PASSEPARTOUT®
Software e servizi gestionali

WebAPI

Gianluca Suzzi

Riferimenti

Manuale WebAPI in Area Sviluppatori:
https://areariservata.passepartout.net/AreaRiservataPA/Resources/Download?id=11132

Test tools consigliati:

- **Insomnia** - https://insomnia.rest/download
- Postman
- SoapUI

Esempi:

Export di HTTP Request per Insomnia presenti in area download: https://download.passepartout.cloud/CondivisaPartner/WebAPI

PASSEPARTOUT
Software e servizi gestionali

Cos'è

WebAPI è una collezione di API Passepartout basate su tecnologia Web tramite protocollo https. Il servizio è fruibile utilizzando richieste https che implementano il pattern architetturale REST: **R**epresentational **S**tate **T**ransfer per questo motivo WebAPI può essere definito RESTful.

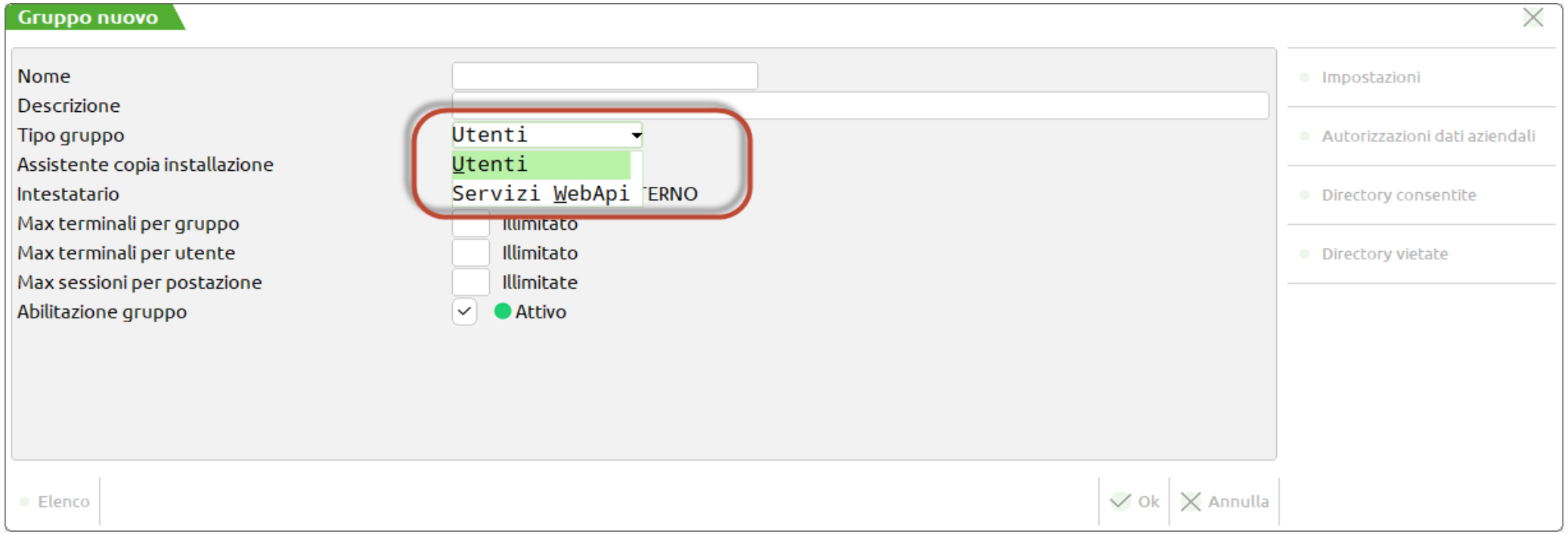
Sono implementati tutti i metodi CRUD (**C**reate, **R**ead, **U**pdate, **D**elete) per mezzo dei metodi http **POST**, **GET**, **PUT**, **DELETE**.

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

PASSEPARTOUT
Software e servizi gestionali

Come si attiva

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.



PASSEPARTOUT
Software e servizi gestionali

Come si usa

Accesso al servizio HTTPS in locale

In installazioni locali il servizio deve essere attivato da configurazione moduli

L'indirizzo per accedere al servizio è:

https://URL:PORTA/webapi

PASSEPARTOUT
Software e servizi gestionali

Come si usa

Accesso al servizio HTTPS in Live

In installazioni Live il servizio è sempre attivo

L'indirizzo per accedere al servizio è:
https://services.passepartout.cloud/webapi/

PASSEPARTOUT
Software e servizi gestionali

Come si usa

Accesso alle risorse ed ai servizi

Per ogni utente WebAPI è disponibile un pool di massimo 3 connessioni contemporanee.

Tramite WebAPI è possibile gestire due canali differenti:

- Accesso alle risorse tramite architettura RESTful:
  https://base_address/webapi/risorse

- Richiesta di servizi tramite metodi POST:
  https://base_address/webapi/servizi

PASSEPARTOUT
Software e servizi gestionali

Come si usa

HTTP Request HEADER Locale (Content-type opzionale)

{

'Authorization': 'Passepartout QURNSA46QQ== QURNSU46QURNSU4=',

'Content-type': 'application/json',

'Coordinate-Gestionale': 'Azienda=DEM Anno=2021 SottoAzienda=1 Magazzino=2'

}

HTTP Request HEADER Live (Content-type opzionale)

{

'Authorization': 'Passepartout QURNSA46QQ== DOMINIO=NOME_DOMINIO',

'Content-type': 'application/json',

'Coordinate-Gestionale': 'Azienda=DEM Anno=2021 SottoAzienda=1 Magazzino=3',

'Cookie': 'WEBAPIID=4416ed4382467e0d; Path=/webapi; …' ***N.B. OBBLIGATORIO!***

}

PASSEPARTOUT
Software e servizi gestionali

Come si usa

Comandi RESTful

GET per leggere una singola risorsa o una collezione di risorse -- http status 200 (OK)
POST per creare una nuova risorsa ---------- http status 201 (Created)
PUT per modificare una risorsa ---------- http status 204 (No Content)
DELETE per eliminare una risorsa ---------- http status 204 (No Content)

GET (no body)

GET https://base_address/webapi/risorse/articoli?qs ---------- 200 OK: lista di tutti gli articoli
GET https://base_address/webapi/risorse/articoli/FUELEX97?qs -- 200 OK: dati dell'articolo

Con i metodi di tipo GET sono ammesse anche Query String (qs):

- **max**: numero massimo di record da restituire (default=100 e non più di 100)
- **next**: prossimo record da cui partire
- **fields**: campi (colonne) che si vuole vengano restituiti
- **info**: se *true* help in linea di tutti i campi che compongono l'archivio

PASSEPARTOUT Software e servizi gestionali

Come si usa

HELP

Utilizzando il metodo GET sulla risorsa WebAPI è possibile ottenere un HELP in linea di tutti gli end point REST disponibili:

GET https://base_address/webapi/HELP -> 200 OK

Solo ed esclusivamente in questo caso non sono richiesti parametri di autenticazione nell'header.

Nel JSON di ritorno sarà presente un Array di elementi, uno per ogni end-point, con tutte le informazioni sui metodi REST disponibili, campi chiave, comando nativo, etc.

Utilizzando l'HELP e, per ogni entità, info=true per le informazioni, di fatto si ha un HELP in linea completo.

PASSEPARTOUT
Software e servizi gestionali

Come si usa

POST (dati in body)

POST https://base_address/webapi/risorse/articoli -> 201 Create: Header Location con path nuovo articolo

PUT (dati in body)

PUT https://base_address/webapi/risorse/articoli/FUELEX97-> 204 No Content

DELETE (no body)

DELETE https://base_address/webapi/risorse/articoli/FUELEX97 -> 204 No Content

PASSEPARTOUT
Software e servizi gestionali

Come si usa

Ricerca avanzata

Dato che i comandi GET non prevedono dati in ingresso, per effettuare ricerche utilizzando filtri più evoluti oltre a quelli in query string si utilizza il metodo POST (sono comunque permesse le Query String)

POST https://base_address/webapi/risorse/articoli/ricerca?qs -> 200 OK: risultato ricerca

Nel body del POST vanno impostati i filtri (tutti in AND tra loro) secondo questo schema:

```
{
"filtri":[
        {
                "campo":"ardes",
                "condizione":"=",
                "valore":"Trek FUELEX"
        },
        {…},
        {…}
]
}
```


PASSEPARTOUT Software e servizi gestionali

Come si usa

Comandi per servizi general purpose

Nel caso si voglia effettuare una richiesta per un servizio generico (es. Collage Server Remoto) si utilizza un altro path dedicato ed esclusivamente con metodo POST

POST https://base_address/webapi/servizi -> 200 OK

```
{
    "cmd": "esec_collage_server_remoto",
    "codice_app": "924148WEBSHAKER",
    "nome_collage": "colws",
    "etichetta_collage": "WAPI"
}
```


PASSEPARTOUT Software e servizi gestionali

Come si usa

Comandi per servizi general purpose

Lo stesso servizio può essere utilizzato per eseguire tutti i comandi RESTful ma specificando il comando e i vari filtri nel body del POST.

Esempio lista clienti:

```
{
        "cmd": "lista_clienti",
        "next": "",
        "max_elementi": 50,
        "info_campi": false,
        "campi":["pccod","pcdes"],
        "dati": {},
        "filtri":[]
}
```

Il comando è recuperabile tramite metodo GET su https:/address/webapi/HELP

PASSEPARTOUT Software e servizi gestionali

PASSEPARTOUT®
Software e servizi gestionali

WebAPI

Gianluca Suzzi