con filtro:

```
POST /webapi/risorse/referenti/clienti/ricerca HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice_cli_for",
      "condizione": "=",
      "valore": "501.00022"
    }
  ]
}
```

Response:

```
{
  "dati": [
    {
      "codice_cli_for": "501.00022",
```

```
      "cod_referente": 5,
      "cod_posizione": 4,
      "nome": "1",
      "nota": "aaa",
      "telefono": "2",
      "cellulare": "aaa",
      "fax": "3",
      "email": "aaa",
      "url": "4",
      "tipo": "R"
    }
  ]
}
```

Entità: Scadenzario

Consultazione dello scadenzario clienti e fornitori. Endpoint di sola lettura con supporto ricerca per analisi delle scadenze di pagamento, rate, effetti e provvigioni. In lettura è possibile selezionare il tipo di scadenzario da consultare — a documento, a partite o extracontabile — tramite il parametro `tipo`.

Ogni record rappresenta un **documento** con le relative rate di scadenza. I campi di tipo array (come `tipo_scadenza`, `dt_sca_pg`, `imp_scadenza`) contengono un elemento per ogni rata del documento, con la struttura `[[indice, valore], ...]`.

Riepilogo Endpoint

Metodo	URL	Descrizione
GET	`/scadenzario?tipo=<tipo>`	Lista scadenzario per tipo (D/P/X)
POST	`/scadenzario/ricerca`	Ricerca scadenzario con filtri

Campi

Campo	Tipo	Chiave	Array	Descrizione
codice	Alfanumerico	Sì	—	Codice conto (cliente/fornitore)
dt_registr_doc	Alfanumerico	Sì	—	Data registrazione documento (AAAAMMGG)
prog_primanota	Numerico	Sì	—	Progressivo primanota
anno_primanota	Numerico	Sì	—	Anno archivio di primanota

Campo	Tipo	Chiave	Array	Descrizione
data_documento	Alfanumerico	—	—	Data documento (AAAAMMGG)
serie	Numerico	—	—	Serie documento
numero	Numerico	—	—	Numero documento
numero_esteso	Alfanumerico	—	—	Numero esteso
causale	Alfanumerico	—	—	Causale documento (es. FE, NE, NC)
sottoazienda	Numerico	—	—	Sottoazienda
numero_valuta	Numerico	—	—	Numero valuta
protocollo_iva	Alfanumerico	—	—	Numero protocollo IVA
nota_documento	Alfanumerico	—	—	Nota documento
imp_provvigione	Numerico	—	—	Importo provvigione (valuta di riferimento)
pr_pagata	Numerico	—	—	Provvigione pagata (valuta di riferimento)
imp_pr_vlt_doc	Numerico	—	—	Importo provvigione (valuta documento)
tipo_scadenza	Alfanumerico	—	250	Tipo scadenza per rata (R=Ricevuta, M=Manuale, ecc.)
dt_sca_pg	Alfanumerico	—	250	Data scadenza/pagamento per rata (AAAAMMGG)
imp_scadenza	Numerico	—	250	Importo scadenza per rata (valuta di riferimento)
imp_sca_vlt_doc	Numerico	—	250	Importo scadenza per rata (valuta documento)
stato_pagamento	Alfanumerico	—	250	Stato pagamento per rata
nota_rata	Alfanumerico	—	250	Nota della rata
conto_eff	Alfanumerico	—	250	Conto emissione effetti
nr_prot_effetti	Numerico	—	250	Numero protocollo distinta effetti
dt_sca_emissio	Alfanumeric	—	250	Data scadenza / emissione

Campo	Tipo	Chiave	Array	Descrizione
ne_eff	o			effetti
id_effetto	Alfanumerico	—	250	Identificativo effetto
conto_presso	Alfanumerico	—	250	Conto presso
banca_appoggio	Numerico	—	250	Codice banca d'appoggio
data_sollecito	Alfanumerico	—	250	Data sollecito
data_recupero	Alfanumerico	—	250	Data recupero
data_legale	Alfanumerico	—	250	Data legale
dt_sca_origine	Alfanumerico	—	250	Data scadenza origine
codice_id_gara	Alfanumerico	—	250	Codice identificativo gara
cod_uni_progetto	Alfanumerico	—	250	Codice unico progetto
codice_agente	Alfanumerico	—	15	Codice agente
imp_pr_vlt_rif	Numerico	—	15	Importo provvigione per agente (valuta di riferimento)
pr_pg_vlt_rif	Numerico	—	15	Provvigione pagata per agente (valuta di riferimento)
imp_pr_doc	Numerico	—	15	Importo provvigione per agente (valuta documento)
conto_sps_prof	Alfanumerico	—	27	Conto sospeso professionista
imp_sps_prof	Numerico	—	27	Importo sospeso professionista
utente_ult_mod	Alfanumerico	—	—	Utente ultima modifica
data_ult_mod	Data e ora	—	—	Data/ora ultima modifica (AAAAMMGG HHMMSS)

Nota: I campi di tipo **array** restituiscono i dati come lista di coppie `[indice, valore]`. Ad esempio `"imp_scadenza": [[1, 396.50], [2, 396.50], [3, 396.50]]` indica tre rate da 396,50 €. Gli array vuoti (`[]`) indicano che il campo non ha valori per quel documento.

Lista Scadenzario

Restituisce l'elenco delle scadenze registrate nello scadenzario.

`GET /scadenzario`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
tipo	string	No	query	Tipo di scadenzario da leggere (vedi tabella sottostante)
max	integer	No	query	Numero massimo di record
next	string	No	query	Token di paginazione
fields	string	No	query	Campi da restituire (separati da virgola)

Il parametro `tipo` seleziona quale scadenzario consultare:

Valore	Tipo di scadenzario	Descrizione
D	A documento	Scadenzario a documento
P	A partite	Scadenzario a partite
X	Extracontabile	Scadenzario extracontabile

Esempio Request

```
GET
/webapi/risorse/scadenzario?tipo=D&max=2&fields=codice,data_documento,serie,numero
,causale,tipo_scadenza,dt_sca_pg,imp_scadenza HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "codice": "501.00001",
      "data_documento": "20180219",
      "serie": 1,
      "numero": 2,
      "causale": "FE",
      "tipo_scadenza": [[1, "R"], [2, "R"], [3, "R"], [4, "R"]],
      "dt_sca_pg": [[1, "20180330"], [2, "20180430"], [3, "20180530"], [4,
"20180630"]],
```

```
      "imp_scadenza": [[1, 396.50], [2, 396.50], [3, 396.50], [4, 396.50]]
    },
    {
      "codice": "501.00001",
      "data_documento": "20180401",
      "serie": 1,
      "numero": 23,
      "causale": "FE",
      "tipo_scadenza": [[1, "M"]],
      "dt_sca_pg": [[1, "20180401"]],
      "imp_scadenza": [[1, 305.25]]
    }
  ],
  "next": "token_paginazione"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Scadenzario restituito
400	Bad Request	Coordinate gestionale mancanti o non valide

Note Operative

- Il parametro `tipo` seleziona il tipo di scadenzario da leggere: `D` (a documento), `P` (a partite), `X` (extracontabile)
- Richiede header `Coordinate-Gestionale` con azienda e anno
- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Supporta selezione campi tramite `fields`
- Versione minima: 85000

Ricerca Scadenzario

Ricerca nello scadenzario con filtri avanzati.

`POST /scadenzario/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
tipo	string	No	query	Tipo di scadenzario su cui cercare: `D` (a documento), `P` (a partite), `X` (extracontabile)
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione

Campo	Tipo	Obbl.	Posizione	Descrizione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Il parametro `tipo` accetta gli stessi valori descritti per la Lista Scadenzario e seleziona lo scadenzario su cui applicare i filtri. Per la sintassi completa dei filtri, vedi Ricerca e Filtri.

Esempio Request

```
POST
/webapi/risorse/scadenzario/ricerca?tipo=D&max=2&fields=codice,data_documento,seri
e,numero,causale,tipo_scadenza,dt_sca_pg,imp_scadenza,stato_pagamento HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "codice",
      "condizione": "=",
      "valore": "501.00001"
    }
  ]
}
```

Esempio Response

```
{
  "dati": [
    {
      "codice": "501.00001",
      "data_documento": "20180219",
      "serie": 1,
      "numero": 2,
      "causale": "FE",
      "tipo_scadenza": [[1, "R"], [2, "R"], [3, "R"], [4, "R"]],
      "dt_sca_pg": [[1, "20180330"], [2, "20180430"], [3, "20180530"], [4,
"20180630"]],
      "imp_scadenza": [[1, 396.50], [2, 396.50], [3, 396.50], [4, 396.50]],
      "stato_pagamento": []
    },
    {
      "codice": "501.00001",
      "data_documento": "20180401",
      "serie": 1,
      "numero": 23,
      "causale": "FE",
      "tipo_scadenza": [[1, "M"]],
      "dt_sca_pg": [[1, "20180401"]],
```

```
      "imp_scadenza": [[1, 305.25]],
      "stato_pagamento": []
    }
  ],
  "next": "token_paginazione"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Il parametro `tipo` seleziona il tipo di scadenzario su cui effettuare la ricerca: `D` (a documento), `P` (a partite), `X` (extracontabile)
- Utile per estrarre scadenze di un singolo conto, per causale, o per intervallo di date registrazione
- I filtri possono operare su qualsiasi campo della risorsa (es. `codice`, `causale`, `data_documento`)
- Per filtrare su campi array (es. `dt_sca_pg`) utilizzare la sintassi con indice: `"indice1": 1`
- Versione minima: 85000

Entità: Ubicazioni

Gestione delle ubicazioni di magazzino. L'entità delle ubicazioni viene gestita in modo del tutto analogo alle altre risorse, con supporto CRUD completo per definire le posizioni fisiche degli articoli all'interno dei magazzini.

Per tutti i metodi (GET, POST, PUT, DELETE) l'entità *ubicazioni* è raggiungibile al seguente path:

`https://base_address/webapi/risorse/ubicazioni`

Riepilogo Endpoint

Metodo	URL	Descrizione
POST	`/ubicazioni`	Inserimento ubicazione
GET	`/ubicazioni`	Lista ubicazioni
POST	`/ubicazioni/ricerca`	Ricerca ubicazioni con filtri
GET	`/ubicazioni/{id}`	Lettura ubicazione
PUT	`/ubicazioni/{id}`	Modifica ubicazione
DELETE	`/ubicazioni/{id}`	Cancellazione ubicazione

Struttura Campi

Campo	Tipo	Chiave	Descrizione
id	Numerico	Sì	Identificatore ubicazione
id_magazzino	Numerico		Codice magazzino
codice	Alfanumerico		Codice ubicazione
codice_alt	Alfanumerico		Codice alternativo ubicazione
descrizione	Alfanumerico		Descrizione ubicazione
tipo	Alfanumerico		Tipo ubicazione (S = Standard, E = Servizio)
stato	Alfanumerico		Stato (T = Attiva, A = Annullata logicamente, B = Bloccata, U = Bloccata in uscita, I = Bloccata in ingresso)
area	Alfanumerico		Area ubicazione (S = Stock, C = Carico, P = Picking, A = Scarico)
quantita	Numerico con virgola		Quantità massima
larghezza	Numerico con virgola		Larghezza
altezza	Numerico con virgola		Altezza
profondita	Numerico con virgola		Profondità
peso	Numerico con virgola		Peso in Kg
temperatura_min	Alfanumerico		Temperatura minima
temperatura_max	Alfanumerico		Temperatura massima
latitudine	Numerico con virgola		Latitudine
longitudine	Numerico con virgola		Longitudine
elevazione	Numerico con virgola		Elevazione
alt_dal_suolo	Numerico con virgola		Altezza dal suolo
stru_art_escl	Alfanumerico		Struttura articoli esclusiva
cod_art_escl	Alfanumerico		Codice articolo esclusivo
catsta_escl	Alfanumerico		Categoria statistica esclusiva

Campo	Tipo	Chiave	Descrizione
grp_merc_escl	Alfanumerico		Gruppo merceologico esclusivo
natura_escl	Alfanumerico		Natura esclusiva
cod_for_escl	Alfanumerico		Codice fornitore esclusivo
cod_percorso	Numerico		Codice percorso (per creare un percorso all'interno del magazzino)
utente_ult_mod	Alfanumerico		Utente ultima modifica
data_ult_mod	Data e ora		Data e ora ultima modifica (formato `AAAAMMGG HHMMSS`)

Inserimento Ubicazione

Crea una nuova ubicazione di magazzino.

`POST /ubicazioni`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id_magazzino	integer	Sì	body	Codice magazzino
codice	string	Sì	body	Codice ubicazione
descrizione	string	Sì	body	Descrizione ubicazione
tipo	string	No	body	Tipo ubicazione (`S` = Standard, `E` = Servizio)
stato	string	No	body	Stato ubicazione (`T`, `A`, `B`, `U`, `I`)
area	string	No	body	Area (`S`, `C`, `P`, `A`)
codice_alt	string	No	body	Codice alternativo
quantita	decimal	No	body	Quantità massima
larghezza	decimal	No	body	Larghezza
altezza	decimal	No	body	Altezza
profondita	decimal	No	body	Profondità
peso	decimal	No	body	Peso in Kg
cod_percorso	integer	No	body	Codice percorso

Esempio Request

```
POST /webapi/risorse/ubicazioni HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
```

```
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "id_magazzino": 27,
  "codice": "C02S01R1",
  "descrizione": "Corsia C02 Scaffale S01 Ripiano R1",
  "tipo": "S",
  "stato": "T",
  "area": "S"
}
```

Esempio Response

```
HTTP/1.1 201 Created
Location: /webapi/risorse/ubicazioni/100
```

Codici di Stato

Codice	Descrizione	Causa
201	Created	Ubicazione creata con successo
400	Bad Request	Dati non validi o campi obbligatori mancanti

Note Operative

- Richiede coordinate gestionale (Azienda e Anno)
- Versione minima: 82001
- Il campo `id` viene assegnato automaticamente dal sistema

Lista Ubicazioni

Restituisce l'elenco delle ubicazioni. Supporta paginazione e selezione campi.

`GET /ubicazioni`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record da restituire
next	string	No	query	Token di paginazione ottenuto dalla response precedente
fields	string	No	query	Campi da restituire (separati da virgola)

Esempio Request

```
GET /webapi/risorse/ubicazioni?max=3&fields=id,codice,descrizione,stato HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response


```
{
  "dati": [
    {
      "id": 1,
      "codice": "C01S01R1",
      "descrizione": "TEST",
      "stato": "U"
    },
    {
      "id": 2,
      "codice": "C01S01R2",
      "descrizione": "Corsia C01 Scaffale S01 Ripiano R2",
      "stato": "A"
    },
    {
      "id": 3,
      "codice": "C01S01R3",
      "descrizione": "Corsia C01 Scaffale S01 Ripiano R3",
      "stato": "T"
    }
  ],
  "next": "57415f33656433..."
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Lista restituita con successo
400	Bad Request	Coordinate gestionale mancanti

Note Operative

- Supporta paginazione tramite `max` e `next` (vedi Paginazione)
- Supporta selezione campi tramite `fields`
- Versione minima: 82001

Ricerca Ubicazioni

Ricerca ubicazioni con filtri (vedi Ricerca e Filtri).

`POST /ubicazioni/ricerca`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
max	integer	No	query	Numero massimo di record
next	string	No	query	Token paginazione
fields	string	No	query	Campi da restituire
filtri	array	Sì	body	Array di oggetti filtro

Esempio Request

```
POST /webapi/risorse/ubicazioni/ricerca HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "filtri": [
    {
      "campo": "stato",
      "condizione": "=",
      "valore": "T"
    }
  ]
}
```

Esempio Response


```
{
  "dati": [
    {
      "id": 3,
      "id_magazzino": 27,
      "codice_alt": "",
      "tipo": "S",
      "descrizione": "Corsia C01 Scaffale S01 Ripiano R3",
      "codice": "C01S01R3",
      "stato": "T",
      "area": "S",
      "quantita": 1.0,
      "larghezza": 0.0,
      "altezza": 0.0,
      "profondita": 0.0,
      "peso": 1.0,
      "temperatura_min": "   0,0",
      "temperatura_max": "   0,0",
      "latitudine": 0.0,
      "longitudine": 0.0,
```

```
      "elevazione": 0.0,
      "alt_dal_suolo": 0.0,
      "stru_art_escl": "",
      "cod_art_escl": "",
      "catsta_escl": "",
      "grp_merc_escl": "",
      "natura_escl": "",
      "cod_for_escl": "",
      "cod_percorso": 0,
      "utente_ult_mod": "0",
      "data_ult_mod": "00000000 000000"
    }
  ]
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ricerca completata
400	Bad Request	Filtri non validi o coordinate mancanti

Note Operative

- Versione minima: 82001
- Supporta tutte le condizioni standard: `=`, `<`, `<=`, `>`, `>=`, `<>`, `contiene`, `inizia_per`

Lettura Ubicazione

Recupera i dati di una specifica ubicazione tramite il suo identificatore.

`GET /ubicazioni/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	Identificatore ubicazione
fields	string	No	query	Campi da restituire

Esempio Request

```
GET /webapi/risorse/ubicazioni/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
{
  "id": 1,
```

```
  "id_magazzino": 27,
  "codice_alt": "123597135A00",
  "tipo": "S",
  "descrizione": "TEST",
  "codice": "C01S01R1",
  "stato": "U",
  "area": "S",
  "quantita": 100.0,
  "larghezza": 12.0,
  "altezza": 13.0,
  "profondita": 5.0,
  "peso": 66.0,
  "temperatura_min": "  -5,0",
  "temperatura_max": "  69,0",
  "latitudine": 43.0,
  "longitudine": 12.0,
  "elevazione": 150.0,
  "alt_dal_suolo": 0.5,
  "stru_art_escl": "2",
  "cod_art_escl": "2.53.03.905",
  "catsta_escl": "200",
  "grp_merc_escl": "ABBIGL",
  "natura_escl": "11",
  "cod_for_escl": "601.00011",
  "cod_percorso": 0,
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20230201 114457"
}
```


Codici di Stato

Codice	Descrizione	Causa
200	OK	Ubicazione trovata
404	Not Found	Ubicazione non trovata

Note Operative

- Versione minima: 82001
- Supporta parametri aggiuntivi in query string per la gestione dei progressivi (vedi sezione Uso Avanzato - Filtri Speciali in Query String)

Modifica Ubicazione

Aggiorna i dati di una ubicazione esistente.

`PUT /ubicazioni/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	Identificatore ubicazione
id	integer	Sì	body	Identificatore ubicazione (deve corrispondere al path)
descrizione	string	No	body	Nuova descrizione
stato	string	No	body	Nuovo stato (`T`, `A`, `B`, `U`, `I`)
area	string	No	body	Nuova area (`S`, `C`, `P`, `A`)
quantita	decimal	No	body	Nuova quantità massima
(altri campi)		No	body	Qualsiasi campo modificabile dalla struttura

Esempio Request

```
PUT /webapi/risorse/ubicazioni/1 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "id": 1,
  "descrizione": "Corsia C01 Scaffale S01 Ripiano R1 (materiale fragile)",
  "stato": "T",
  "peso": 50.0
}
```

Esempio Response


```
{
  "id": 1,
  "id_magazzino": 27,
  "codice_alt": "123597135A00",
  "tipo": "S",
  "descrizione": "Corsia C01 Scaffale S01 Ripiano R1 (materiale fragile)",
  "codice": "C01S01R1",
  "stato": "T",
  "area": "S",
  "quantita": 100.0,
  "larghezza": 12.0,
  "altezza": 13.0,
  "profondita": 5.0,
  "peso": 50.0,
  "temperatura_min": "  -5,0",
  "temperatura_max": "  69,0",
  "latitudine": 43.0,
  "longitudine": 12.0,
  "elevazione": 150.0,
```

```
  "alt_dal_suolo": 0.5,
  "stru_art_escl": "2",
  "cod_art_escl": "2.53.03.905",
  "catsta_escl": "200",
  "grp_merc_escl": "ABBIGL",
  "natura_escl": "11",
  "cod_for_escl": "601.00011",
  "cod_percorso": 0,
  "utente_ult_mod": "626892910",
  "data_ult_mod": "20260617 144500"
}
```

Codici di Stato

Codice	Descrizione	Causa
200	OK	Ubicazione aggiornata
400	Bad Request	Dati non validi
404	Not Found	Ubicazione non trovata

Note Operative

- Versione minima: 82001
- Il campo `id` è obbligatorio nel body della PUT

Cancellazione Ubicazione

Elimina una ubicazione.

`DELETE /ubicazioni/{id}`

Parametri

Campo	Tipo	Obbl.	Posizione	Descrizione
id	integer	Sì	path	Identificatore ubicazione

Esempio Request

```
DELETE /webapi/risorse/ubicazioni/100 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Esempio Response

```
HTTP/1.1 200 OK
```

Codici di Stato

Codice	Descrizione	Causa

Codice	Descrizione	Causa
200	OK	Ubicazione eliminata
404	Not Found	Ubicazione non trovata
409	Conflict	Ubicazione in uso (articoli associati)

Note Operative

- Versione minima: 82001

Uso Avanzato - Filtri Speciali in Query String

Per la GET della singola ubicazione, oltre a specificare l'`id` (obbligatorio), sono previsti dei parametri aggiuntivi facoltativi da inserire in query string per la gestione dei **progressivi degli articoli** presenti in una ubicazione.

Parametro cod_art_progr

Questo parametro controlla la restituzione dei progressivi articolo:

Valore	Comportamento
Parametro assente	Restituisce solo i campi dell'anagrafica ubicazione, senza progressivi
Parametro presente senza valore	Restituisce tutti i progressivi di tutti gli articoli in quella ubicazione
Parametro con codice articolo	Restituisce solo i progressivi dell'articolo specificato in quella ubicazione

> ⚠ **Nota Bene:** Nel caso di caratteri speciali nel codice articolo (compresi slash e backslash), devono essere passati con URL-encoding.

Parametro dettlotti

Vincolato alla presenza di `cod_art_progr`, permette di includere o escludere i dettagli sui lotti:

Valore	Comportamento
`true`	Include i dettagli dei lotti
`false`	Esclude i dettagli dei lotti

Esempi di Lettura con Progressivi

Lettura di tutti i progressivi di una ubicazione:

```
GET /webapi/risorse/ubicazioni/76?cod_art_progr HTTP/1.1
Host: server:porta
```

```
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Lettura dei progressivi di un articolo specifico in una ubicazione:

```
GET /webapi/risorse/ubicazioni/76?cod_art_progr=FUELEX97 HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Lettura dei progressivi con dettagli dei lotti:

```
GET /webapi/risorse/ubicazioni/76?cod_art_progr&dettlotti=true HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

Lettura dei progressivi di un articolo con lotti:

```
GET /webapi/risorse/ubicazioni/76?cod_art_progr=FUELEX97&dettlotti=true HTTP/1.1
Host: server:porta
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Coordinate-Gestionale: Azienda=UB1 Anno=2026
```

> **Nota:** Per ottenere i progressivi di **tutte** le ubicazioni (non solo una singola) è disponibile il servizio dedicato `get_prog_ubicazioni` sul canale Servizi (vedi Servizio: Progressivi di Tutte le Ubicazioni).

Canale Servizi

Introduzione

In tutti i casi in cui l'operazione che si vuole effettuare sul gestionale non identifichi una risorsa ben precisa, ma piuttosto richieda l'esecuzione di una operazione o un insieme di operazioni che dovranno essere svolte da Mexal/Passcom, l'approccio RESTful non è più sufficiente e bisogna adottare un paradigma differente.

WebAPI mette a disposizione un ulteriore canale per la gestione di questi casi identificato con il nome **servizi**.

Differenza tra Endpoint REST e Canale Servizi

Caratteristica	Endpoint REST	Canale Servizi
Paradigma	RESTful (risorse CRUD)	Operazioni/procedure
Metodi HTTP	GET, POST, PUT, DELETE	Solo POST
URL	`/webapi/<entità>`	`/webapi/servizi`
Identificazione	Path + parametri	Campo `cmd` nel body
Uso tipico	Lettura/scrittura singola risorsa	Elaborazioni complesse, calcoli, servizi trasversali

Formato della Request

Le request verso il canale servizi sono **sempre di tipo POST** all'indirizzo:

```
POST https://base_address/webapi/servizi
```

Struttura del Body

Il body della request è un oggetto JSON con i seguenti campi principali:

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	Nome del comando/servizio da eseguire
dati	Object	Dipende	Oggetto contenente i parametri specifici del servizio
next	String	No	Token di paginazione (da valorizzare con il valore ricevuto nella response

Campo	Tipo	Obbl.	Descrizione
			precedente)

Esempio generico

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "nome_comando",
  "dati": {
    "parametro1": "valore1",
    "parametro2": "valore2"
  }
}
```

Paginazione

Alcuni servizi che restituiscono una grossa mole di dati possono paginare il risultato. In questo caso, in fondo al JSON della risposta sarà presente un campo `next` con il valore del record successivo da cui partire nella successiva richiesta.

Il campo `next` dovrà essere inserito nella request successiva **allo stesso livello del campo `cmd`**.

```
{
  "cmd": "nome_comando",
  "next": "valore_ricevuto_dalla_response",
  "dati": {
    ...
  }
}
```

Header obbligatori

Come per tutti i servizi WebAPI, sono richiesti:

- **Authorization**: header di autenticazione Passepartout
- **Coordinate-Gestionale**: per identificare azienda e anno di esercizio

Elenco dei Servizi

I servizi disponibili sono documentati nei capitoli successivi, raggruppati per area funzionale:

- Collage Server Remoto
- Sviluppo Distinta Base
- Avanzamento di Produzione
- Spezza Riga Bolle di Lavorazione
- Condizioni Documento
- Calcolo Esposizione
- Progressivi di Tutte le Ubicazioni
- Totali Documento e Totali Riga
- Località Italiane (CAP e Zone)
- Conteggio Numero Operazioni Contabili
- Pratiche Dichiarativi
- Deleghe Dichiarativi
- Registro Cancellati
- Lettura Dichiarazioni IVA Annuali
- Lettura Comunicazioni LIPE
- Stampa PDF di Documenti
- Inserimento Nuove Righe nei Documenti
- Lista Allegati Docuvision
- Lettura Allegati Docuvision
- Associazioni Docuvision
- Lista Strutture MyDB
- Proprietà Numeratori
- Autenticazione MyDB
- Calendari di Produzione
- Multilotto
- Modifica IBAN e Tipo Invio nelle Deleghe
- Inserimento e Aggiornamento IBAN
- Storia Ordine
- Upload Files

Servizio: Collage Server Remoto

Questo comando permette l'esecuzione di un Collage Server Remoto contenuto in una Passapp, esattamente come avviene con WebShaker (di fatto WebShaker è a tutti gli effetti una "costola" di WebAPI).

È necessario specificare i dati necessari all'individuazione dell'etichetta Collage da eseguire all'interno della Passapp indicando `codice_app`, `nome_collage` e `etichetta_collage`.

Comando: `esec_collage_server_remoto`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"esec_collage_server_remoto"`
codice_app	String	Sì	Codice della Passapp contenente il Collage
nome_collage	String	Sì	Nome del file Collage all'interno della Passapp
etichetta_collage	String	Sì	Etichetta Collage da eseguire
dati	Object	No	Eventuali dati da inviare al Collage

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "esec_collage_server_remoto",
  "codice_app": "924148WEBSHAKER",
  "nome_collage": "colws",
  "etichetta_collage": "WAPI",
  "dati": {}
}
```

Esempio Response

```
{
  <risultato dell'elaborazione>
}
```

Note Operative

- Il risultato dell'elaborazione dipende dalla logica implementata nel Collage Server Remoto.
- Il servizio replica il funzionamento di WebShaker per l'esecuzione di logiche custom lato server.

Servizio: Sviluppo Distinta Base

Servizio per lo sviluppo della distinta base di un articolo.

Nel caso di articolo non a taglie è necessario specificare la quantità come primo elemento dell'array `quantita_taglie`.

Comando: `sviluppo_distinta_base`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"sviluppo_distinta_base"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.codice_articolo	String	Sì	Codice dell'articolo di cui sviluppare la distinta
dati.codice_art_padre	String	No	Codice articolo padre
dati.codice_cliente	String	No	Codice del cliente
dati.data_documento	String	No	Data del documento
dati.codice	Number	No	Numero bolla
dati.cod_sottobolla	Number	No	Numero sottobolla
dati.nr_rif_pf	Number	No	Numero riferimento prodotto finito
dati.id_riga_oc	Number	No	Id riga ordine
dati.quantita_taglie	Array	No	Array di coppie [indice_taglia, quantità] (da 1 a 32)

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "sviluppo_distinta_base",
  "dati": {
    "codice_articolo": "<stringa_codice_articolo>",
    "codice_art_padre": "<stringa_codice_art_padre>",
    "codice_cliente": "<stringa_codice_cliente>",
    "data_documento": "<stringa_data_documento>",
    "codice": <numero_bolla>,
    "cod_sottobolla": <numero_sottobolla>,
    "nr_rif_pf": <numero_rif_prodotto_finito>,
```

```
    "id_riga_oc": <id_riga_ordine>,
    "quantita_taglie": [
      [1, n],
      ...
      [32, n]
    ]
  }
}
```

Note Operative

- Per articoli non a taglie, specificare la quantità come primo elemento dell'array `quantita_taglie` (es. `[[1, 5]]`).
- L'array `quantita_taglie` accetta coppie `[indice, quantità]` con indice da 1 a 32.

Servizio: Avanzamento di Produzione

Servizio per avanzare la produzione creando automaticamente i relativi documenti di carico (CL) e scarico (SL).

Comando: `avanzamento_produzione`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"avanzamento_produzione"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.data_documento	String	Sì	Data documento nel formato AAAAMMGG
dati.codice	Number	Sì	Numero bolla
dati.cod_sottobolla	Number	Sì	Numero sottobolla
dati.nr_rif_pf	Number	Sì	Numero riferimento prodotto finito
dati.tp_operazione	String	Sì	Tipo operazione: "S" (Scarico) o "C" (Carico)
dati.nr_fase	Number	Sì	Numero fase
dati.id_lotto	Array	No	Array id lotti
dati.qta_lotto	Array	No	Array quantità lotti
dati.qta_agg_carico	Number	No	Quantità aggiuntiva carico
dati.nr_colli_lotto	Array	No	Array numero colli lotto

Campo	Tipo	Obbl.	Descrizione
dati.qta_tg_lotto	Array	No	Array quantità taglie lotto
dati.nr_decimali_qta	Array	No	Array numero decimali quantità
dati.id_lotto_sa	Array	No	Array id lotti semilavorati/accessori
dati.qta_lotto_sa	Array	No	Array quantità lotti semilavorati/accessori
dati.nr_colli_lo_sa	Array	No	Array numero colli lotti semilavorati/accessori
dati.qta_tg_lo_sa	Array	No	Array quantità taglie lotti semilavorati/accessori
dati.nr_dev_qta_sa	Array	No	Array numero deviazioni quantità semilavorati/accessori
dati.nr_righe_lo_sa	Array	No	Array numero righe lotti semilavorati/accessori
dati.nr_righe_lotto	Number	No	Numero righe lotto

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "avanzamento_produzione",
  "dati": {
    "data_documento": "20220907",
    "codice": <numero_bolla>,
    "cod_sottobolla": <numero_sottobolla>,
    "nr_rif_pf": <numero_rif_prodotto_finito>,
    "tp_operazione": "S",
    "nr_fase": <numero_fase>,
    "id_lotto": [],
    "qta_lotto": [],
    "qta_agg_carico": 1,
    "nr_colli_lotto": [],
    "qta_tg_lotto": [],
    "nr_decimali_qta": [],
    "id_lotto_sa": [],
    "qta_lotto_sa": [],
    "nr_colli_lo_sa": [],
    "qta_tg_lo_sa": [],
    "nr_dev_qta_sa": [],
    "nr_righe_lo_sa": [],
```

```
    "nr_righe_lotto": 0
  }
}
```

Note Operative

- Il campo `tp_operazione` accetta “S” per Scarico o “C” per Carico.
- Il servizio crea automaticamente i documenti CL (carico) e SL (scarico) corrispondenti.

Servizio: Spezza Riga Bolle di Lavorazione

Funzione che permette di spezzare la quantità di riga del prodotto finito, suddividendo in due righe le quantità totali. È utilizzata per effettuare gli avanzamenti di produzione parziali.

Come per lo sviluppo della distinta base, nel caso di articolo non a taglie è necessario specificare la quantità come primo elemento dell’array `qta`.

Comando: `spezzariga_bl`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"spezzariga_bl"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.codice	Number	Sì	Numero bolla
dati.cod_sottobolla	Number	Sì	Numero sottobolla
dati.codice_pf	String	Sì	Codice articolo prodotto finito
dati.nr_rif_pf	Number	Sì	Numero riferimento prodotto finito
dati.qta	Array	Sì	Array di coppie [indice_taglia, quantità]
dati.stato_riga_orig	String	Sì	Stato della riga di origine
dati.stato_riga_dest	String	Sì	Stato della riga di destinazione
dati.solo_bl	String	No	“S” o “N” — indica se operare solo sulla bolla di lavorazione

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
```

```
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "spezzariga_bl",
  "dati": {
    "codice": <numero_bolla>,
    "cod_sottobolla": <numero_sottobolla>,
    "codice_pf": "<codice_articolo>",
    "nr_rif_pf": <numero_rif_prodotto_finito>,
    "qta": [
      [1, 2]
    ],
    "stato_riga_orig": "<stato_riga_origine>",
    "stato_riga_dest": "<stato_riga_destinazione>",
    "solo_bl": "S"
  }
}
```

Note Operative

- Per articoli non a taglie, la quantità va specificata come primo elemento dell'array `qta` (es. `[[1, 2]]`).
- Il servizio suddivide la riga del prodotto finito in due righe con le quantità specificate, consentendo avanzamenti parziali della produzione.

Servizio: Condizioni Documento

Servizio che permette di ottenere, se presenti, prezzo, sconto e provvigione in base ai parametri impostati.

Comando: `condizioni_documento`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"condizioni_documento"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.tipo	Number	Sì	1: prezzo, sconto e provvigione; 2: sconto e provvigione; 3: solo provvigione
dati.cod_conto	String	Sì	Conto del cliente

Campo	Tipo	Obbl.	Descrizione
dati.codice_articolo	String	Sì	Codice articolo
dati.data_documento	String	Sì	Data documento formato AAAAMMGG
dati.sigla_documento	String	Sì	Sigla documento (es. “OC”)
dati.prezzo	Number	No	Se impostato e tipo=1/2, viene utilizzato come base per calcolo sconti e provvigioni
dati.quantita	Number	No	Valore per lo scaglione delle quantità
dati.coefficiente	Number	No	Se zero lavora in u.m. primaria dell’articolo, se valorizzato in u.m. secondaria
dati.valuta	Number	No	Codice valuta
dati.id_listino	Number	No	ID listino (se zero si usa il listino del cliente)
dati.sconto	String	No	Se impostato e tipo=3, viene utilizzato come base per calcolo provvigioni
dati.id_cat_sconto	Number	No	Se impostata, viene utilizzata al posto di quella del cliente
dati.id_categoria_pr	Number	No	Se impostata, viene utilizzata al posto di quella del cliente
dati.extra	String	No	Se impostato, viene utilizzato per il calcolo
dati.cod_agente	String	No	Se impostato, viene utilizzato questo agente per il calcolo

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "condizioni_documento",
  "dati": {
    "tipo": 1,
    "cod_conto": "501.00085",
    "codice_articolo": "NIPDSWERG001",
    "data_documento": "20211231",
    "sigla_documento": "OC",
```

```
    "prezzo": 0,
    "quantita": 23,
    "coefficiente": 0,
    "valuta": 1,
    "id_listino": 1,
    "sconto": "",
    "id_cat_sconto": 0,
    "id_categoria_pr": 0,
    "extra": "",
    "cod_agente": "601.00001"
  }
}
```

Esempio Response

```
{
  "prezzo": 0.0,
  "sconto": "",
  "provvigione": 0.0,
  "tp_provvigione": [],
  "formula_pr": [],
  "codice_agente": [],
  "quota_pr": [],
  "cond_agente": [],
  "modalita_pr": [],
  "base_impon_calc": [],
  "utente_ult_mod": "",
  "data_ult_mod": ""
}
```

Campi della Response

Campo	Tipo	Descrizione
prezzo	Number	Prezzo calcolato
sconto	String	Sconto calcolato
provvigione	Number	Provvigione calcolata
tp_provvigione	Array	Tipo provvigione (% o T=totale)
formula_pr	Array	Formula provvigione
codice_agente	Array	Codici agente
quota_pr	Array	Quota provvigione ripartita
cond_agente	Array	Condizione agente
modalita_pr	Array	Modalità provvigione
base_impon_calc	Array	Base imponibile di calcolo provvigione
utente_ult_mod	String	Utente ultima modifica
data_ult_mod	String	Data ultima modifica

Servizio: Calcolo Esposizione

Servizio che permette, dato un codice conto e una data di riferimento, di calcolare i dati relativi all'esposizione di un cliente/fornitore.

Comando: `calcolo_esposizione`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"calcolo_esposizione"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_codice_conto	String	Sì	Codice conto del cliente/fornitore
dati.in_alla_data	String	Sì	Data di riferimento formato AAAAMMGG

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "calcolo_esposizione",
  "dati": {
    "in_codice_conto": "501.00022",
    "in_alla_data": "20230101"
  }
}
```

Esempio Response (base)

```
{
  "saldo_cli_for": 51205.6,
  "valore_fido": 0.0,
  "val_ord_bolle": 170212.91,
  "val_ord_netto": 108925.91,
  "val_ord_iva": 132549.65,
  "val_bolle_netto": 61287.0,
  "val_bolle_iva": 74770.14,
  "esposizione": 0.0,
  "dt_esposizione": "20230101",
```

```
  "valore_rischio": 258525.39,
  "fuori_fido": 0.0
}
```

Esempio Response (completa con dati aggiuntivi)

```
{
  "saldo_cli_for": 5129.65,
  "valore_fido": 234.0,
  "val_ord_bolle": 23302500.03,
  "val_ord_netto": 12584.26,
  "val_ord_iva": 15355.99,
  "val_bolle_netto": 23289915.77,
  "val_bolle_iva": 28413697.21,
  "esposizione": 0.0,
  "dt_esposizione": "20230101",
  "valore_rischio": 28434182.85,
  "fuori_fido": 28433948.85,
  "rating_assegnato": "Incerto",
  "numero_insoluti": 0,
  "valore_insoluti": 0.0,
  "fatturato_net_ap": 2500.0,
  "fatturato_ld_ap": 2500.0,
  "fatturato_net_ac": 0.65,
  "fatturato_ld_ac": 0.65,
  "dt_doc_ult_gna": "20230828",
  "doc_ult_consegna": "BC 1/62 "
}
```

Campi della Response

Campo	Tipo	Descrizione
saldo_cli_for	Number	Saldo cliente/fornitore
valore_fido	Number	Valore del fido
val_ord_bolle	Number	Valore ordini e bolle
val_ord_netto	Number	Valore ordini netto
val_ord_iva	Number	Valore ordini con IVA
val_bolle_netto	Number	Valore bolle netto
val_bolle_iva	Number	Valore bolle con IVA
esposizione	Number	Esposizione calcolata
dt_esposizione	String	Data di esposizione
valore_rischio	Number	Valore del rischio
fuori_fido	Number	Importo fuori fido
rating_assegnato	String	Rating assegnato al cliente
numero_insoluti	Number	Numero insoluti

Campo	Tipo	Descrizione
valore_insoluti	Number	Valore insoluti
fatturato_net_ap	Number	Fatturato netto anno precedente
fatturato_ld_ap	Number	Fatturato lordo anno precedente
fatturato_net_ac	Number	Fatturato netto anno corrente
fatturato_ld_ac	Number	Fatturato lordo anno corrente
dt_doc_ult_gna	String	Data documento ultima consegna
doc_ult_consegna	String	Documento ultima consegna

Esempio (Live)

La seguente richiesta è stata eseguita su installazione reale (Azienda UB1, Anno 2026, cliente 501.00001):

Request

```
POST /webapi/servizi HTTP/1.1
Host: gsuzzi:9014
Authorization: Passepartout d2ViYXBpOnc=
Content-Type: application/json
Coordinate-Gestionale: Azienda=UB1 Anno=2026

{
  "cmd": "calcolo_esposizione",
  "dati": {
    "in_codice_conto": "501.00001",
    "in_alla_data": "20260101"
  }
}
```

Response (200 OK)

```
{
  "saldo_cli_for": -3476.42,
  "valore_fido": 0.0,
  "val_ord_bolle": 112794.79,
  "val_ord_netto": 112794.79,
  "val_ord_iva": 137568.03,
  "val_bolle_netto": 0.0,
  "val_bolle_iva": 0.0,
  "esposizione": 0.0,
  "dt_esposizione": "20260101",
  "valore_rischio": 134091.61,
  "fuori_fido": 0.0,
  "rating_assegnato": "",
  "numero_insoluti": 0,
  "valore_insoluti": 0.0,
```

```
  "fatturato_net_ap": 122.0,
  "fatturato_ld_ap": 122.0,
  "fatturato_net_ac": 0.0,
  "fatturato_ld_ac": 0.0,
  "dt_doc_ult_gna": "20250929",
  "doc_ult_consegna": "BC  1/22    "
}
```

Nota: Il campo `saldo_cli_for` negativo indica un credito del cliente nei confronti dell'azienda. Il campo `doc_ult_consegna` indica l'ultimo documento di consegna (bolla BC serie 1, numero 22).

Servizio: Progressivi totale Ubicazioni

Servizio che permette di ottenere i progressivi di tutte le ubicazioni per tutti i magazzini (Magazzino=0) o per il singolo magazzino selezionato (Magazzino=n).

Comando: `get_prog_ubicazioni`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_prog_ubicazioni"`
cod_art_progr	String	No	Se vuoto ritorna il dettaglio di tutti gli articoli; se specificato un articolo ritorna il dettaglio di quell'articolo
dettlotti	Boolean	No	Se `true` ritorna il dettaglio anche dei lotti
campi	Array	No	Elenco dei soli campi da restituire nella response
filtri	Array	No	Array di oggetti filtro (campo, condizione, valore)
next	String	No	Token di paginazione dalla response precedente

Filtri disponibili

I filtri possono essere applicati **esclusivamente** sui seguenti campi:

Campo	Descrizione
`dt_mod_ult_doc`	Data ultima modifica di un progressivo

Campo	Descrizione
`id`	Identificativo dell'ubicazione

> ⚠ **Nota Bene:** Qualsiasi altro campo non è ammesso come filtro. In particolare non va usato il campo `data_ult_mod`, che si riferisce all'anagrafica dell'ubicazione e non ai progressivi.

Sebbene il campo `dt_mod_ult_doc` sia un array, nel filtro deve essere usato come scalare: il significato è "restituisci i soli record di ubicazione in cui almeno un progressivo soddisfa la condizione".

Esempio Request (base)

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024 Magazzino=0

{
  "cmd": "get_prog_ubicazioni",
  "cod_art_progr": "",
  "dettlotti": true
}
```

Esempio Request (con filtro e selezione campi)

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024 Magazzino=0

{
  "cmd": "get_prog_ubicazioni",
  "cod_art_progr": "",
  "dettlotti": true,
  "campi": [
    "id"
  ],
  "filtri": [
    {
      "campo": "dt_mod_ult_doc",
      "condizione": ">",
      "valore": "20230501 162500"
    },
    {
      "campo": "id",
      "condizione": "=",
      "valore": 4
    }
```

```
  ]
}
```

In questo esempio si chiede di avere tutti i progressivi con il dettaglio dei lotti, ma solo delle ubicazioni in cui è cambiato “qualche cosa” dopo il 1° maggio 2023 alle ore 16.25. Il risultato deve mostrare solo gli `id` delle ubicazioni coinvolte.

Paginazione

Se la risposta presenta un elemento `next` significa che il servizio non è riuscito a fornire tutti i dati in una sola request. Bisogna fare una successiva request inserendo, oltre ai dati precedentemente indicati, anche il campo `next` esattamente come è indicato nella response.

```
{
  "cmd": "get_prog_ubicazioni",
  "next": "<valore_next_ricevuto>",
  "cod_art_progr": "",
  "dettlotti": true
}
```

Note Operative

- Il magazzino deve essere impostato nell’header della request nella chiave `Coordinate-Gestionale` (es. `Magazzino=0` per tutti, `Magazzino=1` per il magazzino 1).
- I parametri `cod_art_progr` e `dettlotti` hanno lo stesso significato della corrispondente chiamata REST per singola ubicazione.
- È possibile indicare i soli campi che si vuole vengano restituiti elencandoli nell’oggetto di tipo array `campi`.

Servizio: Totali Documento e Totali Riga

Servizi che permettono di ottenere i totali di un documento di magazzino e i totali di una singola riga.

Metodo: POST
URL: `https://base_address/webapi/servizi`

Totali Documento

Comando: `totali_documento`

Permette di ottenere i totali del documento di magazzino specificato nei dati di input.

Parametri

Campo	Tipo	Obbl.	Descrizione

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"totali_documento"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_sigla	String	Cond.	Sigla del documento (es. “OC”)
dati.in_serie	Number	Cond.	Serie del documento
dati.in_numero	Number	Cond.	Numero del documento
dati.in_cod_conto	String	Cond.	Codice conto
dati.in_documento	Object	Cond.	Oggetto con i dati completi del documento

> ⚠ **Nota Bene:** I campi `{in_sigla, in_serie, in_numero, in_cod_conto}` e `in_documento` sono mutuamente esclusivi. Se vengono forniti dei valori validi per la lettura di un documento esistente vengono usati quelli invece dei campi specificati in `in_documento`.

Per calcolare i totali di un nuovo documento ancora non creato, è necessario inserire in `in_documento` tutti i campi come se si stesse facendo la creazione di un nuovo documento tramite end-point REST.

Oltre ai totali del documento verrà restituito anche un array `totali_righe` con i totali di riga.

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "totali_documento",
  "dati": {
    "in_sigla": "OC",
    "in_serie": 1,
    "in_numero": 400,
    "in_cod_conto": "501.00001",
    "in_documento": {
      <dati_documento>
    }
  }
}
```

Totali di Riga

Comando: `totali_riga_documento`

Permette di ottenere i totali di una riga di un documento.

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"totali_riga_documento"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_sigla	String	Sì	Sigla del documento (es. “OC”)
dati.in_serie	Number	Sì	Serie del documento
dati.in_numero	Number	Sì	Numero del documento
dati.in_id_riga	Number	Sì	ID della riga
dati.in_modo	Number	Sì	Modalità di calcolo (1 o 2)
dati.in_nr_dec_articolo	Number	No	Numero decimali dell’articolo

Valori di in_modo

Valore	Descrizione
1	Calcola per progressivi — i valori (es. quantità e importo) di riga sono in unità di misura primaria e in valuta di gestione azienda
2	Calcola per totale documento — gli stessi valori sono espressi in unità di misura secondaria (se gestita) ed in valuta del documento (se diversa da quella di valuta azienda)

La differenza di metodo è relazionata alle righe in um2 (unità di misura secondaria).

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "totali_riga_documento",
  "dati": {
    "in_sigla": "OC",
    "in_serie": 1,
    "in_numero": 400,
    "in_id_riga": 1,
    "in_modo": 1,
    "in_nr_dec_articolo": 1
  }
}
```

Servizio: Località Italiane (CAP e Zone)

Sono due servizi separati per la lettura delle località italiane:

- **get_tabella_cap** — per l'elenco delle province, località, CAP e comune
- **get_tabella_zone** — per l'elenco degli indirizzi/zone

Metodo: POST
URL: `https://base_address/webapi/servizi`

---

get_tabella_cap

Comando: `get_tabella_cap`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_tabella_cap"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_tipo_ricerca	String	Sì	Tipo di informazione richiesta (vedi tabella sotto)
dati.in_cap	String	No	Filtro per CAP
dati.in_provincia	String	No	Filtro per provincia
dati.in_comune	String	No	Filtro per comune
dati.in_localita	String	No	Filtro per località
next	String	No	Token di paginazione

Valori di in_tipo_ricerca

Valore	Descrizione	Campi restituiti	Filtri possibili
P	Elenco delle province	provincia	—
C	Elenco dei CAP	provincia, cap	provincia
L	Elenco delle località	provincia, cap, località	provincia, cap
M	Elenco dei comuni	provincia, cap, localita, comune	provincia, cap, comune

Esempio Request (con filtri)

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

```
{
  "cmd": "get_tabella_cap",
  "dati": {
    "in_tipo_ricerca": "L",
    "in_provincia": "RN",
    "in_comune": "Rimini",
    "in_cap": "47921",
    "in_localita": "Viserba"
  }
}
```

Esempio Request (tutti i record)

```
{
  "cmd": "get_tabella_cap",
  "dati": {
    "in_tipo_ricerca": "M"
  }
}
```

---

get_tabella_zone

Comando: `get_tabella_zone`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_tabella_zone"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_cap	String	No	Filtro per CAP
dati.in_provincia	String	No	Filtro per provincia
next	String	No	Token di paginazione

Esempio Request (con filtri)

```
{
  "cmd": "get_tabella_zone",
  "dati": {
    "in_cap": "47921",
    "in_provincia": "RN"
  }
}
```

Esempio Request (tutte le zone)

```
{
  "cmd": "get_tabella_zone",
```

```
  "dati": {}
}
```

Paginazione

Entrambi i servizi, in quanto ritornano una grossa mole di dati, potrebbero paginare. In questo caso in fondo al JSON della risposta sarà presente un campo `next` con il valore del record successivo da cui partire nella successiva richiesta.

Il `next` dovrà essere inserito nella request allo stesso livello del campo `cmd`.

Esempio con paginazione (get_tabella_zone)

```
{
  "cmd": "get_tabella_zone",
  "next": "89831",
  "dati": {}
}
```

Servizio: Conteggio Operazioni Contabili

Servizio che restituisce il conteggio del numero di operazioni contabili a fronte di una data di inizio e fine.

Comando: `totali_nr_operazioni_contabili`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"totali_nr_operazioni_contabili"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_da_data_op	String	No	Data inizio nel formato AAAAMMGG
dati.in_a_data_op	String	No	Data fine nel formato AAAAMMGG

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

```
{
  "cmd": "totali_nr_operazioni_contabili",
  "dati": {
    "in_da_data_op": "20230101",
    "in_a_data_op": "20230510"
  }
}
```

Note Operative

- Se `in_da_data_op` e `in_a_data_op` sono vuote vengono restituite tutte le operazioni contabili.

Servizio: Pratiche Dichiarativi

Servizi per la lettura e il filtraggio delle pratiche dei dichiarativi.

Metodo: POST
URL: `https://base_address/webapi/servizi`

Lettura pratica dichiarativi

Comando: `leggi_pratica_dr`

Restituisce le informazioni su una particolare pratica dei dichiarativi.

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"leggi_pratica_dr"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_mod_pratica	String	Sì	Modello dichiarativo (es. “RPF”)
dati.in_pratica	String	Sì	Codice pratica
dati.in_interno	String	Sì	Numero interno
dati.in_tp_soggetto	String	No	Tipo soggetto (es. “D”)

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024
```

```
{
  "cmd": "leggi_pratica_dr",
  "dati": {
    "in_mod_pratica": "RPF",
    "in_pratica": "RMC",
    "in_interno": "1",
    "in_tp_soggetto": "D"
  }
}
```

Lista pratiche dichiarativi

Comando: `get_lista_pratiche_dr`

Restituisce una collection di elementi con le informazioni su tutte le pratiche dei dichiarativi.

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_lista_pratiche_dr"`
dati	Object	No	Oggetto con i filtri (tutti opzionali)

Filtri disponibili

Campo	Descrizione
in_mod_pratica	Filtra le pratiche che appartengono al modello dichiarativo specificato
in_da_pratica	Filtra le pratiche con numero pratica uguale o superiore a quella specificata
in_da_interno	Filtra le pratiche con interno uguale o superiore a quello specificato (a parità di numero pratica)
in_a_pratica	Filtra le pratiche con numero pratica inferiore o uguale a quella specificata
in_a_interno	Filtra le pratiche con interno inferiore o uguale a quello specificato (a parità di numero pratica)
in_cf_dichiarante	Filtra le pratiche relative al dichiarante con il codice fiscale indicato
in_codice_stato_pratica	Filtra per stato pratica (vedi valori sotto)
in_da_data	Filtra le pratiche con data ricevuta uguale o superiore
in_a_data	Filtra le pratiche con data ricevuta inferiore o uguale
in_codice_esito	Filtra per esito ricevuta (vedi valori sotto)

Valori di in_codice_stato_pratica

Valore	Stato
N	(vuoto)
T	Pratica terminata
S	Pratica terminata - Solo stampa IMU
I	Solo stampa IMU
E	Messaggio di errore relativo alla lettura della pratica

Valori di in_codice_esito

Valore	Esito
N	(vuoto)
A	Ricevuta accolta
S	Ricevuta scartata
X	Pratica inviata ma ricevuta non associata

Esempio Request (senza filtri)

```
{
  "cmd": "get_lista_pratiche_dr",
  "dati": {}
}
```

Esempio Request (filtrata per stato)

```
{
  "cmd": "get_lista_pratiche_dr",
  "dati": {
    "in_mod_pratica": "RSC",
    "in_stato": "N"
  }
}
```

Servizio: Deleghe Dichiarativi

Servizi per leggere la lista delle deleghe dei pagamenti unificati F24 e per lo sblocco di deleghe sospese.

Metodo: POST
URL: `https://base_address/webapi/servizi`

Comando: `get_lista_deleghe`

Le deleghe possono essere filtrate passando un elenco di pratiche o aziende, deleghe o un range di date, fino ad arrivare con precisione ad identificare una singola delega.

Filtri disponibili

Campo	Descrizione
in_azienda_pratica	Limita la ricerca: “A” (sole aziende) o “P” (sole pratiche)
in_da_azienda	Sigla azienda uguale o superiore
in_a_azienda	Sigla azienda inferiore o uguale
in_da_pratica	Numero pratica uguale o superiore
in_a_pratica	Numero pratica inferiore o uguale
in_da_interno	Interno uguale o superiore (a parità di numero pratica)
in_a_interno	Interno inferiore o uguale (a parità di numero pratica)
in_cf_dichiarante	Codice fiscale del dichiarante (solo pratiche, no aziende)
in_mod_pratica	Modello dichiarativo: 730, RPF, RSP, RSC, RNC, CNM
in_stato	Stato “sospeso” — accetta valore “S”
in_da_id_delega	ID delega uguale o superiore (formato: GG/MM/AAAA[/PR])
in_a_id_delega	ID delega inferiore o uguale (formato: GG/MM/AAAA[/PR])
in_da_data	Data versamento uguale o superiore
in_a_data	Data versamento inferiore o uguale
in_tipo	Tipo: “Ordinario”, “Semplificato” o “Elide”
in_definitiva	“S” = sole definitive, “N” = solo non definitive
in_accise	“S” = sole con accise, “N” = tutte escluse accise
in_ravvedimento_operoso	“S” = solo ravvedimento, “N” = tutte escluse ravvedimento
in_avviso_bonario	“S” = solo avvisi bonari, “N” = tutte escluse avvisi bonari
in_codice_invio	Modalità invio: S, P, E, O, T, Y, N (vedi tabella sotto)
in_maggiorazione	“S” = con maggiorazione, “N” = senza maggiorazione
in_data_ult_mod	Tutte le deleghe modificate dalla data specificata in poi

Valori di in_codice_invio

Valore	Descrizione
S	CBI Studio
P	CBI Personale
E	Entratel intermediari
O	Contribuente Fisco Online
T	Contribuente Entratel
Y	Entratel Studio

Valore	Descrizione
N	Cartaceo

Array di selezione puntuale

Campo	Descrizione
in_elenco_pratiche	Array di pratiche in cui ricercare le deleghe
in_elenco_aziende	Array di aziende in cui ricercare le deleghe
in_elenco_deleghe	Array di deleghe specifiche da ricercare (dentro in_elenco_pratiche)

Nota: Per le sole pratiche redditi è possibile ricercare deleghe su qualunque pratica senza indicarne una lista tramite `in_elenco_pratiche`. Quando si opera sulle aziende, al contrario, l'etichetta `in_elenco_aziende` deve sempre essere presente.

Esempio 1 — Tutte le deleghe dell'installazione

```
{
  "cmd": "get_lista_deleghe"
}
```

Esempio 2 — Deleghe di una pratica specifica

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "026",
        "in_interno": "1"
      }
    ]
  }
}
```

Esempio 3 — Deleghe in due pratiche di diversa tipologia

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "026",
        "in_interno": "1"
      },
      {
```

```
        "in_mod_pratica": "RSC",
        "in_pratica": "CIM",
        "in_interno": "1"
      }
    ]
  }
}
```

Esempio 4 — Con filtro data versamento

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "026",
        "in_interno": "1"
      },
      {
        "in_mod_pratica": "RSC",
        "in_pratica": "CIM",
        "in_interno": "1"
      }
    ],
    "in_da_data": "20230101",
    "in_a_data": "20230630"
  }
}
```

Esempio 5 — Deleghe in stato “sospeso”

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "026",
        "in_interno": "1"
      }
    ],
    "in_stato": "S"
  }
}
```

Esempio 6 — Deleghe di un’azienda

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
```

```
    "in_elenco_aziende": [
      {
        "in_sigla": "A11"
      }
    ]
  }
}
```

Esempio 7 — Deleghe di 2 aziende

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_aziende": [
      {
        "in_sigla": "A11"
      },
      {
        "in_sigla": "A04"
      }
    ]
  }
}
```

Esempio 8 — Due aziende con filtro data

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_aziende": [
      {
        "in_sigla": "A11"
      },
      {
        "in_sigla": "A04"
      }
    ],
    "in_a_data": "20230630"
  }
}
```

Esempio 9 — Due deleghe specifiche in una pratica

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "026",
        "in_interno": "1",
```

```
        "in_elenco_deleghe": [
          {
            "in_id_delega": "13/12/2023/15"
          },
          {
            "in_id_delega": "16/02/2016/50"
          }
        ]
      }
    ]
  }
}
```

Esempio 10 — Tutte le deleghe di tutte le aziende

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_azienda_pratica": "A"
  }
}
```

Esempio 11a — Filtro per range di id delega in un'azienda (con elenco)

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_da_id_delega": "01/09/2023",
    "in_a_id_delega": "31/12/2023",
    "in_elenco_aziende": [
      {
        "in_sigla": "A11"
      }
    ]
  }
}
```

Esempio 11b — Filtro per range di id delega in un'azienda (con range)

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_da_azienda": "A11",
    "in_a_azienda": "A11",
    "in_da_id_delega": "01/09/2023",
    "in_a_id_delega": "31/12/2023"
  }
}
```

Esempio 12 — Filtro combinato aziende, pratiche e id

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_da_azienda": "A01",
    "in_a_azienda": "A11",
    "in_mod_pratica": "RPF",
    "in_da_pratica": "026",
    "in_da_interno": "1",
    "in_a_pratica": "B01",
    "in_a_interno": "1",
    "in_da_id_delega": "01/08/2021",
    "in_a_id_delega": "31/12/2021"
  }
}
```

Esempio 13 — Aziende con range + tutte le pratiche

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_azienda_pratica": "P",
    "in_da_azienda": "A01",
    "in_a_azienda": "A11"
  }
}
```

Esempio 14 — Paginazione

Prima request:

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_azienda_pratica": "P"
  }
}
```

L'elaborazione si interrompe perché è stato raggiunto uno dei limiti previsti. Viene fornita la chiave per continuare: `"next": "P-730-001-1-D-2016-12-16-07-001-6"`.

Request successiva:

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_azienda_pratica": "P"
  },
  "next": "P-730-001-1-D-2016-12-16-07-001-6"
}
```

Esempio 15 — Filtro per data di ultima modifica

Vengono restituite tutte le deleghe a partire dal 15 Aprile 2025 alle ore 9:00 in poi.

```
{
  "cmd": "get_lista_deleghe",
  "dati": {
    "in_data_ult_mod": "20250415 090000"
  }
}
```

Servizio: Sblocco deleghe

Servizio per lo sblocco di deleghe

Metodo: POST
URL: `https://base_address/webapi/servizi`

Comando: `cambia_stato_delega`

Permette, dato un elenco di pratiche e deleghe, lo sblocco della delega che si trova in stato “sospeso”.

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"cambia_stato_delega"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_elenco_pratiche	Array	Sì	Array di pratiche con le relative deleghe da sbloccare
dati.in_elenco_pratiche[].in_mod_pratica	String	Sì	Modello dichiarativo
dati.in_elenco_pratiche[].in_pratica	String	Sì	Codice pratica
dati.in_elenco_pratiche[].in_interno	String	Sì	Interno
dati.in_elenco_pratiche[].in_elenco_deleghe	Array	Sì	Array con i riferimenti per sbloccare la delega
dati.in_elenco_pratiche[].in_elenco_deleghe[].in_id_delega	String	Sì	ID della delega

Campo	Tipo	Obbl.	Descrizione
dati.in_elenco_pratiche[].in_elenco_deleghe[].in_numero_modulo	String	Sì	Numero modulo

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "cambia_stato_delega",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "A02",
        "in_interno": "1",
        "in_elenco_deleghe": [
          {
            "in_id_delega": "02/01/2023/77",
            "in_numero_modulo": "001"
          }
        ]
      }
    ]
  }
}
```

Codici di esito

codice_esito	Descrizione
0	Nessun errore
1	Delega in uso su altro terminale
2	Delega non è sospesa. Put non eseguita
4	Delega GG/MM/AAAA/PR non trovata. Put non eseguita

Note Operative

- Nell'array `in_elenco_pratiche` devono essere elencate le varie pratiche, e per ogni pratica nell'array `in_elenco_deleghe` devono essere elencati gli oggetti con i riferimenti per sbloccare la particolare delega (`in_id_delega` + `in_numero_modulo`).

- In caso di richieste onerose possono essere restituiti risultati parziali e viene fornita la chiave `next` da cui partire per completare la lettura (paginazione).

Servizio: Registro Cancellati

Servizio per la lettura dei registri con l'elenco dei record eliminati sui vari end-point.

Comando: `get_registro_operazioni_utente`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_registro_operazioni_utente"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_archivi	Array	Sì	Array con l'elenco degli end-point dei quali si vuole avere la lista dei cancellati
dati.in_tipo_operazione	String	Sì	Tipo operazione — al momento solo "C" (cancellati)
dati.in_storico	Boolean	No	Se `true`, viene indicato se il dato cancellato è presente nello storico
dati.in_data_ric_elab	String	No	Data/ora di ultima lettura (formato "AAAAMMGG HHMMSS"). Se omesso prende tutti i record

End-point gestiti

- articoli
- alias-articoli
- clienti
- fornitori
- prima-nota
- movimenti-magazzino
- scadenzario
- anagrafica-unica
- anagrafica-unica-storico
- preventivi
- ordini-clienti

- indirizzi-spedizione
- liste-prelievo
- ubicazioni
- particolarita
- ordini-fornitori
- ordini-matrici
- lotti
- distinte-base
- lavorazione-bolle
- impegni

Informazione storico

L'informazione `in_storico` è disponibile solo per: - Movimenti di magazzino (bolle) - Ordini clienti - Lotti su ordini clienti - Lotti su movimenti di magazzino (lotti su bolle)

Esempio Request (senza data)

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_registro_operazioni_utente",
  "dati": {
    "in_archivi": [
      "clienti",
      "articoli",
      "ordini-clienti",
      "movimenti-magazzino",
      "prima-nota",
      "anagrafica-unica"
    ],
    "in_tipo_operazione": "C"
  }
}
```

Esempio Response (senza data)

```
{
  "data_ora_ultima_lettura": "20240411 121914",
  "registro_operazioni": [
    {
      "archivio": "clienti",
      "elenco_cancellati": []
```

```
    },
    {
      "archivio": "articoli",
      "elenco_cancellati": [
        {
          "codice": "&''UN TEST\\CATTIVO",
          "data_operazione": "20240410 121914"
        }
      ]
    },
    {
      "archivio": "ordini-clienti",
      "elenco_cancellati": []
    },
    {
      "archivio": "movimenti-magazzino",
      "elenco_cancellati": []
    },
    {
      "archivio": "prima-nota",
      "elenco_cancellati": []
    },
    {
      "archivio": "anagrafica-unica",
      "elenco_cancellati": []
    }
  ]
}
```

Questo indica che al momento della chiamata (20240411 121914) risulta cancellato un articolo.

Nota bene: la data non è quella in cui è stato eliminato il record ma quella in cui viene fatta la richiesta.

Esempio Request (con data di ultima lettura)

```
{
  "cmd": "get_registro_operazioni_utente",
  "dati": {
    "in_archivi": [
      "clienti",
      "articoli",
      "ordini-clienti",
      "movimenti-magazzino",
      "prima-nota",
      "anagrafica-unica"
    ],
    "in_tipo_operazione": "C",
    "in_data_ora_ultima_lettura": "20240411 121914"
  }
}
```

Esempio Response (con data di ultima lettura)


```
{
  "data_ora_ultima_lettura": "20240411 122700",
  "registro_operazioni": [
    {
      "archivio": "clienti",
      "elenco_cancellati": []
    },
    {
      "archivio": "articoli",
      "elenco_cancellati": []
    },
    {
      "archivio": "ordini-clienti",
      "elenco_cancellati": []
    },
    {
      "archivio": "movimenti-magazzino",
      "elenco_cancellati": []
    },
    {
      "archivio": "prima-nota",
      "elenco_cancellati": []
    },
    {
      "archivio": "anagrafica-unica",
      "elenco_cancellati": []
    }
  ]
}
```


In questo caso passando come dato anche la data di ultima lettura ricevuta dal comando precedente non viene riportato alcun cancellato in quanto già processato dalla precedente richiesta.

Note Operative

- Al momento è possibile richiedere solo i cancellati (`in_tipo_operazione` = “C”).
- Se `in_data_ric_elab` viene omesso, vengono restituiti tutti i record cancellati per gli archivi specificati.
- Se viene inserita la `data_ric_elab` ritornata dall’ultima richiesta effettuata, mostrerà solo i cancellati da quel momento in poi.
- In caso di ripristino di dati aziendali verrà restituito un errore 400 indicando che il registro non è più consistente.

Servizio: Dichiarazioni IVA Annuali

Servizio per richiedere la lista delle dichiarazioni IVA.

Comando: `leggi_dichiarazione_iva_annuale`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"leggi_dichiarazione_iva_annuale"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_da_azienda	String	Sì	Sigla azienda di inizio range
dati.in_a_azienda	String	Sì	Sigla azienda di fine range
dati.in_da_data_versamento	String	No	Data versamento iniziale (AAAAMMGG)
dati.in_a_data_versamento	String	No	Data versamento finale (AAAAMMGG)
dati.in_da_data_fine_pratica	String	No	Data fine pratica iniziale (AAAAMMGG)
dati.in_a_data_fine_pratica	String	No	Data fine pratica finale (AAAAMMGG)

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "leggi_dichiarazione_iva_annuale",
  "dati": {
    "in_da_azienda": "<sigla_azienda>",
    "in_a_azienda": "<sigla_azienda>",
    "in_da_data_versamento": "AAAAMMGG",
    "in_a_data_versamento": "AAAAMMGG",
    "in_da_data_fine_pratica": "AAAAMMGG",
    "in_a_data_fine_pratica": "AAAAMMGG"
  }
}
```

Note Operative

- I campi `in_da_data_versamento`, `in_a_data_versamento`, `in_da_data_fine_pratica` e `in_a_data_fine_pratica` sono facoltativi:

- Se `in_da_data_versamento` non viene specificato, il servizio lo imposta in automatico alla data 01/01/<primo anno gestito dell'azienda>
- Se `in_a_data_versamento` non viene specificato, il servizio lo imposta in automatico alla data 31/12/<ultimo anno gestito dell'azienda>

⚠ **Nota Bene:** Le aziende vengono aperte con anno terminale indicato nell'header della richiesta, mentre l'azienda indicata nell'header viene ignorata in quanto è un comando sovraziendale.

Servizio: Lettura Comunicazioni LIPE

Servizio per richiedere la lettura delle dichiarazioni LIPE (Liquidazioni Periodiche IVA).

Comando: `leggi_comunicazioni_lipe`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"leggi_comunicazioni_lipe"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_da_azienda	String	Sì	Sigla azienda di inizio range
dati.in_a_azienda	String	Sì	Sigla azienda di fine range
dati.in_da_data_versamento	String	No	Data versamento iniziale (AAAAMMGG)
dati.in_a_data_versamento	String	No	Data versamento finale (AAAAMMGG)
dati.in_trimestre	Number	No	Trimestre (1, 2, 3, 4, -1 = tutti). Se omesso ritorna tutti i trimestri

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "leggi_comunicazioni_lipe",
  "dati": {
    "in_da_azienda": "A01",
    "in_a_azienda": "A22",
    "in_da_data_versamento": "20240701",
```

```
    "in_a_data_versamento": "20240801",
    "in_trimestre": 2
  }
}
```

Note Operative

- I campi `in_da_data_versamento`, `in_a_data_versamento` e `in_trimestre` sono facoltativi:
  - Se `in_da_data_versamento` non viene specificato, il servizio lo imposta in automatico alla data 01/01/<primo anno gestito dell'azienda>
  - Se `in_a_data_versamento` non viene specificato, il servizio lo imposta in automatico alla data 31/12/<ultimo anno gestito dell'azienda>
  - `in_trimestre` può assumere valori numerici (1, 2, 3, 4, -1) e se viene omesso o impostato a -1 ritorna tutti i trimestri

⚠ **Nota Bene:** Verranno mostrate solo comunicazioni per cui è stata impostata una data di versamento. Le aziende vengono aperte con anno terminale indicato nell'header della richiesta, mentre l'azienda indicata nell'header viene ignorata in quanto è un comando sovraziendale.

Servizio: Stampa PDF di Documenti

Servizi che lanciano una stampa PDF lato server del documento specificato nei campi chiave. Il risultato viene tornato nella response con content-type `application/pdf`.

Metodo: POST
URL: `https://base_address/webapi/servizi`

Comandi disponibili

Comando	Tipo documento
`stampa_ordine_cliente`	Ordine cliente
`stampa_ordine_fornitore`	Ordine fornitore
`stampa_preventivo`	Preventivo
`stampa_ordine_matrice`	Ordine matrice
`stampa_movimento_magazzino`	Movimento di magazzino

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	Nome del comando di stampa (vedi tabella sopra)
dati	Object	Sì	Oggetto con i parametri del servizio
dati.sigla	String	Sì	Sigla del documento

Campo	Tipo	Obbl.	Descrizione
dati.serie	Number	Sì	Serie del documento
dati.numero	Number	Sì	Numero del documento
dati.cod_conto	String	Cond.	Codice conto — obbligatorio solo per i movimenti di magazzino
dati.cod_modulo	String	No	Codice modulo di stampa personalizzato

Esempio Request — Stampa ordine cliente

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "stampa_ordine_cliente",
  "dati": {
    "sigla": "OC",
    "serie": 1,
    "numero": 401,
    "cod_modulo": "A"
  }
}
```

Esempio Request — Stampa movimento di magazzino

```
{
  "cmd": "stampa_movimento_magazzino",
  "dati": {
    "sigla": "BC",
    "serie": 1,
    "numero": 1,
    "cod_conto": "501.00001"
  }
}
```

Esempio Response

La response ha `Content-Type: application/pdf` e contiene il file PDF binario del documento stampato.

Note Operative

- Il campo `cod_conto` è obbligatorio solo per i movimenti di magazzino.

- Il campo `cod_modulo` è facoltativo e permette di specificare un modulo di stampa personalizzato.
- Il risultato è un PDF generato lato server.

Servizio: Inserimento Righe nei Documenti

Dato che l'inserimento di nuove righe a seguito di una PUT o anche in fase di trasformazione con POST non restituisce l'id delle righe appena create, è stato inserito un servizio dedicato al solo inserimento di nuove righe all'interno di un documento.

Metodo: POST
URL: `https://base_address/webapi/servizi`

Comandi disponibili

Comando	Tipo documento
`ins_righe_ordine_cliente`	Ordine cliente
`ins_righe_ordine_fornitore`	Ordine fornitore
`ins_righe_preventivo`	Preventivo
`ins_righe_ordine_matrice`	Ordine matrice
`ins_righe_movimento_magazzino`	Movimento di magazzino

Parametri comuni

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	Nome del comando di inserimento
dati	Object	Sì	Oggetto con i parametri del servizio
dati.sigla	String	Sì	Sigla del documento
dati.serie	Number	Sì	Serie del documento
dati.numero	Number	Sì	Numero del documento
dati.cod_conto	String	Cond.	Codice conto — solo per movimenti di magazzino
dati.id_riga	Array	Sì	Array di coppie [posizione, 0]
dati.tp_riga	Array	Sì	Array di coppie [posizione, tipo_riga]
dati.codice_articolo	Array	Sì	Array di coppie [posizione, codice]
dati.quantita	Array	Sì	Array di coppie [posizione, quantità]
dati.cod_iva	Array	Sì	Array di coppie [posizione, codice_iva]
dati.id_mag_riga	Array	Cond.	Array di coppie [posizione, id_magazzino] — per ordini

Campo	Tipo	Obbl.	Descrizione
dati.id_magazzino	Number	Cond.	ID magazzino di testata — per movimenti di magazzino

Campi obbligatori aggiuntivi per movimenti di magazzino

I seguenti campi sono obbligatori nella richiesta ma ininfluenti per il risultato dell'operazione (si possono lasciare vuoti):

```
"sigla_ordine": [],
"serie_ordine": [],
"numero_ordine": [],
"sigla_doc_orig": [],
"serie_doc_orig": [],
"numero_doc_orig": [],
"id_rif_testata": []
```

Logica di posizionamento

La posizione in cui inserire la nuova riga è specificata dall'indice dell'array. Se si ha un documento con 3 righe e si vuole inserire 1 riga tra la seconda e la terza (posizione 3), è necessario specificare come indice 3.

Se si vogliono inserire più righe, la posizione di inserimento sarà sempre riferita al documento originale:

Doc originale	Indice origine	Doc con inserimento	Indice risultante
Riga1	1	Riga1	1
Riga2	2	Riga2	2
		Nuova Riga3	**3**
Riga3	3	Riga3	4
		Nuova Riga4	**5**

Esempio Request — Inserimento righe in ordine cliente

Inserimento di tre righe in posizione 2, 3 e 6 in un documento con 5 righe esistenti:

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "ins_righe_ordine_cliente",
  "dati": {
    "sigla": "OC",
```

```
    "serie": 1,
    "numero": 22,
    "id_riga": [
      [2, 0],
      [3, 0],
      [6, 0]
    ],
    "tp_riga": [
      [2, "R"],
      [3, "R"],
      [6, "R"]
    ],
    "codice_articolo": [
      [2, "ATT002"],
      [3, "ATT002"],
      [6, "ATT002"]
    ],
    "quantita": [
      [2, 1.0],
      [3, 1.0],
      [6, 1.0]
    ],
    "cod_iva": [
      [2, " 22,0"],
      [3, " 22,0"],
      [6, " 22,0"]
    ],
    "id_mag_riga": [
      [2, 1],
      [3, 1],
      [6, 1]
    ]
  }
}
```


Esempio Response — Ordine cliente


```
{
  "sigla": "OC",
  "serie": 1,
  "numero": 22,
  "id_riga": [
    [2, 72],
    [3, 73],
    [8, 74]
  ],
  "tp_riga": [
    [2, "R"],
    [3, "R"],
    [8, "R"]
```

```
  ]
}
```

Le righe 2 e 3 sono state inserite in posizione corretta e quelle presenti nel documento di origine sono state “shiftate”. La riga in posizione 6 si intende che deve essere inserita in ultima posizione, quindi nel documento risultante sarà inserita in posizione 8.

Esempio Request — Inserimento righe in movimento di magazzino

Inserimento di tre righe in un movimento di magazzino composto da testate differenti:

```
{
  "cmd": "ins_righe_movimento_magazzino",
  "dati": {
    "sigla": "BC",
    "serie": 3,
    "numero": 1,
    "cod_conto": "501.00072",
    "id_riga": [
      [3, 0],
      [6, 0],
      [7, 0]
    ],
    "tp_riga": [
      [3, "R"],
      [6, "R"],
      [7, "R"]
    ],
    "codice_articolo": [
      [3, "ATT005"],
      [6, "ATT005"],
      [7, "ATT005"]
    ],
    "quantita": [
      [3, 1.0],
      [6, 1.0],
      [7, 1.0]
    ],
    "cod_iva": [
      [3, " 22,0"],
      [6, " 22,0"],
      [7, " 22,0"]
    ],
    "sigla_ordine": [],
    "serie_ordine": [],
    "numero_ordine": [],
    "sigla_doc_orig": [],
    "serie_doc_orig": [],
    "numero_doc_orig": [],
    "id_rif_testata": [],
```

```
    "id_magazzino": 1
  }
}
```

Esempio Response — Movimento di magazzino

```
{
  "sigla": "BC",
  "serie": 3,
  "numero": 1,
  "cod_conto": "501.00072",
  "sigla_ordine": [],
  "serie_ordine": [],
  "numero_ordine": [],
  "sigla_doc_orig": [],
  "serie_doc_orig": [],
  "numero_doc_orig": [],
  "id_riga": [
    [3, 3],
    [7, 4],
    [8, 5]
  ],
  "tp_riga": [
    [3, "R"],
    [7, "R"],
    [8, "R"]
  ],
  "id_rif_testata": []
}
```

Il documento di origine aveva 5 righe di cui le prime 2 appartenenti ad una testata e le rimanenti 3 ad un'altra. La prima riga viene inserita in posizione 3 e prende come testata quella della riga precedente (prima testata), mentre la 6 e la 7 vengono accodate in fondo e quindi inserite nella seconda testata.

Note Operative

- Per i documenti di tipo ordine e preventivo i campi chiave sono solo sigla, serie e numero.
- Per i documenti di tipo movimenti di magazzino bisogna fornire anche il codice del conto (`cod_conto`).
- Per i documenti di tipo ordine, oltre ai campi obbligatori per i dettagli di riga, va indicato anche l'id del magazzino per ogni riga (`id_mag_riga`).
- Per i documenti di tipo movimento di magazzino, oltre ai campi obbligatori, va indicato anche l'id del magazzino di testata (`id_magazzino`).

Servizio: Lista Allegati Docuvision

Servizio che permette di ottenere la lista di tutti gli allegati Docuvision.

È possibile filtrare i risultati tramite l'elemento `filtri` e selezionare i soli campi che si vogliono leggere tramite l'elemento `campi` (come avviene per gli end-point REST).

Comando: `lista_docdv`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"lista_docdv"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_classe	Number	Sì	Classe del documento (vedi tabella classi)
dati.in_allegati	String	No	"S" = solo allegati, "N" = solo non allegati, "T" = tutti (default)
dati.in_gest_dv_dr	String	No	"S" = usa il DB dei redditi, "N" = DB aziendale
campi	Array	No	Elenco dei soli campi da restituire
filtri	Array	No	Array di oggetti filtro (campo, condizione, valore)

Tabella Classi Docuvision

classe_dv	Descrizione
1	Stampa generica
100	Anagrafica nominativi
200	Bilanci
400	Anagrafica Cliente/Fornitore
600	Operazione contabile
800	Dichiarativi/Versamenti
900	Stampe Fiscali
1000	Agenda di lavoro
1100	Parcella/Notula
1200	Anagrafica articolo
1400	Movimento di magazzino
1450	Controllo di gestione
1500	Ordine/Preventivo/Matrice
1600	Bolla di lavorazione

classe_dv	Descrizione
1700	Anagrafiche MyDB

Parametri di filtro per classe

Oltre ai filtri sui campi dell'archivio è possibile specificare nell'oggetto `dati` anche ulteriori parametri di filtro. I parametri di filtro sono gli stessi che, da gestionale in Docuvision → Gestione documento, compaiono nella maschera "Parametri su classe" (F2, scelta della classe e poi F7).

Classe 1 — Stampa generica

Tipo	Nome Variabile
STR	in_stpg_menu
STR	in_stpg_op_menu
STR	in_stpg_nota
STR	in_stpg_op_nota

Classe 100 — Anagrafica nominativi

Tipo	Nome Variabile
STR	in_nomi_tp_allega
DAT	in_nomi_dt_da
DAT	in_nomi_dt_a

Classe 200 — Bilanci

Tipo	Nome Variabile
NUM	in_bila_anno
STR	in_bila_tipo
STR	in_bila_autore
STR	in_bila_op_autore

Classe 400 — Anagrafica Cliente/Fornitore

Tipo	Nome Variabile
STR	in_pico_tp_conto
STR	in_pico_cod_conto
STR	in_pico_cod_alt
STR	in_pico_op_cod_alt
STR	in_pico_nome_ric
STR	in_pico_op_nome_ric
STR	in_pico_prv

Tipo	Nome Variabile
NUM	in_pico_zona
NUM	in_pico_catsta

Classe 600 — Operazione contabile

Tipo	Nome Variabile
NUM	in_prn_sotto_az
DAT	in_prn_dt_regis
STR	in_prn_cau
STR	in_prn_desc_testa
STR	in_prn_op_desc_testa
STR	in_prn_tp_rstro
NUM	in_prn_serie_prot
NUM	in_prn_nr_prot
NUM	in_prn_nr_doc
DAT	in_prn_dt_doc
STR	in_prn_tp_invio

Classe 800 — Dichiarativi/Versamenti

Tipo	Nome Variabile
NUM	in_dicv_anno
STR	in_dicv_dich
STR	in_dicv_tp_doc
STR	in_dicv_sgl_prat
STR	in_dicv_interno
STR	in_dicv_dich_coniuge
STR	in_dicv_tp_allega

Classe 900 — Stampe Fiscali

Tipo	Nome Variabile
NUM	in_stpf_anno
STR	in_stpf_doc
STR	in_stpf_tp_doc
STR	in_stpf_op_tp_doc
STR	in_stpf_crit_sel
STR	in_stpf_op_crit_sel

Classe 1000 — Agenda di lavoro

Tipo	Nome Variabile
STR	in_agen_stato
DAT	in_agen_dt_lavoro
STR	in_agen_cod_conto
STR	in_agen_prest
STR	in_agen_op_prest

Classe 1100 — Parcella/Notula

Tipo	Nome Variabile
NUM	in_notp_sotto_az
STR	in_notp_tp_doc
NUM	in_notp_serie
NUM	in_notp_numero
DAT	in_notp_dt_doc
STR	in_notp_cod_conto
STR	in_notp_note
STR	in_notp_op_note

Classe 1200 — Anagrafica articolo

Tipo	Nome Variabile
STR	in_arti_cod
STR	in_arti_op_cod
STR	in_arti_desc
STR	in_arti_op_desc
STR	in_arti_cod_alt
STR	in_arti_op_cod_alt
STR	in_arti_catsta
STR	in_arti_tp_allega

Classe 1400 — Movimento di magazzino

Tipo	Nome Variabile
NUM	in_mm_sotto_az
STR	in_mm_sigla
NUM	in_mm_serie
NUM	in_mm_numero

Tipo	Nome Variabile
DAT	in_mm_dt_doc
NUM	in_mm_anno_solare
STR	in_mm_cod_conto
NUM	in_mm_zona
NUM	in_mm_catsta
STR	in_mm_note
STR	in_mm_op_note

Classe 1450 — Controllo di gestione

Tipo	Nome Variabile
STR	in_cdg_tp_anag

Analitici:

Tipo	Nome Variabile
STR	in_cdg_a_cod
STR	in_cdg_a_tp
STR	in_cdg_a_natura
STR	in_cdg_a_qualita

Commesse:

Tipo	Nome Variabile
STR	in_cdg_c_cod
STR	in_cdg_c_padre
STR	in_cdg_c_tp
STR	in_cdg_c_respons
STR	in_cdg_c_zona
NUM	in_cdg_c_cli
NUM	in_cdg_c_dest_merce
NUM	in_cdg_c_age

Aree:

Tipo	Nome Variabile
STR	in_cdg_r_cod
STR	in_cdg_r_padre
STR	in_cdg_r_tp
STR	in_cdg_r_respons

Tipo	Nome Variabile
STR	in_cdg_r_zona

Classe 1500 — Ordine/Preventivo/Matrice

Tipo	Nome Variabile
NUM	in_ordi_sotto_az
STR	in_ordi_sigla
NUM	in_ordi_serie
NUM	in_ordi_numero
DAT	in_ordi_dt_doc
STR	in_ordi_cod_conto
STR	in_ordi_note
STR	in_ordi_op_note

Classe 1600 — Bolla di lavorazione

Tipo	Nome Variabile
NUM	in_blav_cod
STR	in_blav_cod_sottobl
DAT	in_blav_dt_bolla
STR	in_blav_cod_conto
STR	in_blav_nota
STR	in_blav_op_nota

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "lista_docdv",
  "dati": {
    "in_classe": 1200,
    "in_arti_cod": "TEST",
    "in_arti_op_cod": "CONTIENE"
  },
  "campi": [
    "descrizione"
  ],
  "filtri": [
```

```
    {
      "campo": "nr_protocollo",
      "condizione": ">=",
      "valore": "3"
    }
  ]
}
```

In questo esempio si richiedono tutti gli allegati degli articoli che nel codice contengono la parola TEST, con `nr_protocollo >= 3`. Viene tornata solo la descrizione.

Note Operative

- Il parametro numerico `in_classe` deve essere sempre indicato.
- Il parametro `in_allegati` è opzionale (default "T"):
  - **S** = solo i documenti allegati ad archivi rappresentati dalle classi predefinite
  - **N** = solo i documenti non allegati
  - **T** = tutti i documenti indipendentemente se allegati o meno

Servizio: Lettura Allegati Docuvision

Tramite questo servizio è possibile, per ogni archivio del gestionale, leggere i documenti allegati.

Gli allegati vengono restituiti in un `multipart/form-data` in formato RAW ognuno evidenziato con il proprio Content-Type (es. application/pdf, image/jpg, etc.) come da RFC 1341.

Comando: `get_allegato_archivio`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_allegato_archivio"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.tipo	String	Sì	Tipo di allegato richiesto (al momento solo "docuvision")
dati.codice	String	Cond.	Codice dell'entità di cui si vogliono leggere gli allegati (non valorizzare se id_dv > 0)
dati.classe_dv	Number	Sì	Classe del documento Docuvision
dati.id_dv	Number	Cond.	ID del particolare documento (0 o omesso se codice è valorizzato)

Campo	Tipo	Obbl.	Descrizione
dati.numero_versione_dv	Number	No	Versione del documento: -1 = tutte, 0 = ultima, >0 = specifica
next	String	No	Token di paginazione

Mutua esclusività dei parametri

- I campi `codice` e `id_dv` sono mutuamente esclusivi.
- Nel caso `numero_versione_dv > 0` è obbligatorio specificare anche `id_dv`.

Tabella Classi Docuvision

classe_dv	Descrizione
1	Stampa generica
100	Anagrafica nominativi
200	Bilanci
400	Anagrafica Cliente/Fornitore
600	Operazione contabile
800	Dichiarativi/Versamenti
900	Stampe Fiscali
1000	Agenda di lavoro
1100	Parcella/Notula
1200	Anagrafica articolo
1400	Movimento di magazzino
1450	Controllo di gestione
1500	Ordine/Preventivo/Matrice
1600	Bolla di lavorazione
1700	Anagrafiche MyDB

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_allegato_archivio",
  "dati": {
    "tipo": "docuvision",
    "codice": "",
```

```
    "classe_dv": 400,
    "id_dv": 1,
    "numero_versione_dv": 3
  }
}
```

Paginazione

In caso che il particolare archivio contenga più documenti (es. un articolo con diversi datasheet) oppure lo stesso documento ha più revisioni, se vengono superati limiti di tempo (30s) o di spazio (50MB) il risultato verrà paginato restituendo come ultimo oggetto del multipart solo l'elemento `next`.

Esempio di boundary con next nel multipart:

```
--ac70wbp48zghcooa4lol2n2xzuqfohrt
Content-Type: text/plain
Content-Disposition: form-data; name="__next"

9;3
--ac70wbp48zghcooa4lol2n2xzuqfohrt--
```

In questo esempio il documento continua con altri file a partire da `next="9;3"`.

La successiva chiamata va effettuata specificando `next` allo stesso livello di `cmd`:

```
{
  "cmd": "get_allegato_archivio",
  "next": "9;3",
  "dati": {
    "tipo": "docuvision",
    "codice": "",
    "classe_dv": 400,
    "id_dv": 1,
    "numero_versione_dv": 3
  }
}
```

Note Operative

- La response è di tipo `multipart/form-data` con ogni allegato evidenziato con il proprio Content-Type.
- Limiti di paginazione: 30 secondi di tempo o 50MB di spazio.
- Il campo `numero_versione_dv`:
  - **-1** — tutte le versioni
  - **0** — ultima versione
  - **>0** — versione specifica

Servizio: Associazioni Docuvision

Tramite questo servizio è possibile leggere le associazioni fra il documento Docuvision e le entità del gestionale in cui è stato allegato.

Comando: `get_assoc_dv`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_assoc_dv"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.codice	String	Cond.	Codice dell'entità (se valorizzato, è obbligatorio specificare anche la relativa classe)
dati.classe	Number	Cond.	Classe dell'entità in cui il documento è allegato
dati.id	Number	Cond.	ID del particolare documento (può essere 0 o omesso se codice è valorizzato)

Esempio Request — Ricerca per id

Vengono restituite tutte le associazioni in cui è presente il documento con id=15:

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_assoc_dv",
  "dati": {
    "codice": "",
    "classe": 0,
    "id": 15
  }
}
```

Esempio Request — Ricerca per classe

Vengono restituite tutte le associazioni con classe 1200:

```
{
  "cmd": "get_assoc_dv",
  "dati": {
    "codice": "",
    "classe": 1200,
    "id": 0
  }
}
```

Esempio Request — Ricerca per classe e codice

Vengono restituite tutte le associazioni con classe 1200 e codice articolo 121212:

```
{
  "cmd": "get_assoc_dv",
  "dati": {
    "codice": "121212",
    "classe": 1200,
    "id": 0
  }
}
```

Note Operative

- I campi `codice`/`classe` e `id` possono essere usati in combinazione per restringere la ricerca.
- Se `codice` è valorizzato, è obbligatorio specificare anche la relativa `classe`.

Servizio: Lista Strutture MyDB

Tramite questo servizio è possibile leggere e filtrare tutta la lista delle strutture MyDB presenti sui vari archivi.

Comando: `lista_strutture_mydb`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"lista_strutture_mydb"`
info_campi	Boolean	No	Se `true`, viene mostrato l'help in linea di tutti i campi che compongono la struttura
campi	Array	No	Elenco dei soli campi da restituire nella response

Campo	Tipo	Obbl.	Descrizione
filtri	Array	No	Array di oggetti filtro con le stesse condizioni usate per gli end-point REST

Sintassi per campi e filtri

Per indicare i campi negli elementi `campi` o `filtri` bisogna specificare il percorso completo del campo di interesse all'interno della struttura JSON usando i `:` come separatore.

Ad esempio, per usare il campo `codice_campo` all'interno della collection `elenco_campi`:

```
lista_strutture_mydb:elenco_campi:codice_campo
```

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "lista_strutture_mydb",
  "info_campi": true,
  "campi": [
    "lista_strutture_mydb:tipo_archivio"
  ],
  "filtri": [
    {
      "campo": "lista_strutture_mydb:codice_app",
      "condizione": "inizia_per",
      "valore": "924148"
    }
  ]
}
```

Esempio Response

```
{
  "lista_strutture_mydb": [
    {
      "tipo_archivio": "A",
      "codice_app": "010999fatturapa",
      "codice_archivio": "ANAPADOA",
      "desc_archivio": "Fattura XML (Codice Articolo)",
      "ordinamento": "I",
      "chiave_univoca": "N",
      "estensione_univoca": "S",
      "tipo_estensione": "R",
      "elenco_campi": [
```

```
        {
          "codice_campo": "zzz",
          "campo_estensione": "S",
          "etichetta_campo": "Tutti gli articoli",
          "tipo_campo": "*",
          "dimensione_campo": 32,
          "gest_campo_obblig": "S",
          "valore_dft_campo": "",
          "archivio_riportabile": "DMRT",
          "cod_produttore_rel": "",
          "codice_archivio_rel": "ARTT",
          "codice_campo_rel": ""
        }
      ]
    }
  ]
}
```


Note Operative

- Nell'elemento `campi` possono essere indicati gli unici campi che si vogliono ottenere in risposta.
- Nell'elemento `filtri` possono essere specificati dei filtri con le stesse condizioni usate per gli end-point REST.
- Se `info_campi=true` viene mostrato l'help in linea di tutti i campi che compongono la struttura.

Servizio: Proprietà Numeratori

Tramite questo servizio è possibile ottenere, per ogni tipo documento, la propria gestione dei numeratori.

Comando: `get_proprieta_numeratori`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_proprieta_numeratori"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_sigla	String	Sì	Sigla del documento di cui si vogliono le informazioni
dati.in_cod_fornito	String	Cond.	Codice del fornitore — necessario

Campo	Tipo	Obbl.	Descrizione
re			esclusivamente per documenti di tipo FF
dati.in_tipo_documento_el	String	Cond.	Tipo del documento elettronico — necessario esclusivamente per documenti di tipo FF

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_proprieta_numeratori",
  "dati": {
    "in_sigla": "FF",
    "in_cod_fornitore": "601.00001",
    "in_tipo_documento_el": "TD01"
  }
}
```

Esempio Response

```
{
  "max_serie": 1,
  "max_numero_documento": 999999,
  "gest_numerazione_automatica": "N"
}
```

Campi della Response

Campo	Tipo	Descrizione
max_serie	Number	Numero massimo di serie gestite
max_numero_documento	Number	Valore massimo del numeratore
gest_numerazione_automatica	String	“S” o “N” — indica se gestisce la numerazione automatica

Note Operative

- I campi `in_cod_fornitore` e `in_tipo_documento_el` sono necessari esclusivamente per documenti di tipo FF (fatture fornitore).

Servizio: Autenticazione MyDB

Per i soli campi degli archivi MyDB etichettati come “Aut” è possibile gestire l’autenticazione tramite questo servizio.

Il servizio funziona esclusivamente per coppie di credenziali utente+password, quindi se un archivio MyDB contiene più di 2 campi etichettati come “Aut” nel servizio è necessario specificare quali sono i campi sui quali si vuole verificare l’autenticazione.

Comando: `verifica_autenticazione_mydb`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"verifica_autenticazione_mydb"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_codice_archivio	String	Sì	Archivio MyDB contenente i campi di autenticazione
dati.in_nomecampo_login	String	Sì	Campo dell’archivio MyDB designato come campo “username”
dati.in_login_value	String	Sì	Username necessario per l’autenticazione
dati.in_nomecampo_password	String	Sì	Campo dell’archivio MyDB designato come campo “password”
dati.in_password_value	String	Sì	Password associata allo username
dati.in_id	Number	No	ID del record dell’archivio MyDB per il quale si deve verificare l’autenticazione

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "verifica_autenticazione_mydb",
  "dati": {
    "in_codice_archivio": "924148CQWEBAPI@mydb0001",
```

```
    "in_nomecampo_login": "user",
    "in_login_value": "utente1",
    "in_nomecampo_password": "pwd",
    "in_password_value": "pwd1",
    "in_id": 3
  }
}
```

Note Operative

- È possibile specificare il record specifico su cui fare il controllo tramite `in_id`: se viene omesso, il controllo viene fatto su tutti i record e restituisce esito positivo alla prima coppia di credenziali verificata.
- Il servizio funziona esclusivamente per coppie di credenziali (login + password).

Servizio: Calendari di Produzione

Il servizio restituisce la lista di tutti i calendari configurati su Mexal in Produzione → Tabelle → Calendari produzione (Alt+PKA).

Disponibile solo per aziende di livello produzione o superiore.

Comando: `get_calendari_produzione`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	"get_calendari_produzione"

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_calendari_produzione"
}
```

Note Operative

- Il servizio è disponibile solo per aziende di livello produzione o superiore.

- Non richiede parametri aggiuntivi nel campo `dati`.

Servizio: Multilotto

Questo servizio, affiancato agli end-point REST sui lotti, permette di leggere tutti i lotti specificando ulteriori parametri come filtro.

Comando: `get_multilotto`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_multilotto"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_tipo_lotto	String	Cond.	Codice alfanumerico del tipo lotto (es. “aa”)
dati.in_cod_articolo	String	Cond.	Codice articolo
dati.in_tp_cod_utente	Number	No	Modalità di ricerca per codice utente: 0=completo, 1=inizia per, 2=contiene
dati.in_cod_utente	String	No	Codice utente del lotto (case sensitive)
dati.in_includi	Number	No	Tipo inclusione (vedi tabella sotto)
dati.in_nr_magazzino	Number	No	Numero del magazzino (0=tutti i magazzini, da 1 a 255)

Valori di in_includi

Valore	Descrizione
0	Include tutti i lotti del solo archivio corrente, tranne quelli con stato B/N, con data validità inferiore alla data terminale o con quantità 0
1	Include tutti i lotti del solo archivio corrente
2	Come 0 ma solo dei dati nello storico
3	Come 1 ma solo dei dati nello storico
4	Come 0 ma dei dati sia nel corrente che nello storico
5	Come 1 ma dei dati sia nel corrente che nello storico

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_multilotto",
  "dati": {
    "in_tipo_lotto": "<tipo>",
    "in_cod_articolo": "<codice>",
    "in_tp_cod_utente": 0,
    "in_cod_utente": "<codice_utente>",
    "in_includi": 0,
    "in_nr_magazzino": 0
  }
}
```

Note Operative

- `in_tipo_lotto` e `in_cod_articolo` sono mutuamente esclusivi. Se indicati entrambi viene considerato solo `in_cod_articolo`.
- Il campo `in_cod_utente` è facoltativo e case sensitive.
- `in_nr_magazzino`: 0 = tutti i magazzini, da 1 a 255 per selezionare un magazzino specifico.

Servizio: Modifica IBAN e Tipo Invio

Servizio per la modifica massiva di IBAN e/o tipo invio in tutte le deleghe che non risultano definitive o bloccate.

Comando: `cambia_iban_tipo_invio_delega`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"cambia_iban_tipo_invio_delega"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_elenco_aziende	Array	Cond.	Array di aziende con le relative deleghe da modificare (per deleghe contabili)
dati.in_elenco_prat	Array	Cond.	Array di pratiche con le relative

Campo	Tipo	Obbl.	Descrizione
iche			deleghe da modificare (per deleghe dichiarativi)

Per ogni delega nell'array `in_elenco_deleghe`:

Campo	Tipo	Obbl.	Descrizione
in_id_delega	String	Sì	ID della delega (formato GG/MM/AAAA/NN)
in_numero_modulo	String	Sì	Numero modulo
in_dati_delega	Object	Sì	Oggetto con i nuovi dati
in_dati_delega.in_elenco_iban	Array	No	Array con i nuovi IBAN
in_dati_delega.in_codice_invio_delega	String	No	Nuovo codice invio

Esempio Request — Aggiornamento deleghe contabili

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "cambia_iban_tipo_invio_delega",
  "dati": {
    "in_elenco_aziende": [
      {
        "in_sigla": "A03",
        "in_elenco_deleghe": [
          {
            "in_id_delega": "16/02/2026/51",
            "in_numero_modulo": "001",
            "in_dati_delega": {
              "in_elenco_iban": [
                {
                  "in_iban": "IT25H0605524200111111111111"
                }
              ]
            }
          },
          {
            "in_id_delega": "17/11/2025/51",
            "in_numero_modulo": "001",
            "in_dati_delega": {
              "in_elenco_iban": [
                {
```

```
                    "in_iban": "IT25H0605524200111111111111"
                  }
                ],
                "in_codice_invio_delega": "N"
              }
            }
          ]
        }
      ]
    }
}
```

Esempio Request — Aggiornamento deleghe dichiarativi

```
{
  "cmd": "cambia_iban_tipo_invio_delega",
  "dati": {
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "A03",
        "in_interno": "1",
        "in_elenco_deleghe": [
          {
            "in_id_delega": "16/02/2026/51",
            "in_numero_modulo": "001",
            "in_dati_delega": {
              "in_elenco_iban": [
                {
                  "in_iban": "IT25H0605524200111111111111"
                }
              ]
            }
          },
          {
            "in_id_delega": "20/08/2025/51",
            "in_numero_modulo": "001",
            "in_dati_delega": {
              "in_elenco_iban": [
                {
                  "in_iban": "IT25H0605524200111111111111"
                }
              ],
              "in_codice_invio_delega": "N"
            }
          }
        ]
      }
    ]
  }
}
```

Codici di esito

codice_esito	Descrizione
0	Nessun errore
3	Delega GG/MM/AAAA/PR Coordinate bancarie non presenti in tabella: Delega non aggiornata
4	Delega GG/MM/AAAA/PR non trovata. Put non eseguita

Note Operative

- Il servizio modifica solo le deleghe che non risultano definitive o bloccate.
- È possibile modificare sia l'IBAN che il tipo invio nella stessa chiamata.
- Per le deleghe contabili usare `in_elenco_aziende`, per le dichiarativi usare `in_elenco_pratiche`.

Servizio: Inserimento-Aggiornamento IBAN

Il servizio può scrivere una banca con relativo IBAN nell'anagrafica banche per quanto riguarda le aziende, oppure nel quadro GE in caso di pratiche dichiarativi. Inoltre permette anche di modificare l'IBAN abituale.

Comando: `inserisci_aggiorna_iban`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"inserisci_aggiorna_iban"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_elenco_aziende	Array	Cond.	Array di aziende (per operare sull'anagrafica banche)
dati.in_elenco_pratiche	Array	Cond.	Array di pratiche (per operare sulle pratiche dichiarativi)

Per ogni elemento dell'array (azienda o pratica):

Campo	Tipo	Obbl.	Descrizione
in_sigla	String	Sì (az.)	Sigla azienda
in_mod_pratica	String	Sì (pr.)	Modello dichiarativo
in_pratica	String	Sì (pr.)	Codice pratica

Campo	Tipo	Obbl.	Descrizione
in_interno	String	Sì (pr.)	Interno
in_anagrafica_banca	Object	No	Dati anagrafici della banca
in_elenco_iban	Array	No	Array con gli IBAN da inserire/aggiornare

Campi di in_anagrafica_banca

Campo	Tipo	Descrizione
in_abi	String	Codice ABI
in_cab	String	Codice CAB
in_azienda_credito	String	Nome azienda di credito
in_filiale_credito	String	Nome filiale
in_indirizzo	String	Indirizzo
in_cap	String	CAP
in_localita	String	Località
in_provincia	String	Provincia
in_telefono	String	Telefono
in_fax	String	Fax
in_email	String	Email
in_url_homebanking	String	URL homebanking

Campi di in_elenco_iban[]

Campo	Tipo	Descrizione
in_iban	String	Codice IBAN
in_gest_abituale	String	“S” o “N” — gestione IBAN abituale

Codici di esito

codice_esito	Descrizione
0	Nessun errore
3	Delega GG/MM/AAAA/PR Coordinate bancarie non presenti in tabella: Delega non aggiornata
4	Delega GG/MM/AAAA/PR non trovata. Put non eseguita

Note

- Per lavorare sull’anagrafica banche occorre specificare l’array `in_elenco_aziende`.

- Per operare sulle pratiche dichiarativi occorre specificare l'array `in_elenco_pratiche`.
- I dati sono forniti in formato array perché il servizio è stato predisposto per inserimenti multipli ma al momento è possibile inserire **un solo elemento**.
- Se si vogliono modificare solo i campi anagrafici il campo `in_elenco_iban` non deve essere passato.

Esempio Request — Scrittura IBAN nell'anagrafica banche

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "inserisci_aggiorna_iban",
  "dati": {
    "in_elenco_aziende": [
      {
        "in_sigla": "A03",
        "in_anagrafica_banca": {
          "in_abi": "03045",
          "in_cab": "01600",
          "in_azienda_credito": "BANCA AKROS S.P.A.",
          "in_filiale_credito": "SEDE",
          "in_indirizzo": "VIALE EGINARDO, 29",
          "in_cap": "20149",
          "in_localita": "MILANO",
          "in_provincia": "MI",
          "in_telefono": "",
          "in_fax": "",
          "in_email": "",
          "in_url_homebanking": ""
        },
        "in_elenco_iban": [
          {
            "in_iban": "IT41J0304501600411333222555",
            "in_gest_abituale": "N"
          }
        ]
      }
    ]
  }
}
```

Esempio Request — Scrittura IBAN nel quadro GE di una pratica

```
{
  "cmd": "inserisci_aggiorna_iban",
  "dati": {
```

```
    "in_elenco_pratiche": [
      {
        "in_mod_pratica": "RPF",
        "in_pratica": "A05",
        "in_interno": "1",
        "in_anagrafica_banca": {
          "in_abi": "03030",
          "in_cab": "01600",
          "in_azienda_credito": "ROYAL BANK OF SCOTLAND PLC",
          "in_filiale_credito": "MILANO VIA TURATI, 9",
          "in_indirizzo": "VIA TURATI, 9",
          "in_cap": "20121",
          "in_localita": "MILANO",
          "in_provincia": "MI",
          "in_telefono": "",
          "in_fax": "",
          "in_email": "",
          "in_url_homebanking": ""
        },
        "in_elenco_iban": [
          {
            "in_iban": "IT42Z0303001600111333222888",
            "in_gest_abituale": "S"
          }
        ]
      }
    ]
  }
}
```

Servizio: Storia Ordine

Il servizio permette di reperire tutti i documenti relativi ad un dato ordine.

Comando: `get_storico_ordine`

Metodo: POST
URL: `https://base_address/webapi/servizi`

Parametri

Campo	Tipo	Obbl.	Descrizione
cmd	String	Sì	`"get_storico_ordine"`
dati	Object	Sì	Oggetto con i parametri del servizio
dati.in_serie_ordine	Number	Sì	Serie dell'ordine
dati.in_numero_or	Number	Sì	Numero dell'ordine

Campo	Tipo	Obbl.	Descrizione
dine			
dati.in_anno_ordine	Number	Sì	Anno dell'ordine
dati.in_anno_inizio_ricerca	Number	Sì	Anno in cui iniziare la ricerca per i documenti di magazzino derivati dall'ordine

Esempio Request

```
POST /webapi/servizi HTTP/1.1
Host: mioserver:9004
Authorization: Passepartout d2ViYXBpOndlYmFwaQ==
Content-Type: application/json
Coordinate-Gestionale: Azienda=DEM Anno=2024

{
  "cmd": "get_storico_ordine",
  "dati": {
    "in_serie_ordine": 1,
    "in_numero_ordine": 100,
    "in_anno_ordine": 2024,
    "in_anno_inizio_ricerca": 2024
  }
}
```

Campi aggiuntivi nella Response

In uscita, oltre ai campi del documento, saranno restituiti anche i seguenti campi (solo se valorizzati):

Campo	Descrizione
descrizione	Descrizione articolo
qta_preventivo	Quantità preventivo
qta_matrice	Quantità matrice
qta_ordinato	Quantità ordinato
qta_consegnato	Quantità consegnato
qta_residua	Quantità residua

Note Operative

- Il campo `in_anno_inizio_ricerca` indica l'anno in cui iniziare la ricerca per i documenti di magazzino derivati dall'ordine e deve essere sempre >= `in_anno_ordine`.

Servizio: Upload Files

Servizio per effettuare l'upload di file in diversi contesti del gestionale. Il servizio consente di caricare uno o più file associandoli a metadati specifici in base al contesto gestionale indicato.

Comando: `upload_files`

Metodo: POST
URL: `https://base_address/webapi/uploads`

Content-Type: `multipart/form-data`

> ⚠ **Nota Bene:** A differenza degli altri servizi WebAPI che utilizzano Content-Type: application/json, questo servizio richiede Content-Type: multipart/form-data.

Struttura della Richiesta

La richiesta è di tipo **multipart/form-data** e si compone di:

1. **Form metadata** — parte JSON con i metadati da associare ai file. Deve essere il primo form.
2. **Form successivi** — file per i quali viene eseguito l'upload. È possibile allegare più file.

Struttura base del JSON (form metadata)

```
{
  "in_contesto_gestionale": "DV",
  "in_gest_dv_dr": "S",
  "metadati": { ... }
}
```

Parametri comuni

Parametro	Tipo	Obbl.	Descrizione
in_contesto_gestionale	String	Sì	Contesto gestionale. Valori: DV
in_gest_dv_dr	String	No	Effetto solo in contesto DV. Valori: S / N. Ignorato in altri contesti
metadati	Object	Sì	Metadati specifici del contesto

Valori di in_contesto_gestionale

Valore	Descrizione
DV	Docuvision

Comportamento di in_gest_dv_dr

- Ha effetto **esclusivamente** per il contesto DV (Docuvision).

- Se specificato in altri contesti, viene **ignorato**.
- Se non indicato nel contesto Docuvision, il default è “N”.

Contesto Docuvision (DV)

Campi scalari dei metadati

Campo	Tipo	Descrizione
id	numerico intero	Id documento
classe	numerico intero	Classe del documento
tipo_classe	alfanumerico	Tipologia documento della classe
descrizione	alfanumerico	Titolo documento
data_documento	alfanumerico	Data documento
nr_protocollo	alfanumerico	Numero protocollo
dt_protocollo	alfanumerico	Data protocollo
codice	alfanumerico	Codice conto cliente/fornitore
cognome	alfanumerico	Ragione sociale / Cognome
nome	alfanumerico	Nome
partita_iva	alfanumerico	Partita IVA
cod_ufficio_pa	alfanumerico	Codice ufficio PA
nota	alfanumerico	Nota
sigla_azienda	alfanumerico	Sigla azienda
allegato_classi	alfanumerico	Allegato a
codice_fiscale	alfanumerico	Codice fiscale
codice_sdi	num. con decimali	Codice SDI
gestione_doc	alfanumerico	Gestione documento
id_pacchetto	numerico intero	Id pacchetto
stato_documento	alfanumerico	Stato documento
gest_fatt_el	alfanumerico	Tipo fattura elettronica
stato_fatt_el	alfanumerico	Stato fattura elettronica
tipo_import	alfanumerico	Tipo import
tipo_doc_el	numerico intero	Tipologia documento elettronico
tipo_contratto	alfanumerico	Tipo contratto
gest_pubblica	alfanumerico	Pubblica
gest_letto	alfanumerico	Letto
dt_compet_invio	alfanumerico	Data competenza invio

Campo	Tipo	Descrizione
codice_distinta	alfanumerico	Codice distinta
stato_fattura	alfanumerico	Stato fattura

Campi di tipo array dei metadati

Campo	Tipo	Descrizione
numero_versione	array num. intero	Numero versione
ext_file	array alfanumerico	Estensione file
nota_1	array alfanumerico	Nota 1
numero_pagine	array num. intero	Numero di pagine
volume_versione	array num. intero	Volume versione
autore	array alfanumerico	Autore
data_versione	array alfanumerico	Data versione
ora_versione	array alfanumerico	Ora versione
nota_2	array alfanumerico	Nota 2
ext_firma	array alfanumerico	Estensione firma
ext_marca	array alfanumerico	Estensione marca
dt_trasmissione	array alfanumerico	Data trasmissione
ora_trasmissione	array alfanumerico	Ora trasmissione
tipo_ricevuta	array alfanumerico	Tipo ricevuta
id_autore	array num. intero	Id autore
presa_visione	array alfanumerico	Estensione presa visione
prog_versione	array alfanumerico	Release
nome_prodotto	array alfanumerico	Nome prodotto
file_origine	array alfanumerico	Origine file

Esempio Request — Contesto Docuvision

Form metadata (JSON):

```
{
  "in_contesto_gestionale": "DV",
  "metadati": {
    "id": 0,
    "numero_versione": [0, 0],
    "classe": 1400,
    "descrizione": "Titolo documento DV di prova per upload via webapi",
    "nota_1": ["nota rev.1", "nota_rev.2"],
    "nota_2": ["ult.nota rev 1", "ult.nota rev 2"],
    "data_documento": "20260416",
    "cognome": "Moretti",
```

```
    "nome": "Giovanni"
  }
}
```

Form file:

I form successivi al metadata contengono i file binari da caricare. È possibile allegare più file nella stessa richiesta.

Note Operative

- L'header `Authorization` è obbligatorio come per tutti gli altri servizi WebAPI.
- L'header `Coordinate-Gestionale` è richiesto per identificare azienda e anno di esercizio.
- Il parametro `in_contesto_gestionale` è **sempre obbligatorio** indipendentemente dal contesto.
- A differenza degli altri servizi, l'URL è `/webapi/uploads` (non `/webapi/servizi`).
- Il Content-Type deve essere `multipart/form-data` (non `application/json`).

Appendici

Codici di Errore

Questa sezione documenta i codici di errore restituiti dalle Passepartout WebAPI Mexal/Passcom. Ogni risposta di errore è un oggetto JSON con una struttura standard che include un codice HTTP, un codice errore numerico applicativo e dettagli diagnostici.

Struttura dell'oggetto errore

Quando una richiesta non può essere elaborata con successo, la response contiene un oggetto JSON `error` con i seguenti campi:

```
{
  "error": {
    "response-code": 400,
    "response-detail": "6001 - errore gestionale [Aliquota iva obbligatoria]",
    "request-uri": "/webapi/risorse/articoli",
    "response-message": "Bad Request",
    "warsion": "00700",
    "request-method": "POST",
    "request-id": "955f5722-b89b-45ee-a2f4-4fab18216cdb",
    "remote": "192.168.3.180:57716",
    "timestamp": "11/06/2026 09:18:23"
  }
}
```

Campo	Tipo	Descrizione
`response-code`	Integer	Codice HTTP della response (400, 401, 404, 500, ecc.)
`response-detail`	String	Codice errore numerico + descrizione testuale. Formato: `<codice> - <messaggio> [<dettaglio>]`
`request-uri`	String	URI della richiesta che ha generato l'errore
`response-message`	String	Messaggio HTTP standard corrispondente al `response-code`
`warsion`	String	Versione interna del motore WebAPI
`request-method`	String	Metodo HTTP della richiesta originale (GET, POST, PUT, DELETE)
`request-id`	String	UUID univoco della richiesta, utile per il tracciamento nei log del server
`remote`	String	Indirizzo IP e porta del client che ha effettuato la

Campo	Tipo	Descrizione
		richiesta
`timestamp`	String	Data e ora del server in cui si è verificato l'errore (formato: `dd/MM/yyyy HH:mm:ss`)

Nota: Il campo `response-detail` è il più importante per la diagnosi. Contiene sempre un codice numerico seguito dal trattino e dalla descrizione dell'errore. Eventuali dettagli aggiuntivi sono racchiusi tra parentesi quadre.

Codici HTTP utilizzati

Codice	Significato	Uso nelle WebAPI
200	OK	La richiesta è stata elaborata con successo
400	Bad Request	Parametri non validi, campi mancanti, errore gestionale, coordinate assenti
401	Unauthorized	Autenticazione fallita: credenziali assenti, non valide o utente non abilitato
404	Not Found	Endpoint non esistente oppure risorsa (chiave) non trovata nell'archivio
500	Internal Server Error	Errore imprevisto del server, conflitto uso esclusivo, errore pool servizi
408	Request Timeout	Il servizio ha superato il tempo massimo di esecuzione (5 minuti)
503	Service Unavailable	Pool dei servizi esaurito (tutti i terminali dell'utente WebAPI sono occupati)

Codici di errore applicativi

Il campo `response-detail` contiene un codice numerico che identifica la famiglia dell'errore. I codici principali rilevati nelle WebAPI Mexal/Passcom sono:

Codice 1001 — Coordinate gestionale mancanti

HTTP	Formato `response-detail`	Causa
400	`1001 - coordinate gestionale non presenti`	L'header `Coordinate-Gestionale` è assente ma l'endpoint richiede l'accesso a un'azienda specifica

Esempio response:


```
{
  "error": {
    "response-code": 400,
    "response-detail": "1001 - coordinate gestionale non presenti",
    "request-uri": "/webapi/risorse/articoli",
    "response-message": "Bad Request",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "bd7d6d80-2179-49ac-858e-344a3e5bb517",
    "remote": "192.168.3.180:54440",
    "timestamp": "11/06/2026 09:18:35"
  }
}
```


Soluzione: Aggiungere l'header `Coordinate-Gestionale` con almeno `Azienda=XXX Anno=YYYY`. Solo gli endpoint con `coordinate: false` (es. `dati-generali/installazione`, `aziende`, `anagrafica-unica`) non richiedono questo header.

Codice 3002 — Errore pool WebAPI

HTTP	Formato `response-detail`	Causa
401	`3002 - errore in fase di richiesta webapi dal pool [notifica server errore] [Utente o password non validi - Accesso non possibile] [WebapiHttpException]`	Credenziali non valide nell'header `Authorization`
500	`3002 - errore in fase di richiesta webapi dal pool [notifica server errore] [L'utente sul terminale N sta utilizzando una funzione in uso esclusivo...] [WebapiHttpException]`	Conflitto con un altro terminale che ha in uso esclusivo una funzione
503	`3002 - errore in fase di richiesta webapi dal pool [pool esaurito]`	Tutti i servizi del pool utente sono occupati (max 5 simultanei per utente)

Il codice 3002 identifica problemi a livello di **pool dei servizi** — il layer intermedio che gestisce l'allocazione dei terminali Mexal/Passcom all'utente WebAPI.

Esempio — autenticazione fallita (HTTP 401):


```
{
  "error": {
```

```
    "response-code": 401,
    "response-detail": "3002 - errore in fase di richiesta webapi dal pool
[notifica server errore] [Utente o password non validi - Accesso non possibile]
[WebapiHttpException]",
    "request-uri": "/webapi/risorse/articoli?max=1",
    "response-message": "Unauthorized",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "45a192d7-cdf0-4aa8-a9ee-c06eecdab6a8",
    "remote": "192.168.3.180:41934",
    "timestamp": "11/06/2026 09:19:01"
  }
}
```


Esempio — uso esclusivo (HTTP 500):


```
{
  "error": {
    "response-code": 500,
    "response-detail": "3002 - errore in fase di richiesta webapi dal pool
[notifica server errore] [L'utente sul terminale 1 sta utilizzando una funzione in
uso esclusivo che non consente ad altri utenti di connettersi]
[WebapiHttpException]",
    "request-uri": "/webapi/risorse/prima-nota?max=1",
    "response-message": "Internal Server Error",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "e36d0e4d-2db1-4b6a-b957-3916b047b30c",
    "remote": "192.168.3.180:37592",
    "timestamp": "11/06/2026 09:20:08"
  }
}
```


Soluzioni: - **Autenticazione:** Verificare che l'header `Authorization: Passepartout <base64(USER:PWD)>` contenga credenziali valide. Per installazioni Live aggiungere `Dominio=<DOMINIO>`. - **Uso esclusivo:** Attendere che l'utente sul terminale in conflitto esca dalla funzione esclusiva, oppure riprovare dopo qualche secondo. - **Pool esaurito:** Attendere che uno dei servizi si liberi (timeout automatico dopo 5 minuti di inattività) oppure ridurre il numero di richieste parallele.

Codice 5001 — Errore imprevisto inoltro comando

HTTP	Formato `response-detail`	Causa
500	`5001 - errore imprevisto in fase di inoltro comando [<dettaglio>] [WebapiHttpException]`	Interruzione nel dialogo tra il modulo WebAPI e il server

HTTP	Formato `response-detail`	Causa gestionale

Questo errore indica un problema di comunicazione interna tra il modulo WebAPI e il processo Mexal/Passcom. Spesso si manifesta quando il gestionale non riesce a elaborare la richiesta (es. chiave non valida, prerequisito mancante).

Esempio response:

```
{
  "error": {
    "response-code": 500,
    "response-detail": "5001 - errore imprevisto in fase di inoltro comando
[codice conto obbligatorio] [WebapiHttpException]",
    "request-uri": "/webapi/risorse/clienti/INESISTENTE",
    "response-message": "Internal Server Error",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "23bec8e0-18e2-45a2-828b-3dcbf5a6a1c5",
    "remote": "192.168.3.180:49258",
    "timestamp": "11/06/2026 09:19:46"
  }
}
```

Soluzione: Analizzare il testo tra parentesi quadre per capire il prerequisito mancante. Se il problema persiste, verificare la coerenza dei dati e la raggiungibilità del server gestionale.

Codice 6001 — Errore gestionale

HTTP	Formato `response-detail`	Causa
400	`6001 - errore gestionale [<messaggio specifico>]`	Il gestionale ha rifiutato l'operazione per un vincolo funzionale

Questo è il codice più frequente. Il testo tra parentesi quadre descrive la causa specifica dell'errore gestionale. Di seguito i messaggi più comuni rilevati:

Messaggio `[dettaglio]`	Causa	Soluzione
`[azienda XXX non valida]`	La sigla azienda nell'header `Coordinate-Gestionale` non esiste	Verificare la sigla azienda con `GET /webapi/risorse/aziende`
`[Nome campo 'xxx' non valido]`	Un campo nel	Verificare i nomi dei

Messaggio [dettaglio]	Causa	Soluzione
	body JSON non è riconosciuto dall'endpoint	campi con `?info=true` sull'endpoint
`[Nome campo filtro 'xxx' non valido]`	Un campo nel filtro di ricerca non è valido per l'endpoint	Verificare i campi filtrabili con `?info=true`
`[Mancano dei campi chiave: xxx]`	Un campo chiave obbligatorio non è presente nella richiesta POST	Includere tutti i campi chiave necessari per la creazione
`[Aliquota iva obbligatoria]`	Campo `alq_iva` mancante nella creazione di un articolo	Specificare il codice aliquota IVA
`[Unita' di misura mancante]`	Campo `um_principale` mancante nella creazione di un articolo	Specificare l'unità di misura principale
`[Valuta costo standard (0) fuori dai limiti [ 1 - 30 ]]`	Un campo numerico è fuori dall'intervallo ammesso	Correggere il valore secondo i limiti indicati
`[Articolo senza Distinta base]`	Richiesta di sviluppo DBA su un articolo che non ne ha una	Verificare che l'articolo abbia una distinta base configurata
`[Tipo documento obbligatorio]`	Il servizio `totali_documento` richiede il campo `tipo` nel payload	Includere il campo `tipo` (es. `"OC"`, `"FT"`)

Messaggio [dettaglio]	Causa	Soluzione
[cmd xxx sconosciuto]	Il comando specificato nel canale servizi non esiste	Verificare il nome del comando disponibile

Esempio response — campo non valido:

```
{
  "error": {
    "response-code": 400,
    "response-detail": "6001 - errore gestionale [Nome campo 'campo_inesistente'
non valido]",
    "request-uri": "/webapi/risorse/articoli/ART001",
    "response-message": "Bad Request",
    "warsion": "00700",
    "request-method": "PUT",
    "request-id": "911f829c-d073-4b5f-b769-60e95a18995e",
    "remote": "192.168.3.180:49858",
    "timestamp": "11/06/2026 09:17:59"
  }
}
```

Esempio response — azienda non valida:

```
{
  "error": {
    "response-code": 400,
    "response-detail": "6001 - errore gestionale [azienda AZIENDA_INESISTENTE non
valida]",
    "request-uri": "/webapi/risorse/articoli",
    "response-message": "Bad Request",
    "warsion": "00700",
    "request-method": "GET",
    "request-id": "677eba3f-cc82-465e-9090-d49f89ce2c45",
    "remote": "192.168.3.180:57406",
    "timestamp": "11/06/2026 09:17:23"
  }
}
```

HTTP 404 — Endpoint o risorsa non trovata

Gli errori 404 **non contengono un oggetto JSON `error`** strutturato nella maggior parte dei casi. La response è un semplice 404 senza body quando:

- Il path dell'endpoint non corrisponde a nessuna risorsa WebAPI disponibile (es. `/webapi/risorse/endpoint-inesistente`)

- La chiave specificata nel path non corrisponde a nessun record nell'archivio (es. `/webapi/risorse/articoli/CODICE_INESISTENTE`)

Soluzione: Verificare il nome dell'endpoint con `GET /webapi/risorse/help?extended=true` oppure la chiave della risorsa con una richiesta di lista (GET senza chiave).

Riepilogo diagnostico rapido

response-code	Codice nel detail	Famiglia errore	Azione
400	1001	Coordinate mancanti	Aggiungere header `Coordinate-Gestionale`
400	6001	Errore gestionale	Leggere il dettaglio tra `[...]` e correggere i dati
401	3002	Autenticazione	Verificare credenziali e formato header `Authorization`
404	—	Non trovato	Verificare endpoint o chiave risorsa
500	3002	Conflitto pool	Attendere liberazione del terminale in uso esclusivo
500	5001	Errore inoltro	Verificare prerequisiti del comando e raggiungibilità server

response-code	Codice nel detail	Famiglia errore	Azione
408	—	Timeout	Riprovare; ridurre complessità operazione
503	3002	Pool esaurito	Attendere liberazione servizi (max 5 per utente)

Best practice per la gestione errori

1. **Controllare sempre `response-code`** prima di tentare il parsing del body
2. **Analizzare `response-detail`** — il codice numerico identifica la famiglia, il testo tra `[...]` fornisce il dettaglio
3. **Usare `request-id`** per correlare la response di errore con i log lato server in caso di escalation al supporto
4. **Implementare retry con backoff** per errori 500/3002 (uso esclusivo) e 503 (pool esaurito)
5. **Non riprovare** gli errori 400 e 401 senza correggere la richiesta — sono deterministici
6. **Monitorare `warsion`** per verificare la versione del motore WebAPI in caso di comportamenti inattesi

⚠ **Nota Bene:** Il codice 6001 è un "contenitore generico" per tutti gli errori di validazione e vincolo del gestionale Mexal/Passcom. Il messaggio tra parentesi quadre è dinamico e dipende interamente dall'operazione specifica e dallo stato dei dati. Non è possibile elencare esaustivamente tutti i messaggi — vanno interpretati caso per caso leggendo il dettaglio.

Glossario

Di seguito la raccolta alfabetica di tutti i termini tecnici specifici di Passepartout WebAPI utilizzati in questo manuale.

Ambiente Live
Installazione del gestionale Mexal/Passcom presente in server farm Passepartout (cloud), raggiungibile tramite l'indirizzo `https://services.passepartout.cloud/`.

Ambiente locale
Installazione del gestionale Mexal/Passcom presente su un server locale alla rete del cliente (housing).

Back-end
Parte di un programma che elabora i dati acquisiti dal front-end e restituisce i risultati.

Base64
Schema di codifica che converte dati binari o testuali in una stringa di caratteri ASCII. In WebAPI viene utilizzato per codificare le credenziali di autenticazione (USER:PWD) nell'header `Authorization`.

Body
Corpo di una HTTP request in cui è presente il dato da scambiare tra client e server, tipicamente in formato JSON.

Changelog
Registro cronologico delle modifiche apportate alle API, organizzato per versione. Ogni versione è documentata in un file separato nella sotto-cartella `capitoli/05_appendici/changelog/`.

Coordinate-Gestionale
Header HTTP personalizzato di Passepartout WebAPI che specifica il contesto operativo della richiesta: azienda, sotto-azienda, anno di esercizio e magazzino di default.

CRUD
Acronimo di **C**reate, **R**ead, **U**pdate, **D**elete. Indica le quattro operazioni fondamentali sulle risorse, implementate rispettivamente tramite i metodi HTTP POST, GET, PUT, DELETE.

Distinta base
Struttura dati che descrive la composizione di un prodotto finito in termini di fasi di lavorazione e componenti. In WebAPI è gestita tramite l'entità `/distinte-base`.

End-point
Entità alla fine di una connessione; indirizzo URL specifico che identifica una risorsa su cui è possibile eseguire operazioni tramite le API.

Entità
Sinonimo di risorsa ed end-point nel contesto di questo manuale. Rappresenta un archivio del gestionale Mexal/Passcom esposto tramite le API REST.

Front-end
Parte di un programma responsabile dell'acquisizione dei dati di ingresso e dell'interazione con l'utente.

Header
Parte della HTTP request in cui vengono impostati metadati necessari al completamento della richiesta stessa (es. `Authorization`, `Content-type`, `Coordinate-Gestionale`).

HTTP
Hypertext **T**ransfer **P**rotocol. Protocollo di comunicazione utilizzato per il trasferimento di dati sul web.

HTTPS
Hypertext **T**ransfer **P**rotocol over **S**ecure Socket Layer. Versione sicura di HTTP. WebAPI gestisce solo richieste tramite protocollo sicuro HTTPS.

HTTP request
Azione (POST, GET, PUT, DELETE) eseguita su una risorsa identificata da un dato URL.

Installazione
Istanza del programma gestionale Mexal/Passcom, sia essa locale o in cloud.

JSON
Java**S**cript **O**bject **N**otation. Formato standard per l'interscambio di dati tra applicazioni client/server, utilizzato da WebAPI per request body e response.

MDS
Suite di servizi Passepartout attivabile tramite YouPass che include, tra gli altri, il modulo WebAPI.

Pool di servizi
Insieme di connessioni concorrenti disponibili per ciascun utente WebAPI. Ogni utente del gruppo Servizi WebAPI ha a disposizione un massimo di 5 servizi contemporanei.

Query string
Parte di un URL che contiene dati da passare in input ad un servizio, posta dopo il carattere ? e composta da coppie chiave=valore separate da &.

REST / RESTful
Representational **S**tate **T**ransfer. Stile architetturale per la progettazione di servizi web, fortemente orientato alle risorse. WebAPI segue questo paradigma.

Risorsa
Oggetto presente sul gestionale Mexal/Passcom su cui possono essere effettuate operazioni tramite le API (es. articoli, clienti, documenti).

Timeout
Intervallo di tempo massimo di inattività (5 minuti) dopo il quale un servizio WebAPI del pool viene liberato automaticamente, ad eccezione del primo servizio che rimane sempre attivo.

Uso esclusivo
Modalità operativa del gestionale Mexal/Passcom in cui determinate procedure (es. LiveUpdate, anagrafica azienda) richiedono la chiusura di eventuali processi WebAPI attivi per poter proseguire senza interferenze.

WebAPI
Collezione di API Passepartout basate su tecnologia web tramite protocollo HTTPS, che espone le funzionalità del gestionale Mexal/Passcom secondo lo stile architetturale RESTful.

YouPass
Portale Passepartout per l'attivazione e la gestione dei servizi e delle licenze, tramite il quale si abilita la suite MDS che include il modulo WebAPI.

Changelog v3.2

Versione gestionale: 2026G (v888 — 88800)

Nuovi Campi

- Aggiunti due nuovi campi in **testata documento** per l'endpoint Ordini Clienti (`/documenti/ordini-clienti`):

Campo	Tipo	Range	Descrizione
`tipocessione`	Numerico	0–99	Tipo Cessione ai fini del Rimborso Monofase (formato a due cifre)
`variazcontr`	Numerico	0–2	Variazione Contrattuale: 0=nessuna, 1=Conto estimatorio, 2=Variazioni contrattuali

I campi sono disponibili in lettura (`GET`), inserimento (`POST`) e modifica (`PUT`).

> ⚠ **Nota Bene:** I valori vengono memorizzati solo se l'azienda è di tipo RSM (San Marino) con fattura elettronica interna attiva, invio fatture al tributario attivo e il cliente è RSM non privato. Se le condizioni non sono soddisfatte, i campi vengono accettati senza errore ma il valore salvato sarà 0.

- Aggiunti nuovi campi nell'output del servizio dei progressivi delle ubicazioni **solo per Paddy WMS**: get_prog_ubicazioni

Campo	Tipo	Descrizione
`qta_picking`	Numerico	Qauntità di picking
`qta_picking_tg`	Numerico	Qauntità di picking a taglie
`tara_picking`	Numerico	Tara di picking
`colli_picking`	Numerico	Colli di picking

> ⚠ **Nota Bene:** I campi sono disponibili anche per client differenti da Paddy ma sempre valorizzati con array vuoto []

Correzione anomalie

Mancata valorizzazione variabile **conto_corren_b** su lista dell'endpoint *conti* Il problema si presentava solo sulla lista e non nell'interrogazione del singolo conto.

Correzione al manuale

Per l'end-point progressivi-articoli era stata erroneamente indicata la possibilità di inserire parametri in query string come *next*, *max*, *fields*.

Per l'end-point scadenzario non era documentato il parametro di query *tipo*, che permette di selezionare il tipo di scadenzario da consultare: *D* (a documento), *P* (a partite), *X* (extracontabile). Il parametro è ora descritto sia per la lettura della lista (GET) sia per la ricerca (POST).

Per il servizio get_prog_ubicazioni è stato chiarito che i filtri possono essere fatti esclusivamente sui campi *id* e *dt_mod_ult_doc* (data ultima modifica di un progressivo). Non sono ammessi altri campi, compreso *data_ult_mod* che si riferisce all'anagrafica dell'ubicazione.

Changelog v3.1

Versione gestionale: 2026F (v887 — 88700)

Nuovi Campi

Aggiunti nuovi campi dedicati all'integrazione con **Paddy WMS** per la gestione degli abbassamenti da Stock a Picking.

> ⚠ **Nota Bene:** Le funzionalità descritte in questa sezione sono disponibili esclusivamente per installazioni con modulo **Zerodo** attivo nel codice contratto.

- Nuovo parametro di magazzino nell'endpoint Dati Generali (`/dati-generali/parametri-aziendali`):

Campo	Sezione	Tipo	Descrizione
`gest_abbassamenti_magazzino_paddy_wms`	`parametri_magazzino`	Alfanumerico	Attivazione gestione abbassamenti da Stock a Picking per Paddy WMS (S/N)

- Nuovi campi sulla risorsa Articoli, operativi quando il parametro sopra è attivo:

Campo	Tipo	Descrizione
`sco_min_ubic_dft`	Numerico	Scorta minima dell'ubicazione predefinita
`sco_max_ubic_dft`	Numerico	Scorta massima dell'ubicazione predefinita

I campi sono disponibili su tutti i comandi: inserimento (POST), lettura lista e singolo (GET), modifica (PUT).

> ⚠ **Nota Bene:** Per la corretta persistenza dei valori in scrittura, l'articolo deve avere un'ubicazione predefinita di tipo Picking (`area=P`). Senza questo prerequisito, i campi sono leggibili ma la scrittura non persiste.

Correzioni

- **Articoli — indicatori tipo descrizione su righe vuote**: nell'inserimento di articoli di tipo testo variabile (V), gli indicatori di tipo descrizione (`"."` o `","`) venivano anteposti anche alle righe descrittive prive di testo. Vedi Entità: Articoli.

Changelog v3.0

Versione gestionale: 2026E (v886 - 88600)

Nuova versione del manuale

Il manuale WebAPI è stato aggiornato sia nella struttura che nei contenuti inserendo tutti gli end-point presenti corredati con alcuni esempi.

Correzioni

- L'end-point *scadenzario* non gestiva i parametri di query string *max* e *next*
- Nel manuale sono stati inseriti i codici di errore relativi ai servizi *deleghe-dichiarativi*, *modifica-aggiorna-iban*, *inserisci-aggiorna-iban*

Nuovi Campi

Gestione delle variabili riportabili nei documenti lettura/scrittura: *vid_doc_rip_testa*, *vid_doc_rip_riga*.

Entrambi sono array a 2 indici: il primo è "dinamico" (numero videata per testata, oppure riga per la riga), il secondo va da 1 a 26 (max 26 campi per videata).

- Testata: formato [[nr_videata, nr_campo, valore], …] è fino a 26 campi per videata
- Riga: formato [[nr_riga, nr_campo, valore], …] è fino a 26 campi per riga, max 627 righe articolo
- Tipo: sempre Alfanumerico (stringhe)

Changelog v2.34

Versione gestionale: 2026D (v885 - 88500)

Nuovi Campi

Nell'end-point *movimenti-magazzino* sono stati aggiunti i seguenti campi per la gestione del pos e dei ticket:

- pos_carta: Importo pagato tramite POS/carta di credito
- ticket: Importo pagato tramite ticket restaurant
- num_pzticket: Numero di pezzi ticket restaurant utilizzati

Nell'end-point *ordini-clienti/righe* è stato esposto il campo *nr_rif_doc_colleg*

Nuove Funzionalità

Nel servizio *lista_docdv* è stato aggiunto il parametro in_gest_dv_dr. Se impostato a S viene usato il db dei redditi invece di quello aziendale

Changelog v2.33

Versione gestionale: 2026C (v884 - 88400)

Nuovi Servizi

- Servizio per l'upload di file (al momento implementato per Docuvision). Vedi Servizio: Upload Files.

Nuove Funzionalità

- Nel servizio delle condizioni documento è stato aggiunto anche il parametro *cod_agente* per impostare un agente predefinito

Changelog v2.32

Versione gestionale: 2026B (v883 - 88300)

Nuovi Campi

Negli end-point *clienti* e *fornitori* è stato aggiunto il campo *centro_cos_ric*:

```
{
  "nome": "centro_cos_ric",
  "descrizione": "Condizioni generali - Centro di costo/ricavo",
  "tipo": "Numerico"
}
```

Nuove Funzionalità

Negli end-point dei dati-generali sono stati completati tutti i metodi CRUD:

- dati-generali/zone-clienti-fornitori
- dati-generali/nature-articoli
- dati-generali/categorie-statistiche-articoli
- dati-generali/aspetto-esteriore-beni
- dati-generali/categorie-sconti
- dati-generali/categorie-provvigioni
- dati-generali/categorie-prezzi

- dati-generali/categorie-sconti-articoli
- dati-generali/sconti-quantita
- dati-generali/categorie-provvigioni-articoli
- dati-generali/taglie
- dati-generali/fasi-lavorazione

Nuovi Servizi

- E' stato aggiunto il servizio per la lettura della storia ordini: *get_storico_ordine*

Changelog v2.31

Versione gestionale: 2026A (v882 - 88200)

Nuovi Servizi

- E' stato aggiunto il servizio per la scrittura e la modifica dell'IBAN. Per i dettagli si veda Servizio: Inserimento e Aggiornamento IBAN.

Nuovi Campi

- Nel servizio *dichiarazioni_iva_annuali* è stato aggiunto anche il campo *data_fine_pratica*
- Nell'end-point dei movimenti di magazzino e degli ordini è stato esposto il campo relativo alla banca di appoggio: *nr_banca_appo*
- Nell'end-point dei movimenti di magazzino sono stati esposti i campi relativi ai riferimenti dei registratori di cassa telematici (RT):
    - rt_nr_chiusura: num chiusura giornaliera documento di riferimento (Numerico)
    - rt_nr_doc_rif: numero del documento di riferimento (Numerico)
    - rt_data_doc_rif: data del documento di riferimento (Alfanumerico)
    - rt_matr_doc_rif: matricola Registratore Telematico (Alfanumerico)
- Nell'end-point del piano dei conti è stato aggiunto il campo *logis_spl_pay*:
    - logis_spl_pay: Split payment non ente pubblico - Logistica (Alfanumerico)

Correzioni

I servizi di stampa di WebAPI (es. stampa_ordine_cliente) erano affetti dai seguenti problemi:

- non funzionavano con moduli di stampa di tipo PDF-A
- in alcuni casi non impaginavano bene il testo con particolari moduli di stampa

Changelog v2.30

Versione gestionale: 2025I (v881 - 88100)

Nuove Funzionalità

- Il servizio sul calcolo dei totali documento ora restituisce anche i totali di riga

Nuovi Campi

Nell'end-point *articoli* sono stati esposti i seguenti campi:

- data_obbl: Tipo lotto - data validità obbligatoria (Alfanumerico)
- dati_tp_lotto_mod: Tipo lotto - dati tipo lotto modificati (Alfanumerico)
- giorni_autocod: Tipo lotto - giorni per autocodifica (Numerico)
- giorni_manuale: Tipo lotto - giorni per codifica manuale (Numerico)
- giorni_giacenza: Tipo lotto - giorni minimi giacenza magazzino (Numerico)

Correzioni

- Il servizio *stampa_ordine_cliente* funzionava per tutti i tipi di documento non solo per gli OC innescando comportamenti inattesi
- L'indicazione sull'azienda/pratica collegata veniva riportata solo quando il filtro era fatto per range di date

Changelog v2.29

Versione gestionale: 2025H (v880 - 88000)

Nuovi Endpoint

- Aggiunto end-point per gli articoli abbinati. Vedi Entità: Articoli Abbinati.

Nuovi Servizi

- Aggiornamento deleghe non definitive con nuovo IBAN e tipo di invio F24. Vedi Servizio: Modifica IBAN e Tipo Invio nelle Deleghe.

Nuove Funzionalità

- Nell'end-point documenti/lavorazione/bolle in caso di GET di una bolla presente nel piano di lavoro bisogna sommare 10.000.000 al numero della bolla (es. piano di lavoro 5 -> GET 10000005+0)

- Nel servizio sui totali del documento è stata aggiunta la possibilità di fare il calcolo dei totali anche su un nuovo documento. Prima poteva essere fatto solo su un documento esistente.
- Nell'end-point *alias-articoli* sono stati implementati anche i restanti metodi CRUD. Per i comandi GET, PUT e DELETE la chiave si compone con codice_alias+progressivo+codice_articolo (es. GET alias-articoli/test+1+121212)
- Nel servizio sulla lettura delle dichiarazioni iva annuali è stato aggiunto il filtro anche per data_fine_pratica

Nuovi Campi

- Nell'end-point *articoli* sono stati aggiunti i campi relativi alle unità di misura aggiuntive:
    - um_secondaria_A: Unità di misura alternativa A (Alfanumerico)
    - um_secondaria_B: Unità di misura alternativa B (Alfanumerico)
    - coeff_conv_A: Coefficiente conversione A (Numerico con virgola)
    - coeff_conv_B: Coefficiente conversione B (Numerico con virgola)
    - destin_um1: Destinazione unità di misura primaria (Alfanumerico)
    - destin_um2: Destinazione unità di misura alternativa (Alfanumerico)
    - destin_um2_A: Destinazione unità di misura alternativa A (Alfanumerico)
    - destin_um2_B: Destinazione unità di misura alternativa B (Alfanumerico)
- Nell'end-point *dati-generali/parametri-aziendali* è stato aggiunto il campo *gest_altre_unita_di_misura*
- Negli end-point *clienti* e *fornitori* sono stati aggiunti i campi sull'intestazione iban:
    - intesta_iban: Banca appoggio cliente/fornitore - Intestatario Iban bonifici (Alfanumerico)
    - intesta_iban_agg: Ult.banche d'appoggio - Intestatario Iban bonifici (Alfanumerico, dimensione_array: 3)
- Nell'end-point *conti* è stato aggiunto il campo sull'intestatario IBAN:
    - intesta_iban_ban: Intestazione Iban (Alfanumerico)

Correzioni

- In caso di ricerca con filtro su archivi MyDB, l'eventuale NEXT prodotto non funzionava (Sol.79973)

Changelog v2.28

Versione gestionale: 2025G2 (v878B - 87802)

Correzioni

- A causa di uno sporcamento di memoria che avveniva solo su sistemi Linux Rocky9 il servizio WebAPI *leggi_comunicazioni_lipe* si interrompeva con questo errore: “5001 - errore imprevisto in fase di inoltro comando [interruzione inaspettata nel dialogo con il server]” (Sol. 79725)
- Il file usato per la gestione dei NEXT veniva sempre aperto e tenuto in blocco per tutta la durata dell’elaborazione bloccando altre sessioni (Sol. 79699)

Changelog v2.27

Versione gestionale: 2025F3 (v877C - 87703)

Correzioni

- In un OC con riga movimentata in seconda unità di misura e con quantità senza decimali, in trasformazione da sprix/webapi se si evadeva parzialmente la riga introducendo i decimali veniva calcolato erroneamente un residuo intero. Nel caso che l’intero fosse 0 la procedura falliva con “errore gestionale [Quantita’ obbligatoria]”
- Se in una tabella MyDB era definito un campo con ordinamento la lettura del campo via WebAPI falliva con un errore di questo tipo: “ID non valido del campo …”

Modifiche

- Nel servizio spezzariga_bl, nella descrizione del servizio, era stato erroneamente indicato il campo quantita_taglie.
- Aggiunto nel primo paragrafo del capitolo 3 il dettaglio su come interrogare l’help in linea.

Changelog v2.26

Versione gestionale: 2025F (v877 - 87700)

Nuovi Endpoint

- è stato aggiunto l’end-point con la lista delle classi Docuvision solo GET lista: *dati-generali/classi-docuvision*

Nuovi Servizi

- è stato aggiunto il servizio per la ricerca dei lotti specificando ulteriori parametri come codice articolo o codice utente. Vedi Servizio: Multilotto.

Nuove Funzionalità

- Nell'end-point *documenti/movimenti-magazzino* in fase di trasformazione da OC/OX il campo *tipo_stato_riga* accetta come parametro anche la "e" come avviene in emissione/revisione documenti. In questo modo, in caso di evasione parziale di una riga, il residuo rimane sul documento originale (OC/OX) in stato "E".
- Ottimizzata la lettura degli archivi MyDB tramite end-point di ricerca passando da ordini di grandezza di secondi a millisecondi

Correzioni

- Nel caso di POST o PUT su documenti di tipo ordine-cliente o ordine-fornitore in presenza di righe che gestivano più lotti poteva erroneamente essere scritto un lotto appartenente ad un altro articolo senza segnalare alcun errore.
- Nell'end-point articoli (lista) veniva mostrato, facendo *info=true*, il campo *descr_lingua* anche se non effettivamente presente tra i campi disponibili.

Modifiche

- E' stato specificato meglio che il *case_insensitive* si applica per tutti gli operatori che coinvolgono stringhe

Changelog v2.25

Versione gestionale: 2025E1 (v876A - 87601)

Nuovi Campi

Nel servizio *leggi_pratica_dr* sono stati aggiunti, oltre ai già presenti campi sul conteggio immobili, anche quelli sul conteggio dei terreni: *nr_terreni*, *nr_terreni_prop*, *nr_terreni_no_pro*.

Prima era tutto conteggiato come immobile.

Modifiche

Nel servizio *get_allegato_archivio* i campi *id_dv* e *codice* sono mutuamente esclusivi

Changelog v2.24

Versione gestionale: 2025E (v876 - 87600)

Nuovi Campi

Nell'end-point *dati-generali/parametri-aziendali* nella sezione relativa ai parametri di produzione sono stati aggiunti i seguenti campi:

- fissa_prezzi_bl: "Anagrafica"
- gest_avanzamento_parziale_bl_collegate_oc: "N"
- gest_bl_visualizza_pf_per_cliente: "N"
- gest_escludi_da_odp: "N"

Correzioni

- Se un cliente aveva pi di un referente e, tramite webapi, veniva aggiornato uno dei campi, venivano cancellate tutte le altre anagrafiche dei referenti associate al cliente (Sol. 78572)
- La funzione 'gest_quadro_LM' tornava sempre la "N" di quadro non compilato anche nel caso di sua avvenuta compilazione (Sol. 78578)

Changelog v2.23

Versione gestionale: 2025D (v875 - 87500)

Nuovi Campi

Negli end-point dei documenti è stato aggiunto il campo di riga *cod_rif_orig_pf*:

- cod_rif_orig_pf: Riferimento PF origine della BL; solo produzione (Numerico, dimensione_array: dinamico)

Nell'end-point *tipi-lotti-matricole* sono stati aggiunti i seguenti campi:

- data_obbl: Tipo lotto - data validità obbligatoria (Alfanumerico)
- giorni_autocod: Tipo lotto - giorni per autocodifica (Numerico)
- giorni_manuale: Tipo lotto - giorni per codifica manuale (Numerico)
- giorni_giacenza: Tipo lotto - giorni minimi giacenza magazzino (Numerico)

Nuove Funzionalità

- Nel servizio *get_lista_deleghe* è stata aggiunta la possibilità di filtrare a partire da una data di ultima modifica in poi (vedi Lista deleghe)
- L'end-point *dati-generali/tipi-lotti-matricole* è stato completato con tutti i metodi CRUD aggiungendo POST, PUT e DELETE

Modifiche

- Migliorata la gestione in caso di presenza di sotto aziende anche se non viene specificata nell'header della richiesta evitando la riapertura dell'azienda
- Velocizzata dell'86% la lettura di liste MyDB con molti record (centinaia di migliaia)

Correzioni

- Negli end-point dei *referenti* facendo una PUT per cancellare dei campi inserendo stringa vuota ("") i campi non venivano cancellati (Sol. 77921)

Changelog v2.22

Versione gestionale: 2025C (v874 - 87400)

Nuovi Endpoint

- Produzione: tabella con la correlazione delle unità di misura *dati-generali/correlazione-unita-misura*

Nuovi Servizi

- Per le aziende di livello produzione o superiore è stato aggiunto il servizio per leggere i calendari di produzione: *get_calendari_produzione*.

Nuovi Campi

- Nel servizio per ottenere le strutture MyDB (*lista_strutture_mydb*) viene restituito anche il campo *ordinamento*
- Negli end-point dei documenti di magazzino è stato aggiunto il campo *cod_rif_pf_bl*:
    - cod_rif_pf_bl: riferimento PF della BL; solo produzione (Numerico, dimensione_array: dinamico)

Modifiche

- Per l'end-point dei progressivi degli articoli (*articoli//progressivi*), in caso l'articolo non abbia mai avuto nessuna movimentazione viene tornato un json vuoto invece che errore 400
- Per gestire meglio la concorrenza è stata modificata la procedura di paginazione: la logica della gestione dei NEXT rimane identica ma non possono essere salvati per essere riusati per chiamate successive. Per tutti dettagli si veda Paginazione.

Correzioni

- In alcuni casi particolari, quando da WebAPI si eseguiva una trasformazione di un documento inserendo nuove righe, si perdevano le provvigioni su tutte le righe del documento (Sol. 77778)

Changelog v2.21

Versione gestionale: 2025B (v873 - 87300)

Nuovi Servizi

- Aggiunto il nuovo servizio per autenticare i campi di tipo AUT sugli archivi MyDB. Vedi Servizio: Autenticazione MyDB.

Nuovi Endpoint

- Esposta la tabella con la gestione delle lingue straniere: *dati-generali/lingue-straniere*. Sono disponibili tutti i metodi CRUD tranne POST per la ricerca.

Nuove Funzionalità

- Nell'end-point *dati-generali/categorie-statistiche-cli-for* sono stati esposti tutti i metodi CRUD oltre la lista (tranne la ricerca)

Nuovi Campi

- Nell'end-point documenti/lavorazione/bolle è stato aggiunto il campo *cod_art_dbp*:
  - cod_art_dbp: Codice articolo con DBP (Alfanumerico)
- Nell'end-point *anagrafica-contatti* è stato aggiunto il campo *fino_val_spese_sped*:
  - fino_val_spese_sped: Fino a valore spese spedizione (Numerico con virgola)
- Nell'end-point clienti è stato aggiunto il campo *fino_spese_sped*:
  - fino_spese_sped: Vettore - Fino a valore spese di spedizione (Numerico con virgola)
- Nel servizio *leggi_pratica_dr* nel response sono stati aggiunti i campi relativi all'IBAN:
  - elenco_iban
  - iban_abituale
  - modalita_invio
- Nel servizio *spezzariga_bl* è stato aggiunto il nuovo parametro *solo_bl*="S/N" che, se attivo il parametro di produzione "Avanzamento parziale bl collegate a oc", spezza la riga solo sulla BL lasciando invariato l'OC

Correzioni

- Nel servizio *leggi_comunicazioni_lipe* non era gestito correttamente il trimestre a cavallo d'anno (trimestre anno precedente ma versato in anno successivo)
- Nel servizio lista delle deleghe non era gestito correttamente il modello pratica RNC

Modifiche

- Nella sezione relativa alla ricerca nei filtri era stato indicato erroneamente l'indice array come stringa invece che come intero
- Nella sezione relativa alla lettura dei documenti docuvision non era stata documentata la paginazione in caso di superamento dei limiti
- Nella sezione relativa alla lista delle deleghe sono stati specificati i valori disponibili per il campo *in_mod_pratica*

Changelog v2.20

Versione gestionale: 2025A (v872 - 87200)

Nuove Funzionalità

- Nel servizio *lista_docdv* è stata aggiunta la possibilità di filtrare, oltre che sui campi presenti nella risorsa, anche su altri parametri. Si veda Servizio: Lista Allegati Docuvision.
- Nel servizio con la lista delle strutture MyDB sono visibili anche eventuali campi di autenticazione, che saranno però non presenti in chiaro sul particolare archivio MyDB che ne fa uso per motivi di sicurezza. Per la gestione di questi campi verrà implementato un servizio ad hoc.

Nuovi Campi

- Nell'end-point *articoli* è stato aggiunto il campo *peso_medio_unit*
- Nel servizio *lista_strutture_mydb* sono stati aggiunti i metadati *data_ult_mod* e *checksum*
- Nell'end-point *anagrafica-contatti* sono stati aggiunti i seguenti campi:
    - valuta_estera, cat_statistica, zona, addebito_bollo, gest_cig_cup, rg_bolle, rg_effetti, proposto_e_ord, sconto_icz, trasporto_mezzo, cod_vett_abituale, tipo_porto, descr_porto, codice_incoterms, gest_calc_colli, gest_calc_peso, tipo_spese_sped, val_spese_sped, asp_est_beni
    - (per i dettagli su descrizioni e tipi fare GET anagrafica-contatti?info=true)

Changelog v2.19

Versione gestionale: 2024I3 (v871C - 87103)

Correzioni

- Nell'end-point dell'anagrafica dei contatti l'autocodifica non era gestita inserendo come codice 0
- Nell'end-point dell'anagrafica dei contatti mancava il campo *codice_pdc* nella lista
- Nell'end-point delle righe della prima nota mancava il campo *nr_documento*

Changelog v2.18

Versione gestionale: 2024I (v871 - 87100)

Nuovi Endpoint

- Aggiunto l'end-point dell'anagrafica dei contatti completo di tutti i metodi CRUD: *anagrafica-contatti*. Per i dettagli sui campi usare l'help in linea GET /risorse/anagrafica-contatti?info=true

Nuovi Campi

- Nell'end-point *dati-generali/taglie* è stato aggiunto il campo *descr_tg_ext* con il quale è possibile gestire descrizioni di taglie più lunghe di 3 caratteri.
- Negli end-point degli ordini sono stati esposti anche i campi con i riferimenti alle bolle di produzione (solo per singola entità):
  - nr_bl_colleg: num. bolla.lav.collegata*g (Numerico, dimensione_array: dinamico)
  - nr_rif_doc_colleg: num. rif.doc.collegato*g (Numerico, dimensione_array: dinamico)
- Nel servizio della lista delle deleghe ora vengono ritornati anche i riferimenti all'azienda collegata alla pratica o viceversa alla pratica collegata all'azienda:
  - mod_pratica_collegata
  - pratica_collegata
  - interno_collegato
  - sigla_azienda_collegata

Nuovi Servizi

- Servizio per la lettura della configurazione dei numeratori per i vari tipi di documento. Si veda Servizio: Proprietà Numeratori.

Nuove Funzionalità

- Nel servizio della lettura dei cancellati è stato aggiunto un parametro per indicare se la risorsa cancellata è presente o meno nello storico
- Nel servizio per la lettura degli allegati docuvision sono stati aggiunti anche i parametri per identificare univocamente l'allegato e la relativa versione

Changelog v2.17

Versione gestionale: 2024H (v870 - 87000)

Nuove Funzionalità

- Negli end-point dei documenti di magazzino sono gestiti anche i documenti ivati FX/BX/NX/RX
- Negli end-point dei documenti di magazzino sono stati inclusi anche i documenti Deposito Ordine: DO/DX

Nuovi Endpoint

L'end-point degli impegni è stato spostato in *documenti/lavorazione* ed ora gestisce obbligatoriamente anche il parametro *progressivo*:

- progressivo: Progressivo inserimento (Numerico)

In caso di POST con gestione degli impegni multi-riga, se è valorizzato con -1, viene inserito un nuovo impegno assegnando automaticamente il progressivo. In tutti gli altri casi se non viene specificato nulla viene sempre valorizzato a 0.

Nel caso che la gestione degli impegni multi-riga sia disabilitato il parametro deve sempre essere passato con 0 e nella POST per la creazione di un nuovo impegno deve essere omesso.

Esempio: GET documenti/lavorazione/impegni/52+0+1+1+1+FUELEX+0+0

Il vecchio end-point impegni continuerà a funzionare come prima, ma a lungo termine potrebbe essere deprecato. E' pertanto consigliato aggiornare le procedure usando il nuovo end-point.

Nuovi Servizi

- Sono stati implementati i servizi per la lettura degli allegati docuvision. Per tutti i dettagli si veda Servizio: Lettura Allegati Docuvision.
- Servizio per ottenere la lista di tutte le strutture MyDB utilizzate. Per tutti i dettagli si veda Servizio: Lista Strutture MyDB.

Nuovi Campi

- Negli end-point dei documenti sono stati aggiunti i dati di riga per i centri di costo/ricavo:
  - id_causale_riga: Causale -v._MMCMO-/Contropartita -max 32 x Doc. (Numerico, dimensione_array: dinamico)
  - id_ncau_riga: Causale di riga (Numerico, dimensione_array: dinamico)
  - id_ccr_riga: Centro costo ricavo (Numerico, dimensione_array: dinamico)
- Negli end-point dei movimenti di magazzino è stato aggiunto il campo sul tracking number:
  - nr_tracking: Tracking number (Alfanumerico, dimensione_array: dinamico)

- Negli end-point dei movimenti di magazzino sono stati aggiunti i riferimenti ai documenti Deposito Ordine:
    - rif_sgl_ddep: Riferimento sigla documento deposito (Alfanumerico, dimensione_array: dinamico)
    - rif_serie_ddep: Riferimento serie documento deposito (Numerico, dimensione_array: dinamico)
    - rif_nr_ddep: Riferimento numero documento deposito (Numerico, dimensione_array: dinamico)
    - rif_dt_ddep: Riferimento data documento deposito (Alfanumerico, dimensione_array: dinamico)

Correzioni

- Per gli end-point dei clienti e dei fornitori il campo nome_ricerca era gestito in modo errato e non era possibile fare POST o PUT valorizzando questo campo. E' stato inserito un nuovo campo dedicato per i clienti fornitori: *nome_ricerca_cf*
- Nell'end-point con la lista delle strutture articolo la descrizione non era presente: è stato aggiunto il campo *descr_struttura*
- In presenza di un documento trasformato il servizio di lettura righe documenti non valorizzava correttamente le variabili dei lotti.
- La ricerca con filtro su un archivio con numero di campi maggiore di quello della successiva ricerca sempre con filtro, ad esempio prima clienti e fornitori e poi alias articolo, portava all'errore di violazione di protezione di memoria.

Modifiche

- è stato rimosso il capitolo Allegati in quanto obsoleto. Per tutti i dettagli sui campi dei vari end-point è necessario consultare l'help in linea (info=true)

Changelog v2.16

Versione gestionale: 2024G (v866 - 86600)

Correzioni

- In alcuni contesti le chiavi che identificano una particolarità potevano contenere caratteri speciali non utf8

Changelog v2.15

Versione gestionale: 2024F (v865 - 86500)

Nuove Funzionalità

- Nei servizi *get_lista_pratiche_dr* è stata aggiunta la possibilità di usare dei filtri (vedi Lista pratiche dichiarativi)

Nuovi Campi

- Nell'end-point *documenti/lavorazione/bolle/* è stato aggiunto il campo *tp_bl_storico*: Bolla storicizzata (S/N solo lettura)
- Nell'end-point *indirizzi-spedizione* sono stati aggiunti i seguenti campi:
  - rif_esterno_tp: Riferimenti esterni tipo
  - rif_esterno_sgl: Riferimenti esterni sigla
  - rif_esterno_nr: Riferimenti esterni numero
  - rif_esterno_cod: Riferimenti esterni codice
  - rif_esterno_dt: Riferimenti esterni data
- Nel servizio sulla lista delle deleghe (get_lista_deleghe) sono stati esposti i seguenti campi:
  - nr_rata: intero - numero rata a cui fa riferimento la delega
  - nr_rate: intero - numero di rate in cui è stato disposto il pagamento
  - id_delega_origine: stringa - id della delega di origine
  - maggiorazione: flag S/N per sapere se nella delega è stata applicata o meno una maggiorazione.

Nuovi Endpoint

- *risorse/documenti/moduli-stampa*: GET lista e singola risorsa dei moduli di stampa disponibili (es. GET risorse/documenti/moduli-stampa/OC)
- *dati-generali/centri-costo-ricavo*: GET lista e POST ricerca per le tabelle dei centri costo/ricavo

Changelog v2.14

Versione gestionale: 2024E (v864 - 86400)

Nuovi Campi

- Negli end-point delle righe dei documenti (es. ordini-clienti/righe) è stato esposto il campo *sc_merce_doc*: sconto merce del documento.

Nuove Funzionalità

- Nei nuovi servizi di stampa dei moduli pdf è stato aggiunto il parametro per specificare il modulo di stampa: *cod_modulo*. (Vedi Servizio: Stampa PDF di Documenti)

Correzioni

- Il filtro sul campo *data_ult_mod* sull'end-point documenti/movimenti-magazzino/righe/ricerca tornava l'errore "Nome campo filtro 'data_ult_mod' non valido"

Changelog v2.13

Versione gestionale: 2024D (v863 - 86300)

Modifiche

- Nei filtri di ricerca è stata aggiunta la possibilità di controllare una condizione in OR su più valori di un campo. Non è ancora possibile usare la logica OR tra diverse condizioni. Vedi Ricerca e Filtri.

Nuove Funzionalità

- Servizio per ottenere la lista dei record cancellati nei vari end-point: *get_registro_operazioni_utente*. Vedi Servizio: Registro Cancellati.
- Servizi per ottenere la stampa PDF di documenti. Vedi Servizio: Stampa PDF di Documenti.
- Servizio per l'inserimento di righe in un documento con la restituzione dei nuovi id riga. Vedi Servizio: Inserimento Righe nei Documenti.
- Nel servizio *get_lista_deleghe* sono stati aggiunti nuovi filtri:
    - in_tipo: filtra le deleghe per tipo
    - in_definitiva: filtra le deleghe "definitive"
    - in_accise: filtra le deleghe con "accise"
    - in_ravvedimento_operoso: filtra le deleghe di "ravvedimento"
    - in_avviso_bonario: filtra le deleghe relative ad avvisi bonari
    - in_codice_invio: filtra le deleghe in base alla modalità di invio della delega

Nuovi Campi

Negli end-point *ordini-clienti*, *ordini-fornitori* e *preventivi* sono stati aggiunti i seguenti campi:

- tipo_imballo: Imballo variabile in emissione
- dt_inizio_rateo: Data inizio Rateo Risconto
- dt_fine_rateo: Data fine Rateo Risconto

Nell'end-point *liste-prelievo* sono stati esposti i seguenti campi:

- descr_articolo: Descrizione articolo (array)
- nota: Nota documento
- note_documento: Annotazioni documento (array)
- note_articolo: Annotazioni articolo (array)

Modifiche

- Nelle liste, il campo *data_rich_elab* è errato: il nome corretto è *data_ric_elab*

Nota: Nell'end-point dei documenti è stato anche aggiunto il campo *cod_art_anag*: Codice articolo proprietario dell'anagrafica (aggiunto in v2.13.1 - v863A - 86301)

Changelog v2.12

Versione gestionale: 2024C (v862 - 86200)

Nuovi Campi

- Negli end-point *ordini-clienti* e *ordini-fornitori* è stato aggiunto il campo *id_ncau_riga*: Causale di riga
- Per tutti gli end-point che gestiscono metadati, nel caso di GET su lista, viene sempre riportato anche il campo *data_rich_elab* in cui viene riportata la data in cui è stata fatta la richiesta (sarà sempre la stessa in caso di next, in quanto si riferisce alla prima richiesta). Può essere utile nel caso si voglia tenere traccia delle modifiche ed usarla come filtro maggiore o uguale sul campo di data ultima modifica ("data_ult_mod") nella richiesta successiva in modo da avere solo i modificati dalla richiesta precedente alla data della nuova richiesta.
- Nell'end-point *movimenti-magazzino* sono stati aggiunti i riferimenti alle dichiarazioni d'intento:
  - anno_int: Anno ricevimento lettera d'intento (testata)
  - prog_int: Progressivo lettera d'intento (testata)
  - anno_int_riga: Anno ricevimento lettera d'intento (riga)
  - prog_int_riga: Progressivo lettera d'intento (riga)
- Nel servizio *get_lista_deleghe* viene ritornato anche il campo di un carattere *avviso_bonario*.

Nuove Funzionalità

- Servizio per la lettura delle dichiarazioni iva annuali. Vedi Servizio: Lettura Dichiarazioni IVA Annuali.
- Servizio per la lettura delle comunicazioni LIPE. Vedi Servizio: Lettura Comunicazioni LIPE.

Changelog v2.11

Versione gestionale: 2024B (v861 - 86100)

Modifiche

- Aggiunta condizione *inizia_per* nei filtri (vedi Ricerca e Filtri)

Nuovi Campi

- Nell'end-point *articoli*, solo per la lista, è stata aggiunta la descrizione completa come somma dei campi descrizione+descrizione_agg: il campo si chiama *descr_completa*
- Nell'end-point *dati-generali/pagamenti* è stato aggiunto il campo *conto_pagamento*
- Nell'end-point *documenti/movimenti-magazzino* sono stati aggiunti i seguenti campi accessibili solo con GET su singola risorsa (non in lista):
  - cod_ctrpar_acc: codice contropartita acconto
  - cod_ctrpar_ab: codice contropartita abbuono
- Negli end-point sui progressivi degli articoli (*risorse/progressivi-articoli* e *risorse//progressivi*) sono stati aggiunti anche i dettagli sugli impegni per picking:
  - qta_picking
  - qta_picking_tg
  - colli_picking
  - tara_picking
- Nell'end-point *parametri-aziendali* è stata aggiunta la sezione *parametri_contabili*
- Nell'end-point *aziende* è stato aggiunto, in creazione (POST), il parametro sul livello aziendale: *in_livello_azienda*
  - "NULL" - Azienda di livello NULLO
  - "1" - Azienda di livello BASE
  - "1F" - Azienda di livello BASE CON FATTURAZIONE
  - "2" - Azienda di livello ESTESO
  - "3" - Azienda di livello PRODUZIONE
- Nel servizio *avanzamento_produzione* è stato aggiunto il nuovo parametro in ingresso cod_magazzino: solo per operazione "C" indica il magazzino su cui fare il CL

Changelog v2.10

Versione gestionale: 2024A3 (v860C - 86003)

Correzioni

- Nel servizio *get_lista_deleghe* è stato inserito un codice di errore per ogni azienda che per qualche motivo non è stato possibile processare
- Negli end-point *anagrafica-unica* e *anagrafica-unica/storico* a volte la lista andava in errore a causa di una apertura anomala dell'archivio delle divise.

Nuovi Campi

- Nell'end-point *dati-generali/parametri-aziendali* è stato aggiunto anche il parametro per la gestione dello scadenzario a partite

Changelog v2.9

Versione gestionale: 2024A1 (v860A - 86001)

Nuovi Campi

- Nel servizio sullo sblocco delle deleghe (*cambia_stato_delega*) è stato aggiunto il codice di errore oltre il valore

Modifiche

Nota: Il changelog della v2.9 originale conteneva voci che sono state poi rilasciate effettivamente solo in versione 2024B (v2.11). Sono state barrate nel manuale originale.

Nota v2.9.1: La versione v2.9.1 ha specificato che solo l'ultimo punto del changelog v2.9 era effettivamente stato rilasciato. I rimanenti punti sono usciti in versione 2024B.

Changelog v2.8

Versione gestionale: 2024A (v860 - 86000)

Nuovi Campi

Negli end-point dei movimenti di magazzino sono stati aggiunti i seguenti campi:

- gest_bollo_virt: Gestione bollo virtuale nel documento S/N
- dt_inizio_rateo: Data inizio Rateo Risconto
- dt_fine_rateo: Data fine Rateo Risconto

Nell'end-point *articoli* è stata aggiunta la variabile *gest_peso_wms* alfanumerico "S/N": Pesata WMS

Nell'end-point *lotti* è stata aggiunta la variabile *gest_stor_lotto* alfanumerico "S/N": Lotto storicizzato S/N

Nuovi Endpoint

- *dati-generali/parametri-aziendali*: solo GET singola risorsa. Verranno di volta in volta aggiunti nuovi parametri, al momento sono presenti solo alcuni parametri di magazzino e produzione.

Nuove Funzionalità

- Nell'end-point *aziende* è stato aggiunto anche il metodo POST per la creazione di una nuova azienda e il collegamento all'anagrafica unica: vedi Creazione Azienda
- Nel servizio *get_prog_ubicazioni* è stata inserita la possibilità di filtrare anche per *id* oltre che per *data_ult_mod*: vedi Servizio: Progressivi di Tutte le Ubicazioni

Nuovi Servizi

- Servizio per la lettura di un modello dichiarativo: vedi Lettura pratica dichiarativi
- Servizio per ottenere la lista delle pratiche dichiarativi: vedi Lista pratiche dichiarativi
- Servizio per ottenere la lista delle deleghe dichiarativi: vedi Servizio: Deleghe Dichiarativi

Modifiche

- Aggiunta una nota sulla gestione dei documenti in caso di azienda senza gestione taglie (vedi Gestione delle Taglie)
- Come indicato nel changelog della versione 2.6 i seguenti end-point non saranno più raggiungibili. Qualora siano ancora utilizzati è necessario usare il nome corretto sostituendo l'underscore con il trattino ('-' invece che '_'):
  - categorie_statistiche_cli_for
  - categorie_provvigioni_articoli
  - condizioni_generali_pagamenti
  - categorie_sconti_articoli

Changelog v2.7

Versione gestionale: 2023H (v851 - 85100)

Nuovi Campi

- Nell'end-point *prima-nota/righe* è stata aggiunta la data di ultima modifica come per le testate: *data_ult_mod*
- Nell'end-point *articoli* è stata aggiunta la data di ultima modifica delle sole immagini: *dt_variaz_img*
- Nell'end-point *anagrafica-unica* in caso di GET della singola risorsa è disponibile anche la data di inizio del particolare blocco di informazioni xxx_stor_iniz
- Nell'end-point *anagrafica-unica* in caso di GET della singola risorsa è disponibile anche la data di inizio validità e fine validità alla particolare data chiesta (*data_iniz_valid*, *data_fine_valid*). Le date devono essere intese come validità per l'intera anagrafica e non solo dei singoli blocchi di informazioni: *data_iniz_valid* sarà la più grande di tutte le date di inizio validità (xxx_stor_iniz) dei singoli blocchi mentre *data_fine_valid* sarà la più piccola di tutte le date di fine validità (xxx_stor_data) dei singoli blocchi di informazioni.
- Nell'end-point *anagrafica-unica/storico* inserito il campo *tp_dati* con l'informazione sul tipo di dato che è stato modificato (vedi Storico - Tipi di Dati)
- Nell'end-point *documenti/ordini-fornitori/righe* è stato esposto il campo *id_riga_orig* come negli ordini cliente.
- Nell'end-point *lotti* sono stati aggiunti i metadati di utente e data ultima modifica: *utente_ult_mod*, *data_ult_mod*

Nuovi Endpoint

- E' stato aggiunto l'end-point *anagrafica-unica/storico* solo GET e POST per la ricerca con l'elenco di tutte le storicizzazioni

Nuove Funzionalità

- Nell'end-point *anagrafica-unica* è stato aggiunto un nuovo tipo di storicizzazione nel parametro *storicizza* in query string. Storicizza=R aggiorna l'anagrafica e storicizza alla data indicata (vedi dettagli in Modifica con Storicizzazione)

Nuovi Servizi

- Servizio per leggere le informazioni sulle località italiane (CAP, Provincia, Comune, Località, Zone). Vedi Servizio: Località Italiane (CAP e Zone).
- Servizio per il conteggio del numero delle operazioni contabili. Vedi Servizio: Conteggio Numero Operazioni Contabili.

Modifiche

End-point *anagrafica-unica*, corretti i nomi delle seguenti variabili per uniformità con la convenzione dei nomi web-api:

Vecchio nome	Correzione
data_validita	data_fine_valid
anag_pers_fisica	compresa in anag_tipo_sogg
anag_tipo_piva	**eliminata**
anag_partita_iva	anag_piva
anag_rag_sociale	anag_rag_soc
anag_denominazione	anag_denom
res_stesso_sede_leg	res_stesso_dom
leg_soc_cod_fiscale	leg_soc_cod_fis
leg_soc_rag_sociale	leg_soc_rag_soc
docum_rilas	docum_aut_ril
docum_data_rilas	docum_data_ril

Correzioni

- Nell'end-point *alias-articoli* il campo *quantita* gestiva erroneamente solo numeri interi (definito come Numerico). Corretto in "Numerico con virgola".

Changelog v2.6

Versione gestionale: 2023G (v850 - 85000)

Nuovi Campi

Aggiunti i campi per i riferimenti esterni agli ordini di origine:

- rif_sgl_origine: Rif. Documenti Esterni - Sigla documento ordine di origine
- rif_nr_origine: Rif. Documenti Esterni - Num documento ordine di origine
- rif_dt_origine: Rif. Documenti Esterni - Data documento ordine di origine

Aggiunto campo *qta_tg_plus_prod* per la gestione delle taglie sul servizio avanzamento_produzione:

- qta_tg_plus_prod: array per la gestione delle taglie in caso di aggiunta di carico. La somma delle quantità delle taglie deve corrispondere con la quantità indicata nella variabile *qta_agg_carico*

Nuovi Endpoint

- Aggiunto end-point elenco aziende: solo GET della lista delle aziende webapi/risorse/aziende
- Aggiunto end-point scadenzario: solo GET della collection e POST per la ricerca:
  - GET webapi/risorse/scadenzario?tipo=
  - POST webapi/risorse/scadenzario/ricerca
  - Dove indica quale tipo di scadenzario si vuole leggere:
    - D: a Documento
    - P: a Partite
    - X: eXtracontabile
- Aggiunto end-point su prima nota: implementati tutti i metodi di lettura e scrittura. Vedi Entità: Prima Nota.
- Aggiunto end-point anagrafica-unica: implementati tutti i metodi di lettura e scrittura. Vedi Entità: Anagrafica Unica.

Nuovi Servizi

- Calcolo dei totali documento
- Calcolo dei totali di riga
- Calcolo dell'esposizione del cliente

Correzioni

- Nei progressivi degli articoli, se era presente un articolo con un lotto presente su più magazzini la procedura del calcolo dei progressivi scartava il lotto.
- Nell'end-point *articoli* i campi *scorta_minima* e *scorta_massima* erano invertiti.

- Nell'end-point *movimenti-magazzino* non erano stati esposti i campi relativi agli articoli di tipo testo
- Nell'end-point *liste-prelievo* è stato aggiunto un controllo più restrittivo sui campi chiave (sigla, serie, numero, id_riga, prog_riga) i quali dovranno sempre essere indicati in caso di revisione (PUT)

Modifiche

Gli end-point sulle categorie statistiche dei clienti/fornitori, categorie provvigioni, condizioni pagamenti e categoria sconti erano stati nominati con il separatore underscore "_" invece che con il trattino "-". Per uniformità sono stati corretti:

- categorie-statistiche-cli-for
- categorie-provvigioni-articoli
- condizioni-generali-pagamenti
- categorie-sconti-articoli

E' necessario aggiornare tutte le procedure che utilizzano la nomenclatura errata in quanto da Gennaio 2024 non potranno più essere raggiungibili.

Per l'end-point *liste-prelievo*, in caso di modifica (PUT) della sola testata è ora necessario specificare in query string il parametro *solo_testata=true* come avviene già per i documenti. Vedi Modifica Prelievo.

Nota v2.6.1 (v850B - 85002): Nell'end-point *anagrafica-unica* è stata aggiunta la possibilità di storicizzare documenti anche da WebAPI (vedi Modifica con Storicizzazione).

Nota v2.6.2 (v850C - 85003): Nell'end-point *anagrafica-unica* era richiesto l'accesso con azienda anche se non necessario. E' stato corretto ed è possibile usare l'end-point senza dover specificare le Coordinate-Gestionale.

Changelog v2.5

Versione gestionale: 2023F (v845 - 84500)

Nuovi Endpoint

- Aggiunto end-point sulla cartella abbinamenti: (GET lista e POST ricerca) *risorse/dati-generali/cartella-abbinamenti*

Nuovi Campi

In tutti gli end-point del contenitore documenti sono stati aggiunti i seguenti campi:

- sc_merce_doc: sconto merce del documento (solo nell'accesso alla singola risorsa)
- abbuono

Nota v2.5.1 (v845A - 84501): In tutti gli end-point del contenitore documenti sono stati aggiunti i seguenti campi: - codice_cig: Codice CIG per appalti pubblici - codice_cup: Codice CUP per appalti pubblici

Changelog v2.4

Versione gestionale: 2023E2 (v844B - 84402)

Nuovi Endpoint

- Aggiunto end-point sulle categorie sconti degli articoli (solo GET lista): *risorse/dati-generali/categorie_sconti_articoli*
- Aggiunto end-point sulle categorie statistiche dei clienti fornitori (solo GET lista): *risorse/dati-generali/categorie_statistiche_cli_for*
- Aggiunto end-point sulle categorie delle provvigioni degli articoli (solo GET lista): *risorse/dati-generali/categorie_provvigioni_articoli*
- Aggiunto end-point sulle condizioni generali dei pagamenti (solo GET lista): *risorse/dati-generali/condizioni_generali_pagamenti*

Nuovi Campi

Come per i documenti di tipo ordine anche per i movimenti di magazzino sono stati esposti, nelle liste, i campi sui riferimenti esterni:

- rif_sgl_attuale
- rif_nr_attuale
- rif_dt_attuale

Nuovi Servizi

- Servizio per la lettura di tutti gli articoli di tutte le ubicazioni. Per i dettagli si rimanda a Servizio: Progressivi di Tutte le Ubicazioni.

Changelog v2.3

Versione gestionale: 2023E (v844 - 84400)

Nuovi Campi

- Il campo *id_rif_testata* è stato riportato anche negli end-point delle righe dei movimenti di magazzino (movimenti-magazzino/righe)
- In tutti gli end-point dei Documenti è stato esposto anche il campo *cod_mndto_rid*: Codice mandato RID SEPA

Changelog v2.2

Versione gestionale: 2023D (v843 - 84300)

Nuovi Endpoint

- Lista posizione referenti: *risorse/dati-generali/posizione-referenti* solo metodo GET
- DBA articolo: *risorse/dba* sia GET che POST per la ricerca

Nuovi Campi

Aggiunti i campi per la gestione del multi-agente nei documenti (solo per il metodo GET sul singolo documento):

- cod_agente: array alfanumerico con codici multiagente
- cond_agente: array numerico condizione multiagente
- tipo_provv: array alfanumerico con tipo provvigione multiagente
- formula_pr: array alfanumerico con la formula della provvigione per il multiagente (valida solo per il 1° elemento)
- calc_formula_pr: array numerico con il calcolo della formula provvigione (solo lettura)
- quota_ripart_pr: array numerico con la quota di ripartizione delle provvigioni
- modalita_pr: array alfanumerico con la modalit delle provvigioni per multiagente
- imp_calc_pr: array alfanumerico con l'importo del calcolo delle provvigioni: 1 lordo, 2 netto

Nuovi Servizi

- Servizio per il calcolo dei dati relativi all'esposizione di un cliente/fornitore. Vedi Servizio: Calcolo Esposizione.

Changelog v2.1

Versione gestionale: 2023C (v842 - 84200)

Nuovi Endpoint

- Referenti clienti e fornitori:
  - /webapi/risorse/referenti/clienti
  - /webapi/risorse/referenti/fornitori
- Tabella zone clienti/fornitori: /webapi/risorse/dati-generali/zone-clienti-fornitori solo GET della collection
- Tabella banche clienti: /webapi/risorse/banche GET e POST per la ricerca

- Tabella categorie provvigioni clienti/fornitori: /webapi/risorse/dati-generali/categorie-provvigioni solo GET della collection
- Tabella formazione prezzi articoli: /webapi/risorse/dati-generali/categorie-prezzi solo GET della collection
- Tabella categorie sconti clienti/fornitori: /webapi/risorse/dati-generali/categorie-sconti solo GET della collection
- Tabella aspetto esteriore dei beni: /webapi/risorse/dati-generali/aspetto-esteriore-beni solo GET della collection

Nuove Funzionalità

- Per le risorse sotto l'end-point *documenti* è possibile indicare in trasformazione (quindi solo per metodo POST) l'eventuale scorporo dell'iva (es. quando si trasforma da OX a BC). In questo caso è necessario specificare in query string il parametro booleano *gest_scorporo_iva* che può assumere valore *true/false*

Nuovi Campi

- Nell'end-point dei *documenti* (per qualsiasi tipo di documento) nelle righe è ora disponibile il metadato sulla data di ultima modifica come avviene per tutti gli altri end-point: *data_ult_mod*.

Modifiche

- Modificata la gestione dei servizi WebAPI in caso di accesso esclusivo da parte del gestionale. Aggiornato il capitolo sulle modalità di accesso.

Nota v2.1.1 (v842B - 84202): Facendo una richiesta di ricerca con il metodo POST sugli end-point dei documenti (es. documenti/ordini-clienti/ricerca) filtrando per data_ult_mod ritornava il seguente errore: "6001 - errore gestionale [Nome campo filtro data_ult_mod non valido]" (Sol. 70130)

Changelog v2.0

Versione gestionale: 2023B4 (v841D - 84104)

Modifiche

- Nel caso di trasformazione di documenti col metodo POST è ora sufficiente indicare i soli parametri necessari, tutti gli altri dati di riga vengono presi automaticamente dalla riga del documento di origine (come avveniva con la PUT). La trasformazione con PUT non è più ammessa. Vedi Movimenti di Magazzino.
- Nel caso di revisione di qualsiasi end-point è possibile abilitare un controllo lato server (servizi - configurazioni - configurazione moduli - WebAPI) sulla data di ultima modifica in modo da mantenere coerenza tra i dati (vedi Controllo sulla validità dei dati (data ultima modifica))

- E’ stato eliminato il limite di 100 record massimo sulle GET di liste. Ora WebAPI restituisce sempre tutto quello che è in grado di dare entro 30 secondi oppure entro i 50MB di payload: se una delle 2 condizioni non è soddisfatta inizierà a paginare indicando con *next* il prossimo record da cui partire. La presenza di *max* in query string è prioritario a patto che non si violino una delle 2 condizioni sopra citate (vedi Paginazione).

Nuove Funzionalità

- Aggiunta la possibilità di copiare una riga di un documento semplicemente indicando l’indice della riga da cui copiare i dati mediante il nuovo campo array *idx_riga_copia* (vedi Copia Righe Documenti)
- Nell’end-point *documenti/ordini-clienti* è stata aggiunta la possibilità di gestire anche i documenti di tipo OX
- Per tutte le risorse dell’end-point *documenti* è stato aggiunto un parametro in query-string valido per il solo metodo PUT che indica se modificare o meno solo i dati di testata: *solo_testata=true/false* (vedi Uso Avanzato - Struttura Documenti e Operazioni Complesse).

Nuovi Campi

- Nell’end-point *liste-prelievo* è stato aggiunto un nuovo campo di tipo array *gest_tipo_lav* in sola lettura. Può essere scritto esclusivamente da client Zerodo
- Nell’end-point *documenti* è stato aggiunto un nuovo campo *bloccato_in_mag* in sola lettura. Può essere scritto esclusivamente da client Zerodo

Nota v2.0.1: - Nel manuale non è mai stato indicato che è possibile richiedere i dati anche in formato compresso *gzip*. Per ogni http-request è sufficiente specificare nell’header il campo Accept-Encoding:gzip come previsto dallo standard http. WebAPI in questo caso risponderà con dati compressi in formato gzip, sarà poi compito del client decomprimere i dati ricevuti. Vedi Compressione della response. - Aggiornato il capitolo dei documenti indicando come viene gestito l’ordinamento delle righe.

Changelog v1.10

Versione gestionale: 2023A2 (v840B - 84002)

Correzioni

- Nella risorsa articoli era stato inserito due volte il campo *cod_struttura* e poteva assumere valori errati. E’ stato corretto uno dei due nomi in *rif_struttura* che identifica il riferimento alla struttura nel caso di articoli di tipo “B”.

Nuovi Endpoint

- E' stata inserita una nuova risorsa per ottenere i progressivi di tutti gli articoli: *progressivi-articoli*. La risorsa è accessibile solo tramite metodo GET e restituisce tutti i progressivi per tutti gli articoli del magazzino specificato in header, oppure di tutti i magazzini se in header è impostato Magazzino=0. Per i dettagli sull'end-point si veda l'HELP in linea (GET /risorse/help?extended=true)

Nuovi Campi

- In tutti i documenti di magazzino (ordini e movimenti) è stato esposto il campo con il riferimento alla tabella dell'aspetto esteriore dei beni. Il campo si chiama *asp_est_beni*: formato scalare per le collection e formato array per le singole risorse. Per i dettagli fare riferimento all'help in linea (es. GET documenti/ordini-clienti?info=true)

Changelog v1.9

Versione gestionale: 2022J3 (v831C - 83103)

Modifiche

- Negli esempi di trasformazione dei documenti sono stati corretti i valori sugli *id_riga*
- Nell'esempio dello sviluppo distinta base il campo con il codice articolo padre era errato (codice_articolo_padre): è stato corretto in codice_art_padre
- Inserito nei titoli del changelog anche la versione_prodotto reperibile da WebAPI facendo una request sull'end-point *dati-generali/installazione* (es. 83103)

Correzioni

- Per le installazioni locali su macchine Windows la Java Virtual Machine su cui gira il Server Tomcat per la gestione del servizio WebAPI, aveva il charset di default impostato come Windows-1252. Questo comportava che le letture (GET) con WebAPI venivano correttamente codificate in UTF-8, ma le scritture (PUT-POST) venivano gestite in Windows-1252 e quindi in fase di parsing del JSON in ingresso, nel caso di presenza di caratteri speciali, Mexal restituiva un errore sulla codifica.

Nuovi Campi

- Nella risorsa *articoli* è stato aggiunto il campo *gest_macchinario* che, solo per gli articoli di tipo L, può assumere i valori S/N.
- Nella risorsa *movimenti_magazzino* sono stati esposti anche i seguenti ulteriori campi:
    - id_catsta_conto: Categoria statistica cliente/fornitore (cod_cat_sta in anagrafica cliente)
    - id_zona_conto: Zona cliente/fornitore (cod_zona in anagrafica cliente)

- Nella risorsa *movimenti_magazzino/righe* sono stati esposti anche i seguenti ulteriori campi:
  - costo_ult: costo ultimo allo scarico (-1 prende il costo in anagrafica articolo)
  - costo_std: costo standard allo scarico (-1 prende il costo in anagrafica articolo)
  - costo_med_pond: costo medio ponderato allo scarico (-1 prende il costo in anagrafica articolo)

Nota v1.9.1 (v840 - 84000): - Nella risorsa *documenti/lavorazione/bolle/* è stato aggiunto il campo *descr_um*: Descrizione unità di misura (sola lettura) - Al fine di gestire nel path della risorsa anche caratteri speciali come SLASH "/" e BACKSLASH "\", è stata inserita una nuova chiave in query string (*encoding*) per gestire l'encoding di questi caratteri. Per i dettagli si veda Formato dati.

Nota v1.9.2 (v840A - 84001): - Aggiunti maggiori dettagli nella sezione relativa alla gestione del parametro encoding - Se il parametro encoding era utilizzato in un path in cui il codice da decodificare era in mezzo al path e non alla fine (es. immagini articoli) la decodifica andava in errore. - Nella risorsa *documenti/lavorazione/bolle* per la creazione/modifica di un nuovo documento è richiesto come parametro nel body *nr_rif_impegni* invece di *nr_rif_crea*.

Changelog v1.8

Versione gestionale: 2022J (v831 - 83100)

Nuovi Servizi

- Condizioni documento: vedi Servizio: Condizioni Documento

Nuovi Campi

- Nella risorsa *fornitori* è stato aggiunto il campo *assog_ivaxml*: Condizioni generali - Import. op.passive con nat.esenz.IVA N2.1
- Nella risorsa *articoli* sono stati aggiunti due campi di sola lettura:
  - dt_cos_ult_ues: Data costo ultimo op. passive anno (sola lettura)
  - dt_cos_ult_pes: Data costo ultimo anno precedente (sola lettura)
- Nelle risorse *liste-prelievo* ed *ubicazioni* sono state aggiunte informazioni su utente e data di ultima modifica:
  - utente_ult_mod: utente ultima modifica
  - data_ult_mod: data di ultima modifica

Correzioni

- Risorse liste prelievo: aggiunti riferimento ai lotti sulla riga d'ordine (in precedenza id_lotto era sempre vuoto)

Changelog v1.7

Versione gestionale: 2022I (v830 - 83000)

Nuovi Servizi

- Avanzamento di produzione: vedi Servizio: Avanzamento di Produzione.
- Spezza riga bolle di lavorazione: vedi Servizio: Spezza Riga Bolle di Lavorazione.

Nuovi Endpoint

- *mappa-articoli*: è stata esposta la lista con il dettaglio logistico di ogni articolo (es. ubicazione di default, magazzino, etc.). Sono disponibili i metodi GET per lista completa e POST per la ricerca. La risorsa è raggiungibile al seguente end-point: https://base_address/webapi/risorse/mappa-articoli
- *tipi-lotti-matricole*: è stata esposta in sola lettura la lista dei tipi di lotto/matricola. Sono disponibili i metodi GET per lista e singolo elemento e POST per la ricerca. La risorsa è raggiungibile al seguente end-point: https://base_address/webapi/risorse/dati-generali/tipi-lotti-matricole
- *fasi-lavorazione*: è stata esposta la tabella con la descrizione delle fasi di lavorazione. Sono disponibili i metodi GET per lista completa e POST per la ricerca. La risorsa è raggiungibile al seguente end-point: https://base_address/webapi/risorse/dati-generali/fasi-lavorazione

Nuovi Campi

- Aggiunti nella risorsa *liste-prelievo* i campi di tipo array *priorita* e *gest_residuo* modificabili per le sole righe d'ordine (non picking).
- Per ragioni di coerenza semantica sono stati cambiati i nomi dei campi per la gestione dei residui. Dato che la quantità da indicare non indica il residuo ma quella che andrà nel documento trasformato è stato sostituito res con trs:
    - nr_colli_trs
    - quantita_trs
    - qta_taglia_trs
    - qta_lotto_trs
    - nr_colli_lo_trs
    - qta_tg_lo_trs
- Nella risorsa ubicazioni, quando si richiede il dettaglio dei progressivi con *cod_art_progr*, è stata inserita la possibilità di escludere o meno i lotti (per motivi prestazionali e di compatibilità col pregresso) utilizzando in query string la variabile dettlotti=true/false
- Aggiunte le variabili di sola lettura per i totali del documento in tutte le risorse sotto il contenitore *documenti*:
    - tot_doc_validi: indica se i valori specificati nei totali sono validi
    - tot_iva

- tot_documento
- tot_doc_pagare

Modifiche

- Aggiunti esempi nella trasformazione dei documenti: casi con lotti

Changelog v1.6

Versione gestionale: 2022G (v824 - 82400)

Correzioni

- Esportato il campo “descr_art_ext” anche per la risorsa *movimenti-magazzino/righe* (Sol.66278)

Nuove Funzionalità

- Trasformazione documenti (vedi Trasformazione Documenti)

Nuovi Campi

- nr_colli_res: Numero colli residuo
- quantita_res: Quantità / Peso Lordo residuo
- qta_taglia_res: Quantità per taglia residuo
- qta_lotto_res: Lotti - Quantità residuo
- nr_colli_lo_res: Lotti - Colli residuo
- qta_tg_lo_res: Lotti - Quantità per taglia residuo

Nuovi Endpoint

- Liste-prelievo (vedi Entità: Liste Prelievo)
- Giacenze articoli: nella risorsa articoli è stata aggiunta l’entit progressivi. Facendo una GET su questa entità verranno restituite tutte le giacenze dell’articolo selezionato. Es. GET base_address/articoli/COD_ART/progressivi - Verranno restituiti array di n record, e per ogni record è indicata la quantità di giacenza (inventario, carico, scarico) il magazzino, l’eventuale ubicazione.
- Alias-articoli: https://base_address/webapi/risorse/alias-articoli (solo lista e ricerca)
- Imballi: https://base_address/webapi/risorse/dati-generali/imballi

Changelog v1.5

Versione gestionale: 2022C1 (v820a - 82001)

Nuovi Endpoint

- Archivio Lotti: aggiunti relativi riferimenti all'id del lotto anche nelle risorse che lo referenziano
- Archivio Ubicazioni (vedi Entità: Ubicazioni): aggiunti relativi riferimenti all'id dell'ubicazione anche nelle risorse che la referenziano

Nuovi Campi

- Aggiunto campo *tp_articolo* (tipologia articolo) nella risorsa degli impegni e nel servizio per lo sviluppo della distinta base *sviluppo_distinta_base*

Nota v1.5.1 (v820B - 82002): - Nel servizio *sviluppo_distinta_base* sono stati corretti i nomi di 2 variabili: - escludi_da_deposito (mancava la "l") - escludi_da_scarico (mancava la "l") - Nell'entità *dati-generali/particolarita* è stato corretto il nome di una variabile: erano stati erroneamente dichiarati due campi con lo stesso nome (cod_articolo), è stato lasciato *cod_articolo* per indicare il codice dell'articolo mentre è stato aggiunto *cod_art_cli* per indicare il codice articolo associato al cliente/fornitore

Nota v1.5.2 (v821 - 82100): - Nella risorsa *dati-generali/particolarita* è stato aggiunto il campo *descr_art_cli*: Descrizione articolo del Cli/For se "tipo_part" = 'A' - Nella risorsa *ubicazioni* è stato aggiunto il campo *percorso*: Codice percorso utilizzato per creare un percorso all'interno del magazzino - Nella risorsa *documenti/movimenti-magazzino* sono stati aggiunti i seguenti campi: - Campi di testata: - rif_nr_bl: Numero di riferimento bolla di lavoro (Numerico) - rif_nr_sbl: Numero di riferimento sottobolla di lavoro (Numerico) - rif_nr_fs_lav: Numero di riferimento fase di lavorazione (Numerico) - Campi di riga (array): - cod_rif_pf_bl: riferimento PF della BL; solo produzione (Numerico, dimensione_array: dinamico) - dt_doc_bl: data documento della BL; solo produzione (Alfanumerico, dimensione_array: dinamico) - nr_bl: numero BL; solo produzione (Numerico, dimensione_array: dinamico) - nr_sbl: sotto BL; solo produzione (Numerico, dimensione_array: dinamico) - prog_fase_bl: fase BL; solo produzione (Numerico, dimensione_array: dinamico) - Nelle risorse *documenti/ordini documenti/preventivi documenti/matrici* il campo *id_righe_dba* è stato cambiato in *id_rec_dba_dbvo* ma non documentato - Il Content-type è opzionale esclusivamente per i metodi GET e DELETE; per i metodi POST e PUT è obbligatorio

Changelog v1.4

Versione gestionale: 2022B4 (v819D - 81904)

Nuovi Campi

Nella risorsa *articoli* (sia lista che singola entità) sono state aggiunte le seguenti variabili che indicano la presenza o meno di immagini (S/N):

- img_art_disp: indica se l'immagine articolo è presente
- img_cat_disp: indica se l'immagine di catalogo è presente
- img_icona_disp: indica se l'icona è presente

Changelog v1.3

Versione gestionale: 2022B (v819 - 81900)

Correzioni

Correzione refusi di tipi di dato tra liste e singole risorse:

Clienti / Fornitori - "gest_rit_acc": Impostato tipo 'A' S/N in get/put - "alq_regio_sport": Impostato tipo 'F' in lista - "alq_comun_sport": Impostato tipo 'F' in lista - "fido_migliaia": In lista rimane solo "fido" mentre in singola entità "fido" e "fido_migliaia" - "anag_dt_mod": Impostato tipo 'A' in lista

Articoli - "larghezza_ubic": Impostato tipo 'F' in get/put - "altezza_ubic": Impostato tipo 'F' in get/put - "profondita_ubic": Impostato tipo 'F' in get/put - "peso_ubicazione": Impostato tipo 'F' in get/put - "cod_srv_intra": Impostato tipo 'A' numero formattato con '0' a sx in get/put come in lista

Ordini / Preventivi / Matrici - "tp_cambio_vlt": Impostato tipo 'A' S/N in get/put - "tp_cambio_euro": Impostato tipo 'A' S/N in get/put - "gest_mydb_testa": Impostato tipo 'A' S/N in get/put - "gest_mydb_riga": Impostato tipo 'A' S/N in get/put

Movimenti di magazzino - "tp_cambio_vlt": Impostato tipo 'A' S/N in get/put - "tp_cambio_euro": Impostato tipo 'A' S/N in get/put - "gest_mydb_testa": Impostato tipo 'A' S/N in get/put - "tipo_stato_riga": Impostato tipo 'A' in lista - "gest_mydb_riga": Impostato tipo 'A' S/N in get/put

Changelog v1.2

Versione gestionale: 2022A (v818 - 81800)

Correzioni

- Sviluppo distinta base: quantità obbligatoria

Modifiche

- Nuove risorse:
    - Fasi distinte base

    - Archivi MyDB
    - Aggiunta la risorsa “conti” oltre ai clienti e fornitori
- Nuova nomenclatura delle variabili: per tutti i dettagli usare il comando HELP e info=true. Per la conversione tra vecchia e nuova nomenclatura vedere Allegato A
- Corretta sintassi filtri per gestire array a due livelli
- Aggiunti i campi *id_riga_orig* e *cat_annul_riga* per i documenti di magazzino e degli ordini

Changelog v1.1

Versione gestionale: 2021I (v816 - 81600)

Modifiche

- Aggiunto il parametro opzionale Magazzino nell’header della richiesta per permettere di specificare il magazzino di default
- Le liste sono state arricchite di ulteriori dati e gli array possono arrivare fino a 2 livelli invece che 1 solo livello
- Cambiato path del comando HELP e aggiunto filtro *extended* in query string
- Aggiunte nuove risorse: usare il comando HELP per tutti i dettagli
- Cambiato il path di alcune risorse: usare il comando HELP per tutti i dettagli
- Aggiunto limite al numero massimo di utenti appartenenti al gruppo WebAPI: 3
- Aggiunto servizio per lo sviluppo della distinta base
- Eliminata documentazione sull’accesso alle risorse tramite il canale servizi: si consiglia di utilizzare sempre l’accesso con logica REST sul path “risorse”
- Inserita entità allegati: in questa versione solo lettura immagini articoli
- Non è più necessario gestire il cookie per installazioni Live
- Nell’entità di tipo ricerca aggiunta la condizione “contiene”

Nuove Funzionalità

- Nei prossimi rilasci di WebAPI i nomi delle chiavi dei JSON verranno cambiati gestendo un massimo di 15 caratteri. In questo modo i nomi degli elementi risulteranno più parlanti e più aderenti ad una logica RESTful. Nel caso siano già state fatte applicazioni con la nomenclatura attuale, di volta in volta verranno rilasciate delle tabelle di correlazione tra i vecchi ed i nuovi nomi dei vari archivi aggiornati. In ogni caso per i dettagli sui nuovi nomi dei campi si può usare il metodo GET con il parametro *info=true* in query string.

Correzioni

- esec_collage_server_remoto: era stato erroneamente indicato esegui_collage_server_remoto